ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

"ANALISIS DE FUNCIONES CRIPTOGRÁFICAS DE CÓDIGO LIBRE EN LOS PROTOCOLOS SSL Y TLS APLICADO AL PORTAL WEB DE LA JEFATURA PROVINCIAL DE TRÁNSITO DE CHIMBORAZO"

TESIS DE GRADO

PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS

MANUEL ABELARDO HARO MONTERO

FREDY MARCELO GAVILANES SAGÑAY

RIOBAMBA – ECUADOR

- 2009 -

AGRADECIMIENTO

Nuestra más sincera gratitud a las personas que incondicionalmente tendieron su mano generosa y día a día nos encaminaron e impulsaron para alcanzar esta meta, a la Escuela Superior Politécnica de Chimborazo, a la Escuela de Ingeniería en Sistemas a nuestros padres, hermanos, amigos y docentes. De manera especial al Director de Tesis Ing. Danilo Pástor quien ha encaminado la presente investigación. Además agradecemos la colaboración al Capitán Juan Carlos Ortiz Baca, Capitán Carlos Alexander Semblantes Gamboa y a todo el personal de la Jefatura de Chimborazo.

DEDICATORIA

La presente tesis está dedicada principalmente a Nuestro Creador quien nos proporcionó la vida e inteligencia, así también como a nuestros progenitores quienes brindaron su apoyo moral y económico en cada etapa de nuestra carrera.

FIRMAS RESPONSABLES Y NOTAS

DR. ROMEO RODRIGUEZ _____

DECANO DE LA FACULTAD DE

INFORMÁTICA Y ELECTRÓNICA

ING. IVAN MENES _____

DIRECTOR DE LA ESCUELA

DE INGENIERÍA EN SISTEMAS

ING. DANILO PASTOR _____

DIRECTOR TESIS

DR. MARIO PAGUAY _____

MIEMBRO DE TESIS

LCDO. CARLOS RODRIGUEZ _____

DIRECTOR DEL CENTRO

DE DOCUMENTACIÓN

RESPONSABILIDAD DEL AUTOR

Nosotros, Fredy Marcelo Gavilanes Sagñay y Manuel Abelardo Haro Montero, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

_____ _____

Fredy Marcelo Gavilanes Sagñay Manuel Abelardo Haro Montero

ABREVIATURAS

AC	Autoridad Certificadora
AJAX	XML y JavaScript Asíncrono
BD	Base de Datos
CRL	Lista de Certificados Revocados
CSR	Solicitud de un Certificado Digital
DES	Algoritmo de Encriptación Estándar
DNS	Servidor de Nombre de Dominios
GPL	Licencia Pública General
HTTP	Protocolo de Transferencia de Hipertexto
HTTPS	Protocolo Seguro de Transferencia de Hipertexto
JPTCH	Jefatura Provincial de Tránsito de Chimborazo
LGPL	Licencia Pública General Menor
MAC	Código de Autenticación de Mensaje
MD5	Algoritmo de Resumen de Mensaje
MPL	Licencia Pública de Mozilla
NSPR	Netscape Portable Runtime
NSS	Servicios de Red Seguros
PSK	Pre-Shared Key (Secreto compartido)
PHP	Procesador de Hipertexto

RC2	Algoritmo de Clave Simétrica versión 2
RC4	Algoritmo de Clave Simétrica versión 4
SMIME	Extensiones de Correo de Internet de Propósitos Múltiples Seguro
SONIA	Aplicación web para la Administración de Informes y Noticias de forma Segura
SRP	Secure Remote Password
SSL	Protocolo de Capa Segura
TLS	Protocolo de Capa de Transporte Seguro
TSA	Time Stamping Authority
UIAT	Unidad de Investigación de Accidentes de Tránsito
WWW	World Wide Web
XP	Programación Extrema

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE GRAFICOS

INTRODUCCIÓN

Estamos en un mundo de transición en donde la mayoría de las transacciones que se realizaban personalmente, hoy en día las entidades y empresas están automatizando utilizando el medio global más difundido que es la Internet, a través de las aplicaciones web.

Esta transición ha creado la necesidad de desarrollar canales de comunicación seguros para ocultar la información más sensible como son cuentas y claves de usuario, números de cuentas bancarias, números de tarjetas de crédito, en resumen brindar al usuario seguridad, confiabilidad y tranquilidad para realizar sus transacciones.

En el capítulo I Marco Referencial, se propone el estudio de herramientas de administración de funciones criptográficas: OpenSSL, GnuTLS y NSS. Estas herramientas implementan los protocolos SSL y TLS que permiten la comunicación de forma segura en una red.

En el capítulo II Marco Teórico, se realiza un estudio teórico de los conceptos necesarios para entender el funcionamiento y la implementación de los protocolos de transmisión segura SSL y TLS por las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. También se conceptualiza los requerimientos para el desarrollo e implantación de una aplicación web en un servidor seguro.

En el capítulo III Análisis Comparativo, se estudia previamente las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS y se realiza un análisis comparativo para seleccionar la herramienta más idónea para la implantación de la aplicación web “SONIA”.

En el capítulo IV Parte Aplicativa, se pone en práctica los conceptos estudiados y analizados en los capítulos II y III, se utiliza la herramienta más idónea para la implantación de la aplicación web “SONIA” sobre un servidor web seguro.

CAPÍTULO I MARCO REFERENCIAL

1.1 Introducción

En este capítulo se plantea el estudio y análisis de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, debido a que estas herramientas proveen librerías y funciones que implementan los protocolos seguros SSL y TLS, los cuales aculatan la información que es transmitida sobre una red.

Se detalla los lineamientos y directrices que ayudaran a desarrollar el proyecto de una forma eficaz y objetiva para evitar contratiempos y redundancias en las actividades y tareas planificadas para encaminar a la investigación correctamente.

Se definirá las metas principales de este proyecto que se deberán cumplir de acuerdo a una ordenada planificación de recursos como: recursos financieros, recursos humanos, recurso tiempo.

1.2 Problematización

1.2.1. Planteamiento

1.2.1.1 Descripción

Actualmente los sitios Web se han convertido en un punto primordial en el funcionamiento de las instituciones públicas, privadas y de empresas de cualquier índole, es por ello que la evolución en el desarrollo web avanzado en pasos agigantados.

Uno de los aspectos más importantes en el desarrollo de los sitios web es la seguridad, y más aún la transmisión de información, es por ello que actualmente los portales necesitan asegurar la información que es enviada y recibida a los clientes.

La principal debilidad de la transmisión de la información a través del protocolo HTTP, es el envío de la información en texto plano de forma que pueden ser interceptados por terceras partes inescrupulosas que pueden hacer un mal uso de esta información.

El protocolo HTTPS nos permite cubrir la falencia del protocolo HTTP transmitiendo la información encriptado de tal forma que al ser interceptada por cualquier persona no pueda ser entendida, para hacer uso de este protocolo es necesario conocer el funcionamiento de las funciones de encriptación, certificados digitales y CA, que nos permiten realizar este proceso.

1.2.1.2 Análisis

La transmisión de información a través del internet actualmente se realiza en texto plano y esto implica que la información puede ser interceptada con fines maliciosos, en especial si la información que se transmite es sensible como numero de tarjetas de crédito, contraseñas, etc. Para ello es necesario cifrar la información.

Una de las maneras de asegurar la información es utilizando la seguridad basada en los protocolos SSL y TLS, los mismos que nos permiten transmitir la información de una forma cifrada mediante la utilización de llaves públicas y privadas para cifrar la información.

La JPTCH actualmente no consta con portal web, los procesos aún son realizados de forma manual como la presentación de la información a la ciudadanía sobre las actividades que realiza la JPTCH en beneficio de la misma, de igual forma el envío de peritajes por parte de la Unidad de Investigación de Accidentes de Tránsito y la comunicación a los involucrados sobre la actualizaciones de licencias y matriculas.

1.2.2. Formulación

¿Existen estudios comparativos de herramientas de código libre para la administración de funciones criptográficas en los protocolos SSL y TLS para el aseguramiento de la información?

1.2.3. Sistematización

- ¿Existen estudios de comparación sobre herramientas de administración de funciones criptográficas basadas en los protocolos SSL y TLS?
- ¿Cuáles son las ventajas y desventajas de asegurar la información mediante el cifrado?
- ¿Existe un estudio interno de cada herramienta que decida el protocolo más eficiente para la transmisión de la información de forma segura?
- ¿Existen Autoridades Certificadoras de fácil acceso para la obtención de certificados digitales en nuestro medio?

1.3 Justificación

Es importante el análisis de la seguridad de la información basada en los protocolos SSL y TLS ya que la información de cualquier institución es considerada un punto crítico en su estructura.

Proponemos el estudio comparativo de estas tres herramientas de código libre porque podemos observar la estructura y funcionamiento de las mismas y cuantificar las potencialidades y debilidades de cada una.

Una de las maneras utilizadas hoy en día en la seguridad de portales web es la basada en los protocolos SSL y TLS, que utilizan principalmente la criptografía para enviar información cifrada entre el cliente y el servidor y evitar así ataques de terceras partes extrañas e interesadas en la información que se está transmitiendo.

Para ello primero se analizará y luego se comparará proyectos de código libre como: OpenSSL, GnuTLS y NSS, los cuales poseen características de Administración de Funciones Criptográficas a nivel de los protocolos SSL y TLS.

- Análisis de la estructura y funcionamiento de las 3 herramientas de administración de funciones criptográficas: OpenSSL, GnuTLS y NSS.
- Comparación de las 3 herramientas de administración de funciones criptográficas: OPENSSL, GnuTLS y NSS.
- El proyecto no se enfocará en como los protocolos SSL y TLS, transmiten la información.
- No estudiará la seguridad basada en S/MIME, utilizada en la seguridad de envío y recepción de correos electrónicos.

En una institución como es la JPTCH es necesario manejar la información de forma segura, para eso desarrollaremos el portal web de la JPTCH utilizando una de las herramientas comparadas.

Por ser una política de estado el uso de software de código libre fueron elegidas estas tres herramientas, de donde una de ellas será implantada en la JPTCH que es una institución pública del Ecuador.

El portal web nos permitirá acceder a:

- Información general de la JPTCH como es Misión, Visión, Objetivos y datos generales de sus miembros.
- Noticias publicadas por la JPTCH por cada uno de sus departamentos: Ingeniería de Tránsito, UIAT, Relaciones Públicas.

- Información de actualización de matrículas y licencias.
- Información de los accidentes de tránsito registrados por el UIAT.
- Estadísticas por fechas y sectores de los accidentes de tránsito.
- Informes de peritajes realizados por la UIAT de forma segura, únicamente a usuarios específicos.

El presente proyecto no contemplará la implantación de seguridad basada en S/MIME, utilizada en correos electrónicos.

La creación del portal de la JPTCH no se extiende a las demás Jefaturas y Sub-Jefaturas del país.

1.4 Objetivos

1.4.1. General

- Analizar y comparar las herramientas de código libre para la administración de funciones criptográficas en los protocolos SSL Y TLS aplicado al desarrollo del portal web de la JEFATURA PROVINCIAL DE TRÁNSITO DE CHIMBORAZO.

1.4.2. Específicos

- Estudiar conceptos sobre las funciones criptográficas, algoritmos de criptografía, firmas digitales y diferentes tipos de licencias Open Source y GNU.
- Establecer parámetros y métricas para la evaluación de las herramientas OpenSSL, GnuTLS y NSS.
- Analizar y comparar las herramientas de código libre para la administración de funciones criptográficas.
- Desarrollar el portal web de la JEFATURA PROVINCIAL DE TRÁNSITO DE CHIMBORAZO aplicando las herramientas de seguridad.

1.5 Hipótesis

Mediante el análisis y comparación de las herramientas de código libre para la administración de funciones criptográficas: OpenSSL, GnuTLS y NSS en los protocolos SSL y TLS, permitirá seleccionar la herramienta más adecuada para asegurar la transmisión de la información en aplicaciones web.

1.6 Métodos y Técnicas

1.6.1. Método de investigación científica

Este proyecto hará uso del método científico porque nos da un conjunto de reglas y lineamientos que regirá el procedimiento para ejecutar esta investigación.

1.6.2. Técnicas

Observación: Utilizaremos herramientas para observar el trafico de la red y comprobar que realmente la información se transmite de forma cifrada.

Lluvia de ideas: La lluvia de ideas será una de las principales técnicas para recolectar la información y para procesarla, por el mismo hecho que esta investigación se realizará en equipo.

Entrevista: Se realizará entrevistas al personal administrativo de la JPTCH para obtener los requerimientos para el desarrollo e implantación del portal web de esta institución.

Pruebas del sistema: Las pruebas que realicemos en el sistema con las diferentes herramientas a comparar nos permitirán obtener información para poder elegir la más optima.

CAPÍTULO II MARCO TEÓRICO

2.1 Introducción

En el presente capitulo se detalla los conceptos básicos para que permitan un mejor entendimientos del funcionamientos de los protocolos de comunicación seguros SSL y TLS, de las librerías y funciones implementadas, y permitan realizar un mejor estudio de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

Además se detalla las funciones de las Autoridades Certificadoras, y conceptos de los certificados digitales y los algoritmos de cifrado y descifrado tanto simétricos como asimétricos que son una parte muy importante de la implementación de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, y necesarias para la implantación de comunicaciones seguras en los protocolos SSL y TLS.

Las licencias también se encuentran detalladas en este capítulo debido a que conforma una parte fundamental el conocer la situación legal de las diferentes herramientas que utilizan en el presente estudio.

2.2 Conceptos

2.2.1. Metodología XP

El desarrollo del software al igual que cualquier otro producto necesita de un proceso establecido como una metodología que guie al desarrollador hacia el objetivo final al cual debe llegar y de esta forma pueda presentar un producto de mejor calidad en un tiempo óptimo. La Metodología XP es una metodología ágil, moderna y basada en tecnología actual y por esta razón ha sido seleccionada para trabajar en este proyecto.

Empieza con la fase de **Planificación y Análisis,** en donde se recogerá la información más importante desde el punto de vista de la operación de la institución y donde se analizará la funcionalidad que tendrá el sistema. Con el objetivo de tener un vínculo más estrecho para de esa manera, tener mayor confiabilidad y determinar correctamente la funcionalidad, para poder realizar correcciones menores.

En la fase de **Diseño** se empieza a estructurar y armar la aplicación web, es aquí donde se diseña el nivel más bajo del sistema, como lo es el diseño de la base de datos. La estructura de la aplicación web, es decir el diagrama de clases se lo diseña aquí. Opcionalmente se diseña el diagrama de componentes.

Para la fase de **Codificación** se construye el código no solo la codificación de la aplicación web como tal sino la codificación a bajo nivel como en la base de datos, los procedimientos almacenados y las vistas. Además la codificación de las clases de la aplicación web.

Finalmente en la fase de Pruebas todo código generado debe ser puesto tanto a pruebas unitarias como a pruebas globales del sistema, todo esto se puede modificar en posteriores iteraciones.

La metodología XP promueve 4 valores, los cuales se los describe a continuación:

2.2.1.1 Comunicación

El eXtreme Programming se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación: la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio. Cuando dos (o más) personas se comunican directamente pueden no solo consumir las palabras formuladas por la otra persona, sino que también aprecian los gestos, miradas, etc. que hace su compañero. Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva.

2.2.1.2 Coraje

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave.

2.2.1.3 Simplicidad

Dado que no se puede predecir cómo va a ser en el futuro, el software que se está desarrollando; un equipo de programación extrema intenta mantener el software lo más sencillo posible. Esto quiere decir que no se va a invertir ningún esfuerzo en hacer un desarrollo que en un futuro pueda llegar a tener valor. En el XP frases como "...en un futuro vamos a necesitar..." o "Haz un sistema genérico de..." no tienen ningún sentido ya que no aportan ningún valor en el momento.

2.2.1.4 Feedback

La agilidad se define (entre otras cosas) por la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Por este motivo uno de los valores que nos hace más ágiles es el continuo seguimiento o feedback que recibimos a la hora de

desarrollar en un entorno ágil de desarrollo. Este feedback se toma del cliente, de los miembros del equipo, en cuestión de todo el entorno en el que se mueve un equipo de desarrollo ágil.

Estos valores son promovidos mediante la ejecución de las 12 siguientes prácticas:

2.2.1.5 Planificación incremental

La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de feedback que permitan conocer con precisión dónde estamos. Como es lógico, la planificación es iterativa: un representante del negocio decide al comienzo de cada iteración qué características concretas se van a implementar.

2.2.1.6 Testing

La ejecución automatizada de tests es un elemento clave de la XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor cualificada para decidir cuál es la funcionalidad más valiosa.

2.2.1.7 Programación en parejas

La XP incluye, como una de sus prácticas estándar, la programación en parejas. Nadie programa en solitario, siempre hay dos personas delante del ordenador. Ésta es una de las características que más se cuestiona al comienzo de la adopción de la XP dentro de un equipo, pero en la práctica se acepta rápidamente y de forma entusiasta.

2.2.1.8 Refactorización

A la hora de la verdad, el código de la mayor parte de las aplicaciones empieza en un razonable buen estado, para luego deteriorarse de forma progresiva. El coste desorbitado

del mantenimiento, modificación y ampliación de aplicaciones ya existente se debe en gran parte a este hecho.

2.2.1.9 Diseño simple

Otra práctica fundamental de la Programación Extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". Se evita diseñar características extra porque a la hora de la verdad la experiencia indica que raramente se puede anticipar qué necesidades se convertirán en reales y cuáles no. La XP nos pide que no vivamos bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras: lo que parece necesario cambia con frecuencia, es difícil acertar a priori.

2.2.1.10 Integración continua

En muchos casos la integración de código produce efectos laterales imprevistos, y en ocasiones la integración puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La Programación Extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata, en lugar de durante una fase de integración más o menos remota.

2.2.1.11 Cliente en el equipo

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo de desarrollo toma decisiones de negocio críticas. Esto no debería ocurrir, pero a la hora de la verdad con frecuencia no se obtiene feedback del cliente con la fluidez necesaria: el resultado es que se ha de optar por detener el avance de los proyectos, o porque el desarrollo tome una decisión de negocio. Por otra parte, los representantes del negocio también suelen encontrarse con problemas inesperados debido a que tampoco reciben el feedback adecuado por parte de los desarrolladores.

2.2.1.12 Releases pequeñas

Siguiendo la política de la XP de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Éstas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor como para que resulten valiosas para el cliente.

2.2.1.13 Semanas de 50 horas

La Programación Extrema lleva a un modo de trabajo en que el equipo está siempre al 100%. Una semana de 40 horas en las que se dedica la mayor parte del tiempo a tareas que suponen un avance puede dar mucho de sí, y hace innecesario recurrir a sobreesfuerzos -excepto en casos extremos.

2.2.1.14 Estándares de codificación

Para conseguir que el código se encuentre en buen estado y que cualquier persona del equipo pueda modificar cualquier parte del código es imprescindible que el estilo de codificación sea consistente. Un estándar de codificación es necesario para soportar otras prácticas de la XP.

2.2.1.15 Uso de Metáforas

La comunicación fluida es uno de los valores más importantes de la Programación Extrema: la programación en parejas, el hecho de incorporar al equipo una persona que represente los intereses del negocio y otras prácticas son valiosas entre otras cosas porque potencian enormemente la comunicación.

2.2.2. Servidor Web Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por

completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la WWW. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere application server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache en orden de previsualizar y probar código mientras éste es desarrollado.

Microsoft Internet Information Services (IIS) es el principal competidor de Apache, así como Sun Java System Web Server de Sun Microsystems y un anfitrión de otras aplicaciones como Zeus Web Server

La mayor parte de la configuración se realiza en el fichero httpd.conf. Cualquier cambio en este archivo requiere reiniciar el servidor.

La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

El nombre Apache es una marca registrada y puede ser sólo utilizada con el permiso expreso del dueño de la marca.

2.2.2.1 Características

- Corre en una multitud de Sistemas Operativos.
- Es una tecnología gratuita de código fuente abierto.
- Altamente configurable de diseño modular.
- Trabaja con gran cantidad lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar.
- Tiene una alta configurabilidad en la creación y gestión de logs.

2.2.2.2 Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Algunos de estos módulos son:

- mod_ssl: Comunicaciones Seguras vía protocolo TLS.
- mod_rewrite: Reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los

navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).

- mod_dav: Soporte del protocolo WebDAV.
- mod_deflate: Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- mod_auth_ldap: Permite autentificar usuarios contra un servidor LDAP.
- mod_proxy_ajp: Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod_perl: Páginas dinámicas en Perl.
- mod_php: Páginas dinámicas en PHP.
- mod_python: Páginas dinámicas en Python.
- mod_rexx: Páginas dinámicas en REXX y Object REXX.
- mod_ruby: Páginas dinámicas en Ruby.
- mod_aspdotnet: Páginas dinámicas en .NET de Microsoft (Módulo retirado).
- mod_mono: Páginas dinámicas en Mono
- mod_security: Filtrado a nivel de aplicación, para seguridad.

2.2.3. Gestor de Base de Datos MySQL

2.2.3.1 Introducción

MySQL es un sistema que administra base de datos relacionales, multihilo y multiusuario que actualmente es desarrollada por la empresa MySQL AB y Sun Microsystems. Estas herramientas desarrollan MySQL bajo una licencia dual software libre y comercial, lo que quiere decir que se puede adquirir bajo los términos de la licencia GNUGPL o con condiciones específicas para incluirlo en productos privados.

MySQL es propietario y está patrocinado por una empresa privada la cual posee el copyright de la mayor parte del código, por esta razón su licencia dual.

La compañía ofrece soporte técnico y servicios en todo el mundo.

2.2.3.2 Historia

MySQL se basa en el estándar SQL, MySQL es una idea original de la empresa open source MYSQL AB establecida inicialmente en Suecia. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL pero sin sacrificar velocidad, usabilidad o fiabilidad.

2.2.3.3 Características

MySQL tiene soporte para conectarse con muchos lenguajes de programación como: C, C++, C#, Java, Lisp, Perl, PHP, Python, Ruby, etc. Y para lenguajes que tengan soporte para conexiones ODBC.

MySQL es muy utilizado en apliaciones web y de escritorio y por herramientas seguidores de errores como bugzilla.

MySQL funciona sobre una gran cantidad de plataformas como Windows, Linux, OpenBSD, HP-UX, BSD, Solaris, etc.

2.2.3.4 Características de la versión 5.0.22

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones
- Soporte para multiplataforma
- Procedimientos almacenados
- Soporte para triggers
- Cursores
- Vistas actualizables
- Soporte a VARCHAR
- INFORMATION_SCHEMA

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

2.2.3.5 Características adicionales

- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- Uso de multihilos mediante hilos del kernel.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice.
- Tablas hash en memoria temporales.
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.
- Completo soporte para operadores y funciones en cláusulas select y where.
- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación.

- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).
- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.
- En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.
- MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

2.2.4. Lenguaje de Programación Interpretado PHP

PHP es un lenguaje de programación interpretado. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. La más reciente versión principal del PHP fue la versión 5.2.6 de 1 de mayo de 2008.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# VB.NET como lenguajes), a ColdFusion de la compañía Adobe (antes Macromedia), a JSP/Java de Sun Microsystems, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE (entorno de desarrollo

integrado) comercial llamado Zend Studio. Recientemente, CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno integrado de desarrollo para PHP, denominado Delphi for PHP. Existe un módulo para Eclipse uno de los IDE más populares.

2.2.4.1 Características

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

2.2.4.2 Frameworks en PHP

- Zend Framework
- PHP Prado
- Symfony
- CakePHP
- Qcodo
- Kumbia
- PHP4ECore
- CodeIgniter

- Yii Framework
- MfwLite
- Tomates Framework

2.2.4.3 IDEs para PHP

Algunos de los Entornos de Desarrollo Integrados para PHP más conocidos o habituales son:

- Zend Studio: Comercial (Zend).
- PDT, plugin de Eclipse: GPL (Sun).
- Komodo IDE: Libre y gratuito, el IDE es licencia comercial - (Mozilla).
- NuSphere PhpED: Comercial, para Linux y Windows.
- NetBeans: Libre, para Linux y Windows.
- Quanta: GPL y gratuito, para GNU/Linux con QT.
- Bluefish: GPL y gratuito, para GNU/Linux con GTK.
- phpDesigner: Comercial y Freeware, para Linux y Windows.
- Rapid PHP: Comercial para Windows.
- Aptana Studio: GPL, existe una versión comercial.

2.2.5. Tecnología AJAX

Ajax (acrónimo de Asynchronous JavaScript And XML) no es un lenguaje de programación en sí mismo sino la combinación de una serie de tecnologías que permiten una relación cliente-servidor más eficaz agilizando la respuesta de este último. Se considera una técnica de desarrollo web para crear aplicaciones interactivas. Las mismas se ejecutan en las computadoras-clientes lo que brinda ciertas ventajas en lo que respecta a la velocidad e interacción y traspase de contenido. Esta unión de tecnologías está compuesta, entonces, por cuatro tecnologías ya existentes, que operan de modo conjunto, a saber:

- El objeto XMLHttpRequest que permite el intercambio asincrónico con el servidor.

- Un formato que permite transferir datos de vuelta al servidor. Si bien el que más se utiliza es el XML, otros formatos pueden ser utilizados.
- DOM (Document Object Model), al cual los usuarios deben acceder a través de un lenguaje scripting como JavaScript para interactuar con la información solicitada.
- XHTML y hojas en estilo "cascada" (CSS) para el diseño de la página web, es decir, para mostrar el contenido.

La idea principal de la utilización de AJAX reside en habilitar una carga asíncrona de información en una página sin requerir una recarga de la web de modo completo, puesto que las aplicaciones de AJAX permiten trabajar con una serie determinada de datos, por tanto, se reduce la información que se intercambia entre servidor-cliente, y se gana en tiempo y velocidad. Como advertencia, es necesario tener en cuenta las reiteradas incompatibilidades entre servidores por lo general antiguos que no utilizan las tecnologías arriba descriptas dado que es imperativo para su funcionamiento, por ejemplo, tener activado el JavaScript del navegador correspondiente al usuario.

El funcionamiento de estas tecnologías en su conjunto es sencillo. El servidor envía la aplicación en formato HTML, Javascript y CSS a cada usuario o cliente. Es el código Javascript el que pide el contenido a mostrar –el que procesa la respuesta-, y el servidor ejecuta un código que a través del formato XML manda al usuario los datos solicitados. Esta acción se repite cada vez que dicho usuario realice una operación que requiera la muestra de los datos. Por lo tanto, como mencionamos anteriormente, estas peticiones comprenden únicamente a la información que es necesitada y no a la totalidad del contenido. Es extremadamente útil para aquellas ocasiones en donde las peticiones al servidor son continuas. Por lo tanto, la innovación de las aplicaciones de AJAX radica en evitar el tiempo de espera de respuesta del servidor o la recarga constante de la página web. Un claro ejemplo de cómo funcionan las aplicaciones basadas en AJAX es el servicio de correo electrónico Gmail brindado por Google.

Esta técnica de desarrollo web es relativamente nueva (el término data de año 2005) pero bastante utilizada hoy día. La diferencia entre las aplicaciones web tradicionales y las

creadas con AJAX es radical por lo que podemos considerar que estamos tratando con una técnica innovadora en lo que respecta a la programación de aplicaciones web.

2.2.5.1 Navegadores que permiten AJAX

Ha de tenerse en cuenta que ésta es una lista general, y el soporte de las aplicaciones AJAX dependerá de las características que el navegador permita.

- Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores.
- Google Chrome.
- Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él.
- Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores, Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores.
- Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

2.2.5.2 Navegadores que no permiten AJAX

- Opera 7 y anteriores
- Microsoft Internet Explorer para Windows versión 4.0 y anteriores
- Safari, cualquier versión anterior a la 1.2
- Dillo
- Navegadores basados en texto como Lynx y Links
- Navegadores para incapacitados visuales (Braille)

2.2.6. Licencias de Software

2.2.6.1 Introducción

La licencia de software es una especie de contrato, en donde se especifican todas las normas y clausulas que rigen el uso de un determinado programa, principalmente se estipulan los alcances de uso, instalación, reproducción y copia de estos productos.

En el momento en que usted decide descargar, instalar, copiar o utilizar un determinado software, implica que usted acepta las condiciones que se estipulan en la LICENCIA que trae ese programa.

Licenciar un Software es el procedimiento de conceder a otra persona o entidad el derecho de usar un software con fines industriales, comerciales o personales, de acuerdo a las clausulas que en ella aparecen.

Existen varios tipos de licencias como son: GPL, LGPL, MPL, BSD, FREWARE, etc.

2.2.6.2 Licencia OPEN SOURCE

La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

El autor, bajo esta licencia, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación.

Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre.

2.2.6.3 Licencia GPL

La licencia GPL se aplica al software de la FSF (Free Software Foundation) y el proyecto GNU otorga al usuario la libertad de compartir el software y realizar cambios en él. Dicho de otra forma, el usuario tiene derecho a usar el programa, modificarlo y distribuir las versiones modificadas pero no tiene permiso de realizar restricciones propias con respecto a la utilización de ese programa modificado.

La licencia GPL o copyleft (contrario a copyright) fue creada para mantener la libertad del software y evitar que alguien quisiera apropiarse de la autoría intelectual de un determinado programa. La licencia advierte que el software debe ser libre y que el paquete final, también debe ser libre.

2.2.6.4 Licencia GNU Lesser GPL

Si bien la licencia GPL ofrece grandes beneficios, hay algunas veces en la que ofrece ciertas restricciones. Un ejemplo es que un software que utiliza algún componente GPL, debe sí o sí ser licenciado bajo la misma, es decir no se pueden utilizar partes o bibliotecas de software GPL en un software propietario o distribuido bajo otra licencia.

Estas restricciones traen algunos problemas. Por ejemplo si una empresa comercial desea utilizar únicamente una biblioteca GPL pequeña dentro de su software, estaría obligada a distribuir todo su software bajo GPL, lo cual posiblemente no decida hacer y para algunos casos como el de bibliotecas de propósitos generales esto tampoco ayuda a la mejora de la propia biblioteca ya que no sería elegida por ejemplo para convertirse en un estándar. Por eso apareció la licencia LGPL, en un primer momento llamada Library GPL en referencia a que fue especialmente utilizada para bibliotecas, pero luego se popularizó y comenzó a utilizarse inclusive en muchos programas completos debido a sus beneficios comerciales (permite utilizarse junto a software no libre) y cambió su nombre a Lesser GPL que significa GPL menos restrictiva.

2.2.6.5 Licencia MPL

La Licencia Pública de Mozilla es una licencia de código abierto y software libre utilizada por el navegador de Internet Mozilla y sus productos derivados. Cumple completamente con los postulados del open source y del software libre. Sin embargo, la MPL deja abierto el camino a una posible reutilización comercial y no libre del software, si el usuario así lo desea, sin restringir la reutilización del código ni el re-licenciamiento bajo la misma licencia.

Aunque el uso principal de la MPL es servir como licencia de control para el navegador Mozilla y el software relacionado con él, esta licencia es ampliamente utilizada por desarrolladores y programadores que quieren liberar su código.

MPL es una licencia de código abierto y software libre utilizada por desarrolladores y programadores para la liberación de código fuente.

Legalmente no se puede enlazar un módulo cubierto por la licencia GPL con un módulo cubierto por la licencia MPL.

2.2.6.6 Licencia Apache

La licencia Apache (Apache License o Apache Software License para versiones anteriores a 2.0) es una licencia de software libre creada por la Apache Software Foundation (ASF). La licencia Apache (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que permite el uso y distribución del código fuente para software libre y software propietario.

Todo el software producido por la ASF o cualquiera de sus proyectos está desarrollado bajo los términos de esta licencia. Además algunos proyectos que no pertenece a la ASF también siguen la licencia Apache: en enero de 2007, más de 1000 proyectos no pertenecientes a la ASF en SourceForge estaban disponibles bajo los términos de la Licencia Apache.

2.2.7. Cifrado

2.2.7.1 Cifrado de flujo

Es cifrado incremental que convirtiendo el texto en claro en texto cifrado bit a bit. Esto se logra construyendo un generador de flujo de clave. Un flujo de clave es una secuencia de bits de tamaño arbitrario que puede emplearse para oscurecer los contenidos de un flujo de datos combinando el flujo de clave con el flujo de datos mediante la función XOR Si el flujo de clave es seguro, el flujo de datos cifrados también lo será.

2.2.7.2 Cifrado de bloque

Es una unidad de cifrado de clave simétrica que opera en grupos de bits de longitud fija, llamados bloques, aplicándoles una transformación invariante. Cuando realiza cifrado, una unidad de cifrado por bloques toma un bloque de texto en claro como entrada y produce un bloque de igual tamaño de texto cifrado. La transformación exacta es controlada utilizando una segunda entrada la clave secreta. El descifrado es similar: se ingresan bloques de texto cifrado y se producen bloques de texto en claro.

2.2.8. Criptografía Simétrica

La criptografía simétrica es el método criptográfico que usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez ambas tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma.

Un buen sistema de cifrado pone toda la seguridad en la clave y ninguna en el algoritmo. En otras palabras, no debería ser de ninguna ayuda para un atacante conocer el algoritmo que se está usando. Sólo si el atacante obtuviera la clave, le serviría conocer el algoritmo.

Dado que toda la seguridad está en la clave, es importante que sea muy difícil adivinar el tipo de clave. Esto quiere decir que el abanico de claves posibles, o sea, el espacio de posibilidades de claves, debe ser amplio

Actualmente, los ordenadores pueden descifrar claves con extrema rapidez, y ésta es la razón por la cual el tamaño de la clave es importante en los criptosistemas modernos. El algoritmo de cifrado DES usa una clave de 56 bits, lo que significa que hay 2 elevado a 56 claves posibles (72.057.594.037.927.936 claves). Esto representa un número muy alto de claves, pero una máquina computadora de uso general puede comprobar el conjunto posible de claves en cuestión de días. Una máquina especializada puede hacerlo en horas. Algoritmos de cifrado de diseño más reciente como 3DES, Blowfish e IDEA utilizan claves de 128 bits, lo que significa que existen 2 elevado a 128 claves posibles. Esto equivale a muchísimas más claves, y aun en el caso de que todas las máquinas del planeta estuvieran cooperando, tardarían más tiempo en encontrar la clave que la edad del universo.

Algunos ejemplos de algoritmos simétricos son 3 DES AES, Blowfish e IDEA.

El principal problema con los sistemas de cifrado simétrico no está ligado a su seguridad, sino al intercambio de claves. Una vez que el remitente y el destinatario hayan intercambiado las claves pueden usarlas para comunicarse con seguridad, pero ¿qué canal de comunicación que sea seguro han usado para transmitirse las claves? Sería mucho más fácil para un atacante intentar interceptar una clave que probar las posibles combinaciones del espacio de claves.

Otro problema es el número de claves que se necesitan. Si tenemos un número n de personas que necesitan comunicarse entre sí, se necesitan n/2 claves para cada pareja de personas que tengan que comunicarse de modo privado. Esto puede funcionar con un grupo reducido de personas, pero sería imposible llevarlo a cabo con grupos más grandes.

Los algoritmos simétricos utilizan 2 elementos en su funcionamiento básico:

2.2.8.1 IV (Vector de Inicialización)

El vector de inicialización es la cadena con la que empieza cada proceso de encriptación. Aquí el error más común es utilizar la misma cadena al iniciar todas las encriptaciones, es aquí donde se produce el error debido a que le podemos ahorrar el trabajo de descifrar datos al atacante. Esta cadena tiene 16 bytes de longitud.

2.2.8.2 Key (Clave)

Es el componente principal para llevar a cabo el proceso de encriptar y desencriptar información con algoritmos simétricos. Toda la seguridad del funcionamiento de los algoritmos simétricos radica en la clave, en donde se encuentra, como está compuesta y quien tiene acceso a la misma. La longitud de las claves depende del algoritmo.

2.2.8.3 DES

Es un algoritmo criptográfico simétrico, fue escogido por la FIPS en los Estados Unidos en 1976. En su inicio fue un algoritmo controvertido al principio por su longitud de clave relativamente corta y las sospechas sobre la existencia de vigilancia por parte de NSA (National Security Agency).

Actualmente a este algoritmo se lo considera inseguro para la utilización en muchas aplicaciones, debido a longitud de la clave de 56 bits lo cual actualmente es considerado como inseguro en donde se han roto la seguridad en menos de 24 horas.

2.2.8.3.1 Descripción

DES es el algoritmo de encriptación prototipo del cifrado en bloques, es un algoritmo que toma un texto en claro de una longitud fija de bits y lo transforma mediante una serie de complicadas operaciones en otro texto cifrado de la misma longitud. DES utiliza un tamaño de bloque de 64 bits, utiliza también una clave criptográfica para modificar la transformación, de modo que el descifrado sólo puede ser realizado por aquellos que conozcan la clave utilizada en el cifrado. La clave mide 64 bits, pero solo 56 de ellos es utilizada dejando los 8 restantes para comprobar la paridad.

2.2.8.3.2 Estructura Básica

El algoritmo consta de 16 fases idénticas de proceso las cuales se las denomina rondas. También existe una permutación inicial y final las cuales son funciones inversas entre si, aunque las permutaciones inicial y final no son realmente significativas en el proceso de encriptación pero son incluidos porque facilita la carga y descarga de los bloques de datos

sobre el hardware especialmente en la década de los 70. Antes de las rondas el bloque es dividido en 2 mitades de 32 bits y son procesadas alternadamente, este entrecruzamiento se lo conoce como esquema Feistel.

2.2.8.3.3 Generación de claves

Para la generación de claves en el cifrado primero se selecciona 56 bits de la clave de 64 bits iniciales mediante la permutación inicial, los 8 bits son simplemente descartados o utilizados como bits de comprobación de paridad. Los 56 bits de dividen entonces en 3 mitades de 28 bits a continuación cada mitad se trata independientemente. En rondas sucesivas ambas mitades se desplazan hacia la izquierda uno o dos bits y entonces se seleccionan 48 bits de sub-clave mediante la permutación final, 24 bits de la mitad izquierda y 24 de la derecha. Los desplazamientos implican que se utiliza un conjunto diferente de bits en cada sub-clave, cada bit se usa aproximadamente en 14 de las 16 sub-clave.

La generación de claves para el descifrado es similar se lo debe generar las claves en orden inverso.

2.2.8.4 3DES

Este algoritmo se le denomina también Tiple DES y se lo denomina así porque realiza un triple cifrado DES, fue desarrollado por IBM en 1978.

2.2.8.4.1 Algoritmo

Este no es un algoritmo de cifrado múltiple porque no son independientes todas las subclases. Esto es porque DES tiene la característica matemática de no ser un grupo, lo que implica que si se cifra el mismo bloque dos veces con dos claves diferentes se aumenta el tamaño efectivo de la clave.

Su representación matemática es la siguiente manera:

$$C = {E_{DES}}^{k3}({E_{DES}}^{k2}({E_{DES}}^{k1}(M))$$

Donde M es el mensaje a cifrar y k1 k2 k3 las respectivas clases DES.

2.2.8.4.2 Seguridad

Cuando se descubrió que una clave de 56 bits no era suficiente para evitar un ataque por fuerza bruta que lograba romper la seguridad en menos de 24 horas, por ello fue desarrollado 3DES para agrandar el largo de la clave sin necesidad de cambiar de algoritmo de cifrado. Este método de cifrado es inmune al ataque por encuentro a medio camino, doblando la longitud efectiva de la clave a 128 bits, pero en cambio es preciso triplicar el número de cifrado muchísimo más seguro que el DES. Por lo tanto la longitud de la clave será de 192 bits, aunque su eficacia solo sea de 112 bits.

2.2.8.4.3 Usos

Su uso está desapareciendo porque ha sido desplazado por el algoritmo AES. Sin embargo firma de tarjeta de crédito tienen como estándar el 3DES.

2.2.8.5 AES

Advanced Enxryption Standard (AES por sus siglas en inglés) o también denominado Rijndael, es basado en un esquema de cifrado de bloques que actualmente ha sido adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. Se espera que sea difundido, utilizado y analizado exhaustivamente como los fue su predecesor DES (Data Encryption Standard). El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología como un FIPS PUB 197 de los Esrados Unidos el 26 de Noviembre de 2001 después de un proceso de estandarización que duró aproximadamente 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

Los desarrolladores del cifrado fueron los belgas Joan Daemen y Vincent Rijmen, ambos estudiante de la Katholieke Universiteit Leuven, y enviado al proceso de selección AES bajo el nombre Rijindael, el cual abarca los nombres de los inventores.

2.2.8.5.1 Desarrollo

AES fue un refinamiento de un diseño anterior de Daemen y Rijmen, denominado Square que a su vez fue un desarrollo de Shark.

Al contrario de su predecesor DES, Rijindael es una red de sustitución-permutación, no es una red de Feistel. AES es rápido tanto en software como en hardware, es relativamente fácil de implementar, y requiere poca memoria. Como nuevo estándar de cifrado, se está utilizando actualmente a gran escala.

2.2.8.5.2 Descripción del cifrado

AES opera en una matriz de 4*4 bytes, llamada state. Para el cifrado, cada ronda de la aplicación del algoritmo AES excepto la última consiste en 4 pasos:

SubBytes: se realiza una sustitución no lineal donde cada byte es reemplazado con otro de acuerdo a una tabla de búsqueda.

ShiftRows: se realiza una transposición donde cada fila del “state” es rotada de manera cíclica un número determinado de veces.

MixColumns: es una operación de mezclado que opera en las columnas del “state”, combinando los 4 bytes en cada columna utilizando una tranformación lineal.

AddRoundKey: cada byte de “state” es combinado con la clave “round”, cada clave round se deriva de la clave del cifrado utilizando una iteración de la clave.

En rc5 idean la ronda final se reemplaza la fase MixColumns por otra instancia de AddRoundKey.

Hasta el momento se lo considera seguro debido a que no se ha registrado que algún ataque haya logrado descifrar o romper la seguridad de este algoritmo.

2.2.8.6 IDEA

En criptografía, International Data Encryption Algorithm o IDEA (del inglés, *Algoritmo Internacional de Cifrado de Datos*) es un cifrador por bloques diseñado por Xuejia Lai y James L. Massey de la Escuela Politécnica Federal de Zúrich y descrito por primera vez en 1991. Fue un algoritmo propuesto como reemplazo del DES (Data Encryption Standard). IDEA fue una revisión menor de PES (Proposed Encryption Standard, del inglés *Estándar de Cifrado Propuesto*), un algoritmo de cifrado anterior. Originalmente IDEA había sido llamado IPES (Improved PES, del inglés *PES Mejorado*).

IDEA fue diseñado en contrato con la Fundación Hasler, la cual se hizo parte de Ascom-Tech AG. IDEA es libre para uso no comercial, aunque fue patentado y sus patentes se vencerán en 2010 y 2011. El nombre "IDEA" es una marca registrada y está licenciado mundialmente por MediaCrypt.

IDEA fue utilizado como el cifrador simétrico en las primeras versiones de PGP (PGP v2.0) y se lo incorporó luego de que el cifrador original usado en la v1.0 ("Bass-O-Matic") se demostró insegura. Es un algoritmo óptimo en OpenPGP.

2.2.8.6.1 Funcionamiento

IDEA opera con bloques de 64 bits usando una clave de 128 bits y consiste de ocho transformaciones idénticas (cada una llamada un *ronda*) y una transformación de salida (llamada *media ronda*). El proceso para cifrar y descifrar es similar. Gran parte de la seguridad de IDEA deriva del intercalado de operaciones de distintos grupos — adición y multiplicación modular y O-exclusivo (XOR) bit a bit — que son algebraicamente "incompatibles" en cierta forma.

IDEA utiliza tres operaciones en su proceso con las cuales logra la confusión, se realizan con grupos de 16 bits y son:

- Operación O-exclusiva (XOR) bit a bit.
- Suma módulo 216.

- Multiplicación módulo 216+1, donde la palabra nula (0x0000) se interpreta como 216.

(2^{16} = 65536; 2^{16}+1 = 65537, que es primo)

Después de realizar 8 rondas completas se realiza la mitad, obteniendo el resultado en este paso de la ronda:

Este algoritmo presenta, a primera vista, diferencias notables con el DES, que lo hacen más atractivo:

- El espacio de claves es mucho más grande: 2128 ≈ 3.4 x 1038
- Todas las operaciones son algebraicas
- No hay operaciones a nivel bit, facilitando su programación en alto nivel
- Es más eficiente que los algoritmos de tipo Feistel, porque a cada vuelta se modifican todos los bits de bloque y no solamente la mitad
- Se pueden utilizar todos los modos de operación definidos para el DES

2.2.8.6.2 Seguridad

En primer lugar, el ataque por fuerza bruta resulta impracticable, ya que sería necesario probar 10^{38} claves, cantidad imposible de manejar con los medios informáticos actuales.

Los diseñadores analizaron IDEA para medir su fortaleza frente al criptoanálisis diferencial y concluyeron que es inmune bajo ciertos supuestos. No se han reportado debilidades frente criptoanálisis lineal o algebraico. Se han encontrado algunas claves débiles, las cuales en la práctica son poco usadas siendo necesario evitarlas explícitamente. Es considerado por muchos como uno de los cifrados en bloque más seguros que existen.

2.2.8.7 CAST

Es un buen sistema de cifrado en bloques con una clave CAST-128 bits, es muy rápido y es gratuito. Su nombre deriva de las iniciales de sus autores, Carlisle, Adams, Stafford Tavares, de la empresa Northern Telecom (NorTel).

CAST no tiene claves débiles o semidébiles y hay fuertes argumentos acerca que CAST es completamente inmune a los métodos de criptoanálisis más potentes conocidos.

También existe una versión con clave CAST-256 bits que ha sido candidato a AES

2.2.8.8 RC2

El RC2 es un algoritmo de cifrado por bloques de clave de tamaño variable diseñado por Ron Rivest de RSA Data Security (la RC quiere decir *Ron's Code* o *Rivest's Cipher*).

El desarrollo del algoritmo RC2 fue auspiciado por Lotus, ellos investigaban un algoritmo de cifrado personalizado que después de la evaluación de NSA, pudieron ser conocidos como parte del software Lotus. La NSA sugirió unos pequeños cambios con la incorporación de Rivest. Después de algunas negociaciones el algoritmos fue aprobado en 1989. La exportación del mismo se encuentra bajo las regulaciones de la US export regulations para criptografía.

El algoritmo trabaja con bloques de 64 bits y entre dos y tres veces más rápido que el DES en software. Se puede hacer más o menos seguro que el DES contra algoritmos de fuerza bruta eligiendo el tamaño de clave apropiadamente.

El algoritmo está diseñado para reemplazar al DES.

2.2.8.9 RC4

Este es un algoritmo de sistema de cifrado de flujo más utilizado y se utiliza en los protocolos TLS/SSL. Fue excluido enseguida de los estándares de alta seguridad por los criptógrafos y alguno modos de usar este algoritmo lo han llevado a ser un sistema de criptografía muy inseguro. Actualmente no está recomendado en la utilización de algunos sistemas. Aunque algunos sistemas basados en RC4 son los suficientemente seguros para un uso común.

2.2.8.9.1 Descripción

RC4 genera un flujo pseudoaleatorio de bits que al momento de cifrar se combina con el texto plano usando la función XOR (or exclusivo) como en cualquier cifrado Vermam. En la fase de descifrar el mensaje se lo realiza del mismo modo.

Para generar el keystream, el algoritmo de cifrado tiene un estado interno secreto que consiste en:

Una permutación de todos los 256 posibles símbolos de un byte de longitud (lo denominaremos S).

Dos subíndices de 8 bits (los denominaremos “i”, “j”).

La permutación se inicializa con una clave de longitud variable, habitualmente entre 40 y 256 bits usando un algoritmo de programación de claves (KSA Key Scheduling Algorithm).

Una vez completado, el flujo de bits cifrados se genera utilizando un algoritmo de generación pseudoaleatoria.

