ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

TÓPICO DE GRADUACIÓN

"Implementación de un Web Site de Comercio Electrónico utilizando una infraestructura de red segura: Autoridad de Certificación, usando esquema PKI para generación de firmas digitales y certificados".

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

Presentada por:
Víctor Manuel Ponce Díaz
Wilson Fernando Peñafiel Anchundia
Christian Xavier Cobeña Pino

GUAYAQUIL – ECUADOR

Año 2005

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

A la ESPOL, a la Ing. Karina Astudillo, a nuestros profesores por su indispensable apoyo.

D E D I C A T O R I A

A Dios y a mi familia. Dios porque sin Él nada existiría. Mi Familia porque es el motor de mi vida.

Víctor Ponce

D E D I C A T O R I A

A mis padres Fernando y Hanna por su constante e incondicional apoyo, a mi hermana Daniela por su persistente ejemplo de superación y a mis compañeros Víctor y Christian porque juntos formamos un excelente equipo.

Fernando Peñafiel

D E D I C A T O R I A

A mis padres, hermana, y amigos por su apoyo incondicional y comprensión en cada momento de mi vida

Christian Cobeña

TRIBUNAL DE GRADUACIÓN

Ing. Miguel Yapur	Ing. Karina Astudillo
SUB-DECANO DE LA FIEC	DIRECTORA DE TÓPICO
Ing. Albert Espinal	Ing. Cristina Abad
MIEMBRO PRINCIPAL	MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Proyecto, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"

Víctor Ponce

Fernando Peñafiel

Christian Cobeña

RESUMEN

“La única red totalmente segura es aquella que está desconectada”. Sin lugar a dudas, la meta es ambiciosa. La Tecnología es un campo muy amplio que día a día va ampliando los horizontes de la Información, y de la mano con ésta el riesgo que produce un universo ahora unido a través del Internet.

El amplio desarrollo de las nuevas tecnologías informáticas está ofreciendo un nuevo campo de acción a conductas antisociales y delictivas manifestadas en formas antes imposibles de imaginar, ofreciendo la posibilidad de cometer delitos tradicionales en formas no tradicionales.

La mayoría del mundo informático desconoce la magnitud del problema con el que se enfrenta y, generalmente no se invierte ni el capital humano ni económico necesario para prevenir, principalmente, el daño y/o pérdida de la información que, en última instancia es el Conocimiento con que se cuenta.

El motivo del presente proyecto es desarrollar una infraestructura de seguridad suficiente para disminuir al mínimo posible la probabilidad de sufrir pérdidas de Información, y a la vez implementar una solución que permita a

los demás protegerse de los riesgos de seguridad, como es una Autoridad de Certificación.

El primer capítulo “Introducción” explica los conceptos básicos de acerca de la Seguridad de la Información, introduce los temas relacionados a los Certificados Digitales y las Autoridades de Registro y Certificación, además muestra los objetivos del proyecto y la realidad en el Ecuador sobre este tema.

El segundo capítulo “Diseño e Implementación de la Seguridad Física para una Autoridad de Certificación” describe la solución “Ideal” sugerida para una empresa de comercio electrónico, asimismo la solución implementada en el presente proyecto.

El tercer capítulo “Diseño e Implementación de la Aplicación Segura para una Autoridad de Certificación” nos lleva a través del desarrollo de un software de comercio electrónico para la venta de certificados digitales a través del Internet.

El cuarto capítulo “Implementación del Proyecto” presenta la solución desarrollada.

El quinto capítulo “Seguridades Adicionales” sugiere otros mecanismos y técnicas adicionales necesarias para reforzar la Seguridad de la Información.

En el sexto capítulo se establecen las conclusiones originadas a la terminación del proyecto así como las recomendaciones sugeridas a lo largo de su elaboración.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPITULO 1

1 INTRODUCCIÓN.

"Seria bueno que en Ecuador exista mucha más ciencia y tecnología de la que actualmente tenemos. Lamentablemente la economía del país no ha permitido incrementar los niveles de conocimiento internos, pero somos buenos para estar al tanto de lo que pasa en otros lados. Cosas como el Open Source[1] y los temas de Seguridades son ya comunes. La Seguridad de la Información es cada día un riesgo mayor y toma cada vez más acogida. También sería muy bueno que Ecuador tuviese su propio CERT[2] y su propia Autoridad de Certificación"[3].