2.2.8.9.2 Algoritmo de programación de claves

Este algoritmo se utiliza para inicializar la permutación del vector “S”. el subíndice i se define como el número de bytes de la clave y puede estar en el rango 1<=i<=256, típicamente entre 5 y 16 valores correspondientes a un tamaño de clave de entre 40 y 128 bits. Primero el vector S si inicia a la permutación identidad S es entonces procesado por 256 iteraciones similares para el algoritmo principal PRGA, pero mezclándola con bytes de la clave al mismo tiempo.

2.2.8.10 RC5

El RC5 es un algoritmo parametrizable con tamaño de bloque variable, tamaño de clave variable y número de rotaciones variable. Los valores más comunes de los parámetros son 64 o 128 bits para el tamaño de bloque, de 0 a 255 rotaciones y claves de 0 a 2048 bits. Fue diseñado en 1994 por Ron Rivest.

El RC5 tiene 3 rutinas: expansión de la clave, encriptación y desencriptación. En la primera rutina la clave proporcionada por el usuario se expande para llenar una tabla de claves cuyo tamaño depende del número de rotaciones. La tabla se emplea en la encriptación y desencriptación. Para la encriptación sólo se emplean tres operaciones: suma de enteros, o-exclusiva de bits y rotación de variables.

La mezcla de rotaciones dependientes de los datos y de distintas operaciones lo hace resistente al criptoanálisis lineal y diferencial. El algoritmo RC5 es fácil de implementar y analizar y, de momento, se considera que es seguro.

2.2.8.11 BLOWFISH

En criptografía, Blowfish es un codificador de bloques simétricos, diseñado por Bruce Schneier en 1993 e incluido en un gran número de conjuntos de codificadores y productos de cifrado. Mientras que ningún analizador de cifrados de Blowfish efectivo ha sido encontrado hoy en día, se ha dado más atención de la decodificación de bloques con bloques más grandes, como AES y Twofish.

Schneier diseñó Blowfish como un algoritmo de uso general, que intentaba reemplazar al antiguo DES y evitar los problemas asociados con otros algoritmos. Al mismo tiempo, muchos otros diseños eran propiedad privada, patentados o los guardaba el gobierno. Schneier declaró “Blowfish no tiene patente, y así se quedará en los demás continentes. El algoritmo está a disposición del público, y puede ser usado libremente por cualquiera”.

2.2.8.11.1 Diagrama de Blowfish

Blowfish usa bloques de 64 bits y claves que van desde los 32 bits hasta 448 bits. Es un codificador de 16 rondas Feistel y usa llaves que dependen de las Cajas-S. Tiene una estructura similar a CAST-128, el cual usa Cajas-S fijas.

El diagrama muestra la acción de Blowfish. Cada línea representa 32 bits. El algoritmo guarda 2 arrays de subclaves: El array P de 18 entradas y 4 cajas-S de 256 entradas. Una entrada del array P es usada cada ronda, después de la ronda final, a cada mitad del

bloque de datos se le aplica un XOR con uno de las 2 entradas del array P que no han sido utilizadas.

La función divide las entradas de 32 bits en 4 bloques de 8 bits, y usa los bloques como entradas para las cajas-S. Las salidas deben estar en módulo 2^{32} y se les aplica un XOR para producir la salida final de 32 bits.

2.2.8.11.2 Diagrama de la función F de Blowfish

Debido a que Blowfish está en la red Feistel, puede ser invertido aplicando un XOR entre P_{17} y P_{18} al bloque texto codificado, y así sucesivamente se usan las P-entradas en orden reversivo.

La generación de claves comienza inicializando los P-arrays y las cajas-S con los valores derivados de los dígitos hexadecimales de pi, los cuales no contienen patrones obvios. A la clave secreta se le aplica un XOR con las P-entradas en orden (ciclando la clave si es necesario). Un bloque de 64 bits de puros ceros es cifrado con el algoritmo como se indica. El texto codificado resultante reemplaza a P_1 y P_2. Entonces el texto codificado es cifrado de nuevo con la nuevas subclaves, P_3 y P_4 son reemplazados por el nuevo texto codificado. Esto continúa, reemplazando todas las entradas del P-array y todas las entradas de las cajas-S. En total, el algoritmo de cifrado Blowfish correrá 521 veces para generar todas las subclaves, cerca de 4KB de datos son procesados.

2.2.8.12 Camellia

Camellia es un algoritmo de cifrado de bloque que ha tenido una evaluación favorable por algunas organizaciones, incluido el proyecto de la Unión Europea NESSIE y el proyecto japonés CRYPTREC. Este algoritmo de cifrado fue desarrollado conjuntamente a Mitsubishi y la NIT en el 2000, tiene un similar diseño de los elementos de cifrado de cada bloque a los algoritmos MISTY1 y E2 de las compañías antes mencionadas.

Camellia tiene un tamaño de bloque de 128 bits, y puede también utilizar tamaños de clave de 128, 192 o 256 bits. Tiene una interfaz muy similar al algoritmo AES. Trabaja con

el esquema de Feistel de 18 rondas si utiliza un tamaño de clave de 128 bits o 24 rondas si utiliza un tamaño de clave de 192 o 256 bits. Cada 6 rondas se aplica una transformación lógica llamada “FL-function” o su inversa.

Camellia utiliza cuadrados de 8 x 8 con entrada y salida para transformaciones y operaciones lógicas. El cifrado también utiliza barrido de claves de entrada y salida. La capa de difusión utiliza una transformación basada en la función hash md5.

En junio 18 de 2008, el browser Mozilla liberó su versión que soporta el algoritmo Camellia.

El software geli de FreeBSD 7.0 en Noviembre 11 de 2008 ha dado soporte para el algoritmo Camellia gracias a Yoshisato Yanagisawa. Pero el equipo de ingeniería de ingeniería de FreeBSD anunció el soporte para las versiones posteriores a la 6.4 de FreeBSD.

2.2.8.12.1 Análisis de Seguridad

Camellia es uno de los algoritmos de cifrado que puede ser completamente definido por un mínimo de sistemas de polinomios mutivariados. Tanto Camellia como AES las cajas S pueden ser descritas como un sistema de 23 ecuaciones cuadradas en 80 términos. La clave puede ser descrita por 1120 ecuaciones con 768 variables utilizando 3328 términos lineales y cuadrados. El bloque entero de cifrado puede ser descrito por 5104 ecuaciones con 2816 variables utilizando 14592 términos lineales y cuadrados. En total 6224 ecuaciones con 3584 variables utilizando 17920 términos lineales y cuadrados. El número de términos libres es 11696 que es aproximadamente el número para AES. Teóricamente las probabilidades para romper la seguridad de los algoritmos AES y Camellia se la lograría utilizando ataques algebraicos con Linealización Espacio Extendido.

2.2.8.12.2 Estado de la Patente

Aunque el algoritmo es patentado, Camellia está disponible bajo la licencia Royalty-free.

Esto ha permitido que el algoritmo Camellia ha llegado ser parte del proyecto OpenSSL, bajo la licencia Open Source a partir de Noviembre 8 de 2006.

2.2.9. Criptografía Asimétrica

La criptografía asimétrica utiliza un par de claves para cifrar la información, este par de claves son únicos, la una clave es de dominio público por lo que cualquier persona puede saber pero la otra es privada y únicamente debe conocer su propietario. Los métodos criptográficos garantizan que estos pares de claves no se repitan.

Una vez que el remitente utiliza la clave pública para cifrar el mensaje solo el destinatario poseedor de la clave privada puede descifrar el mensaje y de esta forma se consigue la confidencialidad. Si el propietario de la clave privada utiliza esta para cifrar el mensaje todos los que poseen la pública pueden descifrar el mensaje y de esta forma se consigue la identificación y autenticación del remitente. En esta idea se basa la firma electrónica.

Este sistema de criptografía asimétrica permite el intercambio de la clave de los sistemas de cifrado simétrico. Estos sistemas de criptografía asimétrica se basan en funciones de un solo sentido, y su seguridad se basa en la longitud de su clave.

Las desventajas que estos sistemas presentan es la gran cantidad de tiempo que se necesita para cifrar un mensaje en comparación con los sistemas simétricos, además que la longitud de sus mensajes cifrados es mayor a la del mensaje original.

Algunos algoritmos que utilizan claves públicas son: RSA, DHE, SRP, PSK, DSA, Curvas Elipticas (CEE), etc.

2.2.8.13 RSA

El algoritmo RSA realiza un cifrado en bloques y es el más popular y utilizado de los algoritmos asimétricos gracias a su facilidad para el entendimiento y la implementación, básicamente se basa en la teoría de los números primos, este algoritmo fue creado por Ron Rivest, Adi Shamir y Leonard Adleman en 1977.

2.2.8.13.1 Funcionamiento

Los cálculos matemáticos de este algoritmo emplean un número denominado Módulo Público, N, que forma parte de la clave pública y que se obtiene a partir de la multiplicación de dos números primos, p y q , diferentes y grandes (del orden de 512 bits) y que forman parte de la clave privada. La gran propiedad de RSA es que, mientras que N es público, los valores de p y q se pueden mantener en secreto debido a la dificultad de la factorización de un número grande.

También puede ser usado para autenticar un mensaje

2.2.8.13.2 Seguridad

La seguridad del criptosistema RSA está basado en dos problemas matemáticos: el problema de factorizar números grandes y el problema RSA. El descifrado completo de un texto cifrado con RSA es computacionalmente intratable, no se ha encontrado un algoritmo eficiente todavía para ambos problemas. Proveyendo la seguridad contra el descifrado parcial podría requerir la adición de una seguridad padding scheme.

La longitud de la clave puede tener una longitud variable y puede ser tan grande como se desee.

2.2.8.14 Diffie Hellman (DH)

Desarrollado por ***Whitfiled Diffie y Martín Hellman*** en 1976. Este algoritmo es utilizado para el intercambio de claves simétricas entre dos partes que no han tenido contacto previo, para ello utiliza un canal inseguro y de manera anónima.

2.2.8.14.1 Funcionamiento

La generación de claves públicas es la siguiente:

- Se busca un número grande y primo llamado q.
- Se busca a raíz primitiva de q. Para ser raíz primitiva debe cumplir que: amodq, a2modq, a3modq,..., aq-1modq son números diferentes.

- a y q son claves públicas.

Para compartir la clave publica la persona A envía la potencia discreta $YA=\alpha^X modq$ hacia B y B envía la potencia $YB=\alpha^X modq$ hacia A. Y luego cada uno calcula $K= YA^X modq$ y $K= YB^X modq$ con los datos obtenidos en donde las K calculadas por los dos usuarios son iguales por la propiedad distributiva de la multiplicación.

2.2.8.14.2 Seguridad

La seguridad del algoritmo depende de la dificultad del cálculo de un logaritmo discreto. Esta función es la inversa de la potencia discreta, o sea, de calcular una potencia y aplicar una función mod.

- Potencia discreta: Y=Xamodq
- Logaritmo discreto: X=Lnda,q(Y)

2.2.8.15 DSA

Algoritmo de firmas digital, es un estándar del Gobierno Federal de los Estados Unidos para firmas digitales, este algoritmo fue ***propuesto*** por el Instituto Nacional de Normas y Tecnologías de los Estados Unidos para su uso en su Estándar de Firmas Digital (DSS), el algoritmo sirve para firmar y no para cifrar información. Es más lento que el RSA.

2.2.8.15.1 Funcionamiento

El DSA es un algoritmo asimétrico que únicamente se puede utilizar con firma digital Utiliza más parámetros que el RSA y así se consigue un grado de mayor seguridad.

2.2.8.15.2 Generación de llaves

- Elegir un número primo p de L bits, donde $512 \leq L \leq 1024$ y L es divisible por 64.
- Elegir un número primo q de 160 bits, tal que $p-1 = qz$, donde z es algún número natural.
- Elegir h, donde $1 < h < p - 1$ tal que $g = hz(\text{mod } p) > 1$.
- Elegir x de forma aleatoria, donde $1 < x < q-1$.

- Calcular y = gx(mod p).

Los datos públicos son *p, q, g* e *y*. La llave privada es *x*.

2.2.8.15.3 Generación de Firmas

- Elegir un número aleatorio s, donde 1 < s < q.
- Calcular s1 = (gs mod p)mod q.
- Calcular s2 = s-1(H(m)+s1*x)mod q, donde H(m) es la función hash SHA-1 aplicada al mensaje m.
- La firma es el par (s1, s2).

Si s1 o s2 es cero, se vuelve a repetir el procedimiento.

2.2.8.15.4 Seguridad

La longitud de las claves es la principal fuente de seguridad, este algoritmo puede tener una clave entre 512 y 1024 bites de longitud.

2.2.8.16 Curvas Elípticas (CEE)

Las curvas elípticas fueron propuestas por primera vez para ser usadas en aplicaciones criptográficas en 1985 de forma independiente por Miller y Koblitz. Las curvas elípticas en sí llevan estudiándose durante muchos siglos y están entre los objetos más ricamente estructurados y estudiados de la teoría de números.

La eficiencia de este algoritmo radica en la longitud reducida de las claves, lo cual permite su implementación en sistemas de bajos recursos como teléfonos celulares y Smart Cards.

2.2.8.16.1 Funcionamiento

Con el conjunto de puntos G que forman la curva (i.e., todas las soluciones de la ecuación más un punto O, llamado punto en el infinito) más una operación aditiva *+*, se forma un grupo abeliano. Si las coordenadas *x* e *y* se escogen desde un campo finito, entonces estamos en presencia de un grupo abeliano finito. El problema del logaritmo discreto

sobre este conjunto de puntos (PLDCE) se cree que es más difícil de resolver que el correspondiente a los campos finitos (PLD). De esta manera, las longitudes de claves en criptografía de curva elíptica pueden ser más cortas con un nivel de seguridad comparable.

2.2.8.16.2 Seguridad

La realización de las operaciones necesarias para ejecutar este sistema es más lenta que para un sistema de factorización o de logaritmo discreto módulo entero del mismo tamaño. De todas maneras, los autores de sistemas de CCE creen que el PLDCE es significativamente más complicado que los problemas de factorización o del PLD, y así se puede obtener la misma seguridad mediante longitudes de clave mucho más cortas utilizando CCE, hasta el punto de que puede resultar más rápido que, por ejemplo, RSA. Los resultados publicados hasta la fecha tienden a confirmar esto, aunque algunos expertos se mantienen escépticos.

2.2.10. Funciones Hash

Las funciones hash son funciones matemáticas que realizan un resumen del documento a firmar. Su forma de operar es comprimir el documento en un único bloque de longitud fija, bloque cuyo contenido es ilegible y no tiene ningún sentido real. Tanto es así que por definición las funciones hash son irreversibles, es decir, que a partir de un bloque comprimido no se puede obtener el bloque sin comprimir, y si no es así no es una función hash. Estas funciones son además de dominio público.

A un mensaje resumido mediante una función hash y encriptado con una llave privada es lo que en la vida real se denomina ***firma digital***.

Las funciones hash y la firma digital son elementos indispensables para el establecimiento de canales seguros de comunicación, basados en los Certificados Digitales.

Para que una función pueda considerarse como función hash debe cumplir:

- Debe transformar un texto de longitud variable en un bloque de longitud fija, que generalmente es pequeña (algunas son de 16 bits).
- Debe ser cómoda de usar e implementar.
- Debe ser irreversible, es decir, no se puede obtener el texto original del resumen hash.
- Debe ser imposible encontrar dos mensajes diferentes cuya firma digital mediante la función hash sea la misma (no-colisión).
- Si se desea además mantener un intercambio de información con Confidencialidad, basta con cifrar el documento a enviar con la clave pública del receptor.

Las funciones hash más conocidas y usadas son:

2.2.10.1 MD2

Abreviatura de Message Digest 2, diseñado para ordenadores con procesador de 8 bits. Todavía se usa, pero no es recomendable, debido a su lentitud de proceso.

2.2.10.2 MD4

Abreviatura de Message Digest 4, desarrollado por Ron Rivest, uno de los fundadores de RSA Data Security Inc. y padre del sistema asimétrico RSA. Aunque se considera un sistema inseguro, es importante porque ha servido de base para la creación de otras funciones hash. Un sistema de ataque desarrollado por Hans Dobbertin posibilita el crear mensajes aleatorios con los mismos valores de hash (colisiones), por lo que ya no se usa. De hecho, existe un algoritmo que encuentra una colisión en segundos.

2.2.10.3 MD5

Abreviatura de Message Digest 5, también obra de Ron Rivest, que se creó para dar seguridad a MD4, y que ha sido ampliamente usado en diversos campos, como autenticado de mensajes en el protocolo SSL y como firmador de mensajes en el programa de correo PGP. Sin embargo, fue reventado en 1996 por el mismo investigador que lo hizo con MD4, el señor Dobbertin, que consiguió crear colisiones en el sistema MD5, aunque

por medio de ataques parciales. Pero lo peor es que también consiguió realizar ataques que comprometían la no-colisión, por lo que se podían obtener mensajes con igual hash que otro determinado. A pesar de todo esto, MD5 se sigue usando bastante en la actualidad.

2.2.10.4 SHA-1

Secure Hash Algorithm, desarrollado como parte integrante del Secure Hash Standar (SHS) y el Digital Signature Standar (DSS) por la Agencia de Seguridad Nacional Norteamericana, NSA. Sus creadores afirman que la base de este sistema es similar a la de MD4 de Rivest, y ha sido mejorado debido a ataques nunca desvelados. La versión actual se considera segura (por lo menos hasta que se demuestre lo contrario) y es muy utilizada algoritmo de firma, como en el programa PGP en sus nuevas claves DH/DSS (Diffie-Hellman/Digital Signature Standar). Existen cuatro variantes cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (llamándose SHA-2 a todos ellos).

2.2.10.5 RIPEMD-160

Desarrollada por un grupo de investigadores europeos, entre los que se encuentra Hans Dobbertin (el reventador de MD4-MD5) y otros investigadores incluidos en el proyecto RIPE (RACE Integrity Primitives Evaluation). Su primera versión adolecía de las mismas debilidades que MD4, produciendo colisiones, pero las versiones mejoradas actuales son consideradas seguras. Maneja claves muy robustas, normalmente de 160 bits, aunque existen versiones de 128 y se están planteando nuevas de 256 y 320 bits. Es muy rápido, no está patentado y su código fuente es abierto, de libre acceso.

2.2.11. Autoridades Certificadoras

En criptografía una Autoridad de certificación, certificadora o certificante (AC o CA por sus siglas en inglés Certification Authority) es una entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados, utilizados en la firma electrónica, para lo

cual se emplea el cifrado de clave pública. Jurídicamente es un caso particular de Prestador de Servicios de Certificación.

2.2.11.1 Concepto

La Autoridad de Certificación, por sí misma o mediante la intervención de una Autoridad de Registro, verifica la identidad del solicitante de un certificado antes de su expedición o, en caso de certificados expedidos con la condición de revocados, elimina la revocación de los certificados al comprobar dicha identidad. Los certificados son documentos que recogen ciertos datos de su titular y su clave pública y están firmados electrónicamente por la Autoridad de Certificación utilizando su clave privada. La Autoridad de Certificación es un tipo particular de Prestador de Servicios de Certificación que legitima ante los terceros que confían en sus certificados la relación entre la identidad de un usuario y su clave pública. La confianza de los usuarios en la CA es importante para el funcionamiento del servicio y justifica la filosofía de su empleo, pero no existe un procedimiento normalizado para demostrar que una CA merece dicha confianza.

Un certificado revocado es un certificado que no es válido aunque se emplee dentro de su período de vigencia. Un certificado revocado tiene la condición de suspendido si su vigencia puede restablecerse en determinadas condiciones.

2.2.11.2 Modo de funcionamiento

2.2.11.2.1 Solicitud de un certificado

El mecanismo habitual de solicitud de un certificado de servidor web a una CA consiste en que la entidad solicitante, utilizando ciertas funciones del software de servidor web, completa ciertos datos identificativos (entre los que se incluye el localizador URL del servidor) y genera una pareja de claves pública/privada. Con esa información el software de servidor compone un fichero que contiene una petición CSR (Certificate Signing Request) en formato PKCS#10 que contiene la clave pública y que se hace llegar a la CA elegida. Esta, tras verificar por sí o mediante los servicios de una RA (Registration Authority, Autoridad de Registro) la información de identificación aportada y la realización

del pago, envía el certificado firmado al solicitante, que lo instala en el servidor web con la misma herramienta con la que generó la petición CSR.

En este contexto, PKCS corresponde a un conjunto de especificaciones que son estándares de facto denominadas Public-Key Cryptography Standards.

2.2.11.3 La Jerarquía de Certificación

Las CA disponen de sus propios certificados públicos, cuyas claves privadas asociadas son empleadas por las CA para firmar los certificados que emiten. Un certificado de CA puede estar auto-firmado cuando no hay ninguna CA de rango superior que lo firme. Este es el caso de los certificados de CA raíz, el elemento inicial de cualquier jerarquía de certificación. Una jerarquía de certificación consiste en una estructura jerárquica de CAs en la que se parte de una CA auto-firmada, y en cada nivel, existe una o más CAs que pueden firmar certificados de entidad final (titular de certificado: servidor web, persona, aplicación de software) o bien certificados de otras CA subordinadas plenamente identificadas y cuya Política de Certificación sea compatible con las CAs de rango superior.

2.2.11.4 Confianza en la CA

Una de las formas por las que se establece la confianza en una CA para un usuario consiste en la "instalación" en el ordenador del usuario (tercero que confía) del certificado autofirmado de la CA raíz de la jerarquía en la que se desea confiar. El proceso de instalación puede hacerse, en sistemas operativos de tipo Windows, haciendo doble click en el fichero que contiene el certificado (con la extensión "crt") e iniciando así el "asistente para la importación de certificados". Por regla general el proceso hay que repetirlo por cada uno de los navegadores que existan en el sistema, tales como Opera (navegador), Firefox o Internet Explorer, y en cada caso con sus funciones específicas de importación de certificados.

Si está instalada una CA en el repositorio de CAs de confianza de cada navegador, cualquier certificado firmado por dicha CA se podrá validar, ya que se dispone de la clave pública con la que verificar la firma que lleva el certificado. Cuando el modelo de CA

incluye una jerarquía, es preciso establecer explícitamente la confianza en los certificados de todas las cadenas de certificación en las que se confíe. Para ello, se puede localizar sus certificados mediante distintos medios de publicación en internet, pero también es posible que un certificado contenga toda la cadena de certificación necesaria para ser instalado con confianza.

2.2.11.5 Misión de las CA

Finalmente, las CA también se encargan de la gestión de los certificados firmados. Esto incluye las tareas de revocación de certificados que puede instar el titular del certificado o cualquier tercero con interés legítimo ante la CA por e-mail, teléfono o intervención presencial. La lista denominada CRL (Certificate Revocation List) contiene los certificados que entran en esta categoría, por lo que es responsabilidad de la CA publicarla y actualizarla debidamente. Por otra parte, otra tarea que debe realizar una CA es la gestión asociada a la renovación de certificados por caducidad o revocación.

Si la CA emite muchos certificados, corre el riesgo de que sus CRL sean de gran tamaño, lo que hace poco práctica su descarga para los terceros que confían. Por ese motivo desarrollan mecanismos alternativos de consulta de validez de los certificados, como servidores basados en los protocolos OCSP y SCVP.

2.2.11.6 CA de personas y de servidores

Los certificados de "entidad final" a veces designan personas (y entonces se habla de "certificados cualificados") y a veces identifican servidores web (y entonces los certificados se emplean dentro del protocolo SSL para que las comunicaciones con el servidor se protejan con un cifrado robusto de 128 bits).

2.2.11.7 CAs Públicas y Privadas

Una CA puede ser o bien pública o bien privada. Los certificados de CA (certificados raíz) de las CAs públicas pueden o no estar instalados en los navegadores pero son reconocidos como entidades confiables, frecuentemente en función de la normativa del país en el que

operan. Las CAs públicas emiten los certificados para la población en general (aunque a veces están focalizadas hacia algún colectivo en concreto) y además firman CAs de otras organizaciones.

- En Venezuela, la SUSCERTE - MCT (Superintendencia de Servicios de Certificación Electrónica).
- En Perú, el INDECOPI (Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual).
- En Colombia, Certicámara (La Sociedad Cameral de Certificación Digital Certicámara S.A.).
- En Ecuador, el Banco Central del Ecuador.

2.2.11.8 Certificadoras gratuitas

- CAcert.org, Entidad Certificadora administrada por la comunidad
- Albalia Demo CA Certificados de Prueba

En criptografía, X.509 es un estándar UIT-T para infraestructuras de claves públicas (en inglés, *Public Key Infrastructure* o PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

2.2.12. Certificados Digitales

Un Certificado Digital es un documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública.

Si bien existen variados formatos para certificados digitales, los más comúnmente empleados se rigen por el estándar UIT-T X.509. El certificado contiene usualmente el nombre de la entidad certificada, número de serie, fecha de expiración, una copia de la clave pública del titular del certificado (utilizada para la verificación de su firma digital) y la

firma digital de la autoridad emisora del certificado de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación.

2.2.12.1 Formato de certificado digital

Un certificado emitido por una entidad de certificación autorizada, además de estar firmado digitalmente por ésta, debe contener por lo menos lo siguiente:

- Nombre, dirección y domicilio del suscriptor.
- Identificación del suscriptor nombrado en el certificado.
- El nombre, la dirección y el lugar donde realiza actividades la entidad de certificación.
- La clave pública del usuario.
- La metodología para verificar la firma digital del suscriptor impuesta en el mensaje de datos.
- El número de serie del certificado.
- Fecha de emisión y expiración del certificado.

2.2.13. Certificados x.509

2.2.13.1 Historia y uso

X.509 fue publicado oficialmente en 1988 y comenzado conjuntamente con el estándar X.500 y asume un sistema jerárquico estricto de autoridades certificantes (ACs) para emisión de certificados. Esto contrasta con modelos de redes de confianza, como PGP, donde cualquier nodo de la red (no solo las ACs) puede firmar claves públicas, y por ende avalar la validez de certificados de claves de otros. La versión 3 de X.509 incluye la flexibilidad de soporte de otras tecnologías como bridges y mallas. Puede usarse en una web de confianza peer to peer de tipo OpenPGP, pero desde 2004 raramente se usa así.

El sistema X.500 nunca se implementó completamente, y el grupo de trabajo de la infraestructura de clave pública de la IETF, comúnmente conocido como PKIX para *infraestructura de clave pública (X.509)* o PKIX, adaptó el estándar a la estructura más

flexible de Internet. X.509 incluye también estándares para implementación de listas de certificados en revocación (CRLs), y con frecuencia aspectos de sistemas PKI. De hecho, el término *certificado X.509* se refiere comúnmente al Certificado PKIX y Perfil CRL del certificado estándar de X.509 v3 de la IETF, como se especifica en la RFC 3280.

La forma aprobada por la IETF de chequear la validez de un certificado es el Online Certificate Status Protocol (OCSP).

2.2.13.2 Seguridad

En 2005, Arjen Lenstra y Benne de Weger demostraron "como usar colisiones de hash para construir certificados que contienen firmas idénticas y que solo difieren en la clave pública", lo cual alcanzaron usando un ataque de colisiones en la función de hash MD5.

En 2007, Marc Stevens, Arjen Lenstra, y Benne de Weger demostraron "cómo usar colisiones de hash para añadir sufijos a dos ficheros diferentes obteniendo firmas idénticas".

2.2.13.3 Certificados

En el sistema X.509, una autoridad certificadora (AC) emite un certificado asociando una clave pública a un *Nombre Distinguido* particular en la tradición de X.500 o a un *Nombre Alternativo* tal como una dirección de correo electrónico o una entrada de DNS.

2.2.13.4 Estructura de un certificado

La estructura de un certificado digital X.509 v3 es la siguiente:

- Certificado
  - Versión
  - Número de serie
  - ID del algoritmo
  - Emisor
  - Validez
    - No antes de
    - No después de

    - Sujeto
    - Información de clave pública del sujeto
        - Algoritmo de clave pública
        - Clave pública del sujeto
    - Identificador único de emisor (opcional)
    - Identificador único de sujeto (opcional)
    - Extensiones (opcional).
- Algoritmo usado para firmar el certificado
- Firma digital del certificado

Los identificadores únicos de emisor y sujeto fueron introducidos en la versión 2, y las extensiones en la versión 3.

X.509 es la pieza central de la infraestructura de clave pública y es la estructura de datos que enlaza la clave pública con los datos que permiten identificar al titular. Su sintaxis se define empleando el lenguaje ASN.1 (*Abstract Syntax Notation One*) y los formatos de codificación más comunes son DER (*Distinguished Encoding Rules*) o PEM (*Privacy-enhanced Electronic Mail*). Siguiendo la notación de ASN.1, un certificado contiene diversos campos, agrupados en tres grandes grupos:

- El primer campo es el subject (sujeto), que contiene los datos que identifican al sujeto titular. Estos datos están expresados en notación DN (Distinguished Name), donde un DN se compone a su vez de diversos campos, siendo los más frecuentes los siguientes; CN (Common Name), OU (Organizational Unit), O (Organization) y C (Country). Un ejemplo para identificar un usuario mediante el DN, es el siguiente: CN=david.comin O=Safelayer, OU=development, C=ES. Además del nombre del sujeto titular (subject), el certificado, también contiene datos asociados al propio certificado digital, como la versión del certificado, su identificador (serialNumber), la CA firmante (issuer), el tiempo de validez (validity), etc. La versión X.509.v3 también permite utilizar campos opcionales (nombres alternativos, usos permitidos para la clave, ubicación de la CRL y de la CA,etc.).
- En segundo lugar, el certificado contiene la clave pública, que expresada en notación ASN.1, consta de dos campos, en primer lugar, el que muestra el

algoritmo utilizado para crear la clave (ej. RSA), y en segundo lugar, la propia clave pública.

- Por último, la CA, ha añadido la secuencia de campos que identifican la firma de los campos previos. Esta secuencia contiene tres atributos, el algoritmo de firma utilizado, el hash de la firma, y la propia firma digital.

2.2.13.5 Extensiones de archivo de certificados

Las extensiones de archivo de certificados X.509 son:

- .CER - Certificado codificado en CER, algunas veces es una secuencia de certificados
- .DER - Certificado codificado en DER
- .PEM - Certificado codificado en Base64, encerrado entre "-----BEGIN CERTIFICATE-----" y "-----END CERTIFICATE-----"
- .P7B - Ver .p7c
- .P7C - Estructura PKCS#7 SignedData sin datos, solo certificado(s) o CRL(s)
- .PFX - Ver .p12
- .P12 - PKCS#12, puede contener certificado(s) (público) y claves privadas (protegido con clave)

PKCS #7 es un estándar para firmar o cifrar datos (ellos lo llaman "sobreado"). Dado que el certificado es necesario para verificar datos firmados, es posible incluirlos en la estructura SignedData. Un archivo .P7C es simplemente una estructura SignedData, sin datos para firmar.

PKCS #12 evolucionó del estándar PFX (Personal inFormation eXchange) y se usa para intercambiar objetos públicos y privados dentro de un archivo.

Un archivo .PEM puede contener certificados o claves privadas, encerrados entre las líneas BEGIN/END apropiadas.

2.2.13.6 Ejemplo de certificado X.509 y del proceso de validación

Como ejemplo de un certificado X.509, se encuentra aquí una decodificación (generada con OpenSSL) de uno de los certificados viejos de www.freesoft.org. El certificado real tiene un tamaño de alrededor de 1 KB. Fue emitido (firmado) por Thawte (desde que fue adquirido por Verisign), tal como se indica en el campo Emisor. El tema contiene bastante información personal, pero la parte más importante es el nombre común (CN) de www.freesoft.org - esta es la parte que debe coincidir con la terminal que se está autenticando. A continuación viene una clave pública RSA (módulo y exponente público), seguido de la firma, computada tomando un hash MD5 de la primera parte del certificado y cifrándola con la clave privada RSA de Thawte.

```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting
cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/Email=server-certs@thawte.com
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
                OU=FreeSoft,
CN=www.freesoft.org/Email=baccala@freesoft.org
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
                   33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
                   66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
                   70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
                   16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
                   c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
                   8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
                   d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
                   e8:35:1c:9e:27:52:7e:41:8f
               Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
       92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
       ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
       d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
       0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
       5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
```

```
        8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
        68:9f
```

Obsérvese que al final del certificado se encuentra la firma de mismo. Para estampar esta firma, la autoridad certificadora calcula un hash MD5 de la primera parte del certificado (la sección de *Data*: los datos del mismo más la clave pública) y cifra ese hash con su clave privada, que mantiene en secreto. Ahora supongamos que un cliente se conecta a *www.freesoft.org* y el sitio le devuelve el certificado anterior, con la intención de probar su identidad. Para validar este certificado, necesitamos el certificado de la autoridad certificadora, que fue el emisor del primero (Thawte Server CA). Se toma la clave pública del certificado de la autoridad certificadora para decodificar la firma del primer certificado, obteniéndose un hash MD5. Este hash MD5 debe coincidir con el hash MD5 calculado sobre la primera parte del certificado. En ese caso el proceso de validación termina con éxito. Si no, la validación no tiene éxito, y no puede asegurarse que el certificado de *www.freesoft.org* está vinculado con esa clave pública. Dicho de otro modo, es posible que *www.freesoft.org* no sea quien dice ser. A continuación se muestra el certificado de la CA:

```
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number: 1 (0x1)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting
cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/Email=server-certs@thawte.com
       Validity
           Not Before: Aug  1 00:00:00 1996 GMT
           Not After : Dec 31 23:59:59 2020 GMT
       Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting
cc,
                OU=Certification Services Division,
                CN=Thawte Server CA/Email=server-certs@thawte.com
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:0c:
                   68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:da:
                   85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:06:
                   6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:e2:
                   6a:0c:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:6b:
                   29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:90:
```

```
                6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
                5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
                3a:c2:b5:66:22:12:d6:87:0d
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Basic Constraints: critical
            CA:TRUE
Signature Algorithm: md5WithRSAEncryption
    07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8e:ca:d9:
    a8:6f:49:1a:e6:da:51:e3:60:70:6c:84:61:11:a1:1a:c8:48:
    3e:59:43:7d:4f:95:3d:a1:8b:b7:0b:62:98:7a:75:8a:dd:88:
    4e:4e:9e:40:db:a8:cc:32:74:b9:6f:0d:c6:e3:b3:44:0b:d9:
    8a:6f:9a:29:9b:99:18:28:3b:d1:e3:40:28:9a:5a:3c:d5:b5:
    e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
    b2:75:1b:f6:42:f2:ef:c7:f2:18:f9:89:bc:a3:ff:8a:23:2e:
    70:47
```

Este es un ejemplo de un certificado auto firmado. Nótese que el CN del Emisor y el CN del Asunto son iguales. No hay forma de verificar este certificado, salvo que se comprueba contra sí mismo. En este caso, hemos alcanzado el fin de la cadena de certificados. ¿Cómo es entonces que este certificado se hace confiable? Simple: se configura manualmente. Thawte es una de las autoridades certificantes raíz reconocida por Microsoft y Netscape. Este certificado ya viene con el navegador web (probablemente pueda encontrarse listado como "Thawte Server CA" en las opciones de seguridad) y es confiable por defecto. Como certificado confiado globalmente de larga vida (nótese la fecha de vencimiento) que puede firmar prácticamente cualquier cosa (nótese la falta de limitaciones), su clave privada correspondiente debe ser una de las más ocultas del mundo.

2.2.14. Firmas Digitales

La firma digital hace referencia, en la transmisión de mensajes telemáticos y en la gestión de documentos electrónicos, a un método de cifrado que asocia la *identidad* de una persona o de un equipo informático al mensaje o documento. En función del tipo de firma, puede, además, asegurar la *integridad* del documento o mensaje.

La firma electrónica, como la firma ológrafa (autógrafa, manuscrita), puede vincularse a un documento para identificar al autor, para señalar conformidad (o disconformidad) con

el contenido, para indicar que se ha leído o, según el tipo de firma, garantizar que no se pueda modificar su contenido.

2.2.14.1 Terminología

Los términos de firma digital y firma electrónica se utilizan con frecuencia como sinónimos, pero este uso en realidad es incorrecto.

Mientras que firma digital hace referencia a una serie de métodos criptográficos, firma electrónica es un término de naturaleza fundamentalmente legal y más amplia desde un punto de vista técnico, ya que puede contemplar métodos no cifrados.

Un ejemplo claro de la importancia de esta distinción es el uso por la Comisión europea. En el desarrollo de la Directiva europea 1999/93/CE que estable un marco europeo común para la firma electrónica empezó utilizando el término de firma digital en el primer borrador, pero finalmente acabó utilizando el término de firma electrónica para desacoplar la regulación legal de este tipo de firma de la tecnología utilizada en su implementación.

2.2.14.2 La teoría

La firma digital de un documento es el resultado de aplicar cierto algoritmo matemático, denominado función hash, a su contenido y, seguidamente, aplicar el algoritmo de firma (en el que se emplea una clave privada) al resultado de la operación anterior, generando la firma electrónica o digital. El software de firma digital debe además efectuar varias validaciones, entre las cuales podemos mencionar:

- Vigencia del certificado digital del firmante,
- Revocación del certificado digital del firmante (puede ser por OCSP o CRL),
- Inclusión de sello de tiempo.

La función hash es un algoritmo matemático que permite calcular un valor resumen de los datos a ser firmados digitalmente. Funciona en una sola dirección, es decir, no es posible,

a partir del valor resumen, calcular los datos originales. Cuando la entrada es un documento, el resultado de la función es un número que identifica inequívocamente al texto. Si se adjunta este número al texto, el destinatario puede aplicar de nuevo la función y comprobar su resultado con el que ha recibido. Ello no obstante, este tipo de operaciones no están pensadas para que las lleve a cabo el usuario, sino que se utiliza software que automatiza tanto la función de calcular el valor hash como su verificación posterior. Ver Gráfico II.1.

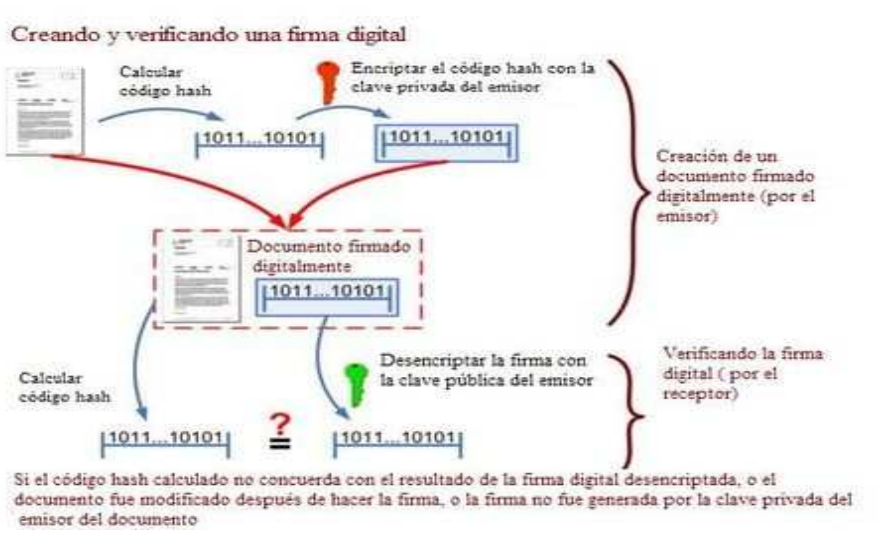

Gráfico II.1: Creación de una firma digital

2.2.14.3 Formato de la firma electrónica

Las normas TS 101 733 y TS 101 903 definen los formatos técnicos de la firma electrónica. La primera se basa en el formato clásico PKCS#7 y la segunda en XMLDsig firma XML especificada por el consorcio W3C.

Bajo estas normas se definen tres modalidades de firma:

Firma básica. Incluye el resultado de operación de hash y clave privada, identificando los algoritmos utilizados y el certificado asociado a la clave privada del firmante. A su vez puede ser "attached" o "detached", "enveloped" y "enveloping"

Firma fechada. A la firma básica se añade un sello de tiempo calculado a partir del hash del documento firmado por una TSA (Time Stamping Authority)

Firma validada o firma completa. A la firma fechada se añade información sobre la validez del certificado procedente de una consulta de CRL o de OCSP realizada a la Autoridad de Certificación.

La firma completa libera al receptor de la firma del problema de ubicar al Prestador de Servicios de Certificación y determinar los procedimientos de validación disponibles.

2.2.14.4 Aplicaciones

- Mensajes con autenticidad asegurada
- Mensajes sin posibilidad de repudio
- Contratos comerciales electrónicos
- Factura_Electrónica
- Desmaterialización de documentos
- Transacciones comerciales electrónicas
- Invitación electrónica
- Dinero electrónico
- Notificaciones judiciales electrónicas
- Voto electrónico
- Decretos ejecutivos (gobierno)
- Créditos de seguridad social
- Contratación pública
- Sellado de tiempo

2.2.15. Protocolo SSL

2.2.15.1 Introducción

El protocolo SSL (Secure Sockets Layer, Capa de Sockets Seguro) fue desarrollado por Netscape, el principal objetivo de este protocolo es proveer de privacidad y confiabilidad a la comunicación entre aplicaciones cliente servidor. Este protocolo fue diseñado con un propósito general y es por esta razón se adapta fácilmente a varias aplicaciones como son transmisión segura, copia segura, acceso remoto en forma segura, correo seguro, comercio electrónico entre otras.

El protocolo SSL se divide en cuatro protocolos: El protocolo Handshake, el protocolo Change Cipher Spec, y el protocolo Alert que están ubicados en la capa de Aplicaciones, y el último protocolo Record que se encuentra sobre la capa TCP del modelo TCP/IP.

El protocolo SSL Handshake es donde se realiza la negociación, en esta fase el cliente y el servidor intentan consensuar los parámetros básicos de la sesión y de la conexión si ambos interlocutores no se pone de acuerdo en lo que se refiere al cifrado y autenticación no se podría producir ninguna transferencia de la información.

En la negociación se define la versión del protocolo usada y los algoritmos de cifrado que se van a aplicar, además cualquiera de los interlocutores pueden solicitar la autenticación del otro como parte del proceso de negociación por último la fase de negociación crea un conjunto de claves que se comparten usando técnicas de cifrado de clave pública (cifrado asimétrico).

El Protocolo SSL Change Cipher Spec se usa para señalizar cambios de estrategia de cifrado es decir se puede cambiar entre un algoritmo de cifrado y otro.

Se podría usar por ejemplo en transacciones HTTP criticas en cualquier momento cuando uno de los dos interlocutores estima que la seguridad puede estar comprometida es posible volver a negociar la utilización de una nueva especificación de seguridad.

El protocolo de Alerta que permite informar al interlocutor sobre circunstancias excepcionales: Mensajes no esperados, MAC incorrecto, Error de descompresión, Error de negociación, Certificado corruptos o caducados, etc.

Usando mensajes de alerta para cada una de los casos mencionados anteriormente.

La capa de registros de SSL o protocolo record recibe datos no interpretados en bloques de tamaño arbitrario y lleva a cabo las siguientes operaciones: Fragmentación, compresión y Cifrado.

Un sitio web se puede identificar que es seguro si su URL comienza con https:// en lugar del http://. SSL proporciona servicios de cifrado de datos, autenticación de servidores, integridad de mensajes, y opcionalmente autenticación del cliente en conexiones TCP/IP.

2.2.15.2 Funcionamiento básico del Protocolo SSL

El funcionamiento del protocolo SSL se podría dividir en dos fases, la fase de negociación y la fase de transmisión de datos.

2.2.15.2.1 Fase de negociación

La fase de negociación, handshake o también conocido como apretón de manos inicia el cliente al solicitar al servidor una comunicación segura enviándole un mensaje con parámetros como la versión del protocolo, una lista de algoritmos de cifrado y de igual forma una lista de algoritmos de compresión que el cliente soporta.

El servidor responde al cliente igualmente con un mensaje que contiene los siguientes parámetros: Un certificado otorgado por alguna Autoridad Certificadora (CA) y los algoritmos de cifrado y compresión seleccionados de la lista enviada por el cliente y adicionalmente envía la clave pública del servidor.

El cliente verifica el certificado enviado por el servidor, y si este es correcto responde con un mensaje que contiene la posible clave secreta para la transmisión de los datos, toda esta información se envía al servidor de forma cifrada con la clave pública del servidor.

Tanto el servidor como el cliente comparten la información que se ha enviado con la clave secreta acordada y están listos para transmitir los datos de manera segura mediante el uso del protocolo Record.

El algoritmo que se usa en la fase de negociación es asimétrico y se usa las claves públicas para cifrar la información y las claves privadas para descifrar en cada uno de los extremos.

Si la comunicación se suspende no es necesario realizar otra vez todo este proceso, únicamente se comprueba los datos de la sesión y la conexión actual.

2.2.15.2.2 Fase de transmisión de datos

El protocolo Record es el encargado de transmitir los datos entre el cliente y el servidor usando los parámetros acordados en la fase de negociación.

Para la transmisión de datos en este protocolo se usan algoritmos de encriptación simétricos con claves que van modificándose para cada siguiente paquete transmitido.

2.2.15.3 Fundamentos del Protocolo SSL

El Protocolo SSL para cumplir con su objetivo principal que es el de brindar una comunicación segura entre dos aplicaciones y en especial en una red tan grande e insegura como es la red del Internet se fundamenta en las siguientes tecnologías:

- Criptografía de claves simétricas y asimétricas
- Códigos de autenticación de mensajes (MACs).
- Certificados digitales X.509

2.2.15.4 Arquitectura del Protocolo SSL

El protocolo SSL está formado por cuatro sub-protocolos: El protocolo Handshake, protocolo Change Cipher Spec, protocolo Alert y el protocolo Record ubicado en el modelo TCP debajo de la capa de Aplicación y sobre la capa de TCP como indica el Gráfico II.2.

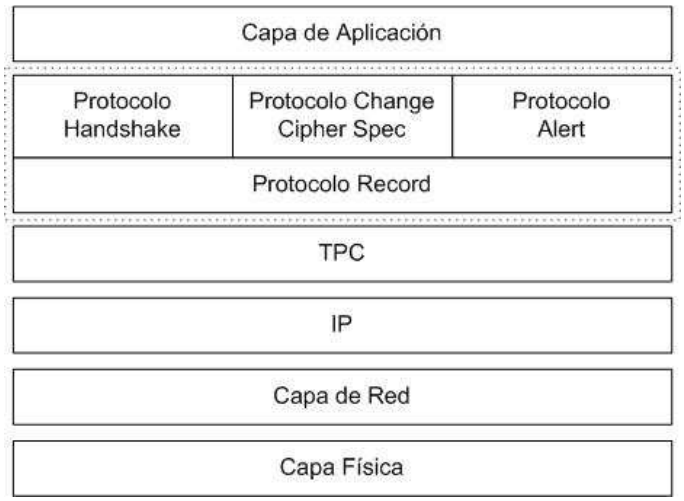

Gráfico II.2: Arquitectura del Protocolo SSL

El formato de los mensajes que utiliza el protocolo SSL tiene tres campos: Type, Lenght y Content.

Type: Es el tipo de mensaje (1 byte).

Length: Longitud del mensaje en bytes (3 bytes).

Content: Parámetros asociados al mensaje (>= 1 byte).

Gráfico II.3: Formato de los mensajes del Protocolo SSL

2.2.15.5 Protocolo SSL Handshake

El protocolo Handshake se encarga de la negociación para el establecimiento de parámetros antes de realizar una conexión segura, esa negociación consiste en una serie de mensajes enviados entre el cliente y el servidor en un dialogo de cuatro fases establecimiento de capacidades de seguridad, autenticación e intercambio de clave del servidor, autenticación e intercambio de clave del cliente y finalización en cual su objetivo es permitir al cliente y al servidor una Autenticación mutua y negociar los algoritmos

utilizados para calcular el MAC y el cifrado, Claves criptográficas utilizadas y parámetros adicionales.

2.2.15.5.1 Primera fase: Establecimiento de capacidades

En esta fase el cliente envia un mensaje *hola* al servidor con los datos para iniciar la conexión como la versión del protocolo soportada por el cliente, un numero aleatorio para proteger ataques de repetición, el id de la sesión el cual puede ser cero si desea establecer una nueva conexión, una lista de conbinaciones de algoritmos criptográficos soportados por el cliente en orden decreciente por preferencia esta lista contiene los algoritmos de intercambio de claves, algoritmos de cifrado, MAC y parametros de los algoritmos, también se envia una lista de los métodos de compresión soportados por el cliente.

El servidor responde con un mensaje *hola* que contiene la versión elegida por el servidor, un numero aleatorio independiente del enviado por el cliente, el id de la sesión que es igual que el del cliente, y los métodos de cifrado y encriptación seleccionados de la lista enviada por el cliente.

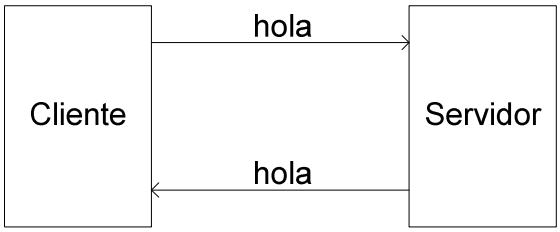

Gráfico II.4: Establecimiento de capacidades del Protocolo SSL Handshake

2.2.15.5.2 Segunda fase: Autenticación e intercambio de clave del servidor

El único mensaje obligatorio es el *hola mensaje terminado*, no tiene parámetros e indica el final de la segunda fase, después de enviarlo queda a la espera de respuesta del cliente.

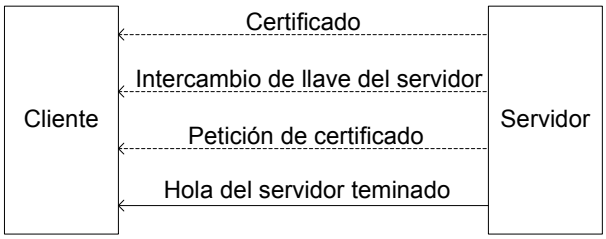

Gráfico II.5: Autenticación e intercambio de clave del servidor del Protocolo SSL Handshake

2.2.15.5.3 Tercera fase: Autenticación e intercambio de clave del cliente

Es el único mensaje obligatorio en esta fase es el *intercambio de llave de cliente*, El contenido depende del tipo de intercambio de clave según el algoritmo asimétrico utilizado.

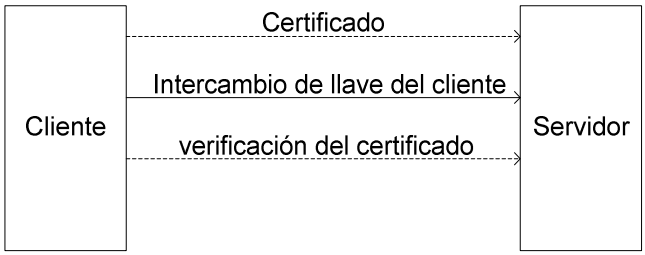

Gráfico II.6: Autenticación e intercambio de clave del cliente del Protocolo SSL Handshake

2.2.15.5.4 Cuarta fase: Finalización

El cliente envía dos mensajes al servidor: especificación de cambio de cifrador, terminado.

El primer mensaje sirve para cambiar de estado a la conexión de pendiente a listo. El segundo mensaje sirve para comprobar que el intercambio de clave y proceso de autenticación se realizaron con éxito.

A partir de este momento se puede comenzar a intercambiar información del nivel de aplicación.

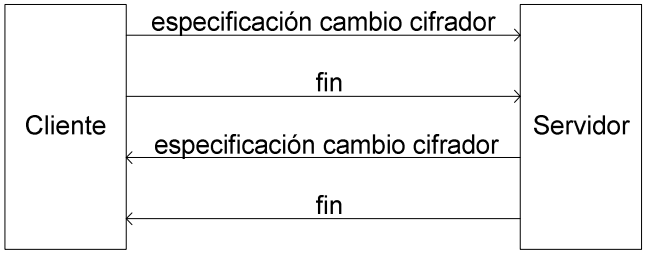

Gráfico II.7: Finalización del Protocolo SSL Handshake

2.2.15.6 Protocolo SSL Change Cipher Spec

Este es uno de los tres protocolos específicos utilizados por el protocolo SSL Record, consiste en transmitir un byte con valor 1, esto hace que en el receptor copie el valor pendiente de CipherSpec al valor actual y cambie el estado de la conexión de pendiente a activo.

2.2.15.7 Protocolo SSL Alert

Este protocolo comunica mensajes de alerta al otro extremo de la línea de la comunicación, es decir comunicara de los errores en el cliente al servidor y viceversa.

Los mensajes de alerta tienen dos parámetros: Nivel de gravedad y tipo de alerta.

Nivel de gravedad: Puede tomar dos valores: 1 que representa una advertencia, y 2 que representa un error fatal.

Tipo de alerta: Los tipos de alerta pueden ser los siguientes:

- unexpected_message (F): Recibe mensaje incorrecto.
- bad_record_mac (F): Recibe MAC incorrect.
- descomression_failure (F): Demasiado largo, error al descomprimir.
- handshake_failure (F): No hay acuerdo de negociación de parámetros.
- illegal_parameter (F): Campo handshake fuera de rango o inconsistente.
- close_notify(W): El transmisor notifica que no enviará mas mensajes. Se utiliza para cerrar la conexión en los dos sentidos (es enviado por ambas partes).

- no_certificate (W): No está disponible un certificado apropiado.
- bad_certificate (W): Certificado recibido está corrupto.
- unsuported_certificate (W): Tipo de certificado recibido no soportado.
- certifícate_revoked (W): Certificado ha sido revocado por el firmante.
- certifícate_expired (W): Certificado ha expirado.
- certifícate_unknown (W): Otros errores relativos a certificados.

2.2.15.8 Protocolo SSL Change Cipher Spec

Este es uno de los tres protocolos específicos utilizados por el protocolo SSL Record, consiste en transmitir un byte con valor 1, esto hace que en el receptor copie el valor pendiente de CipherSpec al valor actual y cambie el estado de la conexión de pendiente a activo.

2.2.15.9 Protocolo SSL Record

El protocolo SSL Record proporciona dos servicios a las conexiones Confidencialidad e Integridad. Confidencialidad pues el protocolo Handshake define una clave secreta compartida usada para cifrado convencional de los datos. Integridad porque el protocolo Handshake define una clave secreta compartida usada para generar un Código de Autenticación de un Mensaje (MAC).

El protocolo Record de SSL toma los datos de la aplicación a enviar y los fragmenta en bloques manejables de una longitud máxima de 16384 bytes, luego de manera opcional comprime los bloques antes fragmentados, los bloques pequeños no deben incrementar en más de 1024 bytes. A este bloque añade un MAC utilizando un algoritmo similar a HMAC. Luego de realizar este proceso cifra todo el bloque resultante, es decir el bloque comprimido y el MAC, a esto se le añade una cabecera para finalmente introducir el bloque resultante en un segmento TCP y enviarlo. EL receptor realiza el proceso inverso, primeramente descifra, luego verifica la integridad del mensaje utilizando el MAC para posteriormente descomprimir y ensamblar los bloques.

2.2.15.9.1 Algoritmo generador de MAC

El algoritmo que genera el MAC es un hash que puede ser MD5 o SHA-1.

2.2.15.9.2 Cifrado de bloque

En el cifrado de bloque no puede incrementar la longitud más de 1024 bytes.

Los algoritmos permitidos para realizar el cifrado en bloque en este protocolo son los siguientes:

- IDEA con clave de 128 bits.
- RC4 con clave de 40 bits.
- DES con clave de 40 y 56 bits.
- 3DES con clave de 168 bits (tres claves de 56 bits).
- RC4 con clave de 40 y 128 bits.

Antes de cifrar puede ser necesario añadir un relleno para cumplir con la condición que la longitud total del paquete debe ser múltiplo de la longitud del bloque cifrado.

El relleno de n más un bytes debe ser igual a bytes de relleno mas la longitud del relleno.

Longitud total es igual a longitud de los datos más longitud del MAC más Longitud del relleno mínimo.

2.2.15.9.3 Cabecera del protocolo SSL Record

La cabecera del protocolo SSL Record tiene los siguientes campos: Content type, Major versión, Minor Version y Compressed Length.

Content type*:* Este campo tiene una longitud de 8 bits, Este campo indica el protocolo de alto nivel usado para procesar el bloque, puede tomar los siguientes valores.

- change_cipher_spec: protocol SSL
- alert: protocolo SSL
- handshake: protocolo SSL

- application_data: HTTP, FTP, etc.

Major versión**:** Este campo tiene una longitud de 8 bits e indica la versión mayor de SSL en uso, para SSL v3 el valor es 3.

Minor versión: Este campo tiene una longitud de 8 bits e indica la versión de SSL en uso, para SSL v3 su valor es 0.

Compressed Length: Este campo tiene una longitud de 16 bits e indica la longitud del bloque comprimido o no comprimido.

2.2.15.9.4 Formato del protocolo SSL Record

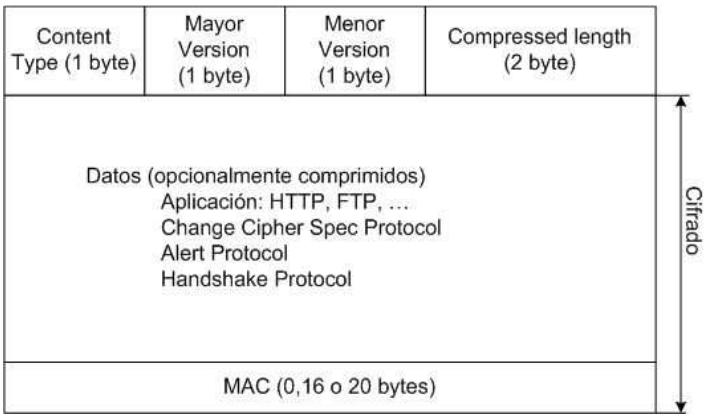

Gráfico II.8: Formato del Protocolo SSL Record

2.2.16. Protocolo TLS

2.2.16.1 Introducción

El protocolo TLS surge como necesidad de estandarizar un protocolo que provea de seguridad entre el cliente y el servidor a través del internet, debido a que el protocolo SSL es un protocolo creado y patentado por Netscape.

Es por ello que la IETF (Internet Engineering Task Force) define y describe en 1999 el protocolo denominado TLS.

Básicamente el protocolo TLS en su versión 1.0 tiene un funcionamiento muy similar al protocolo SSL versión 3.0, en muchos documentos se cita que el protocolo TLS 1.0 es un protocolo SSL 3.1.

El principal objetivo del protocolo TLS es proporcionar privacidad e integridad de los datos entre las 2 partes (Cliente/Servidor). El protocolo TLS al igual que el protocolo SSL está estructurado por 3 capas: la capa de protocolo TLS Record, la capa de protocolo TLS handshake y la capa Protocolo de Alerta.

El protocolo TLS inicia en 1999 con la versión 1.0, en el año 2006 se lanza la versión TLS 1.1 que es actualmente la más utilizada en las diversas aplicaciones como GnuTLS, y en Agosto de 2008 es lanzada la versión TLS 1.2 pero actualmente no es muy utilizada y en aplicaciones como GnuTLS viene deshabilitado por defecto.

Siendo una evolución del protocolo SSL 3.0 utiliza un sistema criptográfico mixto basado en lo que se denomina como cifrado simétrico y cifrado asimétrico, además soporta menos tipos de certificados x.509.

2.2.16.2 Funcionamiento Básico

Para iniciar la transmisión de datos entre el Cliente y el Servidor, primero negocien los parámetros para el intercambio de datos.

2.2.16.3 Arquitectura

El protocolo TLS principalmente se compone de 3 capas: la capa del Protocolo TLS Record, la capa del Protocolo TLS Handshake y la capa del protocolo el Protocolo de Alerta. Estas capas interactúan entre sí.

Para una mejor compresión de las capas del protocolo TLS se las puede representar en el gráfico II.9.

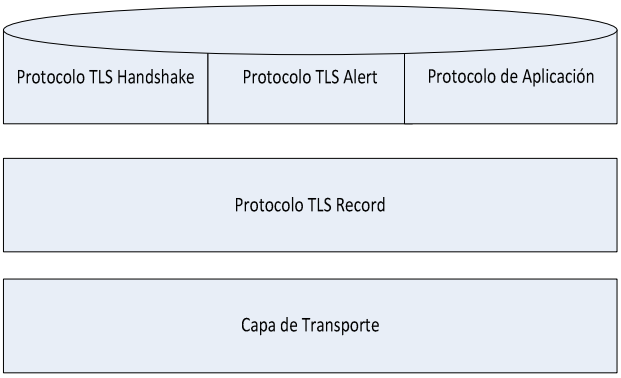

Gráfico II.9: Arquitectura del Protocolo TLS

2.2.16.4 Protocolo TLS Record

El protocolo TLS Record es utilizado por los protocolos que se encuentran en la capa de Transporte. El protocolo TLS Record ofrece encriptación simétrica, autenticidad de datos y compresión que puede ser utilizado de manera opcional.

El protocolo TLS Record es el que proveedor de la comunicación segura. El propósito de esta capa es cifrar, autenticar y opcionalmente comprimir los paquetes.

El protocolo TLS Record tiene inicialmente parámetros Nulos, lo cual nos indica que no existe encriptación y que el MAC no está siendo utilizado. La encriptación y la autenticación inicia justo después que la capa del protocolo handshake ha finalizado su negociación.

2.2.16.4.1 Algoritmos de Encriptación utilizados en la capa del Protocolo TLS Record

Confidencialmente la capa de Protocolo TLS Record utiliza algoritmos de encriptación de bloques simétricos como 3DES, AES, o algoritmos stream como ARCFOUR_128.

Para el cifrado se utiliza algoritmos de encriptación que utilizan una sola clave secreta, tanto para cifrar como para descifrar los datos. Así ha utilizado el protocolo TLS un número aleatorio de bloques que son añadidos a los datos, para prevenir el eavesdropping.

Este protocolo soporta los siguientes algoritmos de cifrado:

- 3DES_CBC 3DES_CBC es el algoritmo de cifrado de bloque utilizado con triple encriptación.
- ARCFOUR_128 el algoritmo arcfour es el más rápido para cifrar un stream.
- ARCFOUR_128 está conformado con una clave de 40 bits que se lo puede considerar como débil.
- AES_CBC el algoritmo AES o RIJNDAEL es un algoritmo de cifrado de bloque que reemplaza al caduco algoritmo DES. Tiene un tamaño de bloque de 128 bits y utiliza en modo CBC. Aunque esto no es oficialmente soportado por el protocolo TLS.

Y los siguientes algoritmos MAC

- MAC_MD5 el algoritmo MD5es un algoritmo hash diseñado por Ron Rivest. Contiene una salida de datos de 128 bits.
- MAC_SHA el algoritmo SHA es un algoritmo diseñado por la NSA. Contiene una salida de datos de 160 bits.

2.2.16.4.2 Algoritmos de Compresión Utilizados en la capa del protocolo TLS Record

La capa del protocolo TLS Record soporta también algoritmos de compresión.

En esta capa se incluyen algoritmos de compresión realmente buenos, cuando se transmite texto plano u otro formato que sea compatible su compresión, estos algoritmos de compresión son muy útiles en túneles con alta banda ancha en TLS, y en casos donde el tráfico de la red tiene que ser optimizado.

Los algoritmos de compresión que soporta esta capa son los siguientes:

- DEFLATE
- LZO

2.2.16.5 Debilidades del Protocolo TLS

Algunas debilidades que pueden afectar la seguridad de la capa del protocolo TLS Record han sido encontradas en la versión 1.0 del Protocolo TLS. Estas debilidades pueden ser aprovechados por atacantes que:

- El protocolo TLS ha separado las alertas para las librerías “decryption_failed” y “bad_record_mac”
- La razón de la falla de la desencriptación puede ser detectado midiendo los tiempos.
- El cuarto paquete que se encripta es el último bloque del paquete previo encriptado.

2.2.16.6 Protocolo TLS Alert

El protocolo TLS Alert ofrece señalización a los otros 2 protocolos. Puede informar las causas de las fallas y otros errores en el funcionamiento del protocolo TLS. El protocolo de Alerta se encuentra una capa superior a la capa del protocolo TLS Record.

El protocolo TLS Alert permite enviar señales. Estas señales son utilizadas para informar acerca de las fallas en el funcionamiento del protocolo. Algunas de estas señales son utilizadas internamente por el protocolo y la capa del protocolo de aplicaciones y otras señales se refieren solamente a la capa del protocolo de aplicaciones. Una alerta o aviso incluye en un parámetro en el cual indica el nivel de riesgo, que un error puede ser fatal o simplemente una advertencia.

Los errores fatales siempre terminan la conexión, así está información no se filtra. Ud. Debe tomar extremos cuidados para que la información no se filtre a un potencial atacante, como son los archivos log del sistema.

2.2.16.7 Protocolo TLS Handshake

Es protocolo Handshake es el responsable de administrar los parámetros de negociación de seguridad, aquí se efectúa el intercambio de claves y la autenticación.

El protocolo TLS Handshake es el responsable de la negociación para el conjunto de cifrado, para el intercambio inicial de claves y la autenticación de las 2 partes. Esto es controlado por la capa de aplicación, así que el programa de capa superior tiene que configurar los parámetros requeridos.

CAPÍTULO III ANALISIS COMPARATIVO

3.1 Introducción

En el presente capítulo se realiza un estudio previo de cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS para tener una idea clara y precisa de su arquitectura, de los protocolos implementados y de los algoritmos que utilizan para que de esta manera se presente de una manera organizada la información de las herramientas.

También se determina y se detalla los parámetros e indicadores que serán el punto de partida para ejecutar el análisis comparativo y determinar la herramienta más idónea para la puesta en práctica de la parte aplicativa del presente proyecto.

Para llevar a cabo el análisis comparativo se construirá módulos de prueba, que consisten en entornos definidos y descritos por los desarrolladores del presente proyecto y servirán como sustento para la calificación que les sea asignada a cada una de las herramientas.

Se obtendrán los resultados finales a partir de la tabulación de la calificación asignada a cada uno de los indicadores de los parámetros de evaluación de las herramientas, se hará uso también de gráficos estadísticos que determinarán a simple vista la herramienta más idónea para la puesta en marcha de la parte aplicativa del presente proyecto.

3.2 Herramientas de administración de funciones criptográficas a comparar.

- OpenSSL
- GnuTLS
- NSS

3.3 Análisis de la Herramienta de Administración de Funciones Criptográficas OpenSSL.

3.3.1. Introducción

El proyecto OpenSSL es un esfuerzo conjunto para el desarrollo de una implementación robusta y segura de nivel comercial, con todas las características y de tipo Open Source de los protocolos SSL y TLS, además de bibliotecas con propósitos de cifrado. Su principal diferencia con un paquete SSL se puede apreciar en el eslogan presente en su página web: "¿Porque comprar un paquete SSL como una caja negra, cuando puede obtener una abierta y gratis?".

OpenSSL es una implementación Open Source de los protocolos SSL y TLS y librerías de cifrado para desarrollo de algoritmos de cifrado y descifrado, certificados x.509, firmas digitales además ofrece la posibilidad de programar con C/C++ aplicaciones de seguridad y autoridades de certificación utilizando OpenCA. La aplicación OpenSSL es multiplataforma adaptándose a una gran variedad de sistemas operativos Unix, Linux, Windows, etc.

La principal herramienta de OpenSSL es la herramienta de línea de comandos openssl que nos permite administrar autoridades certificadoras, certificados digitales, claves públicas y claves privadas. La herramienta openssl nos provee de una gran cantidad de comandos y opciones para el desarrollo de las funciones mencionadas anteriormente.

3.3.2. Funcionalidad

El proyecto OpenSSL implementa los protocolos SSL (SSLv2 y SSLv3) y TLS v1. OpenSSL está configurado por defecto para utilizar el protocolo SSLv3, las demás versiones y protocolos pueden ser configurados en sus respectivos archivos de configuración.

OpenSSL transmite la información cifrada haciendo uso de los algoritmos de cifrado simétricos como son: Blowfish, Camellia, DES, RC2, RC4, RC5, IDEA, AES. De igual forma OpenSSL tiene una configuración por defecto que puede ser cambiada manualmente o automáticamente al momento de establecer un canal seguro de comunicación.

Los algoritmos asimétricos funcionales en el proyecto OpenSSL son: RSA, DSA, Diffie-Hellman key Exchange. Estos algoritmos permiten intercambiar la clave privada para los algoritmos de cifrado asimétricos de forma segura.

OpenSSL para la administración de firmas y certificados digitales implementa las funciones hash como: MD5, MD2, SHA, MDC-2.

Para utilizar las funciones mencionadas anteriormente la herramienta OpenSSL tiene implementado una herramienta de línea de comandos llamada openssl, esta herramienta tiene a su vez una amplia lista de comandos y opciones para cada una de las funciones.

3.3.3. Arquitectura

La herramienta OpenSSL consta de cuatro subprogramas y librerías: c_rehash, openssl, y las librerías libcrypto y libssl.

3.3.3.1 C_rehash

Es un guión perl que revisa todos los ficheros de un directorio y añade enlaces simbólicos apuntando a sus valores aleatorios (hash values).

3.3.3.2 Openssl

Es un programa de línea de comandos para utilizar varias funciones de cifrado de la librería “crypto” de OpenSSL. La herramienta openssl puede ser utilizada para:

- Creación y administración de claves privadas, claves públicas y parámetros.
- Operaciones de cifrado con claves públicas.
- Creación de certificados X. 509, CSRs y CRLs.

- Encriptación y desencriptación con cifrado.
- Calculo de clave de mensajes.
- Test de cliente y servidor con SSL/TLS.
- Administración de firmas digitales para encriptar e-mails con S/MIME.
- Control de tiempo de respuesta, generación y verificación.

3.3.3.3 Libcrypto

Implementa una amplia gama de los algoritmos de cifrado utilizado en varios estándares de Internet. Los servicios facilitados por esta librería son utilizados por las implementaciones OpenSSL de los protocolos SSL, TLS y S/MIME, y también están siendo utilizados para implementar en SSH, OpenPGP, y otros estándares cifrado.

Libcrypto está compuesta por un número de sub-librerías que implementan los algoritmos en forma general, la funcionalidad incluye encriptación simétrica y asimétrica, administración de certificados, cifrado de funciones hash y cifrado para generación de números pseudo-aleatorios.

Librería SSL: Implementa los protocolos SSL (SSL v2/v3) y TLS v1. Posee una completa API.

3.3.4. Instalación

La herramienta OpenSSL viene instalada por defecto en la distribución de Centos5 con la versión openssl-0.9.8b-8.3.el5. Si se desea trabajar con una versión actualizada es necesario usar la herramienta *yum* con la siguiente sintaxis:

```
yum update openssl.
```

Para actualizar desde un archivo *rpm* ejecutamos el siguiente comando:

```
rpm –Uvh openssl-version-actual.
```

También es posible realizar la compilación del código fuente mediante el archivo *config* que viene en la distribución de *OpenSSL-versión-actual.tar.gz* y realizar una instalación personalizada:

```
tar –xzvf openssl-version-actual.tar.gz
cd openssl-version-actual
./config
make
make test
make install
```

3.3.5. Configuración

La configuración por defecto de la herramienta OpenSSL contiene todas las variables con valores respectivos para un correcto funcionamiento, no es necesario modificar estos valores.

El archivo de configuración se encuentra en la siguiente dirección:

```
/etc/pki/tls/openssl.cnf
```

Si la instalación es realizada compilando el código fuente y no se modifica la variable *prefix*, el archivo de configuración se encuentra en:

```
/usr/local/ssl/openssl.cnf
```

3.3.6. Protocolos Implementados

El paquete de herramientas OpenSSL soporta los siguientes protocolos:

- SSL V 2.0.
- SSL V 3.0.
- TLS V 1.0.

3.3.7. Algoritmos de cifrado

Los algoritmos de clave privada (simétricos) que utiliza OpenSSL para el cifrado de la información son:

- Blowfish.
- DES.

- CAST.
- IDEA.
- RC2.
- RC4.
- RC5.
- Camellia

Los algoritmos de clave pública (asimétricos) son:

- RSA.
- DSA.
- Diffie-Hellman key Exchange.

Las Funciones Hash que se utilizan para firmar los certificados digitales son:

- HMAC.
- MD2.
- MD4.
- MD5.
- MDC2.
- RIPEMD.
- SHA.

3.3.8. Portabilidad

OpenSSL está diseñado para ser instalado en diferentes plataformas como son:

Plataformas derivadas de Unix como Linux.

Plataformas Win32

Plataformas OpenVMS

3.3.9. Usabilidad

El proyecto OpenSSL además de presentar una herramienta muy amigable llamada *OpenSSL* que se ejecuta en línea de comandos con una gran cantidad de opciones y parámetros para la gestión de certificados, establecimiento de la comunicación y la transmisión de la información de forma segura también nos facilita el uso de sus librerías para utilizar en otros programas:

- Apache-ssl: Integración de SSL para apache por Ben Laurie.
- Apache mod_ssl: Integración de SSL para apache por Ralf S. Engelschall.
- pkcs12: Una utilidad para el formato PKCS#12 por Stephen Henson.
- OpenCA: Propone una CA basado en OpenSSL.
- CashCow: Transacciones de pagos por internet via OpenSSL.
- pyCA: Implementa una CA implementado con Phyton y OpenSSL.
- OpenSSH: Un puerto de OpenSSL basado en SSH derivado de OpenBSD.

3.3.10. Soporte Técnico

En el sitio oficial podemos encontrar direcciones individuales de los desarrolladores y direcciones de cada grupo de desarrolladores disponibles para brindar ayuda a los usuarios de este proyecto tanto para usuarios comerciales o no comerciales.

Además existe en muchos blogs de ayuda tanto en inglés como en español.

3.3.11. Situación Legal

OpenSSL se basa en la librería SSLeay desarrollada por Eric A. Young and Tim J. Hudson. El paquete de herramientas OpenSSL es licenciado bajo los términos de la licencia Apache-style, la cual significa básicamente que usted es libre de obtener y usar con fines comerciales y no comerciales sujetos a condiciones simples de la licencia.

3.4 Análisis de la Herramienta de Administración de Funciones Criptográficas GnuTLS.

3.4.1. Introducción

GnuTLS es un proyecto desarrollado como un conjunto de librerías que provee una capa segura. Este conjunto de librerías podría ser utilizada por cualquier aplicación cliente servidor que lo requiera, sin embargo debe aceptar las licencias bajo las cuales se encuentra desarrollado el proyecto.

Sin embargo el proyecto tiene su propia implementación la cual se denomina gnutls que provee la implementación sobre el protocolo TLS versión 1.1, administración de certificados x.509 y OpenPGP.

GnuTLS tiene 3 partes que trabajan independientemente entre sí: la parte del Protocolo TLS, la parte del Cifrado y la parte de la administración de los certificados.

3.4.2. Funcionalidad

GnuTLS contiene un estado global que inicia una vez que se inicializan las funciones globales. La estructura global también contiene funciones que se utilizan para asignar memoria, y algunas estructuras que necesitan la librería ASN.1.

La estructura de las credenciales es utilizada para los métodos de autenticación, como la autenticación de certificados, entre otras cosas contiene: claves privadas que son solo conocidas por el cliente que solicita y el servidor que responde, certificados, intercambio de claves públicas entre el cliente y el servidor para establecer un hasdshake o acuerdo, etc.

Una sesión en GnuTLS contiene todo lo necesario para realizar la administración de una conexión segura, como las principales librerías que se necesitan. Esta sesión tiene un único ID. Los ID de sesiones necesitan un motor de base de datos para guardar los parámetros de sesión. Ver Gráfico III.1.

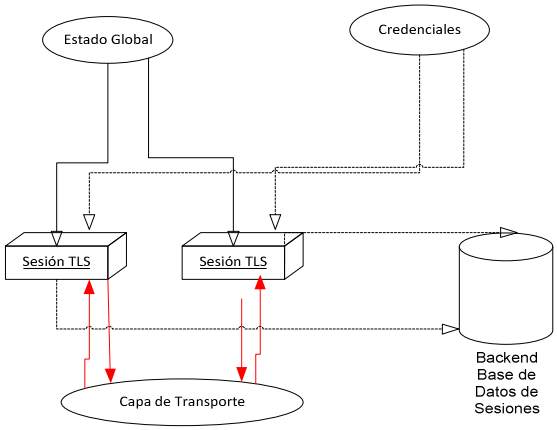

Gráfico III.1: Funcionamiento de la herramienta GnuTLS

3.4.2.1 Manejo de Errores

En GnuTLS las funciones retornan un valor entero. Una función se ha ejecutado correctamente cuando retorna un valor positivo o el valor cero. Pero cuando retorna un valor negativo quiere decir que se ha producido un error, el cual puede ser fatal (interrumpiendo la ejecución de GnuTLS) o simplemente terminando la función incorrectamente ejecutada.

Los errores que son fatales terminan la conexión inmediatamente y envían y reciben mensajes notificando que han sido deshabilitados.

GnuTLS maneja de diferente manera la asignación de objetos, dependiendo del grado de criticidad de los datos que contienen. Sin embargo por motivos de rendimiento las funciones de la memoria predefinidas no borran los datos sensibles de la memoria, ni protege los objetos que están siendo escritos a la swap.

La librería Libgcrypt de la cual depende GnuTLS tiene una asignación segura de las funciones disponibles. Estas deberían ser utilizadas en caso de que no se consideren segura la asignación de memoria swap.

3.4.3. Arquitectura

GnuTLS se puede entender básicamente como una librería que ofrece una interfaz (API) para el acceso a los protocolos de comunicación segura, en el caso particular los protocolos SSL 3.0 y TLS1.1. A esto se suma un framework para autenticación y para la generación de claves públicas.

Entre las principales características que nos ofrece la librería GnuTLS están:

- Soporte para los protocolos SSL y TLS.
- Soporte para certificados X.509 y OpenPGP.
- Soporte para administración y verificación de certificados.
- Soporte para autenticación SRP para el protocolo TLS.
- Soporte para autenticación PSK para el protocolo TLS.
- Soporte para mecanismos de extensión para el protocolo TLS.
- Soporte para métodos de Compresión para el protocolo TLS.

GnuTLS está conformado, esencialmente de 3 partes ver Gráfico III.2: la parte del Protocolo TLS, la parte de los Certificados y la parte del backend de Cifrado, una ilustración que nos puede ayudar a visualizar se presenta a continuación:

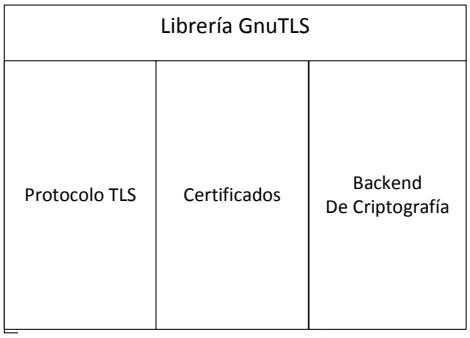

Gráfico III.2: Arquitectura de la herramienta GnuTLS

La parte del Protocolo TLS: es la implementación del protocolo TLS 1.1 que se encuentra totalmente implementada dentro de la librería GnuTLS.

La parte de los Certificados: consiste en la creación de los certificados y la verificación de funciones que se encuentra parcialmente implementado en la librería GnuTLS y la otra parte en la librería "Libtasn1" de ASN.1.

La parte del backend de Cifrado: la cual está implementada en la librería "Libgcrypt".

3.4.4. Instalación

El sistema operativo Centos5 instala por defecto la herramienta GnuTLS con la versión gnutls-1.4.1-2, pero únicamente se instala las librerías y las cabeceras.

Para actualizar esta instalación se puede usar la herramienta *yum* con la siguiente sintaxis:

```
yum update gnutls.
```

Para actualizar desde un archivo *rpm* ejecutamos el siguiente comando:

```
rpm –Uvh gnutls-version-actual.
```

Para realizar una instalación personalizada es necesario realizar la compilación del código fuente el cual se puede obtener desde http://www.gnutls.org/download.html

GnuTLS necesita de las librerías libgcrypt y libgpg-error, y opcionalmente de las librerías libz y lzo.

El archivo gnutls-version.tar.bz2 de la versión más actual se copia en el directorio `/usr/local` y posteriormente se ejecuta los comandos.

```
tar -xjvf gnutls-version.tar.bz2
cd gnutls-version.tar.bz2
./configure
make
make check
make install
```

La instalación por defecto se realiza bajo `/usr/local/lib`, `/usr/local/include`, `/usr/local/bin`, etc. Esta configuración podemos modificar con las opciones de --

prefix=PREFIX donde PREFIX es una variable que contiene el directorio bajo el cual se desea instalar.

Para mayor información de las opciones de *configure* digitar:

```
./configure –help
```

3.4.5. Configuración

La configuración de la herramienta GnuTLS viene realizada por defecto y únicamente se puede observar sus parámetros mediante las herramientas libgcrypt-config para la librería libgcrypt, libgnutls-config para gnutls.

3.4.6. Protocolos Implementados

GnuTLS soporta varios protocolos entre ellos:

- SSL 3.0 implementado en 1996, no soporta el protocolo SSL 2.0 por motivos de seguridad.
- TLS 1.0 implementado en Enero de 1999, es muy similar al protocolo SSL 3.0.
- TLS 1.1 implementado en Abril de 2006, es el protocolo actualmente implementado por defecto
- TLS 1.2 implementado en Agosto de 2008, soporta este protocolo pero viene deshabilitado por defecto por su falta de difusión.

3.4.7. Algoritmos de cifrado

En GnuTLS están involucrados algunos algoritmos para el intercambio de claves, dependiendo del tamaño de las claves GnuTLS soporta los siguientes algoritmos asimétricos:

- Anon-RSA.
- RSA.
- RSA EXPORT.
- DHE-RSA.

- DHE-DSS.
- SRP-DSS.
- SRP-RSA.
- SRP.
- PSK.
- DHE-PSK.
- ECC.

Los algoritmos simétricos que soporta son los siguientes:

- AES-256 CBC.
- AES-128 CBC.
- 3DES CBC.
- DES CBC.
- RC4-128 CBC.
- RC4-40.
- RC2-40.
- Camellia.

Las funciones hash que soporta GnuTLS son los siguientes:

- Mac_md5.
- mac_sha.

3.4.8. Portabilidad

GnuTLS está diseñado para ser instalado en diferentes plataformas como son:

Plataformas derivadas de Unix como Linux.

Plataformas Win32.

3.4.9. Usabilidad

Principalmente el conjunto de librerías gnutls es utilizado por aplicaciones extendidas al proyecto mismo, entre las aplicaciones más importantes se encuentran las siguientes:

- Hydra: Servidor http de gran velocidad de respuesta.
- QStream: Sistema de investigación para vídeo de alta calidad.
- Exim: Agente de envío de mensaje.
- Dovecot: Sistema de servidores de correo POP3 e IMAP.
- Retawq: Navegador Web interactivo para terminales de texto.
- Evolution: Cliente mail.
- Centericq: cliente de mensajerías instantáneo que soporta varios protocolos.
- Lynx: Navegador web de texto.
- Tntnet: Servidor de aplicaciones web escritas en C++.

3.4.10. Soporte Técnico

El proyecto GnuTLS es desarrollado por la comunidad, y todos los desarrolladores son bienvenidos para contribuir bajo ciertas condiciones, para ello se puede contactar a la Free Software Foundation su mail de contacto es el siguiente: gnu@gnu.org.

Pero algunas compañías ofrecen los servicios de soporte técnico pero con un costo económico dependiendo de la compañía contratada. Una compañía reconocida que brinda soporte es “Simon Josefsson Datakonsult” de Suecia su web es: http://josefsson.org/ y su mail de contacto: simon@josefsson.org.

La información que es enviada al proyecto no se la puede comprobar, el proyecto GnuTLS presenta la información como la recibe y no garantiza que sea correcta. También contiene un listado general de las personas que han contribuido y no se lista las habilidades específicas de las personas que han contribuido. Si quien necesita soporte técnico necesita contactar a una persona se proporciona su modo de contacto y la decisión es de quien lo contacta.

3.4.11. Situación Legal

Algunas de las librerías de GnuTLS se encuentra licenciado bajo GNU General Public License y otra parte de las librerías con la licencia GNU Lesser General Public License.

3.5 Análisis de la Herramienta de Administración de Funciones Criptográficas NSS.

3.5.1. Introducción

NSS es un proyecto de código abierto desarrollado por la Fundación Mozilla cuyo principal objetivo es desarrollar un conjunto completo de librerías que implemente los protocolos SSL, TLS y S/MIME. Además de una amplia variedad de algoritmos de cifrado simétricos, asimétricos y funciones hash para brindar seguridad a aplicaciones cliente servidor.

NSS tiene un conjunto de librerías compartidas que implementan los protocolos, los algoritmos y funciones hash de forma totalmente independiente. NSS incorpora herramientas que ayudan a la administración de las distintas funcionalidades que NSS implementa. Además NSS comparte sus librerías para el desarrollo de aplicaciones extendidas.

3.5.2. Funcionalidad

NSS es un conjunto de librerías open source que provee de herramientas que facilitan la implementación de seguridad en aplicaciones cliente servidor, NSS tiene un conjunto de herramientas que facilitan la administración de certificados digitales, firmas digitales, administración de claves públicas y privadas entre las más importantes:

certutil 2.0: Administra la base de datos de certificados y claves (cert7.db y key3.db).

cmsutil 1.0: Ejecuta operaciones básicas CMS como cifrado, descifrado y firma de mensajes.

crlutil: Administra la lista de certificados revocados CRLs.

dbck 1.0: Analiza y repara la base de datos de certificados.

modutil 1.1: Administra la base de datos de los módulos PKCS11 (secmod.db). Añade módulos y modifica las propiedades de módulos existentes.

pk12util 1.0: Importa y exporta certificados y claves entre la base datos de certificados y claves, y archivos en formato PKCS12.

signtool 1.3: Crea firmas digitales.

signver 1.1: Verifica las firmas en los objetos firmados digitalmente.

ssltap 3.2: Realiza una solicitud al proxy por un servidor SSL y despliega el contenido del mensaje intercambiado entre el cliente y el servidor. La herramienta ssltap no desencripta los datos, pero si muestra los mensajes clienteHello, serverHello, etc. y los datos de conexión como la versión del protocolo, serie de cifrado, etc. Esta herramienta en muy utilizada para depuraciones.

3.5.2.1 Manejo de errores

NSS provee el manejo de errores desde la instalación, al momento de compilar el código podemos testear antes de ubicar las librerías, los include y los ejecutables podemos hacer uso de los test que NSS nos provee.

Luego al momento de utilizar las herramientas podemos observar los archivos logs y de acuerdo los errores registrados podemos observar la base de datos de errores de NSS la descripción de cada uno de ellos, también tenemos una herramienta ssltap que se encarga de descomponer el mensaje TCP para poder observar únicamente los registros SSL y handshake.

3.5.2.2 Manejo de Memoria

NSS administra la memoria mediante la herramienta PLArenPool de NSPR, PLArenaPool llama a cada bloque de memoria como PLArena, PLArenaPool registra la memoria libre en listas para su fácil distribución al momento de que el programa solicite asignar más memoria.

El uso de esta herramienta trae consigo dos consecuencias que se analizan a continuación:

Al finalizar la ejecución del programa, todos los bloques de memoria de la lista simplemente podrían desaparecer, esto dificulta para decir si los bloques de memoria están perdidos o están en la lista libre.

Las herramientas de análisis de escape podrían frecuentemente reportar el error de la llamada para la asignación de los bloques de memoria desaparecidos.

Para resolver este problema que habitualmente hace que NSS sea más lento es necesario en la función main() poner la variable NSS_DISSABLE_ARENA_FREE_LIST=1.

3.5.3. Arquitectura

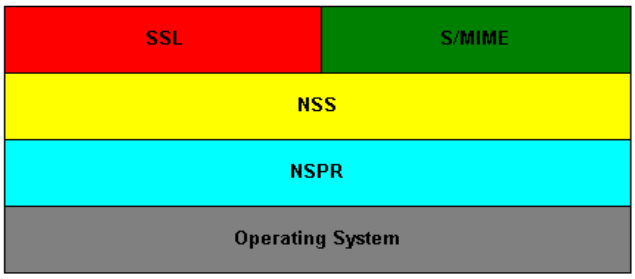

Gráfico III.3: Arquitectura de la herramienta NSS

NSS tiene un conjunto de librerías estáticas y compartidas, las librerías compartidas que exportan funciones públicas son:

3.5.3.1 Librería SSL

Soporta el núcleo de las operaciones SSL, implementa los protocolos SSL v2 y v3, y TLS. El protocolo SSL permite una autenticación mutua entre el cliente y el servidor, y el establece una conexión autentificada y cifrada. SSL se ejecuta sobre el protocolo TCP/IP y bajo los protocolos HTTP, IMAP, LDAP, NNTP.

3.5.3.2 Librería S/MIME

Soporta el núcleo de las operaciones S/MIME, provee de funcionalidades S/MIME mediante APIs que pueden ser integradas con una cantidad de generadores y modificadores MIME, podría soportar las características de S/MIME versión 3.

3.5.3.3 Librería NSS

Soporta el núcleo de las operaciones de cifrado, provee de funcionalidades para utilizar en aplicaciones que soporta SSL como: funciones de certificado, funciones de cifrado y otras funciones.

3.5.3.4 NSPR

Estas tres librerías que componen NSS utilizan el NSPR (Netscape Portable Runtime) el mismo que permite la portabilidad de NSS en varios sistemas operativos.

3.5.4. Instalación

La herramienta NSS viene instalada por defecto en la distribución de Centos5, y para ser actualizada o para realizar otra instalación desde los repositorios simplemente es necesario utilizar la herramienta YUM, o si deseamos un paquete RPM podemos obtener desde los instaladores de la distribución Centos5.

3.5.3.5 Dependencias

La herramienta *gmake* con la versión 3.75 o mayor

La herramienta *perl* con la versión 5.003 o mayor

Para actualizar o instalar desde las herramientas YUM o RPM

```
yum install nss

yum update nss

rpm –ivh nss-3.11.5-1.el5.i386.rpm
```

```
rpm –Uvh nss-3.11.5-1.el5.i386.rpm
```

3.5.3.5.1 Para instalar desde compilado el código fuente

El primer paso es descargarse el paquete *nss-3.12-with-nspr-4.7.tar.gz* desde el sitio oficial de NSS, luego descomprimir bajo el directorio del usuario. Este paquete además del NSS también tiene adjunto el NSPR que es un framework que le permite al NSS ser multiplataforma.

Luego se ubica bajo el siguiente path `/root/nss-3.12/mozilla/security/nss` y se ejecuta el siguiente comando `gmake nss_build_all`

NSS no se instala de la misma forma que otras herramientas a pesar de tener los archivos *Makefile*, NSS compila el código y crea una carpeta llamada *dist* en donde se almacenan las librerías, las cabeceras y los archivos ejecutables. Estos archivos se pueden distribuir en los directorios respectivos de forma manual bajo los siguientes directorios */usr/lib, /usr/bin y /usr/include,* si la instalación ya existe se puede reemplazar los archivos.

3.5.5. Configuración

No existe un archivo de configuración para la herramienta NSS ya que únicamente son librerías, cabeceras y herramientas, pero su principal desenvolvimiento están en sus librerías compartidas que ofrecen funcionalidades para aplicaciones extendida seguras.

3.5.6. Protocolos Implementados

Las librerías de la herramienta NSS implementa o soporta los siguientes protocolos:

- SSL versión 2
- SSL versión 3
- TLS versión 1

3.5.7. Algoritmos de cifrado

Los algoritmos de cifrado de claves públicas o asimétricas que implementa el proyecto NSS son los siguientes:

- RSA
- DSA
- ECDSA
- Diffie-Hellman
- EC Diffie-Hellman

Los algoritmos de cifrado de clave privada que implementa el conjunto de librerías del proyecto NSS son los siguientes:

- AES
- 3DES
- DES
- RC2
- RC4

Las Funciones Hash que soporta NSS son los siguientes:

- SHA-1
- SHA-256
- SHA-384
- SHA-256
- SHA-512
- MD2
- MD5
- HMAC

3.5.8. Portabilidad

NSS ha sido certificado en varias plataformas, entre las más importantes constan:

- AIX 4.3
- HP-UX 11.0
- Red Hat Linux 6.0
- Solaris 2.6 o más
- Windows NT
- Windows 2000

3.5.9. Usabilidad

Las librerías NSS es utilizado en el desarrollo de aplicaciones tanto cliente como servidor, las cuales son muy difundidas, entre las más importantes se encuentran las siguientes:

- Productos clientes Mozilla, incluido la Suite Mozilla, Firefox y Thunderbird.
- Navegadores de Netscape.
- Comunicator AOL y Messenger AOL (AIM).
- Aplicaciones cliente Open Source como Evolution, Gaim y OpenOffice.org 2.0.
- Productos Servidor para Red Hat como: Red Hat Directory Server, Red Hat Certificate System, y mod_nss módulo SSL para el servidor web Apache.
- Productos Servidor para Sun Java Enterprise System, incluyendo Sun Java System Web Server, Sun Java System Directory Server, Sun Java System Portal Server, Sun Java System Messaging Server, y Sun Java System Application Server.

3.5.10. Soporte Técnico

El proyecto NSS es desarrollado por la Fundación Mozilla, por tanto cualquiera de los desarrolladores de la comunidad Mozilla que tengan un conocimiento suficiente pueden dar soporte técnico, además de los desarrolladores propios que han contribuido a la creación del conjunto de librerías que conforman el NSS.

Sin embargo Netscape Comunications Corporation por haber contribuido con sus desarrolladores en la elaboración de las librerías de NSS tienen un mejor conocimiento y se podría contactar a los que han contribuido con el proyecto para obtener soporte técnico.

3.5.11. Situación Legal

NSS se encuentra licenciado bajo las siguientes licencias:

- MPL Mozilla Public License
- GNU General Public License
- GNU Lesser General Public License

Sin embargo el código fuente está sujeto a las regulaciones de la Administración de Exportaciones de los Estados Unidos y otras leyes y no puede ser exportado o re-exportado a países como Cuba, Irán, Libia, Corea del Norte, etc.

3.6 Determinación de los Parámetros de Comparación

- Parámetro 1: Instalación
- Parámetro 2: Seguridad
- Parámetro 3: Funcionalidad
- Parámetro 4: Portabilidad
- Parámetro 5: Soporte Técnico y Situación Legal

3.7 Descripción de los Parámetros de Comparación

3.7.1. Parámetro1: Instalación

En el presente parámetro se llevará a efecto el análisis comparativo de la instalación de cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, se comparará los tiempos de ejecución en cada uno de los pasos de la instalación. Para llevar a efecto la instalación se hará uso del código fuente de cada

herramienta mediante las herramientas que facilita el sistema operativo antes mencionado como: `gmake` y `make`.

3.7.2. Parámetro 2: Seguridad

En este parámetro se analizará y comparará la seguridad que provee cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, basándose en los protocolos de seguridad que soportan, la longitud de las claves de los algoritmos de cifrado simétrico para la transmisión de la información de forma segura, la longitud de las claves de los algoritmos de cifrado asimétrico para el intercambio de claves privadas y la longitud de las claves para las funciones hash para comprobar la integridad del mensaje.

3.7.3. Parámetro 3: Funcionalidad

El parámetro funcionalidad se realizará el análisis y comparación de la generación de las funciones implementadas por cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, para la generación claves, generación de certificados digitales.

3.7.4. Parámetro 4: Portabilidad

La portabilidad se centra especialmente en realizar un análisis comparativo acerca de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS desde el punto de vista de la facilidad de implantar en las diferentes plataformas o sistemas operativos. Además las aplicaciones extendidas que utilizan el conjunto de funciones que proveen las librerías compartidas de las herramientas.

3.7.5. Parámetro 5: Soporte Técnico y Situación Legal

En este módulo se analizará y comparará la documentación y las condiciones de uso de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. En relación a documentación se tomará en cuenta la información publicada en los sitios oficiales de los respectivos proyectos, la ayuda incorporada a cada herramienta mediante

la utilidad MAN. Las condiciones de uso serán analizadas desde el punto de vista de las licencias de distribución y sus restricciones.