[1] Open Souce: Fuente Abierta – Software que cumple con ciertos requisitos como Libre Distribución, Código Abierto, Licencia Independiente de Tecnología, entre otros.
2 CERT: Computer Emergency Response Team. Es un Equipo de Seguridad para la Coordinación de Emergencia en Sistemas Computacionales.
[3] Fuente: Equipo que desarrolló el presente proyecto, denominado "Autoridad de Certificación ECCert".

Las transacciones por Internet se han vuelto muy comunes, el uso del navegador y el desarrollo de sistemas adaptados completamente para la web se visualizan como el futuro de las aplicaciones que tradicionalmente estaban atadas a un instalador. Las nuevas tecnologías apuntan a tener sistemas completos, con acceso a datos y reportes de gerencia, sin barreras geográficas ni de distribución, y todo esto a través del Internet.

Es por esto que existe la necesidad de proteger nuestra información, tanto la que se encuentra dentro de nuestra organización, como la que viaja a través del Internet, que al cumplir con la necesidad de estar a la vista del mundo, puede crear un agujero por el cual un intruso siempre puede estar intentando acceder a ella.

La protección para nuestra Información, para nuestros equipos y servidores de datos y de aplicativos puede llegar a ser excesivamente cara. Actualmente existen muchos proveedores de equipos de seguridad, unos proporcionan más ventajas que el otro, pero en definitiva todos coincidimos en que es realmente necesario protegernos.

Es por esto que el presente proyecto intenta crear una conciencia de los potenciales del Open Source y el Software sin costo para la protección de nuestras redes y servicios empresariales, manteniendo un balance con el uso de equipos y aplicativos con costo, pero que en conjunto tratan de ser una solución óptima para una organización en busca de la seguridad de sus Sistemas de Información.

Además brindamos un conjunto de servicios que permitirán a otras empresas aumentar sus niveles de seguridad, todo esto mediante la emisión y administración de Certificados Digitales, proporcionado por nuestra Autoridad de Certificación mediante una aplicación de comercio electrónico cuyo producto es el Certificado Digital[4] y cuyos servicios giran alrededor de los servicios de seguridad y el soporte del Certificado Digital.

Y finalmente en nuestro proyecto brindamos esquemas de diseño de redes, una arquitectura de software muy útil, basada en patrones de diseño que garantizan la seguridad al nivel de todas las capas de la aplicación y acceso a datos, y además proporcionamos consejos adicionales, como políticas de seguridad y el uso de antivirus, que en conjunto proporcionan la base para que las organizaciones incorporen la Seguridad de la Información rápidamente.

1.1 Antecedentes

La criptografía con Claves Privada/Pública[5] es una poderosa herramienta en la encriptación de datos y por la tanto, es una tecnología de seguridad muy útil en la actualidad. El problema de ésta

[4] Certificado Digital: Es un documento electrónico que contiene datos identificadores de una persona o entidad y la clave (llave) pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Autoridad Certificadora.
[5] Criptografía con Claves Privada/Publica: Es una ciencia matemática usada para proveer confidencialidad y autenticidad en el intercambio de información usando algoritmos criptográficos que trabajan con claves públicas y privadas.

radica en la distribución de la clave pública[6]. Una manera segura de intercambiarlas es directamente de persona a persona, pero ésta no cumple los requerimientos de trabajo distribuido de los sistemas actuales. Pero existe otra manera, esto es, mediante un Certificado Digital.

La idea de un Certificado Digital se basa en firmas digitales: el dueño de alguna llave puede criptográficamente firmar un conjunto de datos. La firma digital asegura que los datos fueron generados por el propietario de alguna llave y que ésta información no ha sido modificada desde el momento que se firmó digitalmente.

Un Certificado no es otra cosa que un caso especial de un documento firmado que dice: “Certifico que la llave pública que aparece en éste documento pertenece a la entidad que aquí se menciona. Firma AC”. En donde AC podría ser cualquiera que tenga una clave pública y en cuya palabra se confíe, pero lo más común es que sea una Autoridad de Certificación.