3.8 Determinación de los indicadores de los parámetros de comparación

3.8.1. Parámetro 1: Instalación

- Dependencias
- Personalización
- Información sobre instalación
- Tiempo de descompresión
- Tiempo de compilación
- Tiempo de construcción
- Tiempo de instalación

3.8.2. Parámetro 2: Seguridad

- Protocolos
- Longitud de clave de los algoritmos de cifrado simétrico
- Longitud de clave de los algoritmos de cifrado asimétrico
- Longitud de clave de las funciones hash

3.8.3. Parámetro 3: Funcionalidad

- Generación de claves de los algoritmos simétricos
- Generación de claves de los algoritmos asimétricos
- Implementación de Autoridades Certificadoras

3.8.4. Parámetro 4: Portabilidad

- Plataformas
- Aplicaciones extendidas

3.8.5. Parámetro 5: Soporte Técnico y Situación Legal

- Ayuda en línea del sitio oficial
- Páginas de ayuda MAN
- Restricciones
- Licencias

Descripción de los indicadores de los parámetros de comparación

3.9.1. Parámetro 1: Instalación

3.9.1.1 Dependencias

El indicador dependencias evaluará la cantidad de dependencias que necesita cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. Mientras mayor sea el número de dependencias menor será el valor de su evaluación.

3.9.1.2 Personalización

La personalización es un indicador que evaluará el grado de configurabilidad al momento de instalar cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. Se evaluará las opciones y argumentos del script `config`, al momento de realizar la compilación del software.

3.9.1.3 Información sobre la instalación

Este indicador evaluará la disponibilidad de información adjunta en las distribuciones de cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. Específicamente se evaluará la existencia y la estructura de los archivos README e INSTALL, los cuales detallan los pasos para ejecutar una instalación.

3.9.1.4 Tiempo de descompresión

El tiempo de descompresión se refiere al tiempo que tarda en descomprimirse el código fuente de las herramientas, que por lo general está disponible en formato bz2, gz, tar.gz, entre otros.

3.9.1.5 Tiempo de compilación

El tiempo de compilación es el tiempo que toma al script `config` en compilar el código fuente según las especificaciones y opciones dadas al momento de ejecutar el script en cada herramienta.

3.9.1.6 Tiempo de construcción

El indicador tiempo de construcción se refiere al tiempo que tarda al ejecutar el comando `make` para construir el código antes compilado por el script `config`.

3.9.1.7 Tiempo de instalación

El tiempo de instalación es el tiempo que tarda el comando `make install` en ubicar los diferentes archivos como pueden ser librerías, cabeceras y ejecutables construidos por el comando `make` en sus respectivos directorios.

3.9.2. Parámetro 2: Seguridad

3.9.2.1 Protocolos

Este indicador nos permitirá evaluar los protocolos implementados en cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS en la transmisión de la información.

3.9.2.2 Longitud de clave de los algoritmos de cifrado simétrico

En el indicador longitud de clave de los algoritmos de cifrado simétrico se evaluará la seguridad de los principales algoritmos implementados en cada una de las herramientas

de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. Para evaluar la seguridad de cada algoritmo se tomará en cuenta la longitud de las claves de los principales algoritmos implementados.

3.9.2.3 Longitud de clave de los algoritmos de cifrado asimétrico

El indicador longitud de clave de los algoritmos de cifrado asimétricos implementados en cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS evaluará la seguridad en base a la longitud de las claves de los algoritmos asimétricos implementados.

3.9.2.4 Longitud de clave de las funciones hash

Este indicador evalúa la seguridad desde el punto de vista de la integridad, que cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS implementan basándose en la longitud de clave que soportan las funciones hash.

3.9.3. Parámetro 3*:* Funcionalidad

3.9.3.1 Generación de claves de los algoritmos simétricos

El indicador generación de claves de los algoritmos simétricos desde el punto de vista de la funcionalidad evalúa la cantidad y calidad de funciones implementadas para cumplir con su propósito en cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

3.9.3.2 Generación de claves de los algoritmos asimétricos

Este indicador evaluará las funciones implementadas por cada herramienta para la generación de claves de los algoritmos de los algoritmos asimétricos basándose en la cantidad y calidad.

3.9.3.3 Implementación de Autoridades Certificadoras

Este indicador evaluará la función de generación autoridades certificadoras mediante cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

3.9.4. Parámetro 4: Portabilidad

3.9.4.1 Plataformas

Este indicador evaluará que plataformas son las que soportan las herramientas de administración de funciones criptográficos OpenSSL, GnuTLS y NSS, además se analizará según la información publicada en sus respectivos sitios web oficiales.

3.9.4.2 Aplicaciones extendidas

El indicador aplicaciones extendidas se refiere a todas las aplicaciones que no pertenecen a cada uno de los proyectos de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, aquí se evaluará la cantidad de las aplicaciones externas que las utilizan.

3.9.5. Parámetro 5: Soporte técnico y Situación Legal

3.9.5.1 Ayuda en línea del sitio oficial

Este indicador se basa en la calidad de la información publicada en sitios web por cada herramienta de administración de funciones criptográficas OpenSSL, GnuTLS y NSS. Se evaluará dicha información según la estructura que estas presentan.

3.9.5.2 Páginas MAN

La mayoría de herramientas desarrolladas para las diferentes distribuciones del sistema operativo Linux presentan ayuda mediante la utilidad MAN, por esta razón se evaluará la existencia de esta utilidad en las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

3.9.5.3 Restricciones

Por ser herramientas que ocultan la información, las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS se encuentran bajo las leyes de exportaciones de los países de origen, en las cuales se estipulan los países de destino que no pueden utilizar dichas herramientas, aquí se evaluará las condiciones que se adapten más al presente proyecto.

3.9.5.4 Licencias

Este indicador evaluará las libertades de las que gozan cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, se tomará en cuenta los aspectos relacionados con el desarrollo del presente proyecto.

3.10 Descripción del Entorno de Pruebas

Para desarrollar el análisis comparativo de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS en los módulos de prueba se ejecutará bajo el siguiente entorno:

3.10.1. Entorno de Pruebas Hardware

- Procesador Intel Core 2 Duo 1.60 GHz
- Memoria RAM 512 MB
- Disco Duro 80 GB
- Tarjeta de red 10/100 MB

3.10.2. Entorno de Pruebas Software

- Sistema Operativo CentOS 5
- Kernel 2.6.28-8.el5

3.11 Determinación de los Módulos de Pruebas

- Módulo 1: Instalación de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.
- Módulo 2: Protocolos utilizados por las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS para la transmisión de la información en la red.
- Módulo 3: Ejecución de las sub-herramientas de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.
- Módulo 4: Plataformas en las que funcionan las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.
- Módulo 5: Soporte técnico oficial.

3.12 Descripción de los Módulos de Pruebas

3.12.1. Módulo 1: Instalación de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

Este módulo de pruebas desarrollará la instalación de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS tomando como base el entorno Hardware definido anteriormente, con la siguiente configuración:

Idioma del Teclado: Latinoamericano

Paquetes Instalados:

- Desarrollo
    - Bibliotecas de desarrollo
    - Herramientas de desarrollo
- gcc-gfortran – 4.1.1-52.el5.i386
- gcc-gnat – 4.1.1-52.el5.i386
- gcc-objc – 4.1.1-52.el5.i386

- make-3.81-1.1

Cortafuegos: Deshabilitado

SELinux: Deshabilitado

Es mediante este módulo de prueba que se desarrollará la evaluación del parámetro Instalación, específicamente de sus indicadores: Dependencias, Personalización, Información sobre instalación, Tiempo de descompresión, Tiempo de compilación, Tiempo de construcción y Tiempo de instalación, de cada una de las herramientas de administración de funciones criptográficas: OpenSSL, GnuTLS y NSS.

3.12.2. Módulo 2: Protocolos utilizados por las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS para la transmisión de la información en la red.

Este módulo pretende evaluar los protocolos que implementan cada una de las herramientas de administración de funciones criptográficas haciendo uso de la herramienta Ethereal que permite capturar paquetes que se trasmiten en una red.

Para desarrollar este módulo se ha creado un formulario con varios campos como se muestra en la siguiente imagen, la misma que será implantada en los servidores seguros configurados con cada una de las herramientas en cuestión.

Gráfico III.4: Aplicación web de prueba

Con el desarrollo de este módulo se analizará el parámetro de seguridad descrito anteriormente y que se basará en los siguientes indicadores.

- Protocolos
- Longitud de clave de los algoritmos de cifrado simétrico
- Longitud de clave de los algoritmos de cifrado asimétrico
- Longitud de clave de las funciones hash

3.12.3. Módulo 3: Ejecución de las sub-herramientas de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

Este módulo pretende ejecutar las principales sub herramientas que presentan cada una de las herramientas de administración de funciones criptográficas, y analizar la funcionalidad de las mismas.

Para medir el grado de funcionalidad de las sub herramientas se tendrá en cuenta la facilidad de uso y la información de uso disponible de cada herramienta.

Para evaluar el parámetro de la funcionalidad se basará en los indicadores siguientes.

- Generación de claves de los algoritmos simétricos
- Generación de claves de los algoritmos asimétricos
- Implementación de Autoridades Certificadoras

3.12.4. Módulo 4: Plataformas en las que funcionan las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

Este módulo analiza las portabilidad de las herramientas OpenSSL, GnuTLS y NSS para los diferentes tipos de sistemas operativos que en la actualidad existen, A demás se analizan las aplicaciones externas a estas herramientas que hacen uso de las librerías que implementan funciones de administración criptográfica o los protocolos seguros. Para cumplir con este propósito se investigará en la información disponible y las fuentes serán publicadas.

La construcción de este módulo de prueba permitirá evaluar el parámetro de la portabilidad basándose en los siguientes indicadores.

- Plataformas
- Aplicaciones extendidas

3.12.5. Módulo 5: Soporte técnico oficial

En el presente módulo de prueba se construirá con el soporte técnico oficial de cada una de las herramientas de administración de funciones criptográficas, el cual principalmente se encuentra en sus sitios web oficiales y en el código fuente de los mismos.

La construcción del módulo de prueba ayudará al desarrollo de los indicadores del parámetro “Soporte técnico y situación legal”, descrito anteriormente, los cuales son:

- Ayuda en línea del sitio web oficial
- Páginas Man
- Restricciones
- Licencias

La ejecución de este módulo de prueba arrojará como resultado la calidad de las herramientas de administración de funciones criptográficas a posteriori (es decir después de la implantación de las herramientas).

3.13 Desarrollo de los Módulos de Prueba

3.13.1. Módulo 1: Instalación de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

3.13.1.1 Instalación de la herramienta de administración de funciones criptográficas OpenSSL

La instalación se encuentra anexada en formato .avi, a continuación se desarrollará los siguientes parámetros:

3.13.1.2 Dependencias

Las dependencias para la instalación de la herramienta de administración de funciones criptográficas OpenSSL se encuentran especificadas en el archivo INSTALL en la dirección ../openssl-0.9.8j/INSTALL a continuación se muestra las dependencias requeridas:

- Make
- Perl 5
- Compilador Ansi C
- Ambiente de desarrollo en forma de librerías C y archivos de cabeceras.
- Un sistema operativo Unix

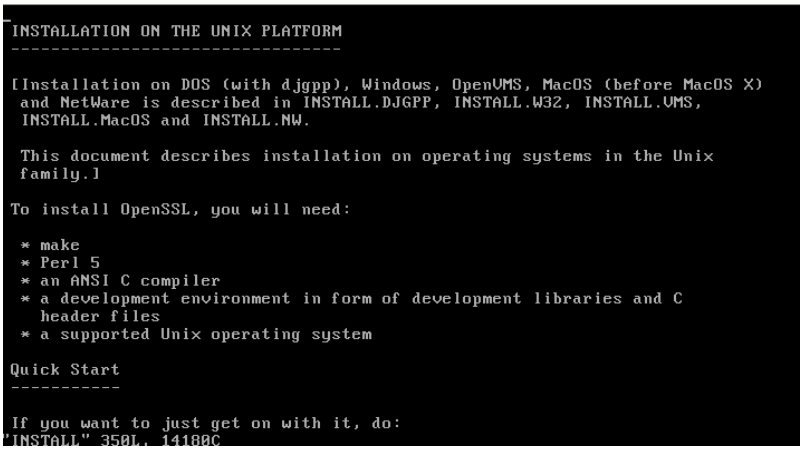

Gráfico III.5: Dependencias de la herramienta OpenSSL

3.13.1.2.1 Personalización

Para realizar la instalación de la herramienta de administración de funciones criptográficas existe una amplia lista de opciones configurables como se muestran, estas se encuentran detalladas en el archivo INSTALL en la dirección ../openssl-0.9.8j/INSTALL.

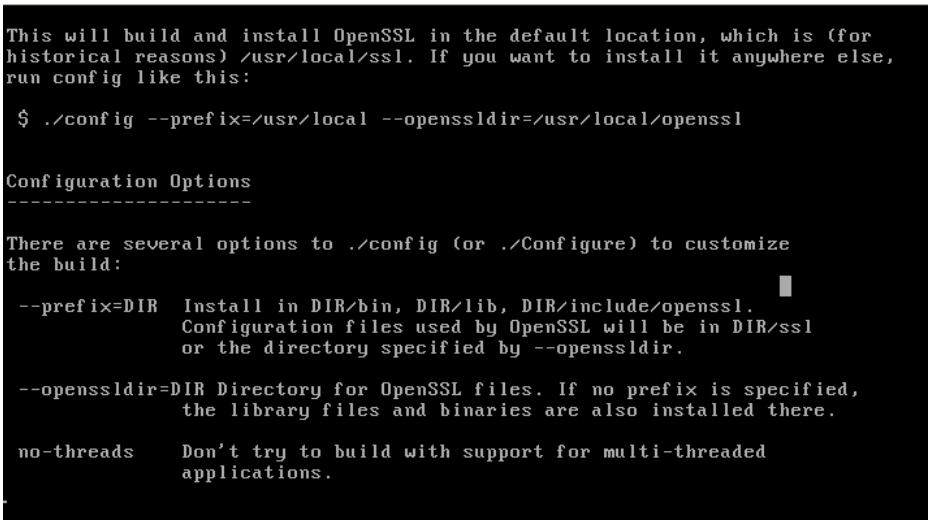

Gráfico III.6: Personalización de la instalación de la herramienta OpenSSL

3.13.1.2.2 Información sobre Instalación

Existe información para la instalación de la herramienta OpenSSL detallada en los archivos README, LICENSE, INSTALL y en la ayuda del script *configure*. Esta información tiene una estructura muy clara el directorio ../openssl-0.9.8j/.

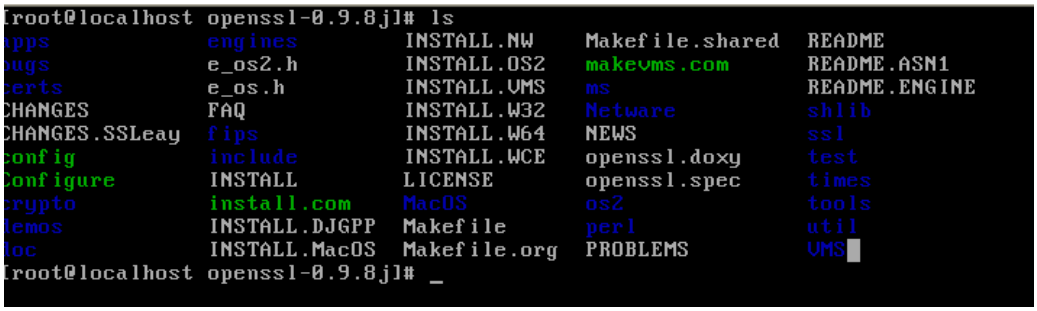

Gráfico III.7: Información sobre la instalación de la herramienta OpenSSL

3.13.1.2.3 Tiempo de descompresión

El tiempo de descompresión de la herramienta de administración de funciones criptográficas OpenSSL es de 9 segundos, puede comprobarlo accediendo a los videos anexados al presente proyecto.

3.13.1.2.4 Tiempo de compilación

El tiempo de compilación de la herramienta de administración de funciones criptográficas OpenSSL es de 8 segundos.

3.13.1.2.5 Tiempo de construcción

El tiempo de construcción de la herramienta de administración de funciones criptográficas OpenSSL es de 3 minutos y 0 segundos.

3.13.1.2.6 Tiempo de instalación

El tiempo de instalación de la herramienta de administración de funciones criptográficas OpenSSL es de 1 minuto y 5 segundos.

3.13.1.3 Instalación de la herramienta de administración de funciones criptográficas GnuTLS

La instalación de la herramienta de funciones criptográficas GnuTLS se ejecutó en las mismas condiciones citadas anteriormente.

3.13.1.3.1 Dependencias

La herramienta de administración de funciones criptográficas GnuTLS tiene como dependencias la librería libgcrypt, a continuación se muestra una pantalla del archivo README en donde se encuentra especificado las dependencias requeridas:

- Librería Libcrypt
- OpenPGP

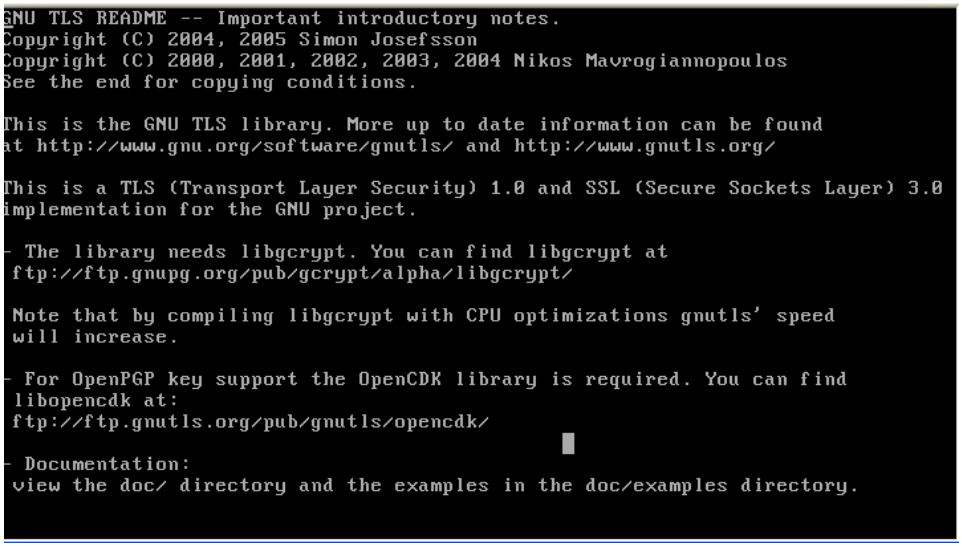

Gráfico III.8: Dependencias de la herramienta GnuTLS

3.13.1.3.2 Personalización

Para la personalización de la instalación de la herramienta existen opciones detalladas en el archivo INSTALL y sobre todo en la ayuda del script *configure.* Esta información se encuentra en el directorio *../gnutls-2.2.4/INSTALL,* aquí se muestra un fragmento.

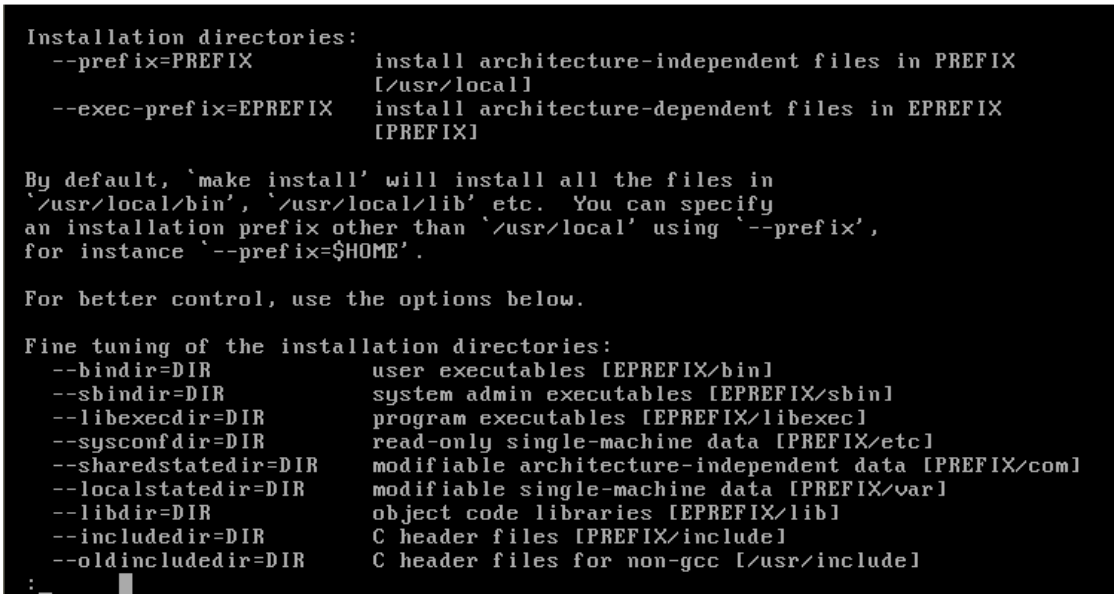

Gráfico III.9: Personalización de la instalación de la herramienta GnuTLS

3.13.1.3.3 Información sobre Instalación

La información de instalación se encuentra bien detallada en los ficheros INSTALL, README y también en la ayuda del script *configure*, toda esta información está bajo el directorio *../gnutls-2.2.4.*

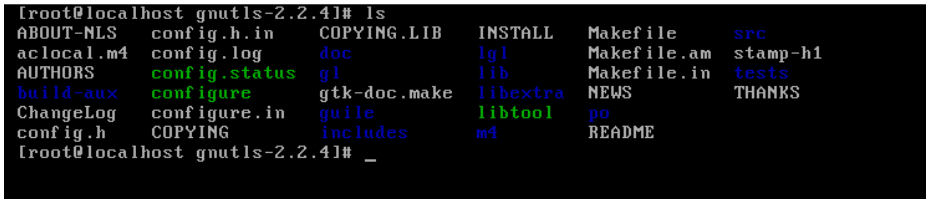

Gráfico III.10: Información sobre la instalación de la herramienta GnuTLS

3.13.1.3.4 Tiempo de descompresión

El tiempo estimado de descompresión del código fuente de la herramienta de administración de funciones criptográficas GnuTLS es 11 segundos. Se lo puede verificar observando los videos anexados al presente proyecto.

3.13.1.3.5 Tiempo de compilación

El tiempo de compilación de la presente herramienta es aproximadamente 54 segundos.

3.13.1.3.6 Tiempo de construcción

El tiempo de construcción de la herramienta GnuTLS por medio de la herramienta make es de 3 minutos y 28 segundos.

3.13.1.3.7 Tiempo de instalación

El tiempo de instalación de la herramienta GnuTLS es de 33 segundos.

3.13.1.4 Instalación de la herramienta de administración de funciones criptográficas NSS

La instalación de la herramienta de administración de funciones criptográficas NSS difiere un poco de las dos herramientas anteriores, debido a que en algunos indicadores de tiempo no es aplicable.

3.13.1.4.1 Dependencias

La principal dependencia para la instalación de la herramienta NSS es el paquete NSPR que le permite la portabilidad para varias plataformas, está dependencia está especificada en el siguiente sitio web de la herramienta.

- http://www.mozilla.org/projects/security/pki/nss/overview.html
- http://www.mozilla.org/projects/security/pki/nss/#info Make

3.13.1.4.2 Personalización

Para la instalación de la herramienta NSS no existe mayor personalización o la información no está muy difundida. La más conocidas y difundidas están detalladas en el sitio web oficial de la herramienta.

- http://www.mozilla.org/projects/security/pki/nss/buildnss_31.html

3.13.1.4.3 Información sobre Instalación

No existe información disponible para ayudar en la instalación como son los archivos README e INSTALL, los únicos que se observan en la búsqueda que se muestra a continuación son archivos README individuales para cada sub-herramienta que no tienen mayor información para realizar la instalación. Ver Gráfico III.10.

3.13.1.4.4 Tiempo de descompresión

El tiempo de descompresión de la herramienta de administración de funciones criptográficas NSS es de 17 segundos.

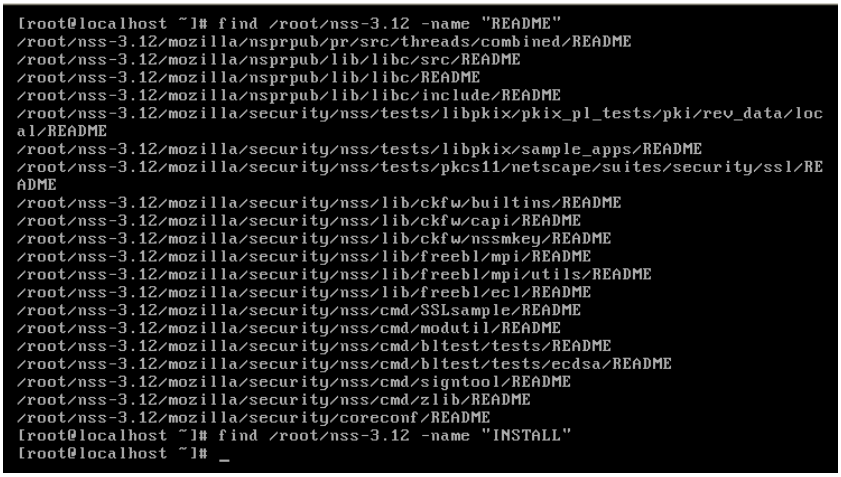

Gráfico III.11: Información sobre la instalación de la herramienta NSS

3.13.1.4.5 Tiempo de compilación

No existe tiempo de compilación de la herramienta de administración de funciones criptográficas NSS, por lo tanto no es aplicable al presente indicador.

3.13.1.4.6 Tiempo de construcción

El tiempo de construcción de la herramienta de administración de funciones criptográficas NSS es de 2minutos y 59 segundos.

3.13.1.4.7 Tiempo de instalación

El indicador de tiempo de instalación de la herramienta NSS no es aplicable, debido a que la instalación de esta herramienta se la tiene ejecutar manualmente.

3.13.2. Módulo 2: Protocolos utilizados por las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS para la transmisión de la información en la red.

3.13.2.1 Módulo 2 implementado sobre OpenSSL

3.13.2.1.1 Protocolos

Paquetes capturados con un servidor de prueba seguro configurado con OpenSSL, como se puede observar el protocolo seguro que utiliza es SSLv3.

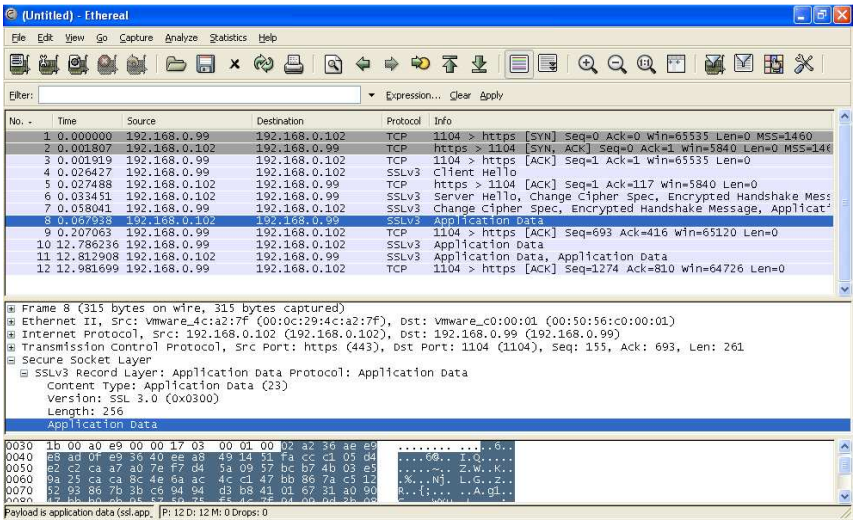

Gráfico III.12: Paquetes capturados con Ethereal desde el servidor con OpenSSL

3.13.2.2 Módulo 2 implementado sobre GnuTLS

3.13.2.2.1 Protocolos

Paquetes capturados con un servidor de prueba seguro configurado con GnuTLS.

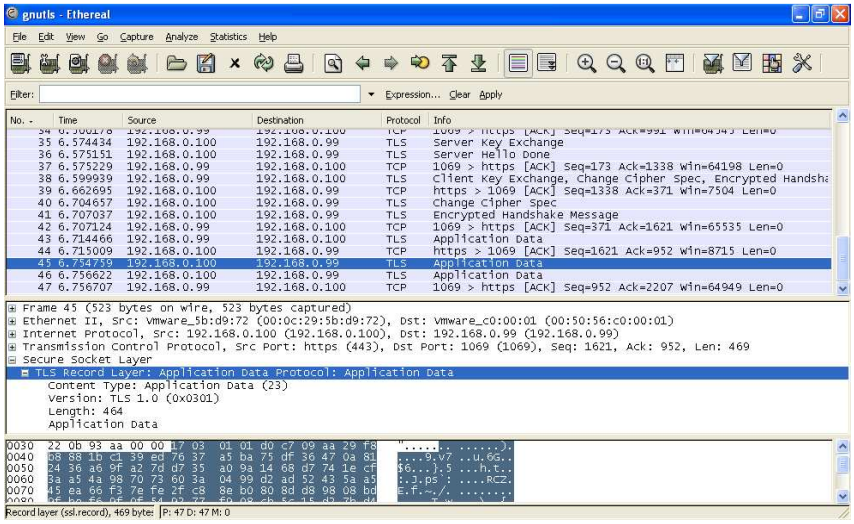

Gráfico III.13: Paquetes capturados con Ethereal desde el servidor con GnuTLS

3.13.2.3 Módulo 2 implementado sobre NSS

3.13.2.3.1 Protocolos

Paquetes capturados con un servidor de prueba seguro configurado con NSS.

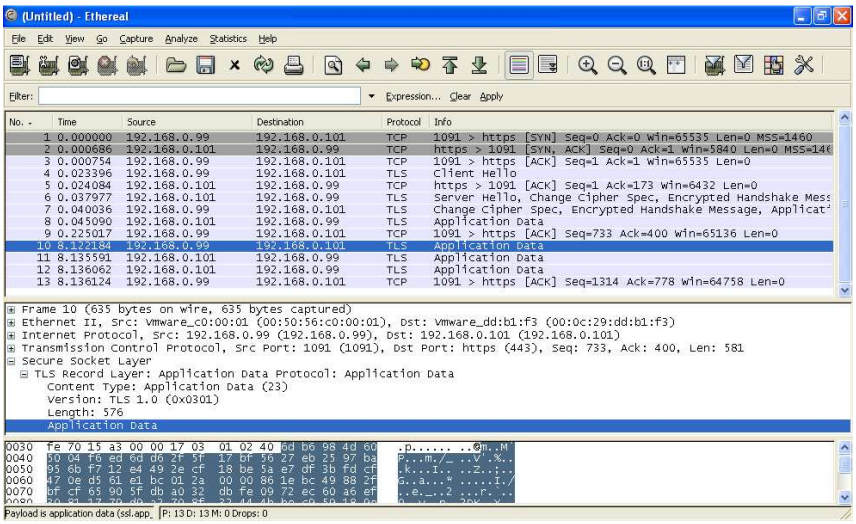

Gráfico III.14: Paquetes capturados con Ethereal desde el servidor con NSS

3.13.2.4 Módulo 2 implementado sobre NSS, GnuTLS y NSS.

3.13.2.4.1 Longitud de clave de los algoritmos de cifrado simétrico, asimétrico, y funciones hash

Como indica el estudio previo de las herramientas en el capitulo anterior, las tres herramientas han implementado un común numero de algoritmos simétricos, asimétricos y funciones hash, basándose principalmente por naturaleza en aquellos que tienen patentes libres y más seguros. Por lo que se considera innecesario un nuevo estudio de este punto.

3.13.3. Módulo 3: Ejecución de las sub herramientas de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

3.13.3.1 Ejecución de las sub herramientas de OpenSSL

3.13.3.1.1 Generación de claves con algoritmos simétricos

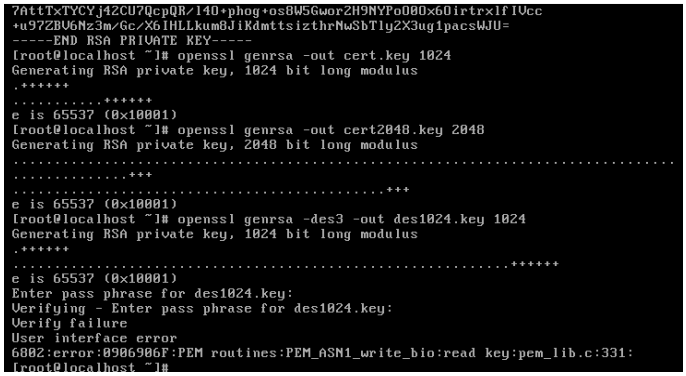

Gráfico III.15: Generación de claves privadas con OpenSSL

3.13.3.1.2 Generación de claves de los algoritmos asimétricos

Gráfico III.16: Generación de claves públicas con OpenSSL

3.13.3.1.3 Implementación de Autoridades Certificadoras

Con la CA que creamos a continuación podemos firmar cualquier certificado que necesitemos hacerlo con la diferencia que no son certificados aprobados por una CA publica que pueda corroborar la veracidad, y son únicamente para pruebas.

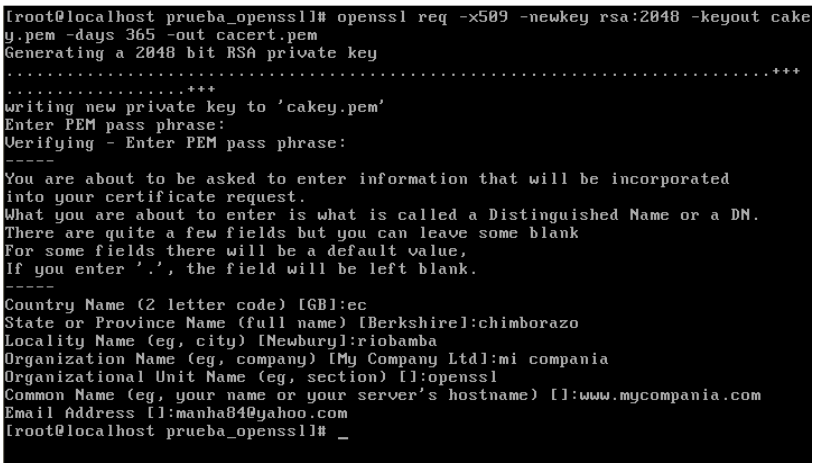

Gráfico III.17: Implementación de una CA con OpenSSL

3.13.3.1.4 Creación de un certificado digital

Para crear un certificado digital es necesario generar antes una clave privada y una petición del certificado que se le puede generar de la siguiente manera.

Con el primer comando generamos la clave privada y con el segundo generamos una petición del certificado, lo podemos observar en el Gráfico III.18.

La herramienta OpenSSL tiene un archivo de configuración para la generación de certificados digitales ubicado bajo el directorio de instalación ssl para reducir los parámetros de configuración es posible crear un archivo de configuración personalizado como se muestra en el Gráfico III.19.

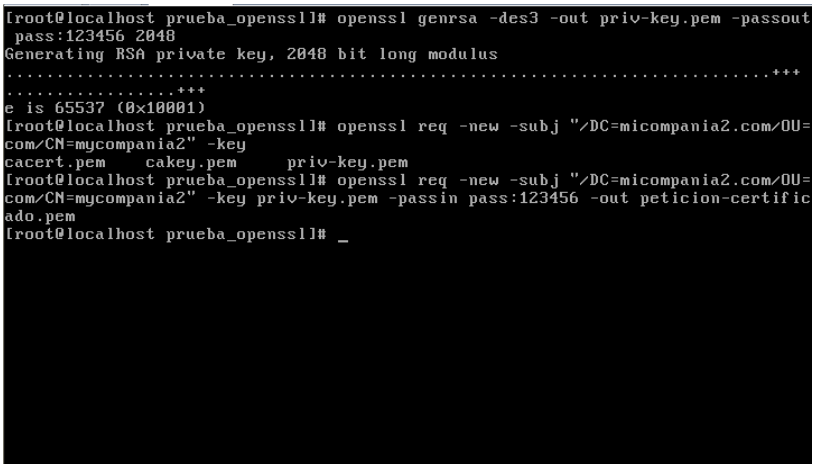

Gráfico III.18: Petición de certificado con OpenSSL

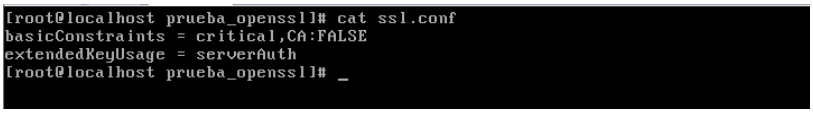

Gráfico III.19: Creación de un archivo de configuración para generar un certificado con OpenSSL.

De esta forma se utiliza el archivo de configuración anterior para generar el certificado digital firmado por una CA propia, los mismos que pueden ser utilizados para crear un servidor seguro con el módulo mod_ssl.

```
[root@localhost prueba_openssl]# openssl x509 -CA cacert.pem -CAkey cakey.pem -r
eq -in peticion-certificado.pem -days 365 -extfile ssl.conf -sha1 -CAcreateseria
l -out cert.pem
Signature ok
subject=/DC=micompania2.com/OU=com/CN=mycompania2
Getting CA Private Key
Enter pass phrase for cakey.pem:
[root@localhost prueba_openssl]# _
```

Gráfico III.20: Generación de un certificado digital con OpenSSL

El comando **ca** es una aplicación pequeña que permite administrar un CA, esta herramienta permite firmar solicitudes de certificados y administrar una base de datos en texto plano que registra la lista de certificados y su estado, también permite crear la CRLs. Su funcionalidad esta explicada en la página oficial de la herramienta. Esta herramienta puede ser ejecutada de la siguiente manera.

Para ejecutar el comando **ca** es necesario crear un archivo de configuración o usar el que viene por defecto, para este caso creamos uno nuevo.

```
policy          = policy_any

email_in_dn     = no

name_opt        = ca_default
cert_opt        = ca_default
copy_extencions = none

[ policy_any ]
countryName             = optional
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = optional
emailAddress            = optional



####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = /root/demoCA          # directorio de la CA
certs           = $dir/certs            # directorios de certificados
crl_dir         = $dir/crl              # lista de certificados rebocados
database        = $dir/index.txt        # archivo index de la base de datos
                                        # several ctificates with same subject.
new_certs_dir   = $dir/newcerts         # lugar por defecto para los
                                        # certificados nuevos

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
private_key     = $dir/private/cakey.pem# The private key
RANDFILE        = $dir/private/.rand    # private random number file

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
```

Gráfico III.21: Herramienta ca de OpenSSL

Ahora se ejecuta el comando ca para crear un nuevo certificado.

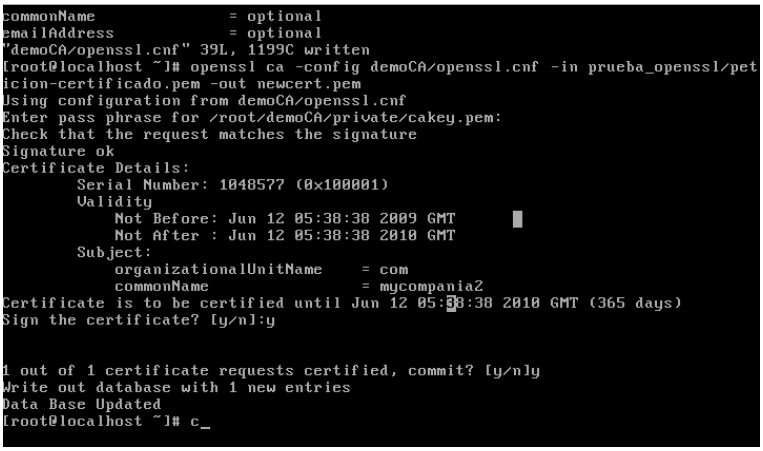

Gráfico III.22: Creación de un certificado con la herramienta ca de OpenSSL

3.13.3.1.5 Generando un CRL

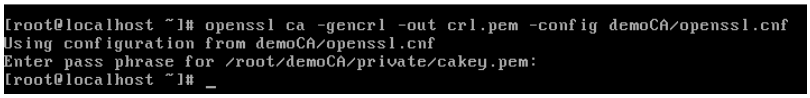

Gráfico III.23: Creación de un CRL con OpenSSL

Para la parte de los test se puede conectar a un servidor seguro mediante los comandos s_client y s_server como se presenta en el Gráfico III.24 y el Gráfico III.25..

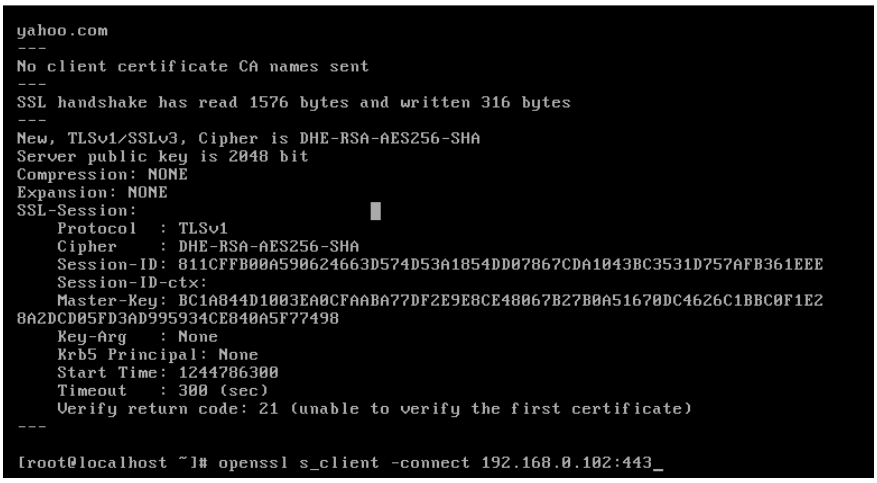

Gráfico III.24: Test de un cliente con OpenSSL

Gráfico III.25: Test de un servidor con OpenSSL

Y de esta forma se puede ejecutar muchos comandos más de la herramienta OpenSSL que permiten cumplir con las tareas detalladas anteriormente, es necesario decir que los comandos de la herramienta OpenSSL tiene mucha información disponible tanto en su sitio web oficial, en las páginas de ayuda de la aplicación misma y en otros sitios web por lo que su uso se hace más sencillo.

3.13.3.2 Ejecución de las sub herramientas de GnuTLS

3.13.3.2.1 Generación de claves de los algoritmos simétricos

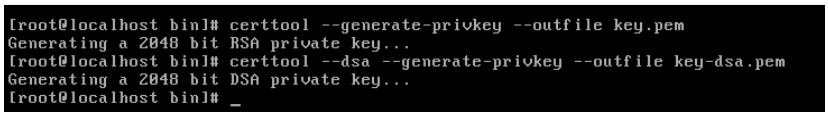

Gráfico III.26: Generación de claves privadas con GnuTLS

3.13.3.2.2 Generación de claves de los algoritmos asimétricos

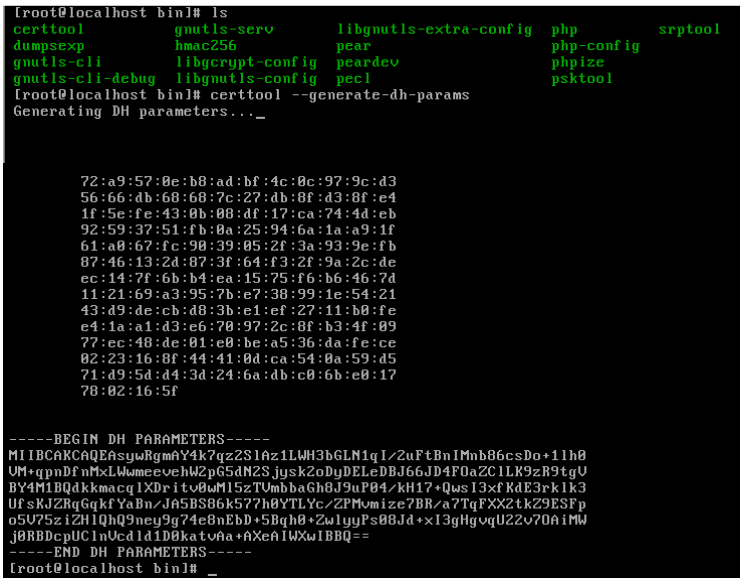

Gráfico III.27: Generación de parámetros de intercambio DH de clave con GnuTLS

Parámetros generados por RSA-EXPORT para el intercambio de clave

```
[root@localhost bin]# certtool --generate-privkey --bits 512
Generating a 512 bit RSA private key...
-----BEGIN RSA PRIVATE KEY-----
MIIBOQIBAAJBAOTj+DZ3MQE5Rs+Fcz1j7cEdctOKOyTh0DV85X1N4T0dPiEyHEK9
9AfiRmjDyDO7Pe3eECbP6/wppzOdzgPTcF0CAwEAAQI/elED2vrlSpBYTy3CqzZm
OyJPz+aybW3pwOSS97/KM3LjReQIavAw06m8UcaIhIt4NG5dPYt13yl9sBLJGvfp
AiEA78sjE0/sPfb3wCoNdEyFCbYDgi/Kutwv6+2cM48E9qkCIQD0XDDfg+WFOp4Z
87yUvq1qzC6HTWzGg86jLQogOrzglQIgKDFrcqw90/EWe+5QjQLaddM9aU/rckDA
A5KzUO0TDukCIASv0/wlgnx4sFmdFCviMF9RBT9xG7n/AzKcBYqytba1AiEAzIyg
sA8y8FNe1kgvRcRS4rXeTfrgXGh9z395crNmEd8=
-----END RSA PRIVATE KEY-----
[root@localhost bin]# _
```

Gráfico III.28: Generación de parámetros de intercambio RSA de clave con GnuTLS

3.13.3.2.3 Implementación de Autoridades Certificadoras

Generación de certificados firmado por sí mismo (CA)

```
[root@localhost bin]# certtool --generate-privkey --outfile /root/ca-key.pem
Generating a 2048 bit RSA private key...
[root@localhost bin]# certtool --generate-self-signed --load-privkey ca-key.pem
--outfile ca-cert.pem_
```

Gráfico III.29: Generación de en certificado digital auto firmado con GnuTLS

Para generar un certificado usando la clave privada

```
[root@localhost bin]# certtool --generate-certificate --load-privkey key.pem --o
utfile cert.pem --load-ca-certificate ca-cert.pem --load-ca-privkey ca-key.pem
Generating a signed certificate...
Please enter the details of the certificate's distinguished name. Just press ent
er to ignore a field.
Country name (2 chars): _
```

Gráfico III.30: Generación de un certificado con GnuTLS

Para generar una solicitud de certificado

```
[root@localhost bin]# certtool --generate-request --load-privkey key.pem --outfi
le request.pem
Generating a PKCS #10 certificate request...
Country name (2 chars): ec
Organization name:
Organizational unit name:
Locality name:
State or province name:
Common name:
UID: haro
Enter a challenge password:
[root@localhost bin]# _
```

Gráfico III.31: Generación de una solicitud de certificado con GnuTLS

Generar un certificado usando la solicitud previa

```
[root@localhost bin]# certtool --generate-certificate --load-request request.pem
 --outfile cert1.pem  --load-ca-certificate ca-cert.pem --load-ca-privkey ca-key
.pem
Generating a signed certificate...
Enter the certificate's serial number in decimal (default: 1250022786): 009
Trailing garbage ignored: `9'


Activation/Expiration time.
The certificate will expire in (days): 365


Extensions.
Does the certificate belong to an authority? (y/N): n
Is this a TLS web client certificate? (y/N): y
Is this also a TLS web server certificate? (y/N): y
Enter the dnsName of the subject of the certificate: prueba.com
Enter the dnsName of the subject of the certificate:
Will the certificate be used for signing (DHE and RSA-EXPORT ciphersuites)? (y/N
): y
Will the certificate be used for encryption (RSA ciphersuites)? (y/N): y
X.509 Certificate Information:
        Version: 3
        Serial Number (hex): 00
```

Gráfico III.32: Generación de un certificado usando solicitud previa con GnuTLS

Ver la información de un certificado

```
                ab:f5:31:71:63:92:0e:a6:f4:62:59:40:0e:56:32:5f
                8a:4f:d7:fa:1b:e5:12:4f:45:f8:e2:00:aa:3a:b5:39
[root@localhost bin]# certtool --certificate-info --infile cert.pem
X.509 Certificate Information:
        Version: 3
        Serial Number (hex): 00
        Issuer: C=ec,L=riobamba,ST=riobamba,CN=manuel haro,UID=mharo,EMAIL=manha
84@yahoo.com
        Validity:
                Not Before: Tue Aug 11 20:25:07 UTC 2009
                Not After: Wed Aug 11 20:25:11 UTC 2010
        Subject: C=ec,L=Chimborazo,ST=riobamba,CN=manuel haro,UID=mhm,EMAIL=manh
a84@yahoo.com
        Subject Public Key Algorithm: RSA
                Modulus (bits 2048):
                        ae:bc:ab:a2:74:47:a5:df:1b:ca:19:45:fb:5d:ac:38
                        0d:b7:8f:15:69:35:eb:6a:1e:81:4c:08:ab:4a:f8:8f
                        25:cc:f7:c1:7d:f9:30:8e:aa:15:c6:eb:e6:c7:e3:cc
                        f0:85:d7:c3:af:04:e9:74:4d:cc:47:a7:8f:ca:1a:b3
                        09:48:d0:2f:e5:08:b5:af:b0:02:c8:e4:51:99:2b:d3
                        6f:6c:ae:94:5c:b3:7b:f0:85:4d:58:33:8d:7f:33:11
                        21:ee:82:aa:d0:5b:24:c4:20:3c:a5:44:3f:c4:1a:5e
                        e4:ae:f1:e4:d6:fd:66:fb:56:b8:7c:7b:77:1d:be:47
                        52:b5:e1:25:7c:d4:2b:91:3e:d9:81:b4:55:eb:26:ff
                        8d:7d:a3:c0:e0:27:cd:c9:ce:e4:34:e4:c6:a7:fb:92
```

Gráfico III.33: Consulta de la Información del certificado con GnuTLS

Para generar una estructura PKCS#12 usando una clave y un certificado previo

```
[root@localhost bin]# certtool --load-certificate cert.pem --load-privkey key.pe
m --to-p12 --outder --outfile key.p12
Generating a PKCS #12 structure...
Loading certificate list...
Loaded 1 certificates.
Enter a name for the key: 123456
Enter password:
[root@localhost bin]# _
```

Gráfico III.34: Generación de parámetros de intercambio de clave con GnuTLS

Crear una CRL vacía

```
[root@localhost bin]# certtool --generate-crl --load-ca-privkey ca-key.pem --loa
d-ca-certificate ca-cert.pem
Generating a signed CRL...
Loading certificate list...
Update times.
The next CRL will be issued in (days): 365

X.509 Certificate Revocation List Information:
        Version: 2
        Issuer: C=ec,L=riobamba,ST=riobamba,CN=manuel haro,UID=mharo,EMAIL=manha
84@yahoo.com
        Update dates:
                Issued: Tue Aug 11 20:44:11 UTC 2009
                Next at: Wed Aug 11 20:44:11 UTC 2010
        No revoked certificates.
        Signature Algorithm: RSA-SHA
        Signature:
                6f:85:17:af:68:a1:27:85:d6:7c:60:5a:9a:aa:1a:e8
                03:12:bb:28:03:e4:6c:67:53:ea:c4:a6:24:33:24:1e
                2f:99:ca:1f:39:71:fc:ad:77:86:b0:32:6c:63:fb:31
                3e:dd:72:6f:58:c4:9a:67:85:59:5e:97:61:43:94:a0
                d8:12:76:9e:c1:41:22:86:ae:82:4d:f8:9e:ca:fb:47
                1b:08:34:3f:31:6c:ff:e5:50:7f:ce:66:b1:8a:bf:1f
[root@localhost bin]#
```

Gráfico III.35: Creación de una CRL vacía con GnuTLS

Creación de una CRL que contiene certificados revocados

```
[root@localhost bin]# certtool --generate-crl --load-ca-privkey ca-key.pem --loa
d-ca-certificate ca-cert.pem --load-certificate cert
cert1.pem  cert.pem   certtool
[root@localhost bin]# certtool --generate-crl --load-ca-privkey ca-key.pem --loa
d-ca-certificate ca-cert.pem --load-certificate cert1.pem
Generating a signed CRL...
Loading certificate list...
Loaded 1 certificates.
Update times.
The next CRL will be issued in (days): 100

X.509 Certificate Revocation List Information:
        Version: 2
        Issuer: C=ec,L=riobamba,ST=riobamba,CN=manuel haro,UID=mharo,EMAIL=manha
84@yahoo.com
        Update dates:
                Issued: Tue Aug 11 20:47:11 UTC 2009
                Next at: Thu Nov 19 20:47:11 UTC 2009
        Revoked certificates (1):
```

Gráfico III.36: Creación de una CRL con certificados revocados con GnuTLS

Listar los certificados revocados

```
[root@localhost bin]# certtool --verify-crl --load-ca-certificate ca-cert.pem

CA certificate:
        Subject: C=ec,L=riobamba,ST=riobamba,CN=manuel haro,UID=mharo,EMAIL=manh
a84@yahoo.com
```

Gráfico III.37: Listar los certificados revocados con GnuTLS

3.13.3.3 Ejecución de las sub herramientas de NSS

3.13.3.3.1 Generación de claves de los algoritmos simétricos y asimétricos

NSS crea claves privadas y públicas de forma conjunta, este es un ejemplo de la creación de un par de claves privadas y públicas, estas son administradas mediante una base de datos.

Gráfico III.38: Creación de claves privadas y públicas con NSS

3.13.3.3.2 Implementación de Autoridades Certificadoras

Creando una base de datos de certificados.

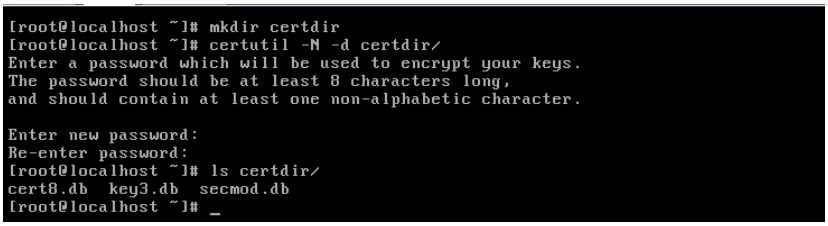

Gráfico III.39: Creación de una base de datos de Certificados con NSS

Creando una solicitud de certificado

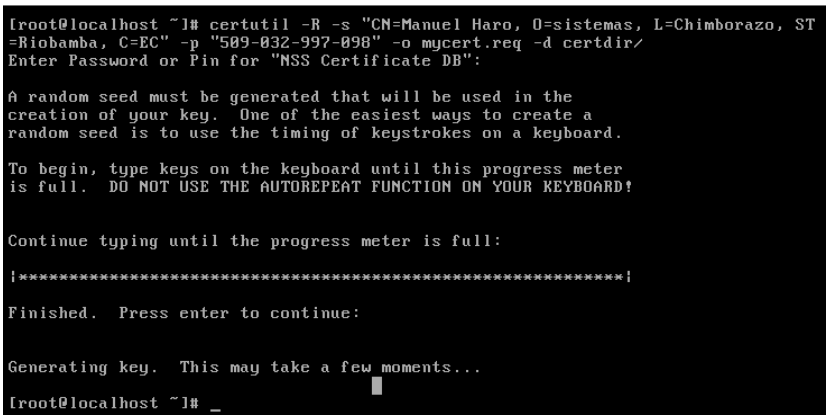

Gráfico III.40: Creación de una solicitud de certificado con NSS

Creando un certificado

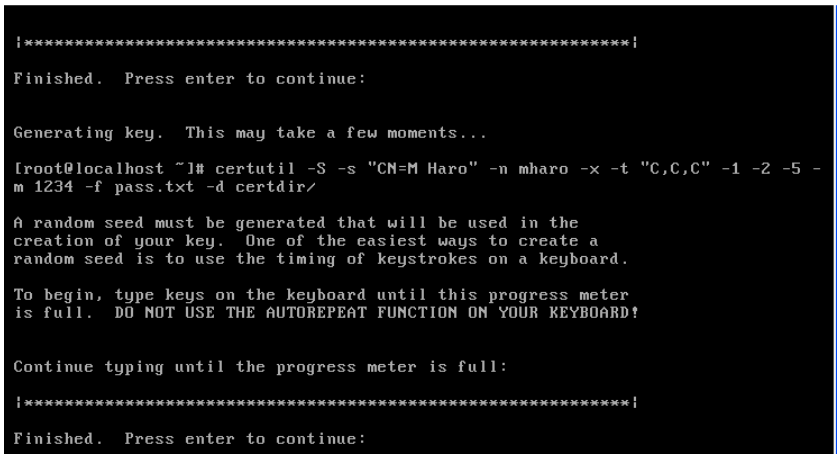

Gráfico III.41: Creación de un certificado con NSS

Añadiendo un certificado a la base de certificados.

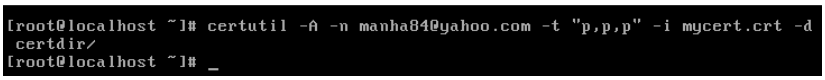

Gráfico III.42: Añadiendo certificados a la base de datos con NSS

Para ver los datos de un certificado.

Gráfico III.43: Ver los datos de un certificado con NSS

Lista los certificados de la base de datos

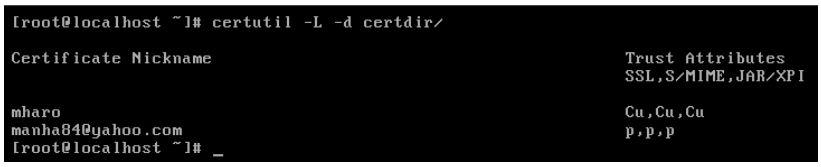

Gráfico III.44: Listar los certificados de una base de datos con NSS

Validar un certificado

Gráfico III.45: Validación de un certificado NSS

3.13.4. Módulo 4: Plataformas en las que funcionan las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

3.13.4.1 OpenSSL

3.13.4.1.1 Plataformas

Según el archivo README de OpenSSL, está diseñado para ser instalado en diferentes plataformas como son:

Plataformas derivadas de Unix como Linux, Plataformas Win32, Plataformas OpenVMS.

```
INSTALLATION
------------

To install this package under a Unix derivative, read the INSTALL file.  For
a Win32 platform, read the INSTALL.W32 file.  For OpenVMS systems, read
INSTALL.VMS.

Read the documentation in the doc/ directory.  It is quite rough, but it
lists the functions; you will probably have to look at the code to work out
how to use them. Look at the example programs.
```

Gráfico III.46: Plataformas para OpenSSL

3.13.4.1.2 Aplicaciones extendidas

Están listados en el sitio web oficial de OpenSSL más de 70 aplicaciones que usan las librerías y funciones de administración de funciones criptográficas implementadas por OpenSSL.

Este listado esta publicado en la siguiente dirección:

- http://www.openssl.org/related/apps.html

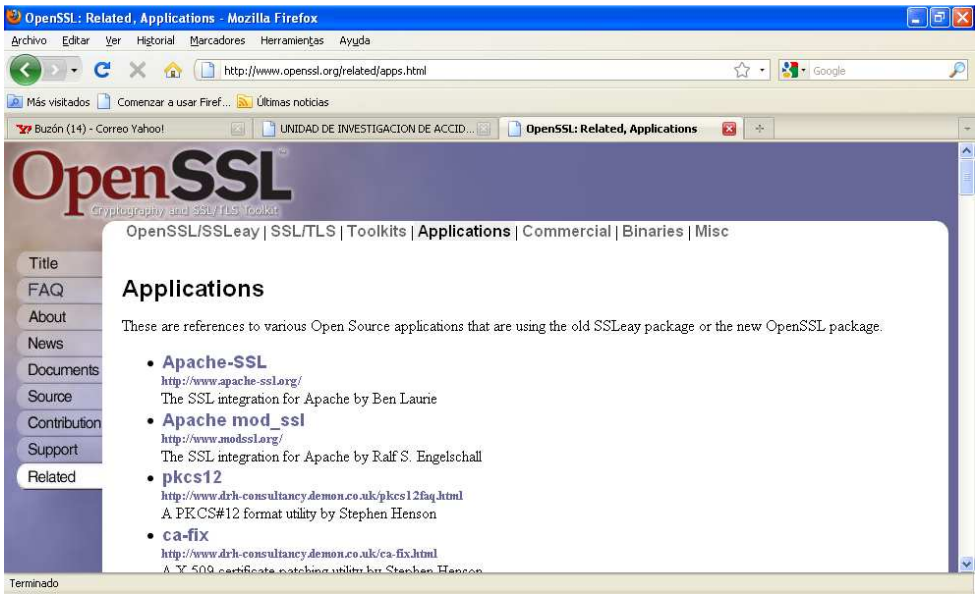

Gráfico III.47: Aplicaciones extendidas de OpenSSL

3.13.4.2 GnuTLS

3.13.4.2.1 Plataformas

Está diseñado para ser instalado en la mayoría de plataformas Linux y Windows, esta información está detallada en la siguiente dirección web.

- http://www.gnu.org/software/gnutls/gnutls.html

3.13.4.2.2 Aplicaciones extendidas

Existen 35 aplicaciones que usan GnuTLS oficialmente las mismas que están listadas en el sitio web oficial de la herramienta GnuTLS.

Las aplicaciones se encuentran listadas en la siguiente URL.

- http://www.gnu.org/software/gnutls/programs.html

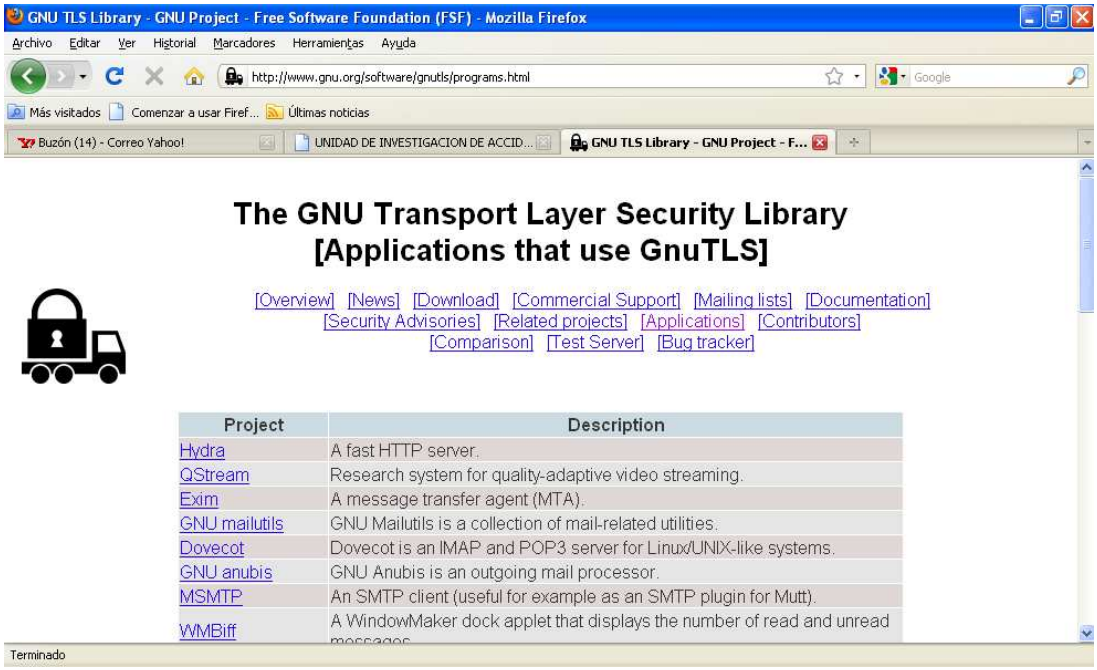

Gráfico III.48: Aplicaciones extendidas de GnuTLS

3.13.4.3 NSS

3.13.4.3.1 Plataformas

Los sistemas operativos en los que se puede instalar se muestran en la dirección web http://directory.fedoraproject.org/wiki/Mod_nss#What_platforms_does_it_support.3F.

Gráfico III.49: Plataformas para NSS

3.13.4.3.2 Aplicaciones extendidas

NSS lista 14 aplicaciones que hacen uso de forma oficial las librerías implementadas por la herramienta de administración de funciones criptográficas NSS como se muestra en el Gráfico III.50.

3.13.5. Módulo 5: Soporte técnico oficial.

Para tener una mejor visualización y entendimiento haremos uso del recurso gráfico, específicamente de pantallas capturadas tanto de los sitios web oficiales de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS, como de los comandos ejecutados acerca de la ayuda incorporada en el código fuente de cada una.

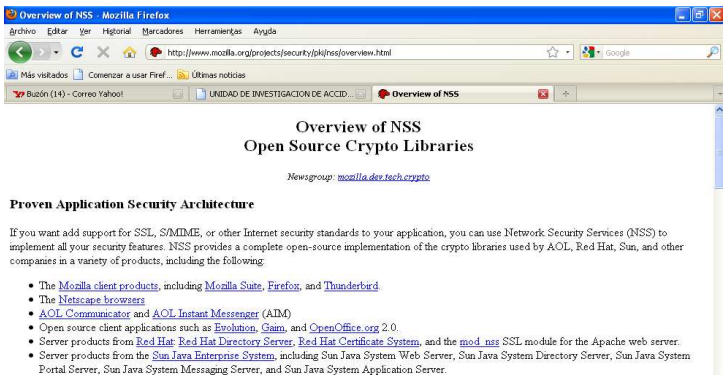

Gráfico III.50: Aplicaciones extendidas de NSS

3.13.8.1 Soporte técnico oficial de la herramienta de administración de funciones criptográficas OpenSSL

Para el desarrollo del módulo de prueba de la herramienta de administración de funciones criptográficas OpenSSL, se basará principalmente de su sitio web oficial, así como la documentación incluida en su código fuente.

3.13.8.1.1 Sitio web oficial de la herramienta OpenSSL

La dirección del sitio web de la herramienta de administración de funciones criptográficas OpenSSL es: http://www.openssl.org la misma que se muestra a continuación:

Gráfico III.51: Sitio Web oficial de la herramienta OpenSSL

La documentación existente en este sitio web contiene lo siguiente:

- Página web de documentación de la herramienta OpenSSL.
- Página web de documentación de la librería SSL/TLS de la herramienta OpenSSL.
- Página web de documentación de la librería Crypto de la herramienta OpenSSL.

- Página web de documentación de la HOWTO para introducción de conceptos teóricos.
- Página web de datos y documentación relacionada al soporte de FIPS140 (Estándar Federal de Procesamiento de Información) en la herramienta OpenSSL.
- Página web de documentación variada de sitios web de otras herramientas relaciones a la herramienta OpenSSL.

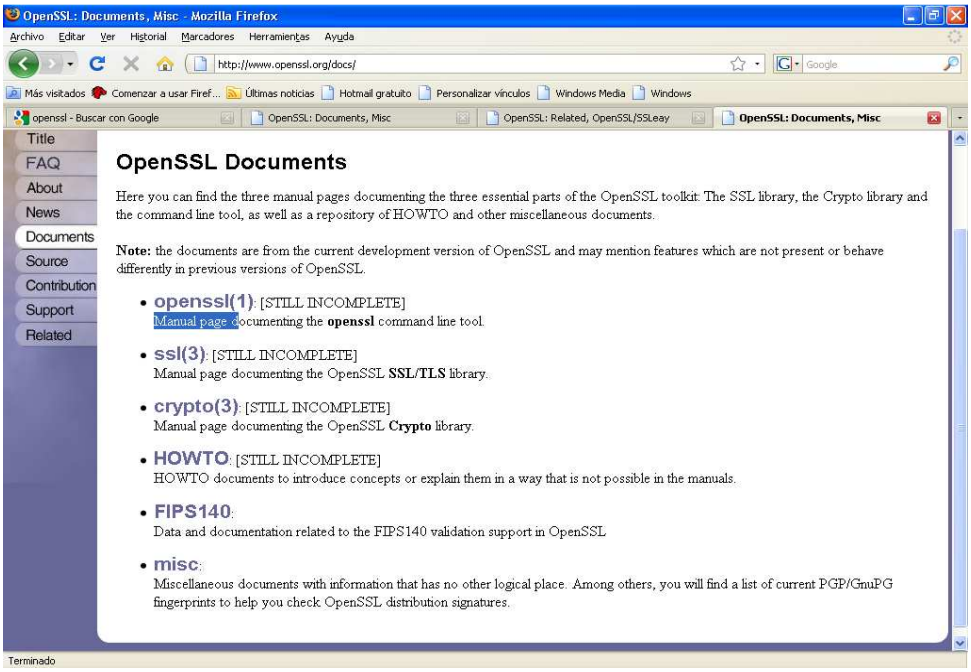

Gráfico III.52: Documentación del Sitio Web de la Herramienta OpenSSL

3.13.8.1.2 Páginas MAN de la herramienta OpenSSL

A continuación se puede observar que la herramienta de administración de funciones criptográficas OpenSSL contiene su utilidad MAN para ayudar a entender la estructura del entorno de comandos de la herramienta, ver en el Gráfico III.53.

3.13.8.1.3 Restricciones de la herramienta OpenSSL

La herramienta de administración de funciones criptográficas OpenSSL se encuentra licenciado bajo “Apache-style License”, la cual básicamente le da un significado de libre y

que pueda ser exportado o utilizado para propósitos tanto comerciales como no comerciales.

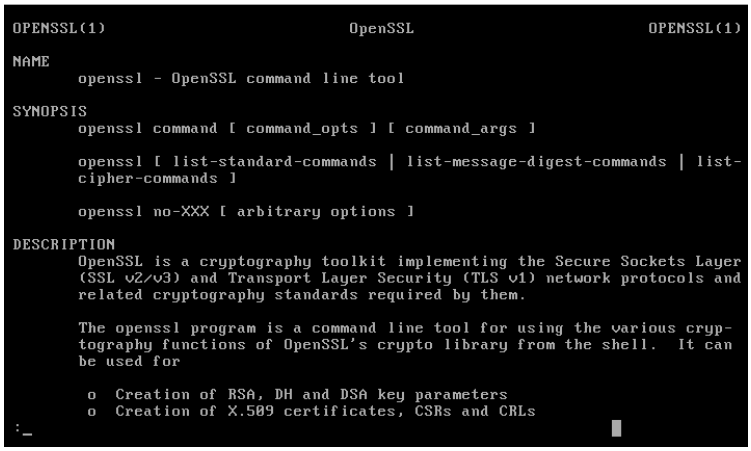

Gráfico III.53: Página MAN de la herramienta OpenSSL

3.13.8.1.4 Licencias de la herramienta OpenSSL

La licencia de las librerías de la herramienta OpenSSL es: "Apache-style License".

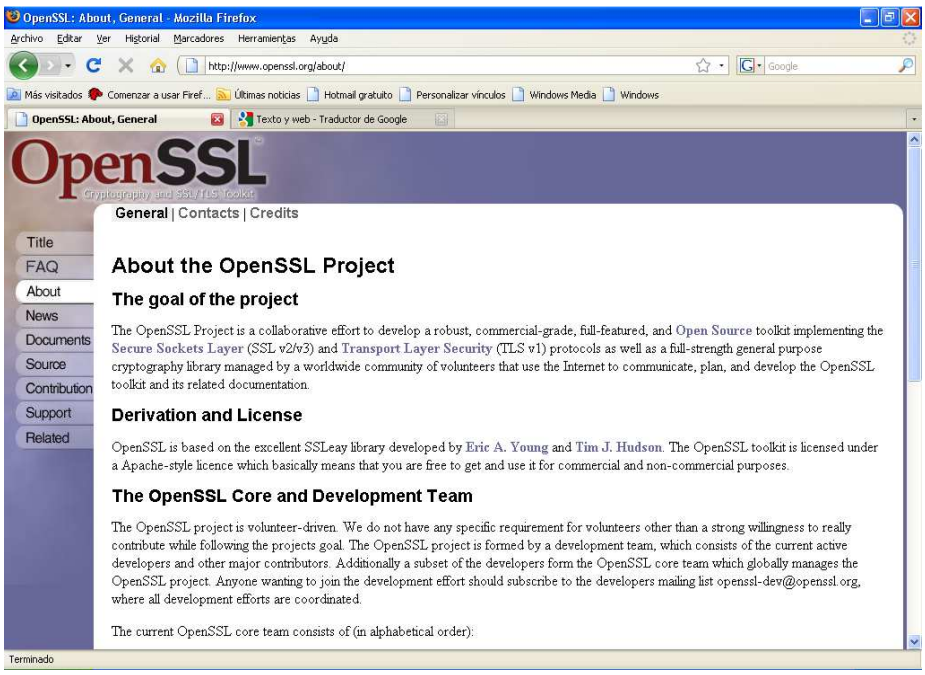

Gráfico III.54: Licencia de la Herramienta OpenSSL

3.13.8.2 Soporte técnico oficial de la herramienta de administración de funciones criptográficas GnuTLS

En el desarrollo del módulo de prueba para la herramienta de administración de funciones criptográficas GnuTLS se fundamentará principalmente en el soporte técnico del sitio web oficial de esta herramienta, además de la información incluida en el código fuente.

3.13.8.2.1 Sitio web oficial de la herramienta GnuTLS

La herramienta de administración de funciones criptográficas GnuTLS tiene su sitio web oficial en la dirección: http://www.gnu.org/software/gnutls, al ingresar a esta dirección se le presenta el siguiente home page.

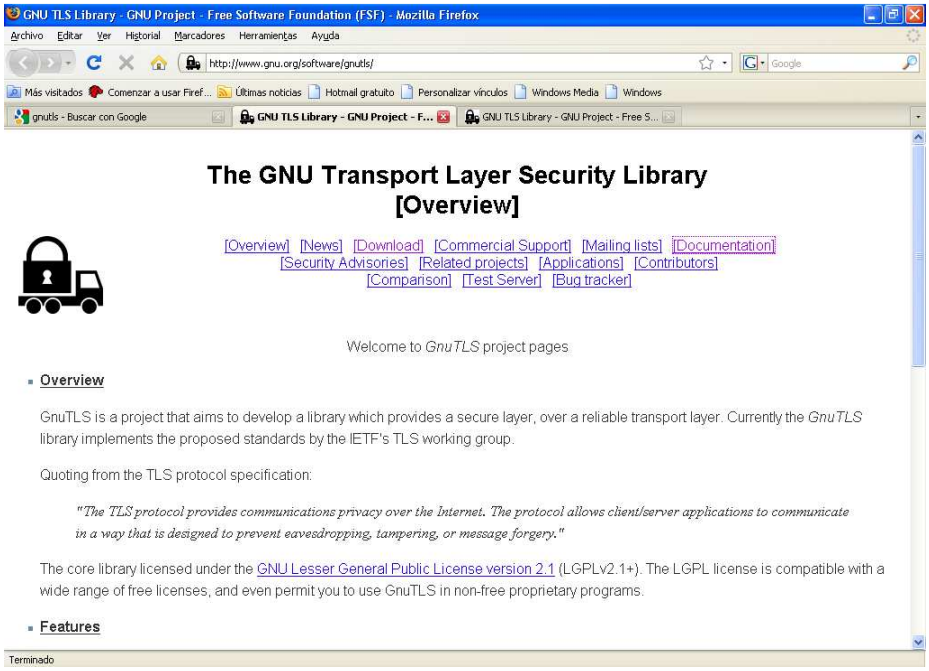

Gráfico III.55: Sitio Web Oficial de la herramienta GnuTLS

La documentación descrita en el sitio web oficial de la herramienta de administración de funciones criptográficas GnuTLS contiene la siguiente información:

- Un manual completo de la herramienta GnuTLS, desarrollado en los formatos: HTML, PDF.

- Un manual de la API (Interfaz de Programación de Aplicación) de la herramienta GnuTLS en formato HTML.
- Un manual de ayuda para desarrollo en entorno GNOME.

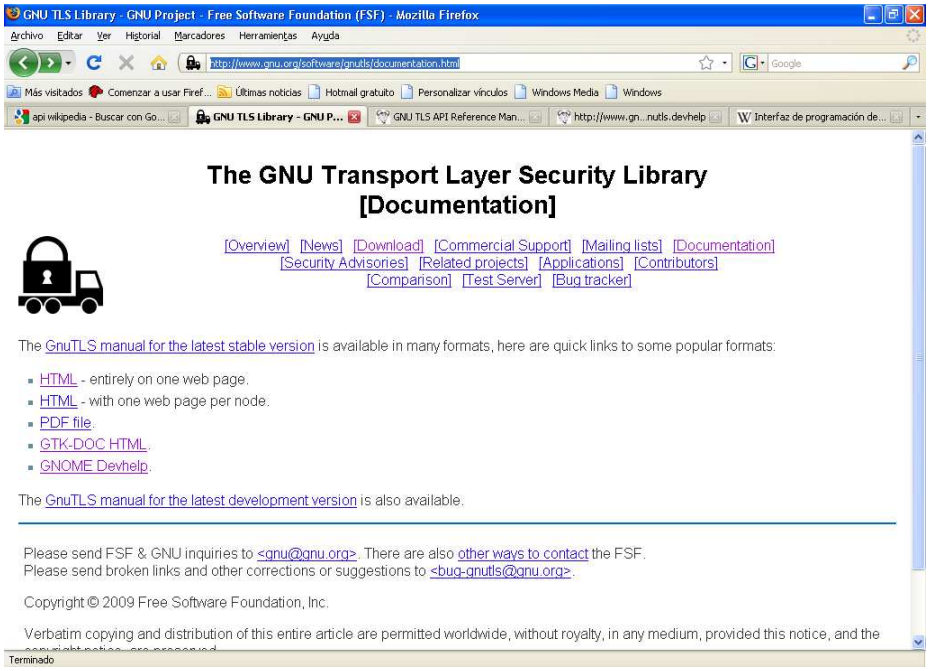

Gráfico III.56: Documentación del Sitio Web de la herramienta GnuTLS

3.13.8.2.2 Páginas MAN de la herramienta GnuTLS

A continuación se puede visualizar la utilidad MAN de la herramienta GnuTLS para que los usuarios puedan digitar de una manera más eficaz los comandos propios de la herramienta, ver el Gráfico III.57.

3.13.8.2.3 Restricciones de la herramienta GnuTLS

La herramienta de administración de funciones criptográficas GnuTLS se encuentra licenciado bajo GNU LGPL y también bajo GNU GPL en las cuales no muestra alguna restricción explícita acerca de la exportación del código fuente.

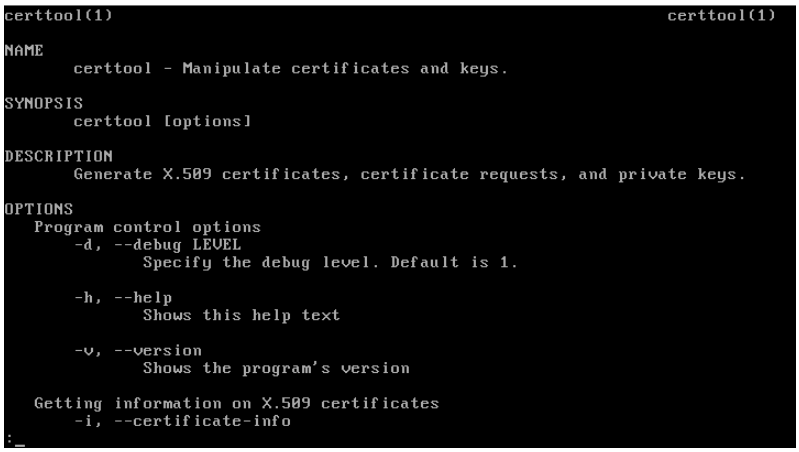

Gráfico III.57: Página MAN de la herramienta GnuTLS

3.13.8.2.4 Licencias de la herramienta GnuTLS

Las dos licencias principales GnuTLS son:

- GNU LGPL (GNU Lesser Library Public License).
- GNU GPL (GNU Library Public License).

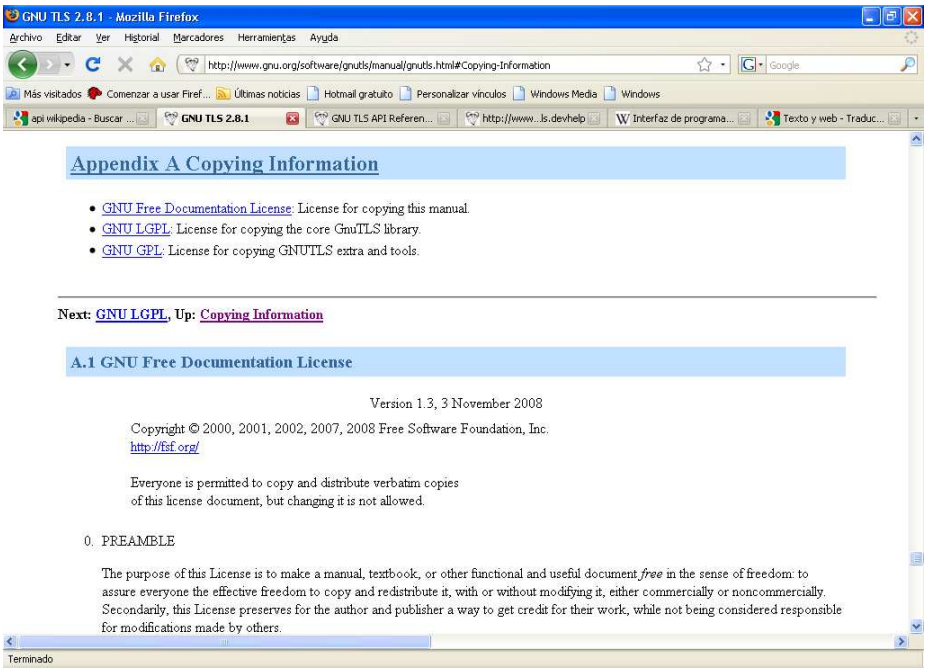

Gráfico III.58: Licencias de la herramienta GnuTLS

3.13.8.3 Soporte técnico oficial de la herramienta de administración de funciones criptográficas NSS

Para desarrollar el módulo de prueba de la herramienta de administración de funciones criptográficas NSS, se hará uso de la información publicada en el sitio web oficial y en la información proporcionada en el código fuente.

3.13.8.3.1 Sitio web oficial de la herramienta NSS

El sitio web oficial de la herramienta de administración de funciones criptográficas NSS es: http://www.mozilla.org/projects/security/pki/nss, al ingresar a este sitio se le presenta la siguiente pantalla:

Gráfico III.59: Sitio Web Oficial de la herramienta NSS

La documentación que provee el sitio web oficial de la herramienta de administración de funciones criptográficas NSS es la siguiente:

- Una vista general de la herramienta NSS.
- Introducción a la criptografía de clave pública.

- Historia de la herramienta NSS.
- Un manual de las APIs de la herramienta NSS.
- Un manual de las herramientas incluidas en la librería NSS y otros detalles técnicos.
- Información de los certificados de los CAs precargadas.

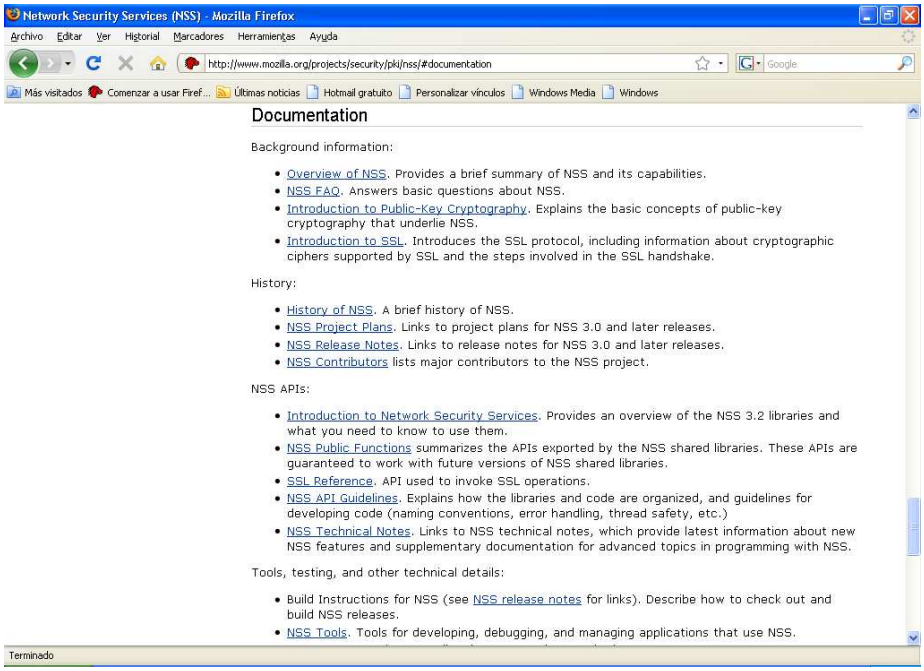

Gráfico III.60: Documentación del Sitio Web de la herramienta NSS

3.13.8.3.2 Páginas MAN de la herramienta NSS

A continuación se muestra que la herramienta de administración de funciones criptográficas NSS no posee la utilidad MAN de su entorno de desarrollo, por lo tanto en el análisis comparativo no se podrá aplicar una calificación.

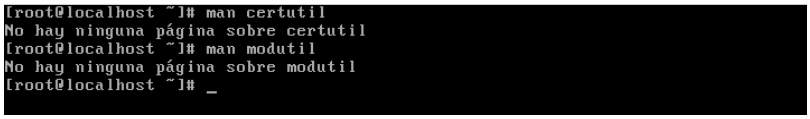

Gráfico III.61: Página MAN de la herramienta NSS

3.13.8.3.3 Restricciones de la herramienta NSS

Debido a que la herramienta de administración de funciones criptográficas NSS está bajo las regulaciones de la administración de exportaciones y otras leyes de Estados Unidos tiene principalmente 4 restricciones:

- Su código fuente no puede ser exportado ni re-exportado a ciertos países, entre ellos: Cuba, Irán, Libya, Corea del Norte, Sudan y Siria .
- No puede ser exportado o re-exportado a personas o entidades que se encuentran en la lista de personas o entidades prohibidas por la oficina de industria y seguridad de los Estados Unidos.
- No puede ser exportado o re-exportado a personas o entidades que se encuentran en la lista de personas o entidades prohibidas por la oficina de control de activos extranjeros.
- No puede ser exportado o re-exportado a personas o entidades involucradas en tecnología nuclear o de armas químicas o biológicas.

3.13.8.3.4 Licencias de la herramienta NSS

La herramienta NSS se encuentra bajo un triple licenciamiento:

- Mozilla Public License
- GNU LGPL (GNU Lesser Library Public License).
- GNU GPL (GNU Library Public License).

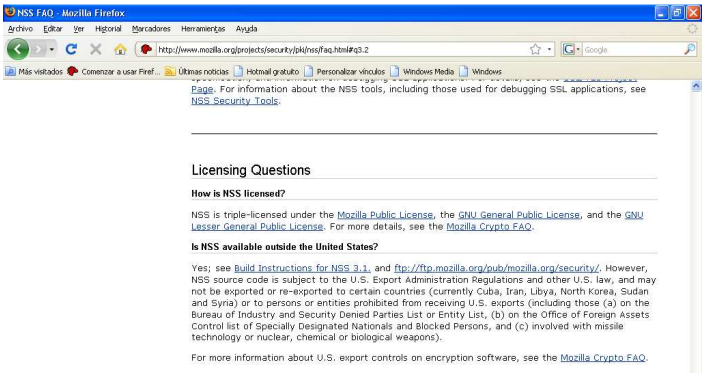

Gráfico III.62: Licencias de la herramienta NSS

3.14 Valorización

Para evaluar los indicadores antes definidos en los módulos correspondientes se empleará la siguiente matriz de valorización con los valores cuantitativos y cualitativos.

Valor cualitativo	**Valor cuantitativo**
Insuficiente	1
Regular	2
Bueno	3
Muy bueno	4
Excelente	5
NA	Sin valor cuantitativo

Tabla III.1: Valorización para el Análisis Comparativo

3.14.1. Insuficiente

Esta calificación se asignará cuando la herramienta no cumpla con el objetivo del indicador. Dicho de otra manera cuando la herramienta no contenga la característica del indicador. El equivalente en valor cuantitativo será igual a 1.

3.14.2. Regular

Este valor cualitativo se asignará a las herramientas que cumplan de forma deficiente con el objetivo del indicador correspondiente. Su valor cuantitativo será igual a 2.

3.14.3. Bueno

La presente calificación se le asignará a las herramientas que cumplan parcialmente con el objetivo del indicador. El equivalente cuantitativo será igual a 3.

3.14.4. Muy Bueno

El valor cualitativo Muy bueno se asignará a las herramientas que cumpla con casi todos los requerimientos del indicador. El valor cuantitativo será igual a 4.

3.14.5. Excelente

Esta calificación se asignará a las herramientas que cumplan a cabalidad con el objetivo del indicador. Su valor cuantitativo será igual a 5.

3.14.6. NA

Se determina NA cuando el indicador no es aplicable asignar una calificación.

3.15 Evaluación de indicadores de los Parámetros del Análisis Comparativo de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS

3.15.1. Evaluación de los Indicadores del Parámetro 1: Instalación de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS

En la siguiente tabla se muestra la calificación que se asignó a los indicadores del Parámetro 1 de las Herramientas de Administración de Funciones Criptográficas:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Dependencias	Bueno	Muy bueno	Bueno
Personalización	Muy bueno	Muy bueno	Regular
Información sobre instalación	Excelente	Muy bueno	Regular
Tiempo de descompresión	Excelente	Muy bueno	Bueno
Tiempo de compilación	Excelente	Muy bueno	NA
Tiempo de construcción	Muy bueno	Bueno	Muy bueno
Tiempo de instalación	Bueno	Muy bueno	NA

Tabla III.2: Análisis Comparativo Parámetro 1

Podemos observar en lo referente al indicador Dependencias que las 3 herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS son equivalentes debido a que cada herramienta tiene sus propias dependencias como requisito para ser instaladas. El indicador Personalización nos da una idea de flexibilidad de las herramientas, en donde la que se encuentra en bajo nivel es la herramienta NSS, al igual que el indicador de información sobre la instalación que viene en los archivos README e INSTALL del código fuente de las herramientas. En lo referente a los tiempos observamos que la herramienta más óptima en su instalación es OpenSSL seguido de GnuTLS, aunque en la herramienta NSS no se puede aplicar para el tiempo de compilación y el tiempo de instalación.

En la siguiente tabla se asigna los valores cuantitativos de acuerdo a la calificación obtenida en los indicadores de cada una de las herramientas de Administración de Funciones Criptográficas:

Herramientas / Indicadores	**OpenSSL**	**GnuTLS**	**NSS**
Dependencias	3	4	3
Personalización	4	4	2
Información sobre instalación	5	4	2
Tiempo de descompresión	5	4	3
Tiempo de compilación	5	4	NA
Tiempo de construcción	4	3	4
Tiempo de instalación	3	4	NA
Total	**29**	**27**	**14**

Tabla III.3: Valores Cuantitativos Análisis Comparativo Parámetro 1

Análisis Comparativo del Parámetro 1: Instalación de las Herramientas de Funciones Criptográficas OpenSSL, GnuTLS y NSS, se observa claramente el mejor performance de la herramienta OpenSSL, seguido de la herramienta GnuTLS y por último con una clara diferencia la herramienta NSS.

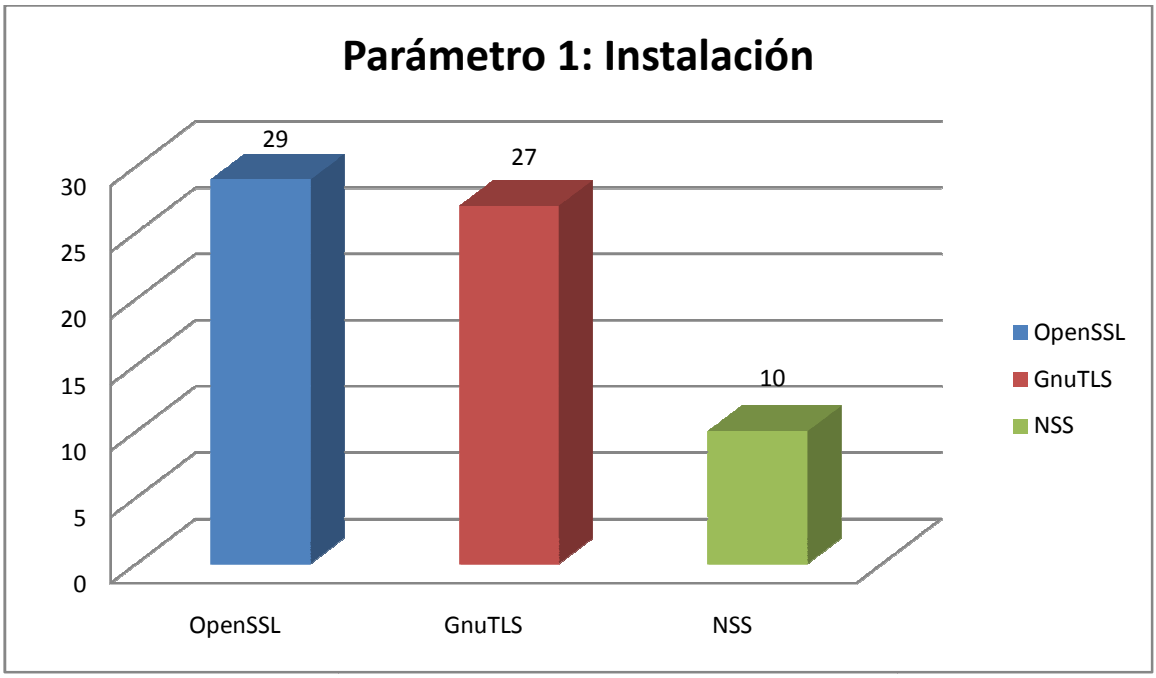

Gráfico III.63: Gráfico Estadístico Análisis Comparativo Parámetro 1

3.15.2. Evaluación de los Indicadores del Parámetro 2: Seguridad de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS

El análisis comparativo del Parámetro 2 lo resumimos en la siguiente tabla:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Protocolos	Excelente	Excelente	Excelente
Longitud de clave de los algoritmos de cifrado simétrico	Muy Bueno	Muy Bueno	Muy Bueno
Longitud de clave de los algoritmos de cifrado asimétrico	Muy Bueno	Muy Bueno	Muy Bueno
Longitud de clave de las funciones hash	Muy Bueno	Muy Bueno	Muy Bueno

Tabla III.4: Análisis Comparativo Parámetro 2

En este parámetro existe una igualdad en la evaluación de los indicadores, ya que las dos herramientas GnuTLS y NSS utilizan el protocolo TLS v1.0 y este protocolo es el equivalente del protocolo SSL v3 que utiliza la herramienta OpenSSL.

En lo que se refiere a la longitud de claves de los algoritmos es irrelevante su comparación ya que el costo de tiempo es directamente proporcional a la longitud de la clave, por lo que el tamaño de cifrado de las claves son usados de acuerdo a la sensibilidad de la información o a su vez de acuerdo a la capacidad de los equipos utilizados.

En la siguiente tabla se asigna la equivalencia cuantitativa obtenida del Análisis Comparativo del Parámetro 2:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Protocolos	5	5	5
Longitud de clave de los algoritmos de cifrado simétrico	4	4	4
Longitud de clave de los algoritmos de cifrado asimétrico	4	4	4
Longitud de clave de las funciones hash	4	4	4
Total	**17**	**17**	**17**

Tabla III.5: Valores Cuantitativos Análisis Comparativo Parámetro 2

A continuación resumiremos el análisis comparativo del Parámetro 2 Funcionalidad de las Herramientas de Funciones Criptográficas OpenSSL, GnuTLS y NSS.