Entonces dada la necesidad de la Seguridad de la Información surge el uso de la Criptografía con clave Privada/Pública proporcionando de ésta forma las seguridades requeridas y junto con este esquema nacen necesariamente las Autoridades de Certificación, que son

[6] Clave Pública: Es la clave que todos conocen para cifrar o descifrar un mensaje. Clave Privada: Es la clave que solo el emisor del mensaje conoce para cifrar o descifrar un mensaje.

empresas totalmente confiables que se dedican al negocio de generar y administrar Certificados Digitales.

1.2 Certificados Digitales, Autoridades de Registro y Autoridades de Certificación.

Es necesario tener una visión sistémica de lo que es la Seguridad de la Información. Para esto, a continuación presentamos unas definiciones comunes en el tema, y que a la vez nos enfoca hacia la verdadera utilidad de los Certificados Digitales, para que se utilizan y como se distribuyen a través de las Autoridades de Certificación.

Para poder afirmar que una comunicación entre dos entidades es segura se deben cumplir cuatro requisitos principales:

Autenticidad: Todas las entidades participantes en la transacción deben estar perfecta y debidamente identificadas antes de comenzar la misma. Se debe estar seguro de que la persona con la que nos comunicamos es realmente quien dice ser, ya que si no podemos estar facilitando datos íntimos y/o sensibles a una persona o entidad no deseada, que puede hacer con ellos luego lo que le venga en gana.

En las comunicaciones "normales" entre dos personas casi siempre se dispone se alguna forma de comprobación de la Autenticidad. Si

hablamos en directo con alguien, sabemos quién es, y si no lo sabemos podemos poner límites a la información que le facilitamos. En una conversación telefónica podemos oír la voz de nuestro interlocutor, y si lo conocemos bien es muy difícil que otra persona se pueda hacer pasar por él.

Pero en los viajes por la red no se tiene ninguna forma efectiva de saber con quién estamos comunicándonos. Podemos acudir a la página de una empresa, ver su dirección de dominio, ver en su página lo que dice, pero ¿cómo podemos saber que es realmente quien dice? Imaginemos una situación extrema: un pirata informático sin escrúpulos crea una página igual en todo a la de nuestro banco y nos manda un correo diciendo que es el director de nuestra sucursal y que hay un problema con una de nuestras cuentas, ofreciendo un enlace para que entremos en las páginas del banco. Pero ese enlace resulta que nos lleva a su página falsificada, en la que nos pide que introducir nuestras claves de acceso. En cuanto lo hagamos y le demos al botón de enviar, el pirata las conocerá, con todo lo que ello puede significar.

Lo ideal en este sentido sería que el cliente en una transacción de compra por Internet sólo debiera garantizar que es el legítimo propietario de la tarjeta de crédito que está usando en la misma, sin tener que hacer pública su identidad, por muchas leyes de protección de datos que estén vigentes.

La **Autenticidad** se consigue mediante el uso de algoritmos de clave pública (Certificados Digitales) y algoritmos simétricos.

Confidencialidad: Se debe estar seguros de que los datos que se envían no pueden ser leídos por otra persona distinta del destinatario final deseado, o que si ocurre esto, el espía no pueda conocer el mensaje enviado. O en su defecto, que cuando consiga obtener los datos éstos ya no le sirvan para nada. Es decir, debemos estar seguros de que ninguna persona ajena a la transacción puede tener acceso a los datos de la misma.

Imaginemos ahora que trabajamos en una empresa y deseamos enviar un correo al director general explicándole el fabuloso contrato que se está a punto de firmar con un cliente. Si algún pirata está a la escucha, puede conocer al momento todos los detalles del trato que vamos a realizar, pudiendo vender esa información a la competencia, lo que puede arruinar el negocio antes de hacerse realidad (¿qué impresión le causaría a un cliente si recibiera información detallada de sus actividades con nosotros por medio de un correo anónimo?).

Lo ideal en este aspecto sería que las entidades implicadas en la transacción no llegaran a conocer más que los datos imprescindibles para realizar su función.

La **Confidencialidad** de consigue en las transacciones electrónicas con el uso de la Criptografía.