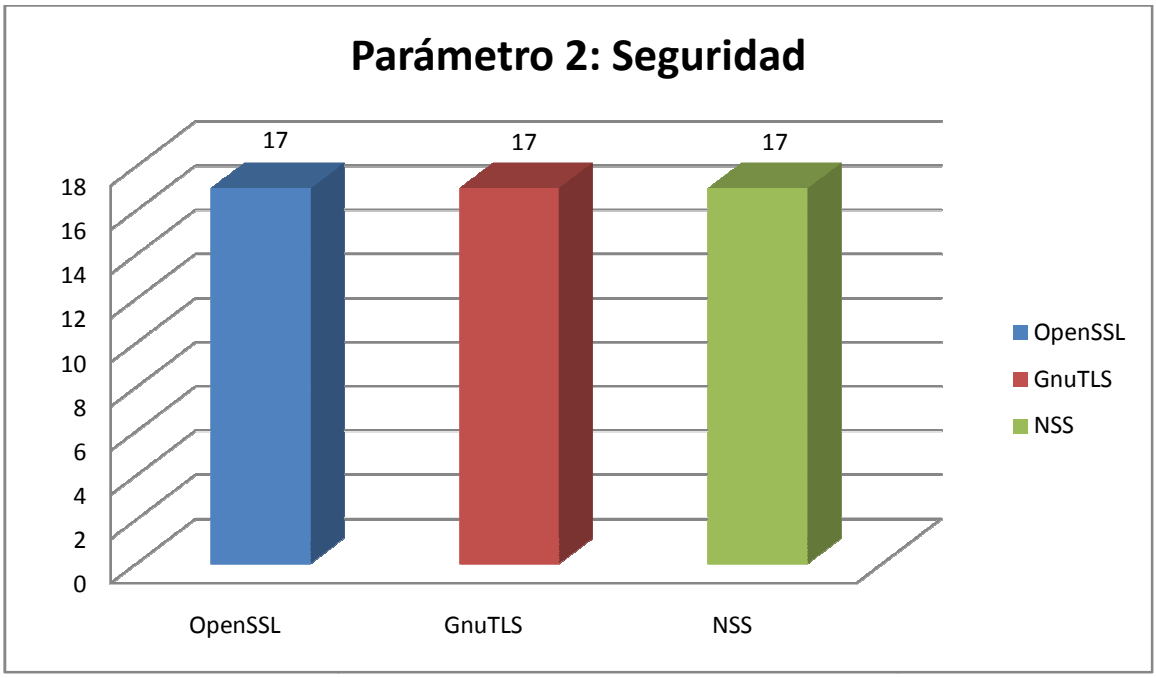

Gráfico III.64: Gráfico Estadístico Análisis Comparativo Parámetro 2

3.15.3. Evaluación de los Indicadores del Parámetro 3: Funcionalidad de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS.

El análisis comparativo del Parámetro 3 lo resumimos en la siguiente tabla:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Generación de claves de los algoritmos simétricos	Excelente	Muy Bueno	Bueno
Generación de claves de los algoritmos asimétricos	Excelente	Muy Bueno	Bueno
Implementación de Autoridades Certificadoras	Excelente	Excelente	Excelente

Tabla III.6: Análisis Comparativo Parámetro 3

La funcionalidad en la generación de claves de los algoritmos simétricos es mejor en la herramienta OpenSSL ya presenta muchos comando para la generación de claves privadas personalizadas, presenta mayor documentación y ayuda que las demás.

La generación de las claves privadas de igual forma OpenSSL posee mayor documentación y ayuda, además las opciones de configuración en la generación de las claves es mayor.

La gestión de AC de las herramientas OpenSSL, GnuTLS y NSS es muy interesante y aunque manejan de forma diferente los certificados digitales las tres presentan mucho interés en la seguridad y seguimiento de los certificados.

Las tres herramientas poseen información sobre la gestión de CA y sus certificados.

A continuación se muestra la equivalencia cuantitativa resultado del Análisis Comparativo del Parámetro 3:

Herramientas / Indicadores	**OpenSSL**	**GnuTLS**	**NSS**
Generación de claves de los algoritmos simétricos	5	4	3
Generación de claves de los algoritmos asimétricos	5	4	3
Implementación de Autoridades Certificadoras	5	5	5
Total	**15**	**13**	**11**

Tabla III.7: Valores Cuantitativos Análisis Comparativo Parámetro 3

Aquí se representa gráficamente el análisis comparativo del parámetro 3: Funcionalidades las Herramientas de Funciones Criptográficas OpenSSL, GnuTLS y NSS.

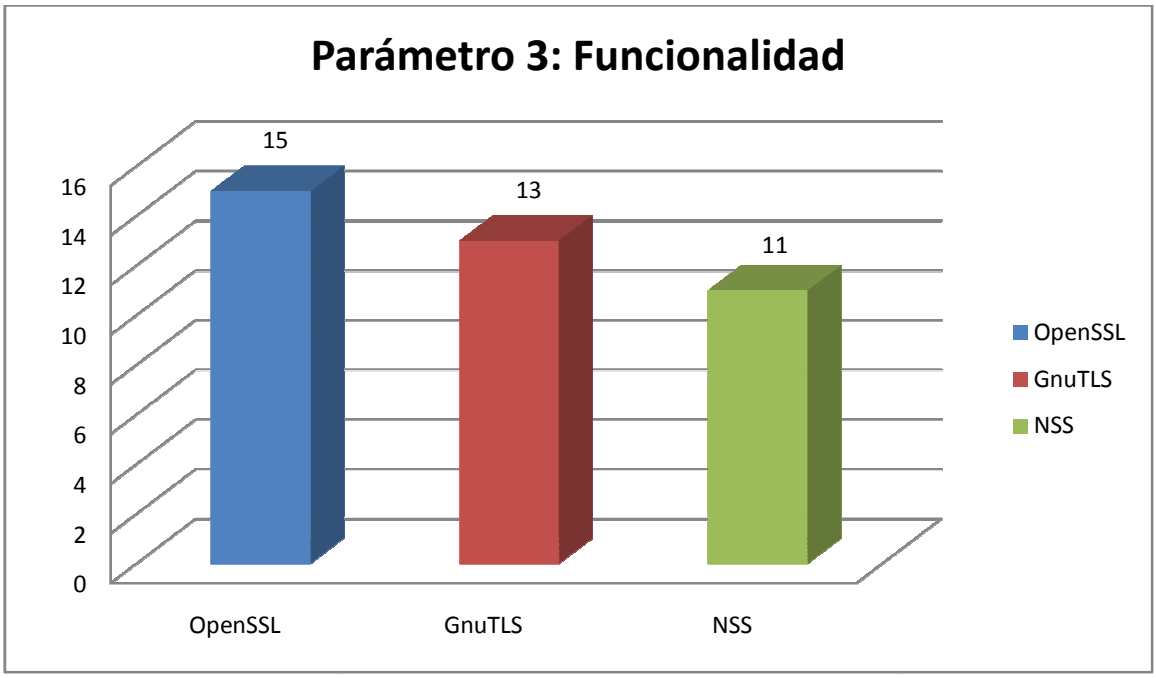

Gráfico III.65: Gráfico Estadístico Análisis comparativo Parámetro 3

3.15.4. Evaluación de los Indicadores del Parámetro 4: Portabilidad de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS

El análisis comparativo del Parámetro 4 lo resumimos en la siguiente tabla:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Plataformas	Excelente	Bueno	Bueno
Aplicaciones extendidas	Excelente	Muy Bueno	Bueno

Tabla III.8: Análisis Comparativo Parámetro 4

OpenSSL está diseñado para ser implantado en mayor número de plataformas y aunque las otras herramientas también son multiplataforma OpenSSL tiene una gran ventaja.

Está muy claro que la herramienta OpenSSL por mayor tiempo en funcionamiento es la que mayor información tiene disponible y por la misma razón es la que hasta la actualidad es la que más se usa para el desarrollo de aplicaciones seguras.

En la siguiente tabla se asigna los valores cuantitativos de acuerdo a la calificación obtenida en el análisis comparativo del Parámetro 4:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Plataformas	5	3	3
Aplicaciones extendidas	5	4	3
Total	**10**	**7**	**6**

Tabla III.9: Análisis Comparativo Parámetro 4

A continuación se muestra la representación gráfica del Parámetro 4: Portabilidad de las herramientas de funciones Criptográficas OpenSSL, GnuTLS y NSS.

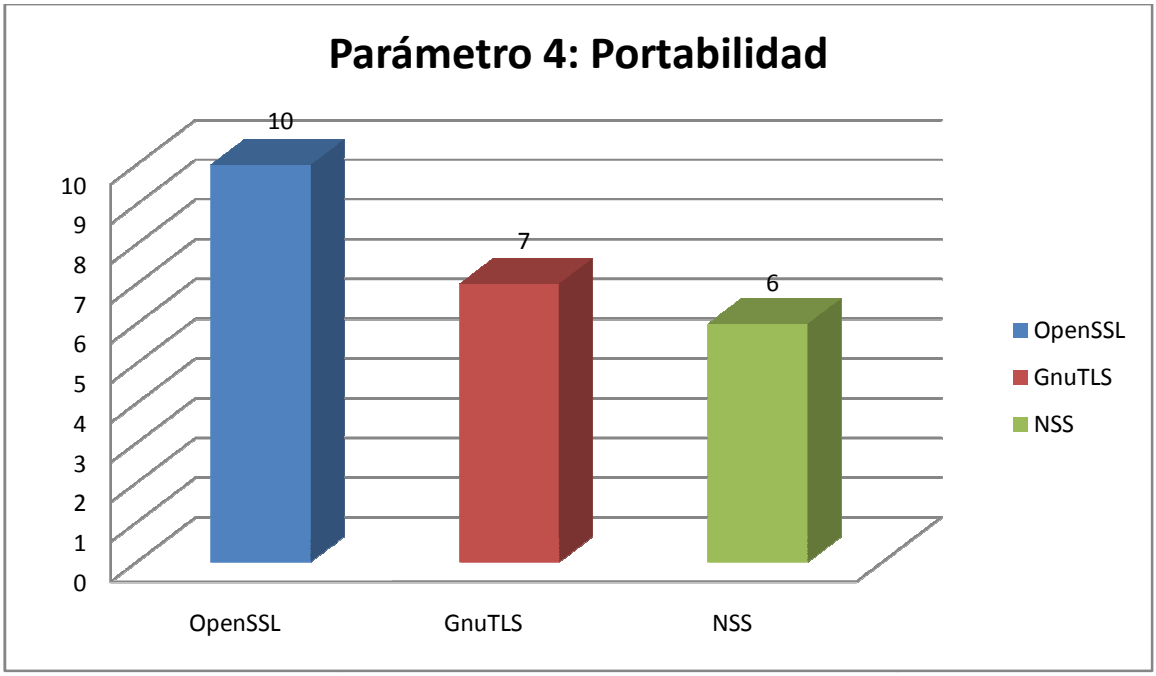

Gráfico III.66: Gráfico Estadístico Análisis Comparativo Parámetro 4

3.15.5. Evaluación de los Indicadores del Parámetro 5: Soporte Técnico y Situación Legal de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS

El análisis comparativo del Parámetro 5 lo resumimos en la siguiente tabla:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Ayuda en línea del Sitio Oficial	Excelente	Bueno	Muy bueno
Páginas de ayuda MAN	Excelente	Excelente	NA
Restricciones	Excelente	Excelente	Bueno
Licencias	Excelente	Excelente	Excelente

Tabla III.10: Análisis Comparativo Parámetro 5

Podemos observar que la herramienta que contiene una mejor estructura de ayuda en línea de su sitio web oficial es la herramienta OpenSSL, debido a que en ella no solo contiene documentación de información de la herramienta como tal sino también información de las librerías principales que utiliza la herramienta OpenSSL y documentación de los conceptos teóricos que se necesita conocer acerca del Protocolo SSL/TLS. En relación a la utilidad MAN se puede observar que el único caso especial es en la herramienta NSS debido a que aquí no se aplica el análisis comparativo. La única herramienta que contiene restricciones explícitas acerca de su código fuente es la herramienta NSS debido a que se encuentra bajo la administración de regulación de exportaciones de los Estados Unidos y por lo tanto su código fuente no puede ser ni exportado ni reexportado a países como Cuba, Irán, Libia, Siria, etc. En términos generales en lo referente al indicador Licencias las tres herramientas se encuentran bajo código libre, debido a que ninguna de ellas restringe la utilización de sus respectivas herramientas.

A continuación se puede apreciar su equivalencia cuantitativa del Análisis comparativo del Parámetro 5:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Ayuda en línea del Sitio Oficial	5	3	4
Páginas de ayuda MAN	5	5	NA
Restricciones	5	5	3
Licencias	5	5	5
Total	**20**	**18**	**12**

Tabla III.11: Valores Cuantitativos Análisis Comparativo Parámetro 5

Aquí se representa gráficamente el análisis comparativo del Parámetro 5: Soporte Técnico y Situación Legal de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS.

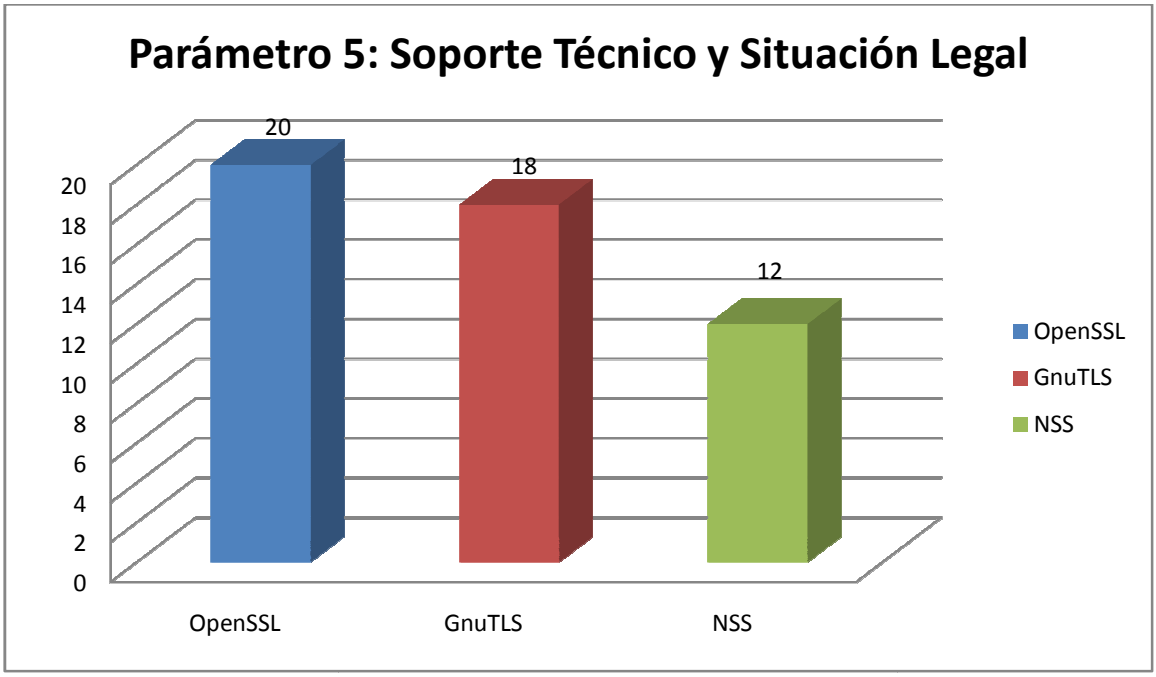

Gráfico III.67: Gráfico Estadístico Análisis Comparativo Parámetro 5

3.16 Matriz de Valorización: Análisis Comparativo de los Indicadores de los Parámetros de las Herramientas de administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS

Luego de experimentar el comportamiento de cada una de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS en los módulos de prueba desarrollados anteriormente podemos cuantificar la calificación de los indicadores de los 5 parámetros que conforman el análisis comparativo, a continuación se presenta la matriz de valorización:

Herramientas / Indicadores	OpenSSL	GnuTLS	NSS
Dependencias	3	4	3
Personalización	4	4	2
Información sobre instalación	5	4	2
Tiempo de descompresión	5	4	3
Tiempo de compilación	5	4	NA
Protocolos	5	5	5
Longitud de clave de los algoritmos de cifrado simétrico	4	4	4
Longitud de clave de los algoritmos de cifrado asimétrico	4	4	4
Longitud de clave de las funciones hash	4	4	4
Generación de claves de los algoritmos simétricos	5	4	3
Generación de claves de los algoritmos asimétricos	5	4	3
Implementación de Autoridades Certificadoras	5	5	5
Plataformas	5	3	3
Aplicaciones	5	4	3

extendidas			
Ayuda en línea del Sitio Oficial	5	3	4
Páginas de ayuda MAN	5	5	NA
Restricciones	5	5	3
Licencias	5	5	5
Total	**84**	**75**	**56**

Tabla III.12: Matriz de Valorización Análisis comparativo de las herramientas OpenSSL, GnuTLS y NSS

Haciendo uso de la estadística analítica podemos representar gráficamente el total de los valores cuantitativos de los parámetros del análisis comparativo de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS:

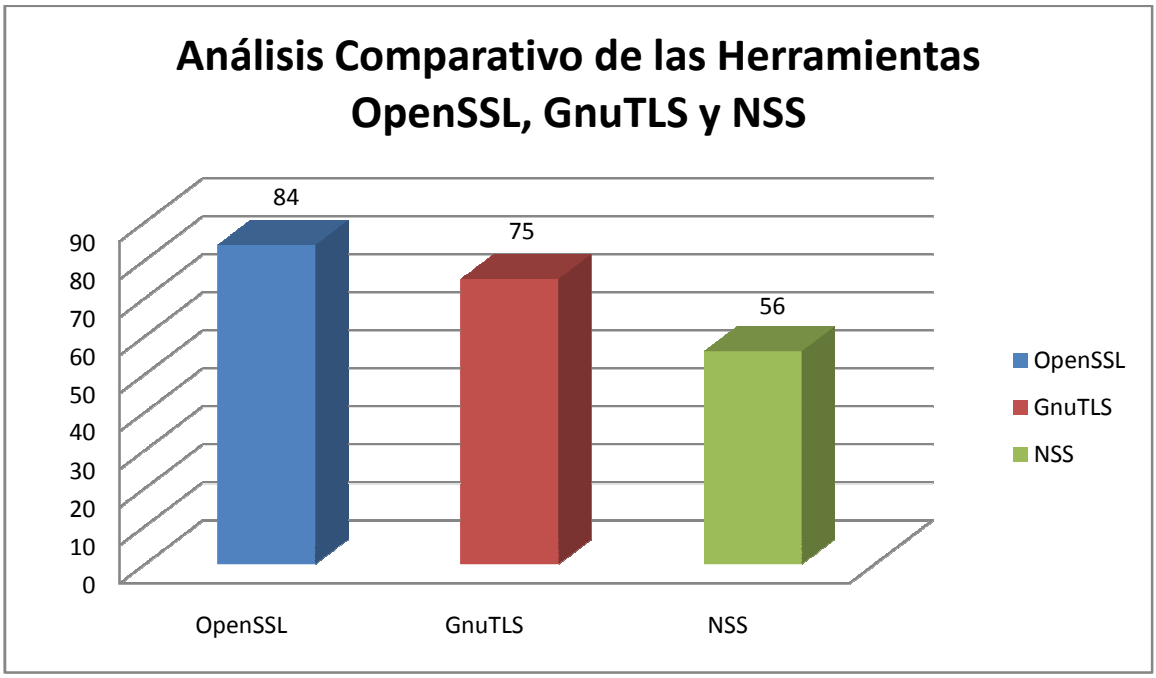

Gráfico III.68: Gráfico Estadístico Análisis Comparativo de las Herramientas OpenSSL, GnuTLS y NSS

3.17 Comprobación de hipótesis y resultados

3.17.1. Hipótesis

Mediante el análisis y comparación de las herramientas de código libre para la administración de funciones criptográficas: OpenSSL, GnuTLS y NSS en los protocolos SSL y TLS, permitirá seleccionar la herramienta más adecuada para asegurar la transmisión de la información en aplicaciones web.

3.17.2. Análisis Comparativo en porcentajes de las Herramientas OpenSSL, GnuTLS y NSS.

Gráfico resultado del Análisis Comparativo de las Herramientas de Administración de Funciones Criptográficas OpenSSL, GnuTLS y NSS expresado en porcentajes:

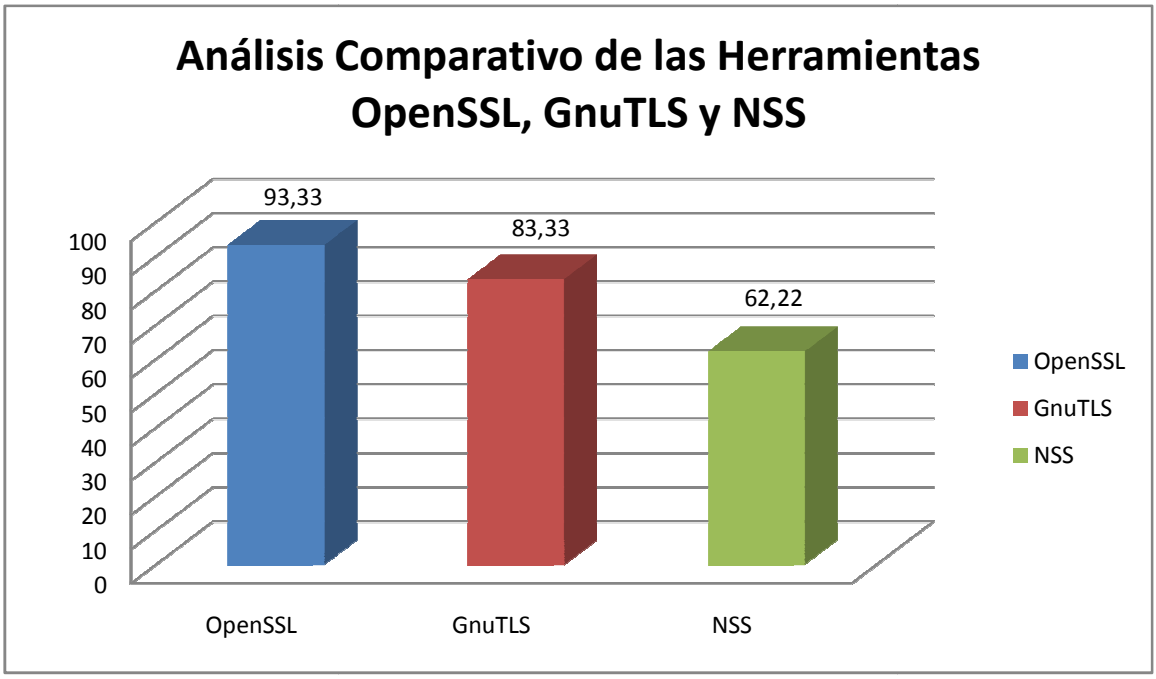

Gráfico III.69: Gráfico Estadístico Análisis Comparativo en porcentajes de las Herramientas OpenSSL, GnuTLS y NSS

3.17.3. Resultados Obtenidos

- OpenSSL es la herramienta que obtiene el mejor resultado con un 93.33% debido a características como la facilidad de uso, difusión, mejor soporte técnico y su funcionalidad. Con un 83.33% GnuTLS está ubicado en la segunda posición y NSS en la tercera posición con un 62.22%.
- Después de un análisis comparativo del parámetro de instalación se puede verificar que la herramienta con mejor performance es OpenSSL por sus tiempos en la instalación, seguido muy de cerca de la herramienta GnuTLS. En lo referente a la herramienta NSS presenta inconvenientes en su análisis porque no se aplica la evaluación a varios indicadores.
- La seguridad de las tres herramientas comparada desde el punto de vista de la fortaleza de sus algoritmos criptográficos y los protocolos implementados por dichas herramientas se puede decir que existe un empate técnico, debido a que GnuTLS y NSS utiliza por defecto el protocolo TLS 1.0 que es equivalente al protocolo SSL 3.0 que utiliza por defecto OpenSSL. En cuanto a la implementación de algoritmos criptográficos y la longitud de sus claves trabajan de forma muy parecida.
- OpenSSL funcionalmente es mejor porque presenta un mayor número de sub herramientas que permite la creación de claves públicas, privadas y certificados digitales firmados por una propia CA de forma más transparente y organizada, GnuTLS está ubicado en segundo lugar por la documentación que presenta de las sub herramientas implementadas, y NSS tiene el tercero ya que su documentación no es muy comprendida por lo que es menos amigable.
- OpenSSL es más portable ya que es posible instalar en un mayor número de sistemas operativos y de una forma más sencilla, además es usado en un mayor número de aplicaciones que implementan seguridad ocultando la información. GnuTLS y NSS le siguen obviamente por tener un menor número de aplicaciones que usan sus librerías lo cual se puede deber a su tiempo de vida.

- En lo referente al soporte técnico se puede deducir que OpenSSL contiene el mejor soporte técnico oficial en línea e información no solo de las librerías de la herramienta, sino también de conceptos previos que nos ayudan a entender mejor el funcionamiento, es la herramienta que contiene menos restricciones desde el punto de vista legal, en segunda posición la herramienta GnuTLS debido a su buena organización de la información y disponible en diferentes formatos en su sitio web oficial. Finalizando el análisis la herramienta NSS debido a las restricciones en la exportación y reexportación en su código fuente.

3.17.4. Conclusión de la Comprobación de la Hipótesis

Después de realizar el análisis comparativo de las herramientas de código libre para la administración de funciones criptográficas: OpenSSL, GnuTLS y NSS en los protocolos SSL y TLS se concluye que la herramienta más idónea en base a los indicadores planteados es OpenSSL con un 93.33% por características como la facilidad de uso, difusión, mejor soporte técnico y funcionalidad en el aseguramiento de la transmisión de información de aplicaciones web.

CAPÍTULO IV APLICACIÓN PORTAL WEB JPTCH

4.1 Introducción

En el presente capítulo se pone en marcha la parte aplicativa del proyecto, es decir la instalación y configuración de la herramienta de administración de funciones criptográficas OpenSSL y el módulo mod_ssl que permite la interacción entre la herramienta y el servidor web apache.

Se desarrolla la aplicación web “SONIA” que es el portal web de la Jefatura Provincial de Tránsito de Chimborazo y cuyo dominio es www.jctch.gov.ec.

Se presenta la documentación de la aplicación web “SONIA” que consiste en: Documentación Técnica, Manual de Instalación y Manual de Usuario.

Para finalizar comprobamos la hipótesis mediante el resultado del análisis comparativo de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS.

4.2 Parte aplicativa de la herramienta de administración de funciones criptográficas OpenSSL.

4.2.1. Instalación de OpenSSL-0.9.8j

Openssl es la herramienta que administra funciones criptográficas y permite crear autoridades certificadoras, claves privadas y certificados digitales para la configuración de un servidor web seguro. Para su instalación se recomienda el proceso siguiente.

Descomprimimos el archivo openssl-0.9.8j.tar.gz bajo el directorio /usr/local/src/

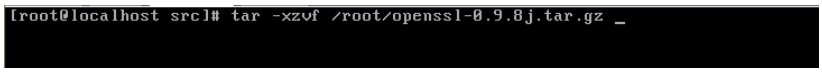

Gráfico IV.1: Descompresión del paquete openssl-0.9.8j.tar.gz

Nos ubicamos bajo el directorio descomprimido /usr/local/src/openssl-0.9.8j y ejecutamos el comando ./config

Gráfico IV.2: Configuración de la instalación de OpenSSL

Luego ejecutamos el comando make, make test y make install.

Gráfico IV.3: Creación e Instalación de OpenSSL

4.2.2. Instalación y configuración del servidor web Apache httpd-2.0.63

Para instalar httpd se puede descargar los paquetes desde la página oficial de apache descomprimir y ubicarse bajo el directorio para ejecutar los siguientes comandos.

```
[root@localhost httpd-2.0.63]# ./configure --enable-mods-shared=all --enable-ssl
=shared --enable-so_
[root@localhost httpd-2.0.63]# make install_

[root@localhost httpd-2.0.63]# make_
```

Gráfico IV.4: Instalación del servidor web Apache

4.2.3. Configuración del servidor web Apache httpd-2.0.63

Para la configuración del servidor es necesario asignar la dirección IP 192.168.0.1 con una máscara de red 255.255.255.0.

El principal archivo de configuración es httpd.conf ubicado en /usr/local/apache2/conf

Primeramente debemos asignar el puerto por el que escuchará las peticiones el servidor que es 80 y que está por defecto.

```
#
#Listen 12.34.56.78:80

Listen 80
```

Gráfico IV.5: Configuración del puerto que escucha el servidor web no seguro

En la sección donde se cargan los módulos debemos incluir el módulo mod_ssl.

```
LoadModule expires_module modules/mod_expires.so
LoadModule headers_module modules/mod_headers.so
LoadModule usertrack_module modules/mod_usertrack.so
LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule version_module modules/mod_version.so
<IfDefine SSL>
LoadModule ssl_module modules/mod_ssl.so
</IfDefine>
LoadModule mime_module modules/mod_mime.so
LoadModule dav_module modules/mod_dav.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule asis_module modules/mod_asis.so
LoadModule info_module modules/mod_info.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
LoadModule imap_module modules/mod_imap.so
LoadModule actions_module modules/mod_actions.so
LoadModule speling_module modules/mod_speling.so
LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule php5_module        modules/libphp5.so
```

Gráfico IV.6: Inclusión del módulo mod_ssl en httpd.conf

Definimos el directorio donde se cargara la aplicación en la variable DocumentRoot.

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
#DocumentRoot "/usr/local/apache2/htdocs"
DocumentRoot "/sonia/sonia"
```

Gráfico IV.7: Configuración del directorio por defecto para cargar la aplicación web

Creamos un directorio virtual para los archivos que serán almacenados para asignar permisos de acceso.

```
#Directorio virtual

Alias /sonia_archivos/ "/sonia_archivos/"
<Directory "/sonia_archivos/">
   #Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
```

Gráfico IV.8: Creación de un directorio virtual en httpd.conf

Para que el servidor pueda mostrar los caracteres especiales es necesario incluir AddDefaultCharset On como se indica.

```
#para controlar los caracteres como las tildes y enies
<Directory />
    AddDefaultCharset On
</Directory>
AddDefaultCharset ISO-8859-1
#
```

Gráfico IV.9: Configuración de httpd.conf para mostrar caracteres especiales como vocales con tildes

Incluimos el archivo de configuración del servidor seguro.

```
#
# Bring in additional module-specific configurations
#
<IfModule mod_ssl.c>
    Include conf/ssl.conf
</IfModule>
```

Gráfico IV.10: Inclusión del fichero ssl.conf en httpd.conf

4.2.4. Instalación del módulo mod_ssl

El módulo mod_ssl vine incluido en el paquete de instalación de apache-2.0.63, es necesario compilar el código con la opción –enable-ssl.

4.2.5. Instalación y configuración de php-5.2.9

Requerimientos:

libjpeg-6b

libpng-1.2.37

freetype-2.3.7

t1lib-5.1.2

Instalación de php-5.2.9 con sus respectivos módulos, los mismos que permitirán el manejo de funciones de compresión, creación de formatos .pdf, interacción con el gestor de base de datos MySQL y edición de fotos en varios formatos.

```
[root@localhost php-5.2.9]# ./configure --with-apxs2=/usr/local/apache2/bin/apxs
 --with-gd --with-gettext --with-jpeg-dir=/usr/local/lib --with-kerberos --with-
mysql --with-pear --with-png-dir=/usr/local/lib --with-zlib --with-zlib-dir=/usr
/local/lib --enable-zip --with-openssl --enable-bcmath --enable-calendar --enabl
e-ftp --enable-magic-quotes --enable-sockets --enable-mbstring --with-iconv --en
able-dba --with-db4 --with-freetype-dir=/usr/include/freetype2/ --with-ttf --ena
ble-gd-native-ttf --with-t1lib=/usr/local/_
```

Gráfico IV.11: Configuración de PHP antes de la instalación

Luego de la configuración del código es necesario ejecutar los comandos make y make install para finalizar la instalación.

Luego de finalizar la instalación se debe copiar el archivo php.ini-dist bajo el directorio lib de la siguiente forma /usr/local/lib/php.ini

```
[root@localhost local]# cp php.ini-dist /usr/local/lib/php.ini_
```

Gráfico IV.12: Configuración de PHP

Posterior es necesario cargar el módulo php en el archivo de configuración httpd.conf

```
LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule php5_module        modules/libphp5.so
```

Gráfico IV.13: Configuración de Apache con PHP

También configuramos httpd.conf para que reconozca los archivos de php.

```
#
AddType application/x-httpd-php .php

AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
```

Gráfico IV.14: Configuración de Apache con PHP

4.2.6. Configuración del servidor seguro HTTPS con la herramienta OpenSSL, el módulo mod_ssl.

4.2.6.1 Creación de una Autoridad Certificadora

La creación de la autoridad certificadora es necesaria cuando no se posee un certificado firmado por alguna AC pública o para la realización de pruebas. En este caso generamos una CA para generar los certificados digitales ya que aún no se posee un certificado digital.

Para esto ejecutamos el comando OpenSSL seguido de otro comando req en el cual solicita un certificado digital x509 con una clave privada generada con el algoritmo rsa utilizando 2048 bits, este comando genera dos archivos, la clave privada y certificado.

```
[root@localhost prueba1]# openssl req -x509 -newkey rsa:2048 -keyout cakey.pem -d
ays 3650 -out cacert.pem
Generating a 2048 bit RSA private key
..+++
..........+++
writing new private key to 'cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:ec
State or Province Name (full name) [Berkshire]:chimborazo
Locality Name (eg, city) [Newbury]:riobamba
Organization Name (eg, company) [My Company Ltd]:JPCTSUCH
Organizational Unit Name (eg, section) []:UIAT
Common Name (eg, your name or your server's hostname) []:manuel haro
Email Address []:manha84@yahoo.com
[root@localhost prueba1]# _
```

Gráfico IV.15: Creación de una CA

4.2.6.2 Creación de una clave privada y un certificado digital

Se genera la clave privada de 2048 bits con el algoritmo 3des la que servirá para generar el certificado digital.

Gráfico IV.16: Generación de la clave privada

4.2.6.3 Generación del certificado digital firmado por una CA propia.

Gráfico IV.17: Generación del certificado digital

Se debe copiar la clave y el certificado digital en un lugar seguro para luego enlazar desde el archivo de configuración ssl.conf

4.2.6.4 Configuración del archivo ssl.conf

El archivo principal de configuración de el servidor seguro es el ssl.conf que se encuentra bajo el directorio /usr/local/apache2/conf/ y que sirve para configurar un servidor virtual que escucha en el puerto 443.

En este archivo es necesario revisar el puerto en el que escucha este servicio, aunque se puede asignar otro puerto que no coincida con los estándares de facto que utilizan los diferentes servicios como mail, ftp, etc.

```
#       but a statically compiled-in mod_ssl.
#
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512

<IfDefine SSL>

#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
# Note: Configurations that use IPv6 but not IPv4-mapped addresses need two
#       Listen directives: "Listen [::]:443" and "Listen 0.0.0.0:443"
#
Listen 443
```

Gráfico IV.18: Puerto en el que escucha el servidor seguro

En esta parte del archivo se configura la variable DocumentRoot en donde se alojará la aplicación y otros parámetros propios y opcionales que se muestran.

También se indica la ruta donde se encuentran alojadas la clave privada y el certificado digital con el que trabajara el servidor, esto se realiza en las variables SSLCertificateFile y SSLCertificateKeyFile.

```
## SSL Virtual Host Context
<VirtualHost _default_:443>
#   General setup for the virtual host
DocumentRoot "/sonia/ssonia/"
#ServerName www.example.com:443
#ServerAdmin you@example.com
ErrorLog /usr/local/apache2/logs/error_log
TransferLog /usr/local/apache2/logs/access_log

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

#   Server Certificate:
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/cert.pem
#SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server-dsa.crt

#   Server Private Key:
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/servidor.key
#SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server-dsa.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
```

Gráfico IV.19: Asignación el certificado digital y la clave privada en el fichero ssl.conf

4.2.6.5 Implantación de la aplicación SONIA

Creación y configuración de los directorios

Para la implantación de la aplicación es necesario crear el siguiente árbol de directorios y los permisos de lectura, escritura y ejecución únicamente para el usuario root, para los otros usuarios lectura y la ejecución.

```
/sonia
      sonia
      ssonia
```

En estos directorios se aloja la aplicación.

Para los passwords de los usuarios de base de datos también es necesario crear el directorio /um que contienen los archivos camP.info y camA.info con los datos de los usuarios y passwords de la base de datos bdSonia, especificado de la siguiente forma:

Gráfico IV.20: Fichero con para conexión a la base de datos

El directorio sonia_archivos contiene los archivos generados por la aplicación y los que se sube desde el usuario.

```
/sonia_archivos
      descargas
      estadísticas
      fotos_jefatura
      fotos_it
      fotos_uiat
      informes_uiat
```

```
      noticias_jefatura

      noticias_it

      noticias_uiat
```

4.2.6.6 Subir los archivos de la aplicación

Con la ayuda de una herramienta se debe subir la aplicación al sitio indicado anteriormente.

4.2.6.7 Restauración de la Base de datos bdSonia.sql

De igual forma se debe restaurar la base de datos con la ayuda de alguna herramienta gráfica o mediante los siguientes comandos.

Para restaurar la base de datos primero debemos crear la base de datos

```
#mysqladmin –u root –p create bdSonia
```

Restauramos la base de datos

```
#mysql –u root –p  - -opt bdSonia  > bdSonia.back
```

4.2.6.8 Ejecución de la aplicación

Una vez implantada la aplicación con los pasos detallados anteriores, configuramos para que esta se inicie automáticamente cuando el servidor se reinicie e iniciamos los servicios.

Para el gestor de base de datos MySQL

```
chkconfig mysqld on

service mysqld start
```

Para el servidor web seguro es necesario generar un archivo con la contraseña para que este inicie sin solicitarla, y se asigna la ruta del archivo en la variable SSLPassPhraseDialog.

Ejemplo del contenido del fichero que almacena la contraseña, por seguridad se debe dar permisos únicamente al usuario root.

```
#!/bin/sh


echo mipassword
```

Variable con la ruta del archivo en el fichero ssl.conf

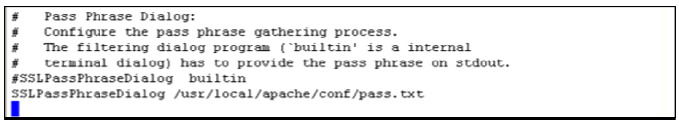

Gráfico IV.21: Configuración del fichero ssl.conf para asignar la contraseña

Para que el servicio de httpd se inicie automáticamente es necesario editar el archivo /etc/rc.local de la siguiente forma.

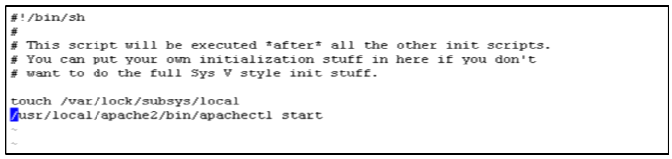

Gráfico IV.22: Configuración del archivo rc.local

Ahora iniciamos el servido httpd o reiniciamos la máquina.

```
/usr/local/apache2/bin/apachectl start
```

El certificado que se muestra en el cliente es el siguiente:

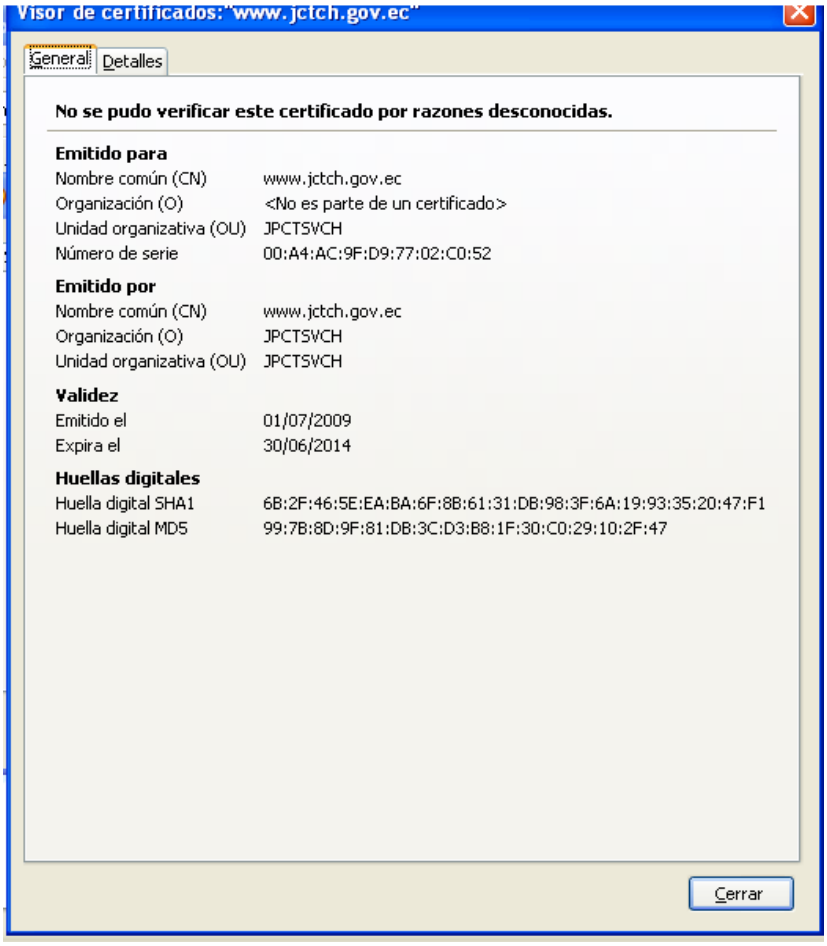

Gráfico IV.23: Detalle del certificado en el cliente

En el Gráfico IV.23 se puede observar los datos del certificado que se obtiene en el browser del cliente, en el cual se especifica que el certificado no ha sido verificado por razones desconocidas, pues la razón es que este certificado no está registrado en una autoridad certificadora.

4.3 Documentación técnica de la aplicación web "SONIA"

4.3.1. Introducción

El presente documento se genera debido a la imperiosa necesidad de documentar técnicamente la aplicación web para la administración de informes de peritaje,

estadísticas, noticias de forma segura utilizando el protocolo TLS denominada “SONIA”, que se desarrolló bajo la metodología XP.

Para empezar se documenta la fase 1 que es la Planificación y Análisis del proyecto, se toma muy en cuenta los riesgos que puede implicar en el desarrollo del proyecto, luego mediante una tabla de valores y un análisis se los clasifica de acuerdo a su grado de probabilidad de impacto. Se define y se detalla tanto los requerimientos funcionales como los no funcionales al igual que los requerimientos del sistema.

En la fase 2 se realiza el diseño de la aplicación web “SONIA “, para ello se diseña la base de datos, diagrama de clases y demás diagramas que ayudan al diseño correcto y conceptual de la aplicación.

En la fase 3 se plasma en código fuente, de acuerdo al diseño de la aplicación.

Finalmente en la fase 4 se realiza pruebas exhaustivas del código desarrollado en la fase previa, en donde si el código fuente no pasa la o las pruebas se lo envía a la fase 3 para su corrección. Dentro de estas cuatro fases muchas veces se las realizará de forma iterativa y un número indeterminado de veces hasta llegar a superar la fase de pruebas.

4.3.2. Planificación y Análisis: Documento SRS

4.3.2.1 Introducción

El propósito del presente documento es la especificación de los requerimientos de software del sitio web de la Jefatura Provincial de Tránsito de Chimborazo, para ello se debe realizar un proceso que ayude a detallar y desarrollar los mismos.

En el desarrollo del proceso se debe tomar muy en cuenta los posibles riesgos que se pueden presentar a los largo del desarrollo del sitio web, teniendo presente su grado de riesgo que pueden implicar grandes cambios dentro de los elementos participantes en la construcción.

La esencia del documento será la formulación y la descripción de los requerimientos funcionales, para lo cual se hará uso de los casos de uso, sus actores, para determinar de una manera clara, concisa y entendible tanto para sus desarrolladores como para sus futuros usuarios, debido a que en este punto se basa el contrato entre los usuarios del sitio web con los desarrolladores.

Además de los requerimientos funcionales se debe contemplar los no funcionales, es decir los requerimientos del sistema y requerimientos de interfaz.

4.3.2.2 Riesgos

4.3.2.2.1 LISTADO DE RIESGO

Listado de Riesgos	
Id	**Descripción**
RI-1	El equipo de desarrollo puede no adaptarse rápidamente a las tecnologías como Ajax, Javascript, sobre las cuales se desea construir el sitio web de la JPTCH.
RI-2	Los desarrolladores pueden tener retrasos en el cronograma de desarrollo debido a un análisis inadecuado del sitio web de la JPTCH.
RI-3	La comunicación entre los desarrolladores del sitio web y el personal policial sea complicada debido a los horarios y actividades exigentes de los mismos.
RI-4	Al estar conectados los equipos de desarrollo al internet su vulnerabilidad es expuesta a los diferentes tipos de peligro existentes en la gran red de información.
RI-5	En los equipos de desarrollo se puede presentar daños imprevistos en el hardware.
RI-6	Cambios en los requerimientos por parte del personal policial de la JPTCH.

Tabla IV.1: Tabla de Riesgos

A continuación se procede a valorar los riegos determinados anteriormente

4.3.2.2.2 Tabla de Valores

Descripción	Valor
Alto	3
Medio	2
Bajo	1

Tabla IV.2: Tabla de Valores de Riesgos

4.3.2.2.3 Análisis de Riesgo

Id	Probabilidad			Impacto	
	%	Valor	Probabilidad	Valor	Impacto
RI-1	50	2	Media	2	Medio
RI-2	70	3	Alta	2	Medio
RI-3	50	2	Media	1	Bajo
RI-4	40	2	Media	1	Bajo
RI-5	30	1	Baja	1	Baja
RI-6	50	2	Media	2	Medio

Tabla IV.3: Tabla de Análisis de Riesgos

4.3.2.2.4 Resultados

Id	Exposición al Riesgo	
	Valor	Exposición
RI-2	6	Alta
RI-1	4	Media
RI-6	4	Media
RI-3	2	Baja
RI-4	2	Baja
RI-5	1	Baja

Tabla IV.4: Tabla de Resultados del Análisis de Riesgos

Como resultado de este análisis se toma la decisión de gestionar los riesgos con exposición Media y Alta

4.3.2.3 Restricciones

Listado de Restricciones	
Id	**Descripción**
RE-1	El desarrollo del presente proyecto se debe mostrar como resultado el sitio web de la JPTCH.
RE-2	El sitio web resultado del presente proyecto no es el sitio web oficial de la JPTCH.
RE-3	El sitio web no contempla un módulo referente a licencias y matriculación de autos debido a la creación del Consejo Nacional de Tránsito.

Tabla IV.5: Tabla de Restricciones

4.3.2.4 Requerimientos Funcionales

4.3.2.4.1 Actores

Listado de Actores		
Id	**Nombre**	**Descripción**
AC-1	ADMINSTRADOR_SITIO	Es el responsable de la seguridad del sitio web.
AC-2	ADMINISTRADOR_DEPARTAMENTO	Es el responsable de la información de cada departamento de la JPTCH.
AC-3	USUARIO_PUBLICO	Es el visitante al sitio web de la JPTCH.
AC-4	ADMINISTRADOR_JPTCH	Administra la información publicada en el sitio web de la JPTCH.
AC-5	ADMINISTRADOR_SIAT	Administra la información publicada en el sitio web del departamento SIAT de la JPTCH.
AC-6	ADMINISTRADOR_RP	Administra la información publicada en el departamento RP del sitio web de la JPTCH.
AC-7	ADMINISTRADOR_IT	Administra la información publicada en el departamento IT del sitio web de la JPTCH.
AC-8	ADMINISTRADOR_FISCALIA	Administra la información receptada db de la JPTCH.

Tabla IV.6: Tabla de Actores de Casos de Uso

4.3.2.4.2 Diagramas de Casos de Uso

- Diagrama de Caso de Uso General

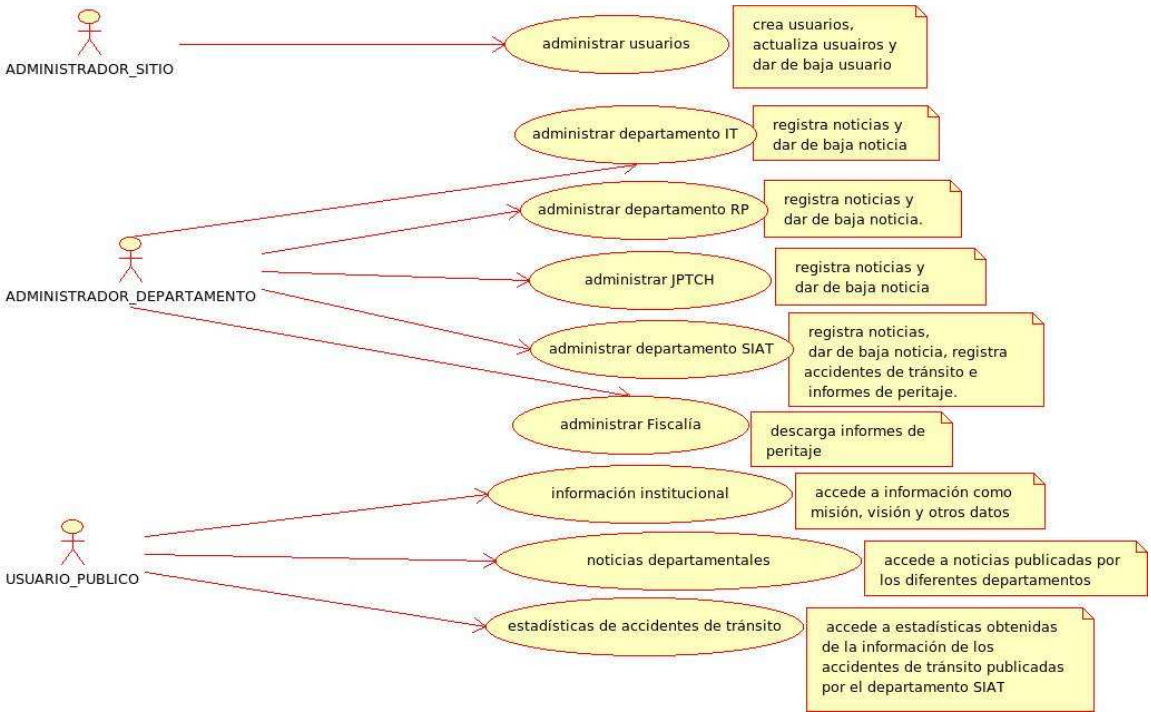

Gráfico IV.24: Diagrama de Caso de Uso General

- Diagrama de Caso de Uso Administración JPTCH

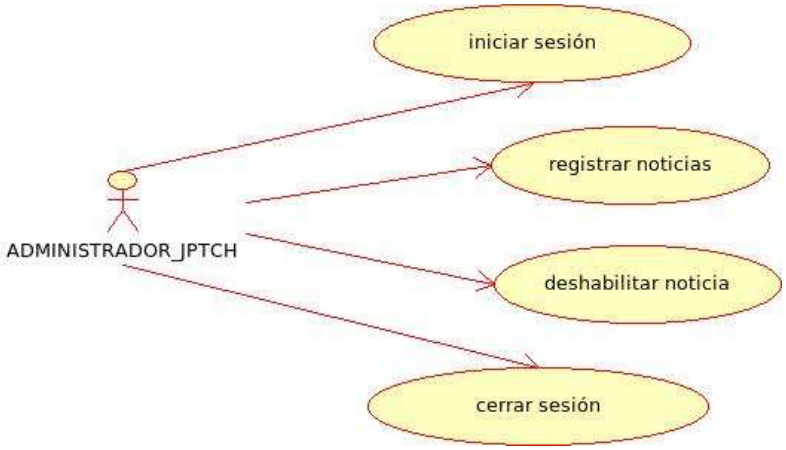

Gráfico IV.25: Diagrama de Caso de Uso Administración JPTCH

- Diagrama de Caso de Uso Usuario Público

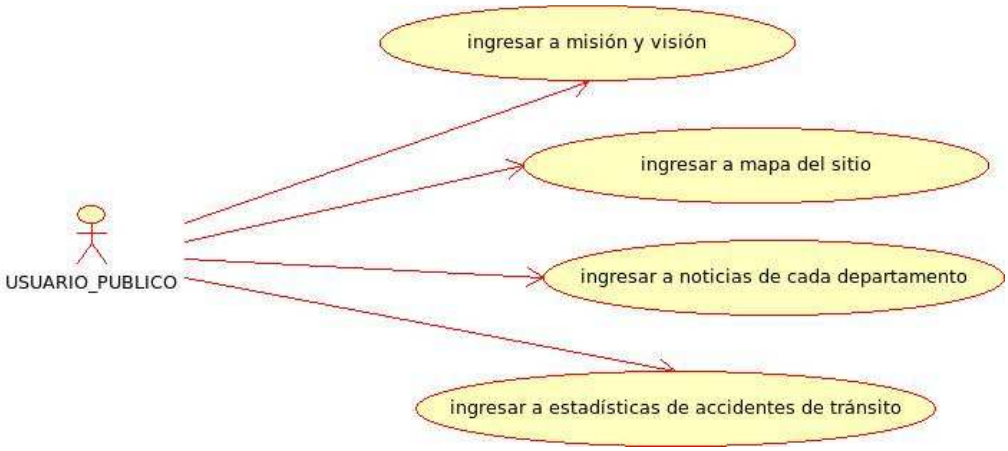

Gráfico IV.26: Diagrama de Caso de Uso Usuario Público

- Diagrama de Caso de Uso Administrar Usuario

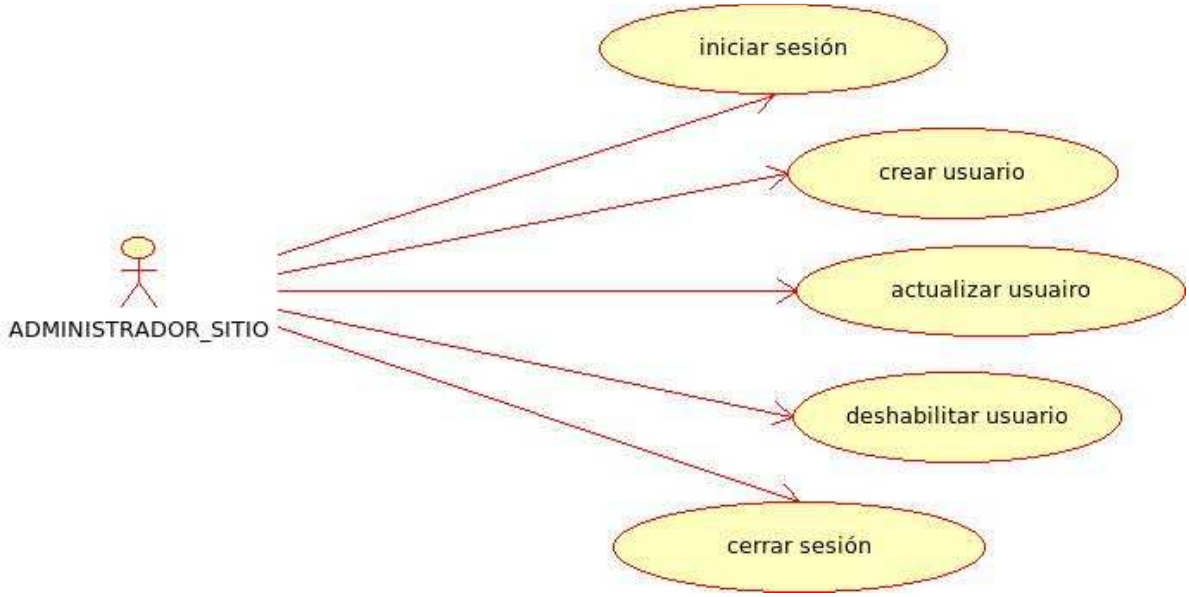

Gráfico IV.27: Diagrama de Caso de Uso Administrar Usuario

- Diagrama de Caso de Uso Administración SIAT

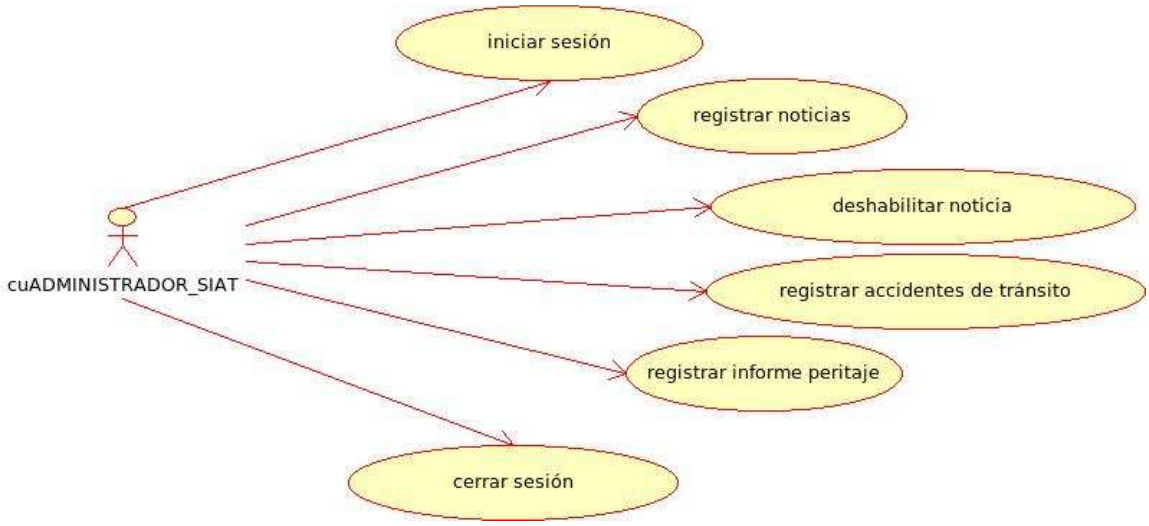

Gráfico IV.28: Diagrama de Caso de Uso Administración SIAT

- Diagrama de Caso de Uso Administración IT

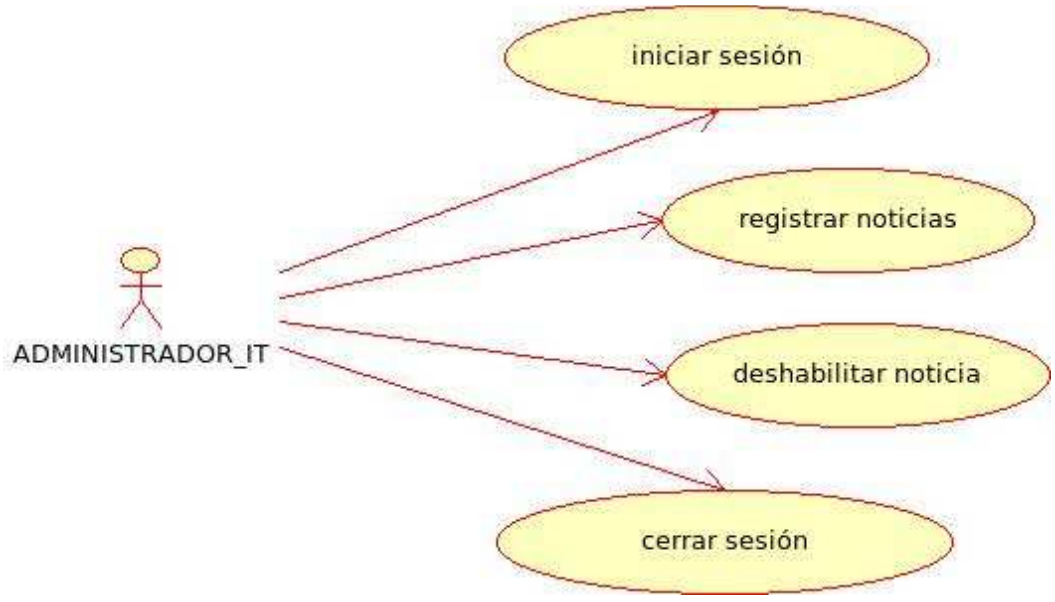

Gráfico IV.29: Diagrama de Caso de Uso Administración IT

- Diagrama de Caso de Uso Administración RP

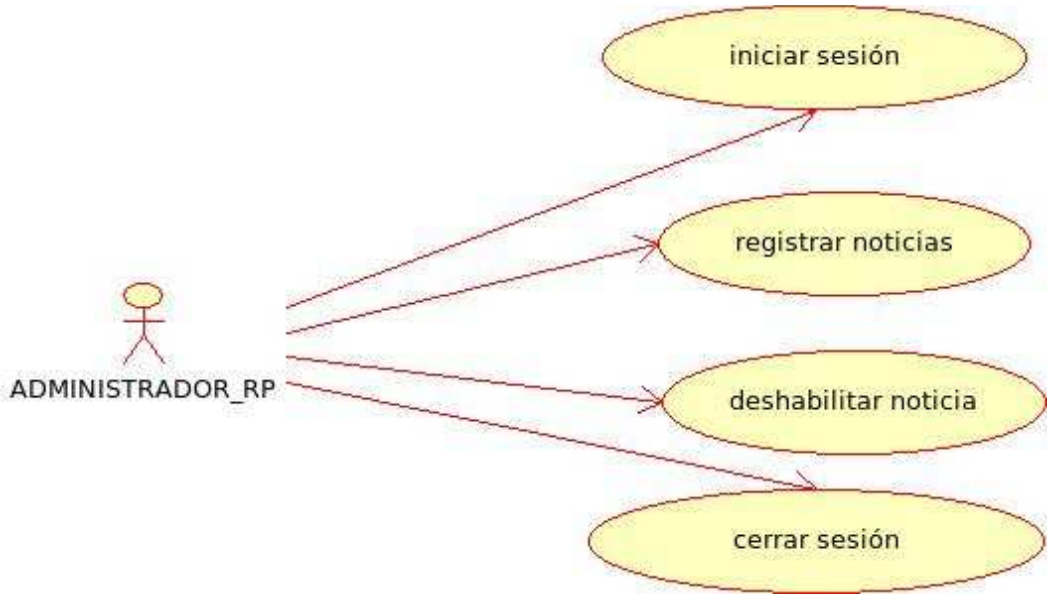

Gráfico IV.30: Diagrama de Caso de Uso Administración RP

- Diagrama de Caso de Uso Administración Fiscalía

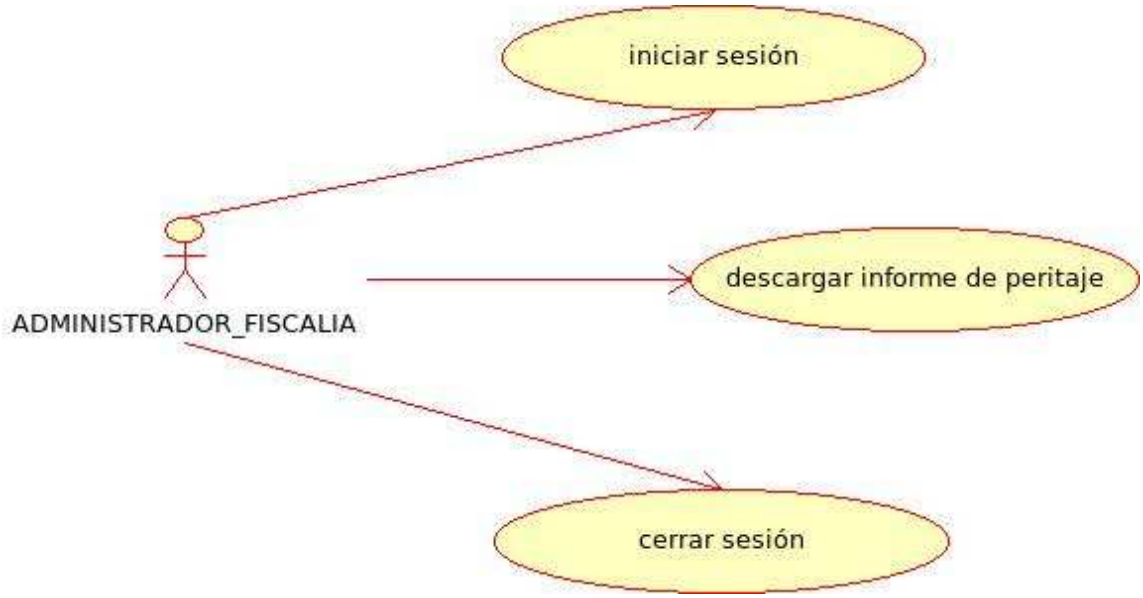

Gráfico IV.31: Diagrama de Caso de Uso Administración Fiscalía

4.3.2.5 Casos de Uso

4.3.2.5.1 Caso de Uso: Iniciar sesión

Identificador Caso de Uso	CU-1
Nombre del Caso de Uso	Iniciar sesión
Actores	ADMINISTRADOR_SITIO ADMINISTRADOR_JPTCH ADMINISTRADOR_SIAT ADMINISTRADOR_IT ADMINISTRADOR_RP ADMINISTRADOR_FISCALIA
Propósito	Ingresar al módulo de administración del sistema.
Visión General	El usuario ingresa al sitio web, ingresa a administrar, el sistema le solicita el nombre de la cuenta, su password y que seleccione el tipo de usuario, el usuario ingresa los datos, el sistema verifica los datos e ingresa al módulo administrar del sitio web.
Tipo	Primario
Referencias	

Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Ingresa al sitio web	2. Presenta la página de inicio
3. Ingresa a administrar	4. Solicita los datos, nombre de la cuenta, el password y el tipo de usuario.
5. Ingresa los datos	6. Ingresa al módulo de administración.
Cursos Alternativos	
Línea 6: Los datos no son correctos, da un mensaje de error y borra los datos ingresados	

Tabla IV.7: Caso de Uso Iniciar sesión

4.3.2.5.2 Caso de Uso: Cerrar sesión

Identificador Caso de Uso	CU-2
Nombre del Caso de Uso	Cerrar sesión
Actores	ADMINISTRADOR_SITIO ADMINISTRADOR_JPTCH ADMINISTRADOR_SIAT ADMINISTRADOR_IT ADMINISTRADOR_RP ADMINISTRADOR_FISCALIA
Propósito	Cerrar la sesión antes de cerrar el sistema.

Visión General	El usuario, una vez que haya iniciado una sesión es necesario cerrar antes de abandonar el sitio, únicamente debe cerrar la sesión.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita cerrar la sesión	2. Cierra la sesión y muestra un mensaje confirmado la sesión está cerrada.

Tabla IV.8: Caso de Uso Cerrar sesión

4.3.2.5.3 Caso de Uso: Crear usuario

Identificador Caso de Uso	CU-3
Nombre del Caso de Uso	Crear usuario
Actores	ADMINISTRADOR_SITIO
Propósito	Crear un usuario del tipo solicitado
Visión General	El usuario ADMINISTRADOR_SITIO solicita al sistema crear un usuario, el sistema solicita los datos del usuario que se va a crear, el usuario ingresa los datos, el sistema valida y verifica los datos, y luego registra al nuevo usuario.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita crear un nuevo usuario	2. Solicita ingresar los datos del nuevo usuario.
3. Ingresa los datos del usuario nuevo	4. Valida y verifica los datos, y registra al nuevo usuario en la BD.
Cursos Alternativos	
Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.	
Línea 4: Si existe un usuario idéntico, se muestra un mensaje de error y no continúa.	

Tabla IV.9: Caso de Uso Crear Usuario

4.3.2.5.4 Caso de Uso: Actualizar usuario

Identificador Caso de Uso	CU-4
Nombre del Caso de Uso	Actualizar usuario
Actores	ADMINISTRADOR_SITIO

Propósito	Actualiza los datos del usuario
Visión General	El usuario ADMINISTRADOR_SITIO solicita actualizar los datos de un usuario en particular, el sistema devuelve los datos del usuario, el usuario ADMINISTRADOR_SITIO actualiza los datos permitidos, el sistema valida y verifica los datos, y actualiza en la BD.
Tipo	Secundario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita actualizar los datos de un usuario dado.	2. Presenta los datos actuales del usuario.
3. Modifica los datos permitidos	4. Valida y verifica los datos, y luego los registra en la BD.
Cursos Alternativos	
Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.	
Línea 4: Si existe un usuario idéntico, se muestra un mensaje de error y no continúa.	

Tabla IV.10: Caso de Uso Actualizar usuario

4.3.2.5.5 Caso de Uso: Deshabilitar usuario

Identificador Caso de Uso	CU-5
Nombre del Caso de Uso	Deshabilitar usuario
Actores	ADMINISTRADOR_SITIO
Propósito	Dar de baja un usuario
Visión General	El usuario ADMINISTRADOR_SITIO solicita deshabilitar una cuenta de usuario, el sistema deshabilita y muestra un mensaje.
Tipo	Primario.
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita deshabilitar un usuario.	2. Muestra un mensaje de confirmación
3. Confirma la solicitud	4. Deshabilita y muestra un mensaje

Tabla IV.11: Caso de Uso Deshabilitar usuario

4.3.2.5.6 Caso de Uso: Registrar noticia

Identificador Caso de Uso	CU-6

Nombre del Caso de Uso	Registrar noticia
Actores	ADMINISTRADOR_JPTCH ADMINISTRADOR_SIAT ADMINISTRADOR_IT ADMINISTRADOR_RP
Propósito	Registra una noticia de acuerdo al departamento correspondiente.
Visión General	El usuario administrador del departamento solicita al sistema registrar una noticia, el sistema le pide que ingrese los datos, el usuario ingresa los datos, el sistema valida estos datos y luego registra en la BD.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita el registro de una noticia	2. Solicita los datos de la noticia
3. Ingresa los datos de la noticia	4. Valida los datos de la noticia y registra en la BD.
Cursos Alternativos	
Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.	

Tabla IV.12: Caso de Uso Registrar Noticia

4.3.2.5.7 Caso de Uso: Deshabilitar noticia

Identificador Caso de Uso	CU-7
Nombre del Caso de Uso	Deshabilitar noticia
Actores	ADMINISTRADOR_JPTCH ADMINISTRADOR_SIAT ADMINISTRADOR_IT ADMINISTRADOR_RP
Propósito	Dar de baja una noticia que ya no se quiere mostrar.
Visión General	El usuario solicita deshabilitar una noticia, el sistema deshabilita y muestra un mensaje.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita deshabilitar una noticia	2. Presenta un mensaje de confirmación
3. Confirma	4. Deshabilita la noticia y presenta un mensaje.

Tabla IV.13: Caso de Uso Deshabilitar noticia

4.3.2.5.8 Caso de Uso: Registrar accidente de tránsito

Identificador Caso de Uso	CU-8
Nombre del Caso de Uso	Registrar accidente de tránsito
Actores	ADMINISTRADOR_SIAT
Propósito	Registra los datos de un accidente de transito
Visión General	El usuario ADMINISTRADOR_SIAT solicita al sistema registrar un accidente, el sistema solicita los datos, el usuario ingresa los datos, el sistema valida en la BD
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita el ingreso de un accidente	2. Solicita los datos del accidente
3. Ingresa los datos del accidente	4. Valida los datos y registra en la BD
Cursos Alternativos	
Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.	

Tabla IV.14: Caso de Uso Registrar accidente de tránsito

4.3.2.5.9 Caso de Uso: Registrar informe de peritaje

Identificador Caso de Uso	CU-9
Nombre del Caso de Uso	Registrar informe de peritaje
Actores	ADMINISTRADOR_SIAT
Propósito	Registrar informe de peritaje en la BD
Visión General	El usuario ADMINISTRADOR_SIAT solicita al sistema registrar un informe de peritaje, el sistema le pide los datos, el usuario ingresa los datos y luego el sistema valida y registra el informe en la BD.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	**Respuesta del Sistema**
1. Solicita el registro de un informe de peritaje.	2. Solicita los datos del informe.
3. Ingresa los datos del informe	4. Valida datos y registra el informe en la BD.
Cursos Alternativos	
Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.	

Tabla IV.15: Caso de Uso Registrar informe de peritaje

4.3.2.5.10 Caso de Uso: Descargar informe de peritaje

Identificador Caso de Uso	CU-10	
Nombre del Caso de Uso	Descargar informe de peritaje	
Actores	ADMINISTRADOR_FISCALIA	
Propósito	Descargar un informe de peritaje	
Visión General	El usuario ADMINISTRADOR_FISCALIA solicita al sistema descargar un informe, el sistema descarga el informe y genera una respuesta de descargado y registra en la base de datos.	
Tipo	Primario	
Referencias		
Curso típico de eventos		
Acción del Actor		**Respuesta del Sistema**
1. Solicita descargar un informe		2. Descarga el informe seleccionado. Y registra un mensaje de descargado o recibido en la BD

Tabla IV.16: Caso de Uso Descargar informe de peritaje

4.3.2.6 Requerimientos Detallados

Para un detalle mejor de las funcionalidades del sitio web se procede a dividir en módulos para un mejor entendimiento tanto de los miembros de la Jefatura Provincial de Tránsito de Chimborazo como de los desarrolladores.

Módulo 1: Jefatura Provincial de Tránsito de Chimborazo

Módulo 2: Departamento SIAT

Módulo 3: Departamento Ingeniería de Tránsito

Módulo 4: Departamento Relaciones Públicas

Módulo 5: Fiscalía

4.3.2.6.1 Módulo 1: Jefatura Provincial de Tránsito de Chimborazo

Req (1): El sitio web debe mostrar información institucional relacionada a la JPTCH.

Req (2): El sitio web debe mostrar noticias relacionadas a la Jefatura Provincial de Tránsito de Chimborazo.

Req (3): El sitio web debe diferenciar a los diferentes tipos de usuario mediante tipos de cuentas de usuario.

Req (4): El sitio web debe permitir ingresar al usuario con su respectiva cuenta de usuario.

4.3.2.6.2 Módulo 2: Departamento SIAT

Req (5): El sitio web debe permitir ingresar al usuario con su respectiva cuenta de usuario.

Req (6): El sitio web debe permitir a la cuenta de usuario respectiva registrar noticias del departamento SIAT de la JTPCH.

Req (7): El sitio web debe mostrar noticias relacionadas al departamento SIAT de la JPTCH.

Req (8): El sitio web debe permitir a la cuenta de usuario respectiva deshabilitar las noticias que se muestran en el sitio.

Req (9): El sitio web debe permitir a la respectiva cuenta de usuario registrar los informes de peritajes de forma segura mediante el protocolo https.

Req (10): El sitio web debe permitir a la respectiva cuenta de usuario registrar los accidentes de tránsito.

Req (11): El sitio web debe permitir al usuario cerrar su sesión cuando lo crea conveniente.

Req (12): El sitio web debe automáticamente cerrar la sesión de la cuenta de usuario después de algún tiempo que no presente actividad el usuario.

4.3.2.6.3 Módulo 3: Departamento Ingeniería de Tránsito

Req (13): El sitio web debe permitir ingresar al usuario con su respectiva cuenta de usuario.

Req (14): El sitio web debe permitir a la cuenta de usuario respectiva registrar noticias del departamento IT de la JTPCH.

Req (15): El sitio web debe mostrar noticias relacionadas al departamento de IT de la JPTCH.

Req (16): El sitio web debe permitir a la cuenta de usuario respectiva deshabilitar las noticias que se muestran en el sitio.

Req (17): El sitio web debe permitir al usuario cerrar su sesión cuando lo crea conveniente.

Req (18): El sitio web debe automáticamente cerrar la sesión de la cuenta de usuario después de algún tiempo que no presente actividad el usuario.

4.3.2.6.4 Módulo 4: Departamento Relaciones Públicas

Req (19): El sitio web debe permitir ingresar al usuario con su respectiva cuenta de usuario.

Req (20): El sitio web debe permitir a la cuenta de usuario respectiva registrar noticias del departamento RP de la JTPCH.

Req (21): El sitio web debe mostrar noticias relacionadas al departamento de RP de la JPTCH.

Req (22): El sitio web debe permitir a la cuenta de usuario respectiva deshabilitar las noticias que se muestran en el sitio.

Req (23): El sitio web debe permitir al usuario cerrar su sesión cuando lo crea conveniente.

Req (24): El sitio web debe automáticamente cerrar la sesión de la cuenta de usuario después de algún tiempo que no presente actividad el usuario.

4.3.2.6.5 Módulo 5: Fiscalía

Req (25): El sitio web debe permitir ingresar al usuario con su respectiva cuenta de usuario.

Req (26): El sitio web de permitir a la cuenta de usuario respectiva descargar el informe de peritaje de forma segura mediante el protocolo https.

Req (27): El sitio web debe permitir al usuario cerrar su sesión cuando lo crea conveniente.

Req (28): El sitio web debe automáticamente cerrar la sesión de la cuenta de usuario después de algún tiempo que no presente actividad el usuario.

4.3.2.7 Requerimientos del Sistema

Listado de Requerimientos de Sistema	
Id	**Descripción**
RS-1	El tiempo de respuesta a cualquier petición no debe superar los 5 segundos.
RS-2	El sistema debe soportar a por lo menos 100 usuarios concurrentes trabajando con la aplicación.
RS-3	Al tratarse de información confidencial, las comunicaciones al servidor web deben ser de forma segura mediante el protocolo SSL.
RS-4	La aplicación deberá poder ser accesible las 24 horas del día, los 365 días del año.
RS-5	La velocidad de respuesta dependerá también del ancho de banda que contrate JPTCH para salir al internet.
RS-6	El tiempo de respuesta del sitio web dependerá del hardware facilitado por la JPTCH.

Tabla IV.17: Tabla de Requerimientos del Sistema

4.3.2.8 Diccionario de Datos

4.3.2.8.1 Acrónimos

Acrónimos	Significado
JPTCH	Jefatura Provincial de Tránsito de Chimborazo
SIAT	Departamento Servicio de Investigaciones de Accidentes de Tránsito de la JPTCH.
IT	Departamento de Ingeniería de Tránsito de la JPTCH.
RP	Departamento de Relaciones Públicas de la JPTCH.
FISCALIA	Fiscalía de Chimborazo que interactúa con la aplicación.

Tabla IV.18: Tabla de Acrónimos

4.3.2.8.2 Diccionario de Datos

Objeto	Atributos	Tipo de Dato	Caracteres permitidos	Descripción
Usuario	Usuario	varchar 25	Caracteres alfanuméricos	Nombre de usuario para iniciar sesión
	Nombre	varchar 30	Caracteres alfabéticos	Nombre real del usuario
	Apellido	varchar 30	Caracteres alfabéticos	Apellido real del usuario
	Dirección	varchar 50	Caracteres alfanuméricos más signos y símbolos de puntuación.	Dirección real del usuario
	Email	varchar 50	Caracteres alfanuméricos más caracteres como (‘@’,’ – ‘ y ‘_’)	Dirección de correo electrónico del usuario
	Tipo	varchar 20	ADMINISTRADOR_SIAT ADMINISTRADOR_RP ADMINISTRADOR _IT ADMINISTRADOR _JTPCH ADMINISTRADOR _FISCALIA ADMINISTRADOR_SITIO	Tipos definidos de usuarios para la administración de la aplicación divididos por departamentos. Donde la terminación de cada tipo especifica al departamento que pertenece.

	Password	varchar 20	Caracteres alfanuméricos más signos y símbolos de puntuación.	Contraseña para seguridad de la administración de la aplicación utilizada por todos los usuarios para iniciar una sesión.
Informe	identificador	Int	Caracteres numéricos ingresados automáticamente por el sistema.	Identificador de los objetos almacenados en la tabla, utilizado únicamente por los desarrolladores.
	Descripción	varchar 100	Caracteres alfanuméricos	Descripción del archivo del informe ingresado
	Archivo	varchar 100	Caracteres alfanuméricos incluido signos de puntuación como guiones y barra inclinadas.	Dirección donde se almacena el archivo del informe ingresado.
Noticia	identificador	int	Caracteres numéricos	Identificador de los objetos Noticias en la tabla respectiva, únicamente para los desarrolladores.
	Descripción	varchar 100	Caracteres alfanuméricos	Descripción de la noticia y que se mostrara como título.
	Foto	varchar 50	Caracteres alfanuméricos incluido signos de puntuación como guiones y barra inclinadas.	Dirección donde se encuentra la foto.
	Archivo	varchar 50	Caracteres alfanuméricos incluido signos de puntuación como guiones y barra inclinadas.	Dirección del archivo donde está redactada la noticia.
Responsable	Ci	varchar 10	Caracteres numéricos	Cedula de identidad del responsable de los informes
	nombre	varchar	Caracteres alfabéticos	Nombre del

		25		responsable
	Apellido	varchar 25	Caracteres alfabéticos	Apellido del responsable
	Dirección	varchar 25	Caracteres alfanuméricos más signos y símbolos de puntuación.	Dirección del responsable
	Email	varchar 25	Caracteres alfanuméricos más caracteres como (‘@’,’ –‘ y ‘_’)	Dirección de correo electrónico del responsable
	Rango	varchar 20	Caracteres alfanuméricos	Rango del policía responsable de realizar el informe.
Accidente	identificador	Int	Caracteres numéricos	Identificador de los objetos Accidentes en la tabla respectiva, únicamente para los desarrolladores.
	Descripción	varchar 50	Caracteres alfanuméricos	Descripción del accidente
	Dirección	varchar 25	Caracteres alfanuméricos más signos y símbolos de puntuación.	Dirección donde se realizo el accidente
	Sector	varchar 25	Caracteres alfanuméricos más signos y símbolos de puntuación.	
	Fecha	date		Fecha en la que ocurrió el accidente.
	Hora	time		Hora que sucedió el accidente.
	Tipo	varchar 25	Caracteres alfabéticos	Tipo del accidente, se puede seleccionar de una lista.

Tabla IV.19: Tabla de Diccionario de Datos

4.3.3. Diseño

4.3.3.1 Diseño de la Base de Datos

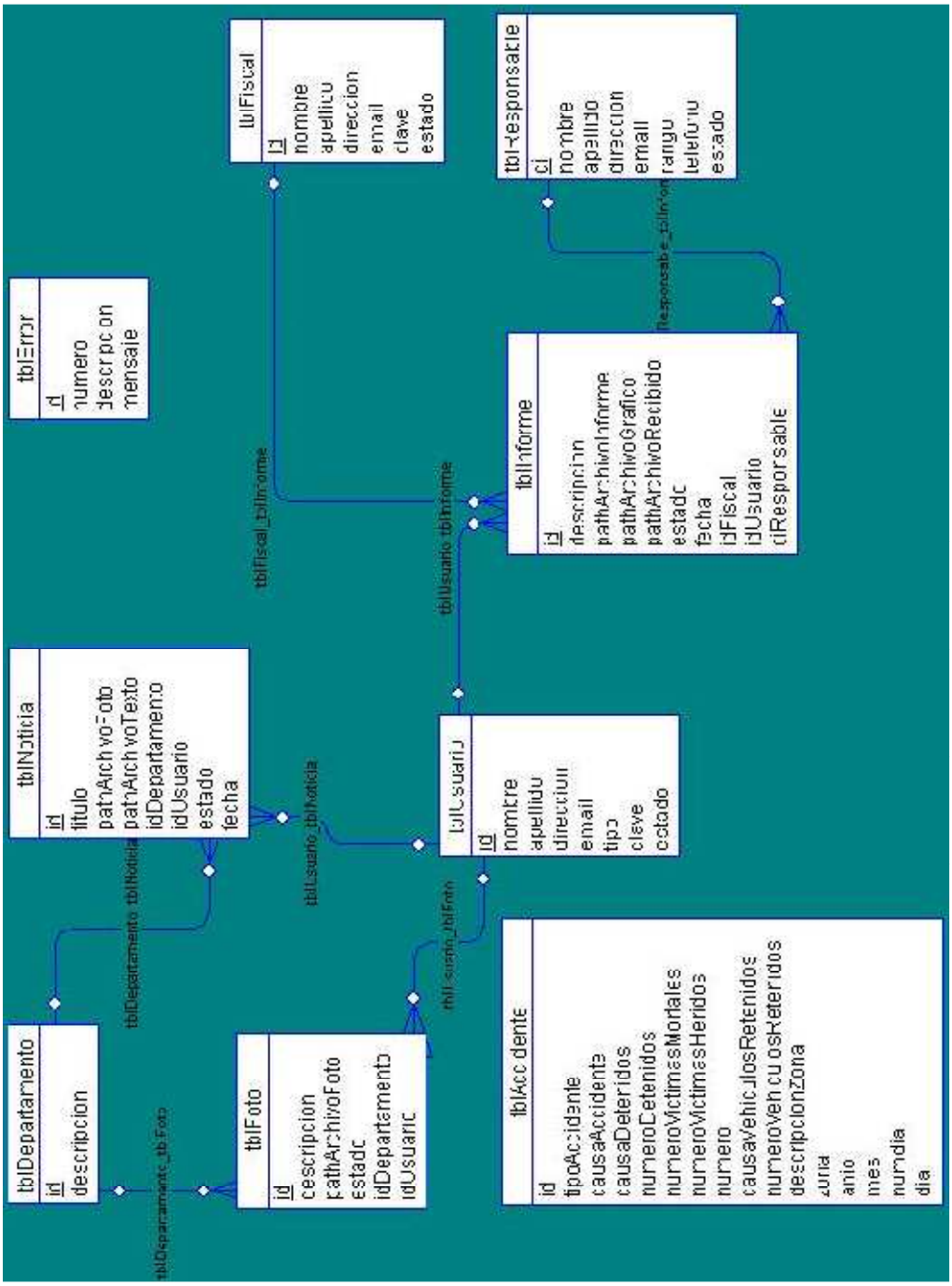

Gráfico IV.32: Diseño de Base de Datos

4.3.3.2 Diagrama de Clases

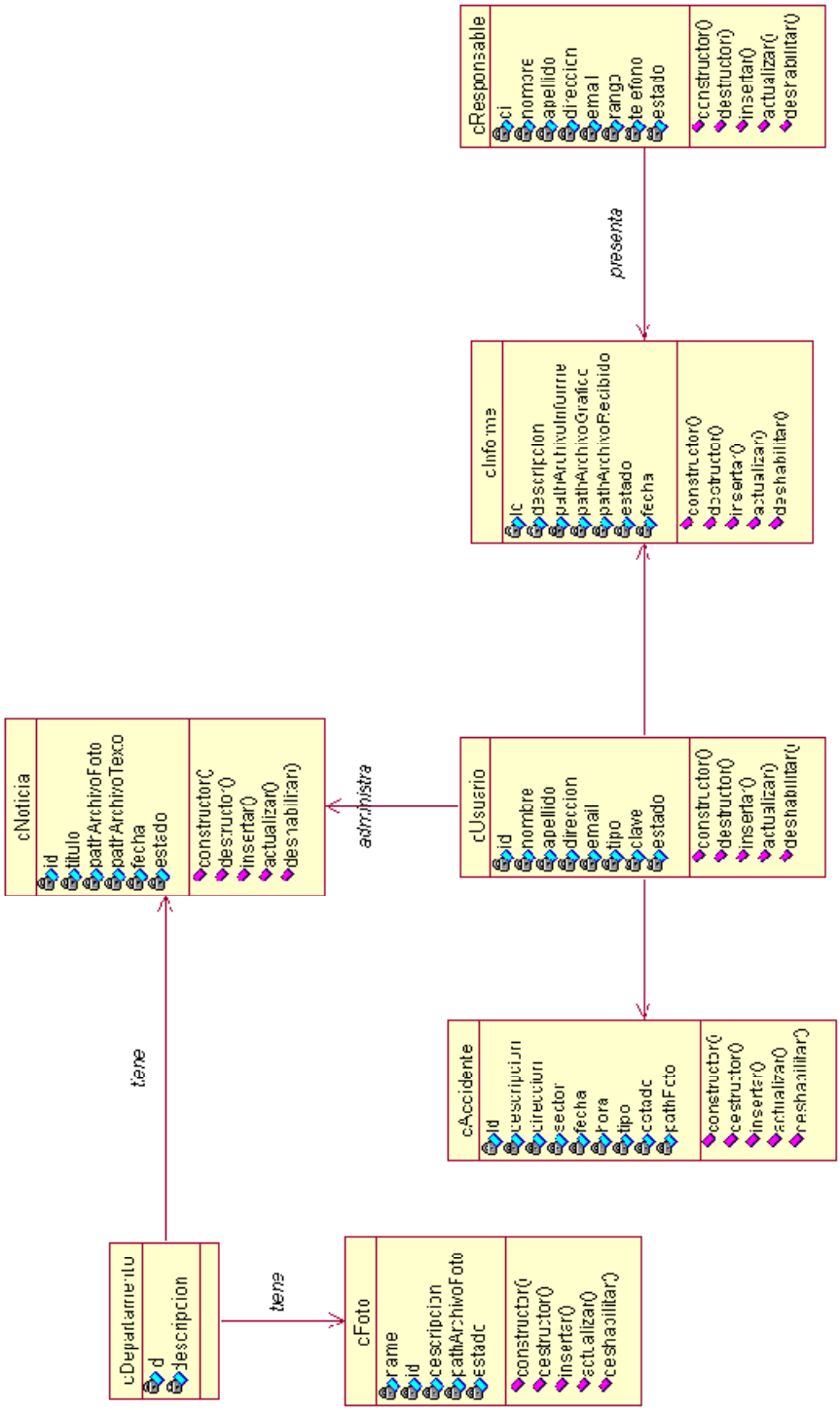

Gráfico IV.33: Diagrama de Clases

4.3.3.3 Diagrama de despliegue de la Aplicación Web SONIA

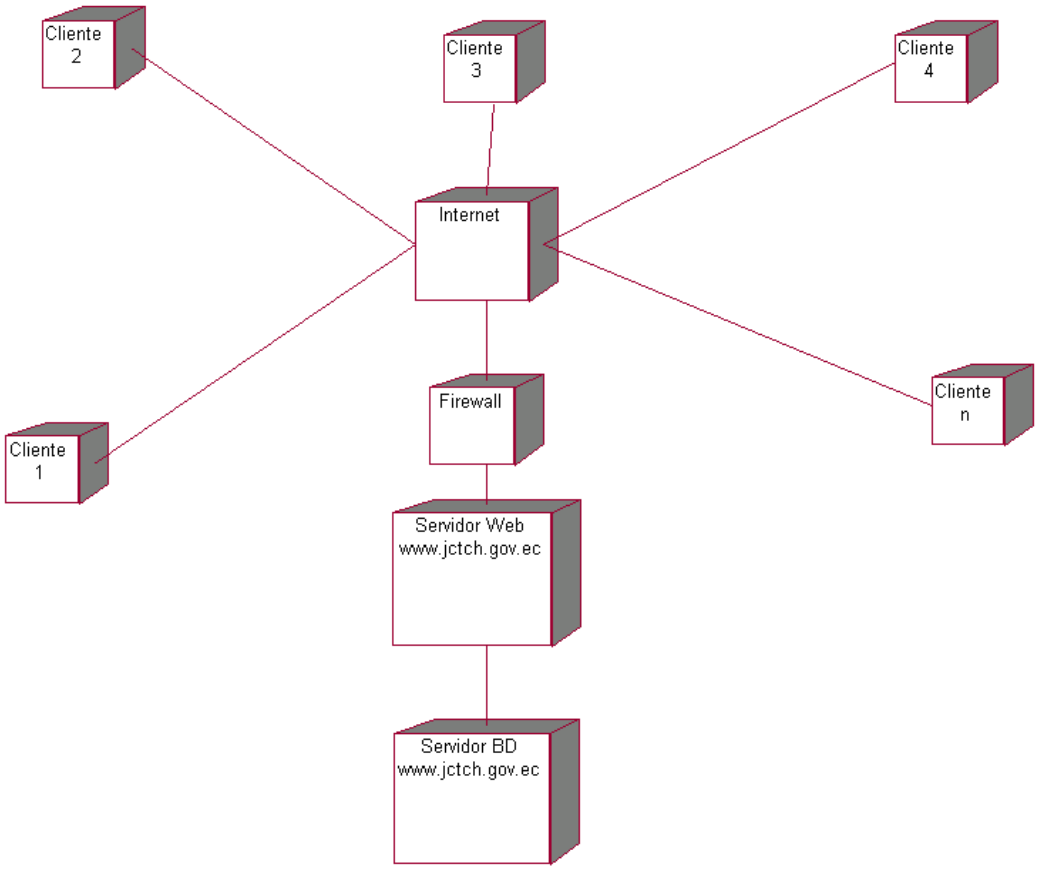

Gráfico IV.34: Diagrama de Despliegue

4.3.3.4 Diagrama de Componentes de la Aplicación Web SONIA

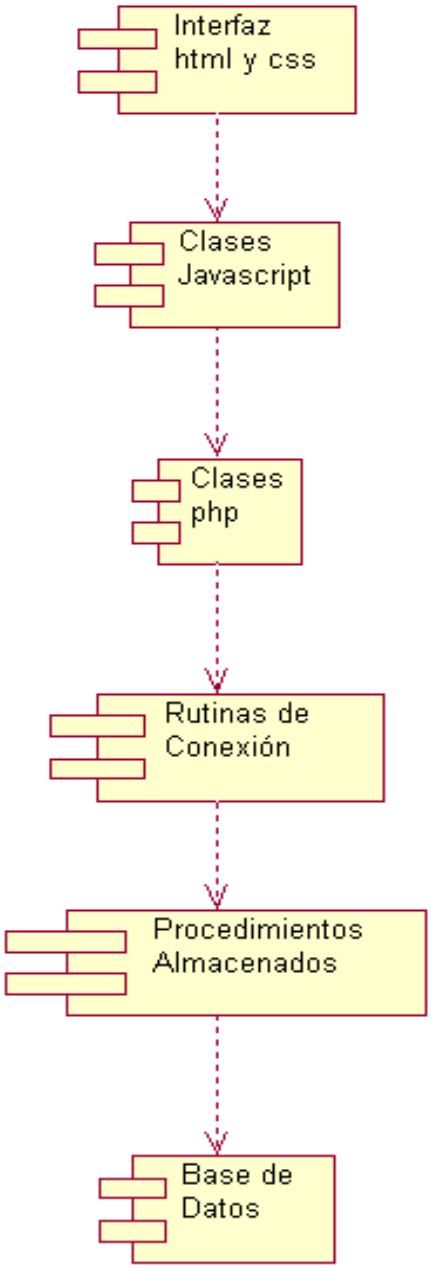

Gráfico IV.35: Diagrama de Componentes

4.3.4. Codificación

La codificación de la aplicación web SONIA es orientada a objetos por lo tanto principalmente detallaremos la codificación de las clases diseñadas en el diagrama de clases desarrollado anteriormente.

4.3.4.1 Clase Usuario

```
<?
class clsUsuario
{
      var $usuario = "";
      var $nombre = "";
      var $apellido = "";
      var $direccion = "";
      var $email = "";
      var $tipo = "";
      var $clave = "";
      var $estado = "";

      //Constructor
      function
clsUsuario($usuario,$nombre,$apellido,$direccion,$email,$tipo,$clave,$est
ado){
            $this->usuario=$usuario;
            $this->nombre=$nombre;
            $this->apellido=$apellido;
            $this->direccion=$direccion;
            $this->email=$email;
            $this->tipo=$tipo;
            $this->clave=$clave;
            $this->estado=$estado;
      }

      function crear($conexion){
      $error = 0;
      if ($this->tipo=="FISCALIA"){
      $pa="CALL patblFiscalInsertar('".$this->usuario."','".$this-
>nombre."','".$this->apellido."','".$this->direccion."','".$this-
>email."','".$this->clave."','".$this->estado."',@res);";
      }else{
      $pa=$pa="CALL patblUsuarioInsertar('".$this->usuario."','".$this-
>nombre."','".$this->apellido."','".$this->direccion."','".$this-
>email."','".$this->tipo."','".$this->clave."','".$this-
>estado."',@res);";
      }
@     $result = mysql_query($pa,$conexion);
      if ($result){
@           $resultado = mysql_query("SELECT @res AS resultado;");
@           $row = mysql_fetch_array($resultado);
            if ($row["resultado"]!=1){
```

```
                  $error = 4;  //error en los datos ingresados en el
formulario
            }
      }
      else{
            $error = 3; //no se puede ejecutar el procedimiento
alamacenado
      }
@     mysql_free_result($result);
      return($error);
      }

      function actualizar($conexion){
            $error = 0;
            $pa="CALL patblUsuarioActualizar('".$this-
>usuario."','".$this->direccion."','".$this->email."','".$this-
>clave."',@res);";
@           $result = mysql_query($pa,$conexion);
            if ($result){
@                 $resultado = mysql_query("SELECT @res AS resultado;");
@                 $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 6;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }

      function actualizarTipoFiscal($conexion){
            $error = 0;
            $pa="CALL patblFiscalActualizar('".$this-
>usuario."','".$this->direccion."','".$this->email."','".$this-
>clave."',@res);";
@           $result = mysql_query($pa,$conexion);
            if ($result){
@                 $resultado = mysql_query("SELECT @res AS resultado;");
@                 $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 6;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }

      function deshabilitar($conexion){
            $error = 0;
            $pa="CALL patblUsuarioDeshabilitar('".$this-
>usuario."',@res);";
@           $result = mysql_query($pa,$conexion);
```

```
            if ($result){
@                 $resultado = mysql_query("SELECT @res AS resultado;");
@                 $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 7;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }
}
?>
```

4.3.4.2 Clase Responsable

```
<?
class clsResponsable
{
      var $ci = "";
      var $nombre = "";
      var $apellido = "";
      var $direccion = "";
      var $email = "";
      var $rango = "";
      var $telefono = "";
      var $estado = "";

      //Constructor
      function
clsResponsable($ci,$nombre,$apellido,$direccion,$email,$rango,$telefono,$
estado){
            $this->ci=$ci;
            $this->nombre=$nombre;
            $this->apellido=$apellido;
            $this->direccion=$direccion;
            $this->email=$email;
            $this->rango=$rango;
            $this->telefono=$telefono;
            $this->estado=$estado;
      }

      function crear($conexion){
            $error = 0;
            $pa="CALL patblResponsableInsertar('".$this->ci."','".$this-
>nombre."','".$this->apellido."','".$this->direccion."','".$this-
>email."','".$this->rango."','".$this->telefono."','".$this-
>estado."',@res);";
            $result = mysql_query($pa,$conexion);
            if ($result){
                  $resultado = mysql_query("SELECT @res AS resultado;");
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
```

```
                        $error = 8;  //error en los datos ingresados en el
formulario
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
@     mysql_free_result($result);
      return($error);
      }

      function actualizar($conexion){
            $error = 0;
            $pa="CALL patblResponsableActualizar('".$this-
>ci."','".$this->nombre."','".$this->apellido."','".$this-
>direccion."','".$this->email."','".$this->rango."','".$this-
>telefono."',@res)";
@           $result = mysql_query($pa,$conexion);
            if ($result){
@                 $resultado = mysql_query("SELECT @res AS resultado;");
@                 $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 9;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }

      function deshabilitar($conexion){
            $error = 0;
            $pa="CALL patblResponsableDeshabilitar('".$this-
>ci."',@res);";
            $result = mysql_query($pa,$conexion);
            if ($result){
                  $resultado = mysql_query("SELECT @res AS resultado;");
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 10;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }
}
?>
```

4.3.4.3 Clase Noticia

```
<?
class clsNoticia
{
	var $id = "";
	var $titulo = "";
	var $pathFoto = "";
	var $pathNoticia = "";
	var $idDepartamento = "";
	var $idUsuario = "";
	var $estado = "";

	//Constructor
	function clsNoticia($id,$tit,$fot,$not,$idD,$idU,$est){
		$this->id=$id;
		$this->titulo=$tit;
		$this->pathFoto=$fot;
		$this->pathNoticia=$not;
		$this->idDepartamento=$idD;
		$this->idUsuario=$idU;
		$this->estado=$est;
	}

	function crear($conexion){
		$error = 0;
		$pa="CALL patblNoticiaInsertar('".$this->titulo."','".$this-
>pathFoto."','".$this->pathNoticia."','".$this-
>idDepartamento."','".$this->idUsuario."','".$this->estado."',@res);";
		$result = mysql_query($pa,$conexion);
		if ($result){
			$resultado = mysql_query("SELECT @res AS resultado;");
			$row = mysql_fetch_array($resultado);
			if ($row["resultado"]!=1){
				$error = 13;  //error en los datos ingresados
			}
		}
		else{
			$error = 3; //no se puede ejecutar el procedimiento
alamacenado
		}
@	mysql_free_result($result);
	return($error);
	}

	function deshabilitar($conexion){
		$error = 0;
		$pa="CALL patblNoticiaDeshabilitar('".$this->id."',@res);";
		$result = mysql_query($pa,$conexion);
		if ($result){
			$resultado = mysql_query("SELECT @res AS resultado;");
			$row = mysql_fetch_array($resultado);
			if ($row["resultado"]!=1){
				$error = 14;  //error en los datos ingresados
			}
		}
```

```
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }
}
?>
```

4.3.4.4 Clase Informe

```
<?
class clsInforme
{
      var $id = "";
      var $descripcion = "";
      var $informe = "";
      var $grafico = "";
      var $recibido = "";
      var $estado = "";
      var $idFiscal = "";
      var $idUsuario = "";
      var $ciResponsable = "";

      //Constructor
      function clsInforme($id,$des,$info,$graf,$rec,$est,$idF,$usu,$ciR){
            $this->id=$id;
            $this->descripcion=$des;
            $this->informe=$info;
            $this->grafico=$graf;
            $this->recibido=$rec;
            $this->estado=$est;
            $this->idFiscal=$idF;
            $this->idUsuario=$usu;
            $this->ciResponsable=$ciR;
      }


      function crear($conexion){
            $error = 0;
            $pa="CALL patblInformeInsertar('".$this-
>descripcion."','".$this->informe."','".$this->grafico."','1','".$this-
>idFiscal."','".$this->idUsuario."','".$this->ciResponsable."',@res);";
            $result = mysql_query($pa,$conexion);
            if ($result){
                  $resultado = mysql_query("SELECT @res AS resultado;");
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 18;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
```

```
@     mysql_free_result($result);
      return($error);
      }

      function deshabilitar($conexion){
            $error = 0;
            $pa="CALL patblInformeDeshabilitar('".$this->id."',@res);";
            $result = mysql_query($pa,$conexion);
            if ($result){
                  $resultado = mysql_query("SELECT @res AS resultado;");
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 19;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }

      function deshabilitarDefinitivo($conexion){
            $error = 0;
            $pa="CALL patblInformeDeshabilitarDefinitivo('".$this-
>id."',@res);";
            $result = mysql_query($pa,$conexion);
            if ($result){
                  $resultado = mysql_query("SELECT @res AS resultado;");
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 19;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }

      function habilitar($conexion){
            $error = 0;
            $pa="CALL patblInformeHabilitar('".$this->id."',@res);";
            $result = mysql_query($pa,$conexion);
            if ($result){
                  $resultado = mysql_query("SELECT @res AS resultado;");
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 19;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
```

```
	}
}
?>
```

4.3.4.5 Clase Foto

```
<?
class clsFoto
{
	var $id = "";
	var $descripcion = "";
	var $pathFoto = "";
	var $estado = "";
	var $idDepartamento = "";
	var $idUsuario = "";

	//Constructor
	function clsFoto($id,$des,$fot,$est,$idD,$idU){
		$this->id=$id;
		$this->descripcion=$des;
		$this->pathFoto=$fot;
		$this->estado=$est;
		$this->idDepartamento=$idD;
		$this->idUsuario=$idU;
	}

	function crear($conexion){
		$error = 0;
		$pa="CALL patblFotoInsertar('".$this-
>descripcion."','".$this->pathFoto."','".$this->estado."','".$this-
>idDepartamento."','".$this->idUsuario."',@res)";
		$result = mysql_query($pa,$conexion);
		//echo mysql_error();
		if ($result){
			$resultado = mysql_query("SELECT @res AS resultado;");
			$row = mysql_fetch_array($resultado);
			if ($row["resultado"]!=1){
				$error = 22;  //error en los datos ingresados
			}
		}
		else{
			$error = 3; //no se puede ejecutar el procedimiento
alamacenado
		}
@	mysql_free_result($result);
	return($error);
	}

	function deshabilitar($conexion){
		$error = 0;
		$pa="CALL patblFotoDeshabilitar('".$this->id."',@res);";
		$result = mysql_query($pa,$conexion);
		if ($result){
			$resultado = mysql_query("SELECT @res AS resultado;");
```

```
                  $row = mysql_fetch_array($resultado);
                  if ($row["resultado"]!=1){
                        $error = 23;  //error en los datos ingresados
                  }
            }
            else{
                  $error = 3; //no se puede ejecutar el procedimiento
alamacenado
            }
      return($error);
      }
}
?>
```

4.3.5. Pruebas

En el presente punto se desarrolla una serie de pruebas para comprobar la funcionalidad de la aplicación web “SONIA”.

Para ingresar a la aplicación web “SONIA”, el usuario debe digitar la dirección: www.jctch.gov.ec

4.3.5.1 Cuenta Administración

Para ingresar al módulo de Administración el usuario debe dar clic en ingresar como se muestra en el Gráfico IV.36.

Se le muestra al usuario una pantalla de autenticación, en donde, si el usuario ingresa de una manera incorrecta los datos se le presenta el respectivo mensaje: Ingreso incorrecto de datos. Como se muestra en el Gráfico IV.37.

Una vez ingresado correctamente en la cuenta administrador, este puede crear usuarios administradores de cada uno de los tipos que permite la aplicación, como: tipo UIAT, tipo JEFATURA, tipo FISCALIA, tipo ADMINISTRADOR, tipo INGENIERÍA DE TRÁNSITO. Para ello el administrador debe ingresar correctamente los datos requeridos, en caso de no cumplir con este requerimiento, el sistema le muestra el mensaje de error personalizado que se visualiza en el Gráfico IV.38.

Solo la cuenta de usuario administrador permite observar los errores que se genera al procesar alguna transacción incorrectamente, el usuario administrador observa los errores como se muestra en el Gráfico IV.39.

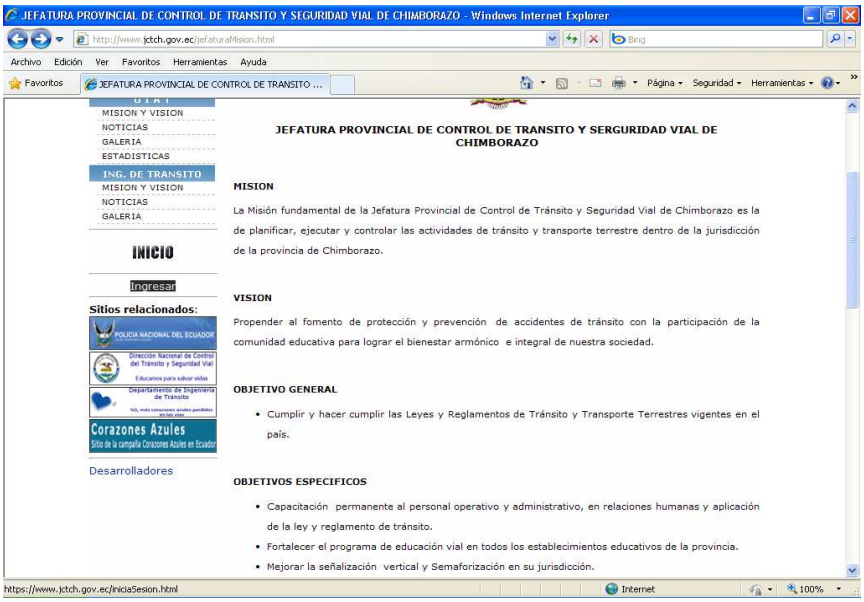

Gráfico IV.36: Prueba Ingreso Módulo Administración

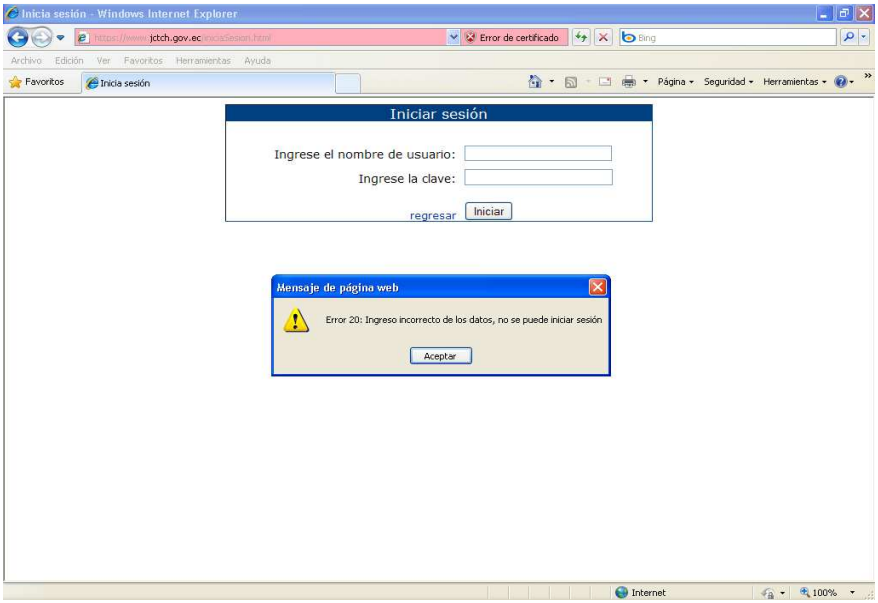

Gráfico IV.37: Prueba de Autenticación

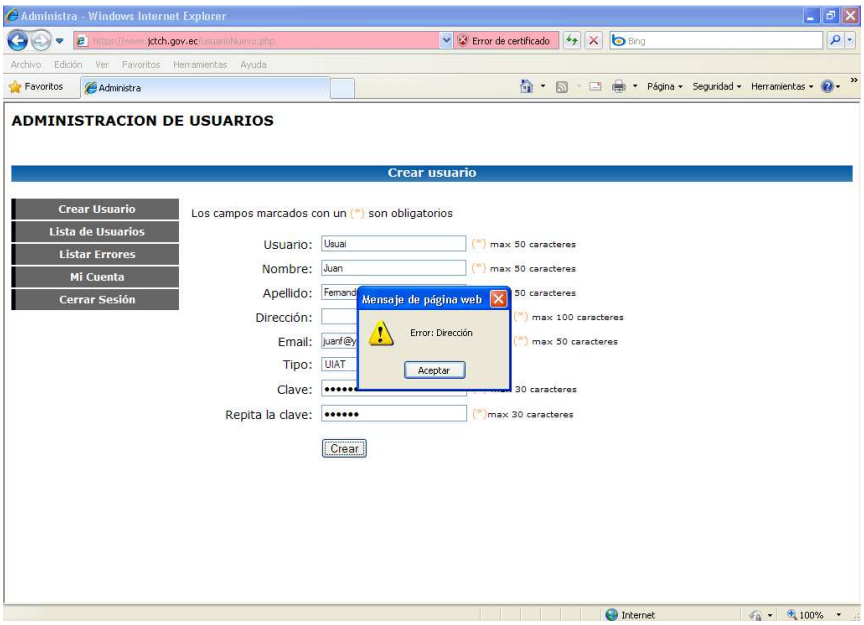

Gráfico IV.38: Prueba Ingreso Datos de Nuevo Usuario

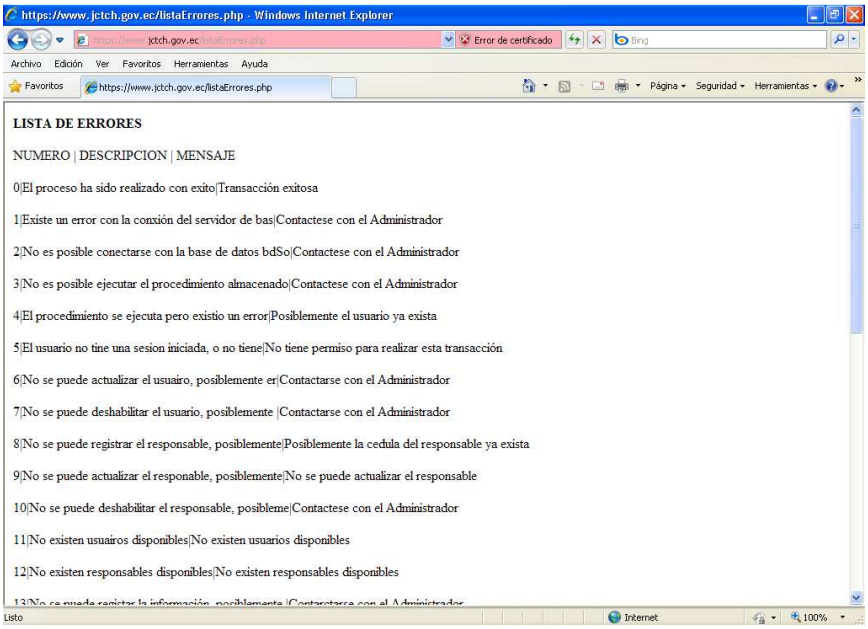

Gráfico IV.39: Prueba Visualización de Errores

Otro error controlado es en la actualización de la cuenta de usuario que se encuentra activo, por ejemplo sino coinciden la clave con su repetición, la aplicación web le muestra el respectivo mensaje de error.

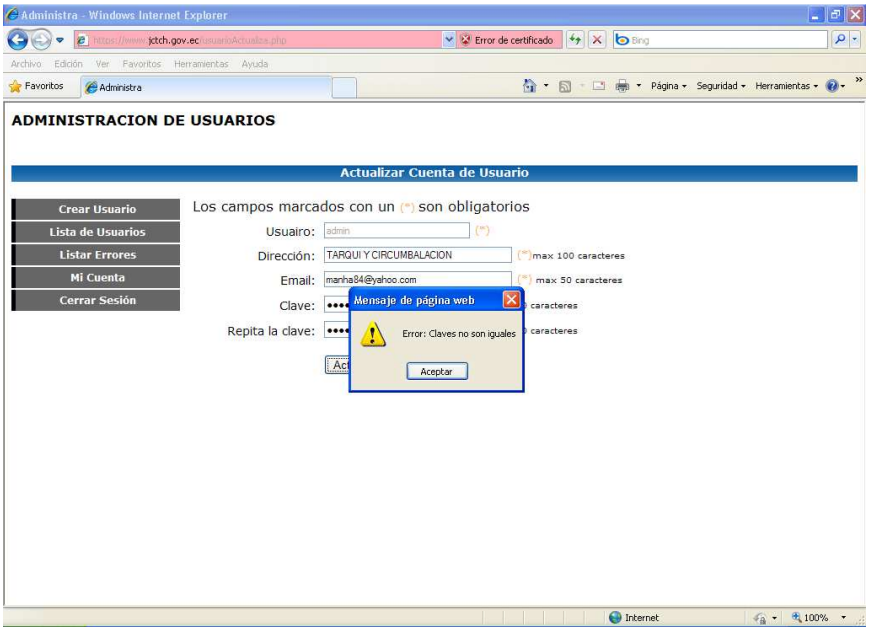

Gráfico IV.40: Prueba Actualización Datos Cuenta Activa

4.3.5.2 Cuenta UIAT

El usuario que ingresa con el tipo de cuenta UIAT, puede ingresar responsables de los informes, que son las personas responsables de la elaboración de los informes ingresados. Si no ingresa correctamente todos los datos requeridos, la aplicación web le muestra el respectivo mensaje de error, como se muestra a continuación.

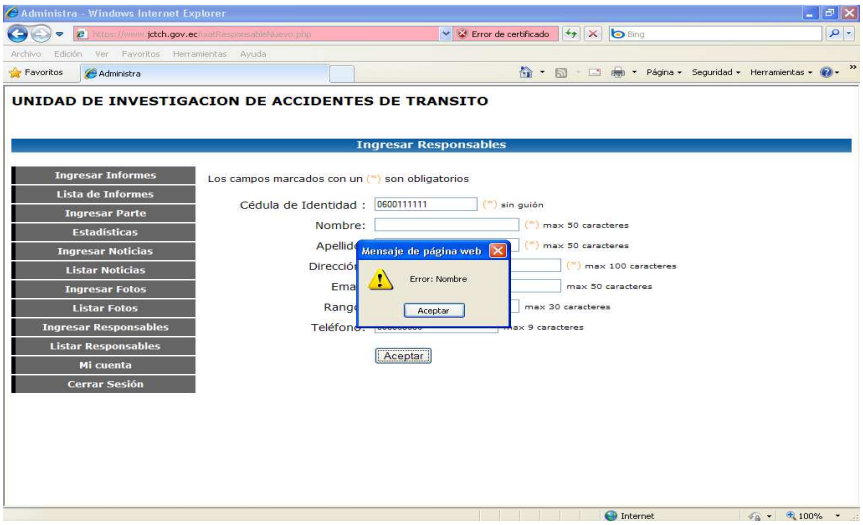

Gráfico IV.41: Prueba Ingreso Responsables de Informes

Una vez ingresado el responsable del informe el usuario de tipo UIAT puede ingresar informes, para ello debe ingresar una descripción obligatoria, un archivo .doc, .docx o .pdf y una imagen, sino lo ejecuta correctamente se muestra el siguiente mensaje de error:

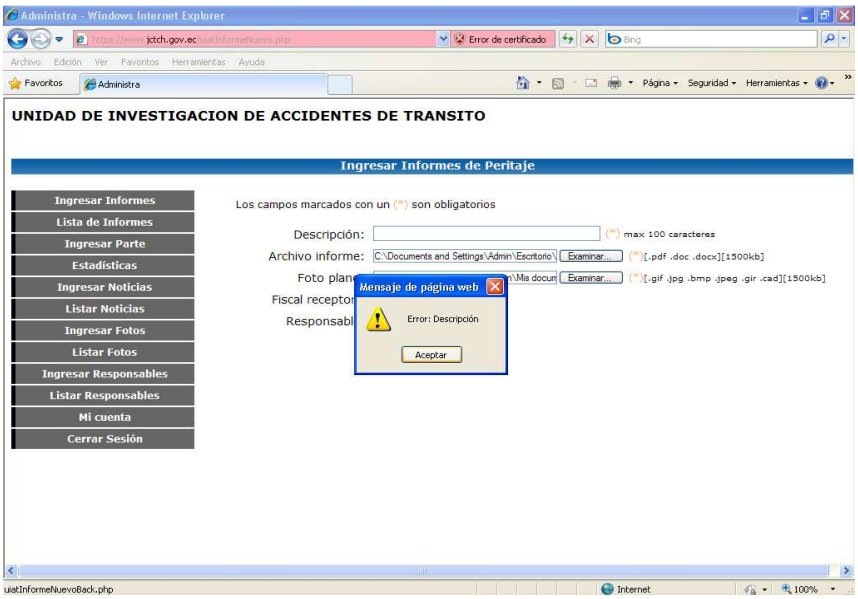

Gráfico IV.42: Prueba Ingreso Informes de Peritaje

Al registrar un parte policial, y, si el usuario de tipo UIAT no ingresa correctamente los datos requeridos se controla visualizando el siguiente mensaje de error.

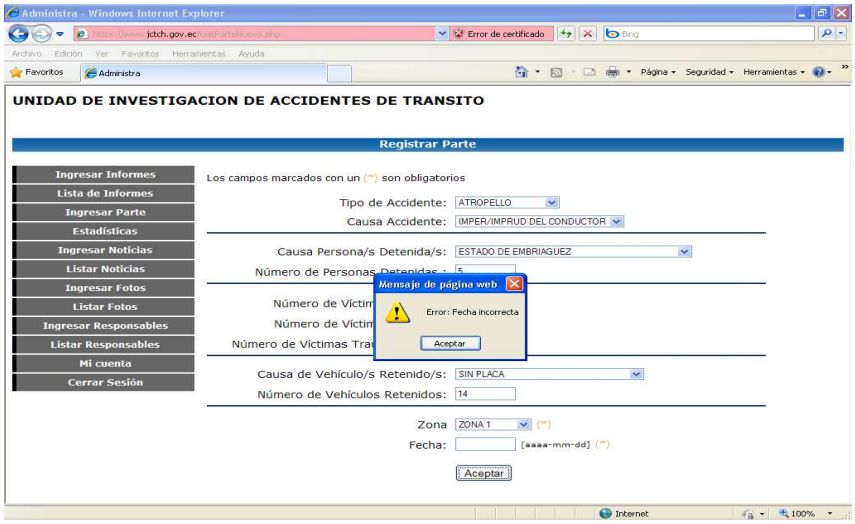

Gráfico IV.43: Prueba Registro Parte Policial

Para visualizar las estadísticas correctamente, el usuario deberá seleccionar el año o las zonas, sino lo realiza correctamente la aplicación web controla este error emitiendo el siguiente mensaje de error.

Gráfico IV.44: Prueba Estadísticas

El usuario e tipo UIAT también podrá publicar noticias relacionadas a las actividades que se realizan como departamento UIAT en beneficio de la comunidad. Para ello deberá ingresar correctamente el título, el archivo en formato .txt, con el texto de la noticia y una imagen que representa la noticia, sino lo hace correctamente se muestra el siguiente mensaje de error en el Gráfico IV.45.

4.3.5.3 Cuenta JEFATURA

En la cuenta jefatura se pueden realizar las mismas transacciones que fueron detalladas en la cuenta UIAT como la publicación de noticias. De la misma forma se puede publicar noticias relacionadas con la jefatura como institución. Aquí se puede ingresar fotos para su galería para ello debe ingresar correctamente una descripción y una foto, sino lo hace correctamente visualizará el siguiente mensaje de error mostrado en el Gráfico IV.35.

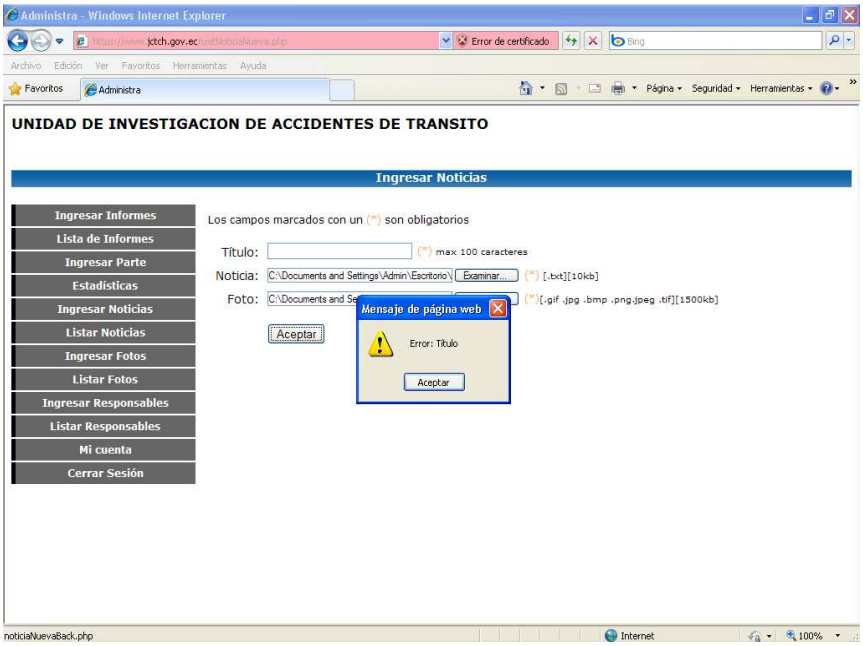

Gráfico IV.45: Prueba Ingreso Noticias

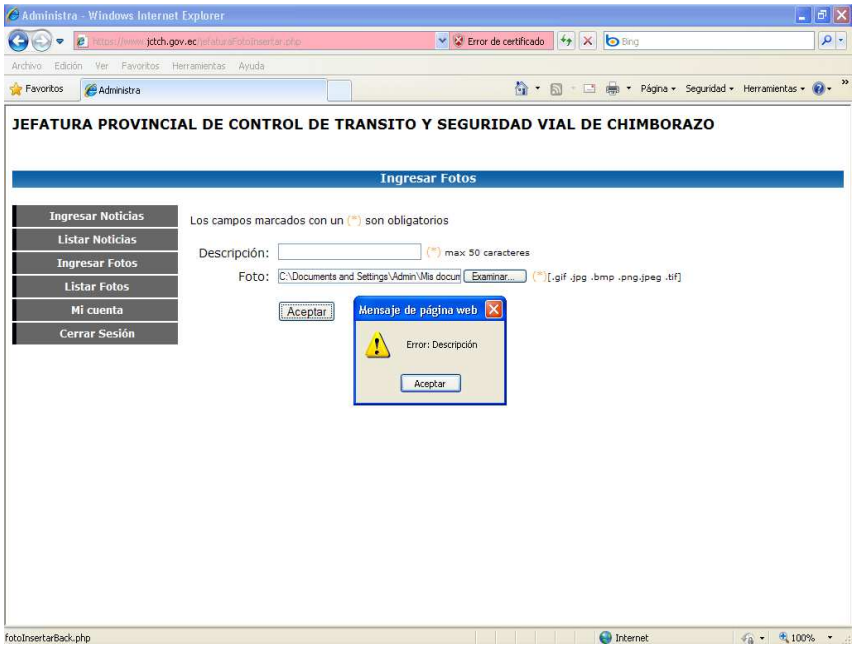

Gráfico IV.46: Prueba Ingreso Fotos

4.3.5.4 Cuenta FISCALIA

Al igual que en la cuenta administrador deberá tener muy en cuenta la forma de la actualización de sus datos. Si el usuario de tipo FISCALIA no selecciona correctamente un informe y lo descarga se le muestra el siguiente mensaje de error.

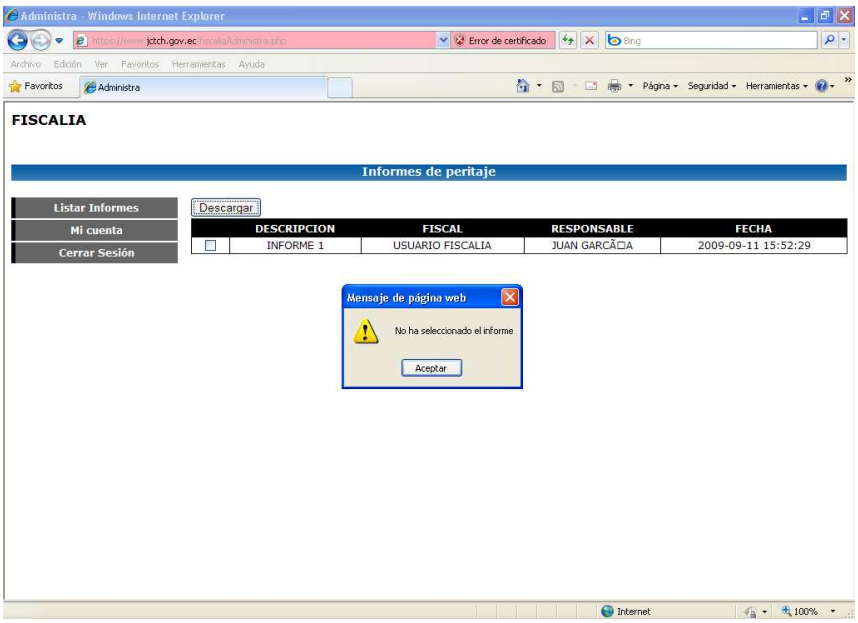

Gráfico IV.47: Prueba Descargar Informes Peritaje

4.3.5.5 Cuenta INGENIERÍA DE TRÁNSITO

Esta cuenta realiza funciones descritas anteriormente en las cuentas: Administración, Uiat, Jefatura, Fiscalía, como por ejemplo: ingresar noticias, ingresar fotos y actualizar los datos de su cuenta. El usuario deberá realizar todas estas transacciones de forma correcta, sino visualizará los mensajes de error mostrados anteriormente.

4.3.5.6 Cuenta Usuario Público

El usuario público no realiza transacciones que requieran de ingreso de datos u otras situaciones en las que pueda caer en errores. Sin embargo al generar estadísticas y no seleccionar correctamente el año y el mes se le muestra el siguiente mensaje de error.

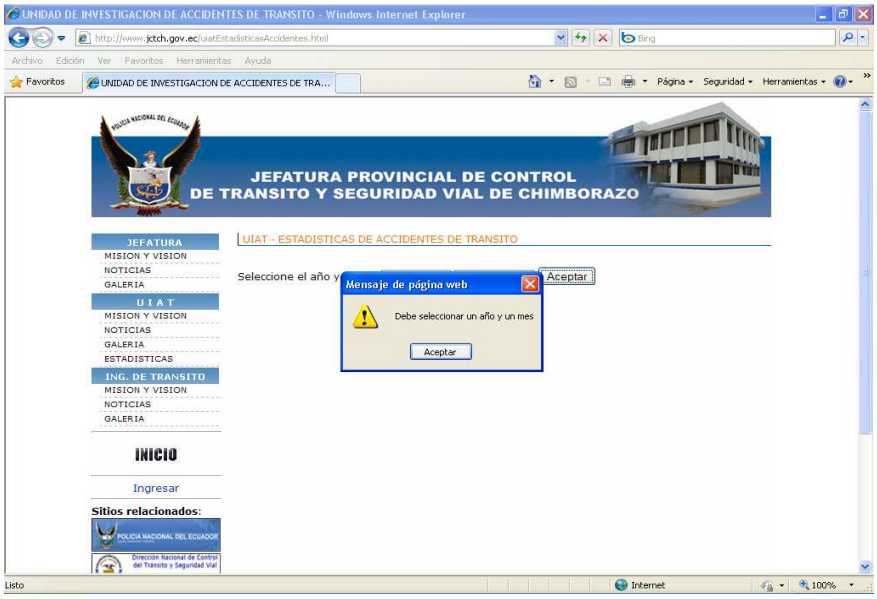

Gráfico IV.48: Prueba Generación de Estadísticas

4.4 Manual de usuario de la aplicación web "SONIA"

Ver Anexo 1

CONCLUSIONES

- Mediante la utilización de parámetros e indicadores se ha logrado ejecutar el análisis comparativo de las herramientas de código libre OpenSSL, GnuTLS y NSS concluyendo que la herramienta más idónea para implantar la aplicación web en un servidor seguro es OpenSSL.
- Se concluye luego de implantar la aplicación web "SONIA" sobre un servidor seguro que la información se transmite a través de la red de forma oculta o cifrada brindando de esta forma mayor seguridad y confiabilidad al usuario al momento de transmitir información sensible.
- En instalación de las herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS se puede concluir que la herramienta OpenSSL posee un mejor rendimiento en lo referente a instalación.
- Desde el punto de vista funcional se concluye que la herramienta OpenSSL presenta una mejor organización en la Generación de claves públicas, privadas y Certificados Digitales.
- Se concluye que la herramienta OpenSSL tiene un mayor grado de portabilidad por tener un mayor soporte para la instalación en diferentes Sistemas Operativos y su difusión en las aplicaciones que implementan seguridad.
- En lo referente al Soporte Técnico oficial en línea se puede concluir que la herramienta OpenSSL tiene una extensa información organizada no solo de sus librerías, sino de conceptos teóricos que ayudan a la mejor comprensión del funcionamiento de los protocolos SSL y TLS.
- Luego de estudiar los conceptos básicos sobre las funciones criptográficas, algoritmos de criptografía, firmas digitales y los diferentes tipos de licencias se puede concluir que algunos de los algoritmos de cifrado están patentados y que para su utilización se debe obtener la patente.
- Se ha concluido que los algoritmos simétricos son utilizados únicamente para cifrar la información a transmitirse y que los algoritmos asimétricos se utilizan para el intercambio de claves privadas.

RECOMENDACIONES

- Actualmente es recomendable utilizar la herramienta de administración de funciones criptográficas OpenSSL debido a su facilidad de uso ya que le permitirá optimizar recursos como el tiempo de instalación y configuración.
- Es recomendable utilizar firmas digitales firmadas por autoridades certificadoras públicas y reconocidas para evitar una posible suplantación de identidad ya que las autoridades certificadores verifican su valides cada vez que un cliente lo solicita, y además controlan que la firma digital no se encuentren en las listas de certificados revocados y de esta forma brindar mayor seguridad a los usuarios.
- Para la instalación de las herramientas de software libre sobre sistemas operativos basados en la plataforma UNIX es recomendable leer cuidadosamente los archivos README e INSTALL que vienen incluidos en los paquetes de instalación.
- Es recomendable investigar las patentes, licencias y restricciones del algoritmo de cifrado que va a utilizar ya sea este simétrico o asimétrico.
- En lo referente a la longitud de clave de los algoritmos de cifrado se recomienda realizar un estudio de los parámetros de seguridad con respecto a los parámetros de eficiencia por cuanto estos son inversamente proporcional.
- Para el estudio de herramientas similares a las herramientas objeto de estudio de la presente investigación es recomendable desarrollar paralelamente la parte teórica con la parte práctica.
- Se recomienda un seguimiento en investigación de estas tres herramientas de administración de funciones criptográficas OpenSSL, GnuTLS y NSS debido al gran desarrollo y crecimiento en el que se encuentran estas herramientas.

RESUMEN

Proyecto de investigación con el objetivo de estudiar y analizar comparativamente herramientas de administración de funciones de cifrado OpenSSL, GnuTLS y NSS, debido a que es importante contar con seguridad en la transmisión de la información vía internet basada en protocolos TLS y SSL que hacen uso de la criptografía para enviar información cifrada entre el cliente y el servidor evitando ataque de terceros.

En la ejecución del análisis comparativo se determinaron y se describieron los siguientes parámetros de comparación y evaluación: Instalación, Seguridad, Funcionalidad, Portabilidad y Soporte Técnico y Situación Legal, con sus respectivos indicadores.

Mediante Estadística Descriptiva se realizó la representación en un diagrama de barras simple de la sumatoria obtenida de la tabulación de todos los parámetros, determinándose que la herramienta más idónea para implantar un servidor seguro es la herramienta OpenSSL que alcanzó un porcentaje de 93.33% seguido de la herramienta GnuTLS con un 83.33% y la herramienta NSS con un 62.22%.

Se determina que la herramienta más idónea en base a los indicadores planteados es OpenSSL con un 93.33% por características como la facilidad de uso, difusión, mejor soporte técnico y funcionalidad en el aseguramiento de la transmisión de información de aplicaciones web.

SUMMARY

Research Project with the aim of comparative analysis of the encrypted administration tools OpenSSL, GnuTLS and NSS, given the importance of secure transmission of information via internet using the TLS and SSL encrypted protocols for information transfer between clients and servers and the avoidance of third party attacks.

The following parameters for comparison and evaluation were identified and described: installation, security, functionality, portability and technical support and legal situation, with relative indicators.

Using descriptive statistics the values attributed to each parameter were summed and represented through a histogram. OpenSSL was found to be the most suitable tool for server security with a percentage of 93.33%, followed by GnuTLS with 83,33% and NSS with 62.22%.

Based on the indicators used OpenSSL was found to be the most suitable tool with 93.33%, for characteristics such as ease of use, diffusion, best technical support and functionality, and security of information transfer for web applications.

GLOSARIO

AJAX (Acrónimo de Asynchronous JavaScript And XML) no es un lenguaje de programación en sí mismo sino la combinación de una serie de tecnologías que permiten una relación cliente-servidor más eficaz agilizando la respuesta de este último.

CA En criptografía una Autoridad de certificación (AC o CA por sus siglas en inglés Certification Authority) es una entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados, utilizados en la firma electrónica, para lo cual se emplea el cifrado de clave pública.

Certificado Digital Es un documento digital mediante el cual un tercero confiable (CA) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública.

Criptografía Asimétrica La criptografía asimétrica utiliza un par de claves para cifrar la información, este par de claves son únicos, la una clave es de dominio público por lo que cualquier persona puede saber pero la otra es privada y únicamente debe conocer su propietario.

Criptografía Simétrica Es el método criptográfico que usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar.

CRLs Una CRL es una lista de certificados que han sido revocados, ya no son válidos y en los que no debe confiar ningún usuario del sistema.

Firma Digital La firma digital hace referencia, en la transmisión de mensajes telemáticos y en la gestión de documentos electrónicos, a un método de cifrado que asocia la identidad de una persona o de un equipo informático al mensaje o documento. En función del tipo de firma, puede, además, asegurar la

integridad del documento o mensaje.

GnuTLS Es un proyecto desarrollado como un conjunto de librerías que provee una capa segura. Este conjunto de librerías podría ser utilizado por cualquier aplicación cliente servidor que lo requiera bajo las licencias en que se encuentra desarrollado el proyecto.

Licencia Es una especie de contrato, en donde se especifican todas las normas y clausulas que rigen el uso de un determinado programa, principalmente se estipulan los alcances de uso, instalación, reproducción y copia de estos productos.

MySQL Es un sistema que administra base de datos relacionales, multihilo y multiusuario que actualmente es desarrollada por la empresa MySQL AB y Sun Microsystems.

NSS Es un proyecto de código abierto desarrollado por la Fundación Mozilla cuyo principal objetivo es desarrollar un conjunto completo de librerías que implemente los protocolos SSL, TLS y S/MIME. Además de una amplia variedad de algoritmos de cifrado simétricos, asimétricos y funciones hash para brindar seguridad a aplicaciones cliente servidor.

OpenPGP Es un estándar de internet para la interoperabilidad de mensajes protegidos con criptografía.

OpenSSL Es una implementación Open Source de los protocolos SSL y TLS y librerías de cifrado para desarrollo de algoritmos de cifrado de cifrado, certificados x.509 y firmas digitales.

PHP Es un lenguaje de programación interpretado. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos.

Protocolo Handshake	Se encarga de la negociación para el establecimiento de parámetros antes de realizar una conexión segura.
Servidor web Apache	El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix, Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.
SONIA	Es una aplicación web creada para la gestión de informes y noticias de la JPTCH y UIAT de Chimborazo implantada sobre un servidor web seguro.
SSL	(Secure Sockets Layer, Capa de Sockets Seguro) fue desarrollado por Netscape, el principal objetivo de este protocolo es proveer de privacidad y confiabilidad a la comunicación entre aplicaciones cliente servidor.
SMIME	Extensiones de Correo de Internet de Propósitos Múltiples Seguro es un estándar para criptografía de clave pública y firma de correo electrónico encapsulado en MIME.
TLS	El protocolo TLS surge como necesidad de estandarizar un protocolo que provea de seguridad entre el cliente y el servidor a través del internet, debido a que el protocolo SSL es un protocolo creado y patentado por Netscape.
X.509	Estándar de certificados digitales y asume un sistema jerárquico estricto de autoridades certificantes (ACs) para emisión de certificados.

BIBLIOGRAFÍA

Criptografía

- **3DES** [en línea]
  <http://csrc.nist.gov/cryptval/des/tripledesval.html#451>
  20081021
- **AES** [en línea]
  <http://csrc.nist.gov/cryptval/aes/aesval.html#420>
  20081023
- **CEE** [en línea]
  <http://es.wikipedia.org/wiki/Criptografía_de_curva_elíptica>
  <http://www.segu-info.com.ar/criptologia/asimetricos.htm>
  20081025
- **Cifrado de flujo** [en línea]
  http://es.wikipedia.org/wiki/Flujo_de_cifrado>
  20081026
- **Cifrado por bloques** [en línea]
  <http://es.wikipedia.org/wiki/Cifrado_por_bloques>
  20081027
- **Criptografía asimétrica** [en línea]
  <http://es.wikipedia.org/wiki/Clave_pública>
  20081026
- **DH** [en línea]
  <http://es.wikipedia.org/wiki/Diffie-Hellman>
  20081027
- **DSA** [en línea]
  <http://csrc.nist.gov/cryptval/dss/dsaval.htm#175>
  20081028
- **Funciones hash** [en línea]

<http://es.wikipedia.org/wiki/Hash>

<http://ict.udlap.mx/people/carlos/is215/ir09.html>

20081029

- **HMAC** [en línea]

  <http://csrc.nist.gov/cryptval/mac/hmacval.html#194>

  20081029

- **RNG** [en línea]

  <http://csrc.nist.gov/cryptval/rng/rngval.html#216>

  20081029

- **RSA** [en línea]

  <http://csrc.nist.gov/cryptval/dss/rsaval.html#177>

  20081029

  <http://es.wikipedia.org/wiki/RSA>

  20081030

- **RSA, DSS, DH** [en línea]

  <http://gaussianos.com/criptografia-cifrado-de-clave-publica-ii/>

  <http://www.eurologic.es/cifrado/clavepub.htm>

  20081031

- **SHA-1,2** [en línea]

  <http://csrc.nist.gov/cryptval/shs/shaval.htm#490>

  20081031

Gestor de Base de Datos MySQL

- **MySQL** [en línea]

  <http://es.wikipedia.org/wiki/MySQL>

  20081013

  <http://www.mysql.com/>

  20081010

<http://www.netpecos.org/docs/mysql_postgres/x57.html>
20081015

Herramientas de Administración de Funciones Criptográfica

- **Herramienta GnuTLS** [en línea]
  <http://www.gnu.org/software/gnutls>
  20081105
- **Herramienta NSS** [en línea]
  <http://en.wikipedia.org/wiki/network_security_services>
  <http://www.mozilla.org/projects/security/pki/nss>
  20081110
- **Herramienta OpenSSL** [en línea]
  <http://csrc.nist.gov/cryptval/140-1/140sp/140sp733.pdf>
  <http://www.openssl.org>
  20081104
- **Referencias de NSS** [en línea]
  <http://www.mozilla.org/projects/security/pki/nss/ref/ssl/>
  20081112
- **Soporte técnico de GnuTLS** [en línea]
  <http://josefsson.org/>
  20090110

Lenguaje de Programación Interpretado PHP

- **PHP** [en línea]
  <http://es.wikipedia.org/wiki/.php>
  20081030
  <http://www.php.net/>

20081029

Licencias software

- **Licencia Apache** [en línea]
  <http://es.wikipedia.org/wiki/Apache_License>
  20090117
- **Licencia GNU GPL** [en línea]
  <http://www.gnu.org/licenses/licenses.es.html>
  20081020
- **Licencia GNU GPL y GNU LGPL** [en línea]
  <http://www.gnu.org/licenses/license-list.es.html>
  20090117
- **Licencia MPL** [en línea]
  <http://es.wikipedia.org/wiki/Mozilla_Public_License>
  20091022
- **Licencias Software Libre** [en línea]
  <http://www.monografias.com/trabajos55/licencias-de-software/licencias-de-software2.shtml>
  20090117

Protocolos de transmisión segura SSL y TLS

- **Documentación de estándares SSL y TLS** [en línea]
  <http://csrc.nist.gov/cryptval/140-1/FIPS1402IG.pdf>
  20081019
  <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>
  20081018
- **Documentos RFC** [en línea]

<http://www.ietf.org/rfc/rfc2440.txt>

20081019

- **Documentos RFC** [en línea]

<http://www.ietf.org/rfc/rfc4880.txt

20081019

- **Especificaciones de SSL** [en línea]

<http://wp.netscape.com/eng/ssl3/>

20081018

- **Protocolo HTTPS** [en línea]

<http://es.wikipedia.org/wiki/HTTPS>

20081018

- **Protocolo SSL** [en línea]

<http://www.apache-ssl.org/>

20081015

<http://www.linuxtotal.com.mx/index.php?cont=info_seyre_001>

20081017

Servidor Web Apache

- **Servidor Apache** [en línea]

<http://es.wikipedia.org/wiki/Apache_http_server>

20081003

<http://www.apache.org/>

20081001

Tecnología AJAX

- **AJAX** [en línea]

<http://es.wikipedia.org/wiki/AJAX>

<http://www.elguruprogramador.com.ar/articulos/que-es-ajax.htm>
20081005

ANEXO 1