Integridad: Es necesario estar seguro de que los datos que enviamos llegan íntegros, sin modificaciones, a su destino final.

En este caso estamos realizando el pedido de un computador de $1.500 a una tienda virtual, introducimos nuestro número de tarjeta de crédito y nuestra dirección de entrega del equipo. Pero si un pirata está a la escucha, intercepta el envío, puede cambiar los datos de la dirección por otros a su gusto y deja que continúe el envío. El resultado será que el pirata disfrutará de un computador que otra persona ha pagado.

La **Integridad** se consigue combinando Criptografía, funciones hash y firmas digitales.

No repudio: Se debe estar seguro de que una vez enviado un mensaje con datos importantes o sensibles el destinatario de los mismos no pueda negar el haberlos recibido. Y en una compra en línea debe garantizarse que una vez finalizada la misma ninguna de las partes que intervienen pueda negar haber participado en ella.

Un caso puede ser que nuestra empresa tiene que enviar un presupuesto antes de una fecha determinada, presupuesto que debe

ser recogido por un empleado de otra empresa, y que éste olvida comunicar a sus superiores dicha recepción. Pasa el plazo y el contrato que esperábamos se lo dan a otra empresa, alegando que no han recibido a tiempo el presupuesto nuestro. Si no disponemos de un medio para atestiguar que el mensaje fue entregado en plazo, nos quedaremos sin contrato y sin poder reclamar.

Lo ideal sería que al finalizar la transacción quedara algo equivalente a un recibo de compra o factura firmado por todas las partes implicadas.

El **No Repudio** se consigue mediante los certificados y la firma digital.

La siguiente tabla resume los requisitos anteriormente mencionados o también llamados Servicios de Seguridad y además con que algoritmo o función se lo puede suplir:

Servicio de Seguridad	Algoritmos / Función que Utiliza
Autenticidad o Autentificación	Clave Simétrica (DES) Clave Pública (RSA - Certificados Digitales)
Confidencialidad	Criptografía (Cualquier algoritmo criptográfico)
Integridad	Criptografía más funciones hash y firmas digitales
No Repudio	Certificados y firmas digitales

Tabla 1.1 Servicios de Seguridad y Algoritmos/Función que utiliza.

Fuente: ECCert
Autor: ECCert

Hasta ahora hemos mencionado algunas palabras claves en el ambiente de la Seguridad de la Información. Para aclarar un poco más estos términos, en la siguiente sección presentamos las definiciones más comunes en el ambiente de la Seguridad de la Información:

Criptografía: La criptografía es una rama de las Matemáticas y de la Informática que se vale de métodos para codificar o cifrar un texto o archivo por medio de un algoritmo y una o más claves. Su fin es el de asegurar tanto la Confidencialidad como la Integridad de la información.

Los algoritmos usados para el cifrado de la información pasan desde los clásicos (como la clave César[7]) a los modernos, los cuales podemos subdividir en tres grupos: de clave simétrica, de clave pública e híbridos.

Cifrado simétrico y asimétrico: Antes de explicar firmas y certificados digitales, es necesario realizar un breve resumen de las técnicas de cifrado, ya que en estas técnicas se basan los Certificados Digitales.

[7] Algoritmo Cesar: Consiste en usar la fórmula n+3, en donde "n" es la letra del alfabeto. Por ejemplo, cuando queremos encriptar la palabra LOCO, cada letra se sustituye por la tercera próxima del alfabeto, es decir ORFR. Para descifrarla tenemos que aplicar el algoritmo a la inversa, es decir n-3.

El cifrado consiste básicamente en la modificación de la información mediante la aplicación de un algoritmo matemático conocido, sobre el mensaje que se desea cifrar utilizando como parámetro una clave en la cual se basa la confidencialidad del mensaje cifrado. Existen dos técnicas de cifrado y una derivada de la combinación de éstas:

Cifrado simétrico: consiste en el uso de la misma clave para cifrar el mensaje y para descifrarlo. Los algoritmos DES, Triple DES, IDEA son ejemplos de este tipo de algoritmos y tienen como ventaja el que son óptimos computacionalmente porque usan claves de tamaños reducidos (por ejemplo DES usa 56 bits de clave). Por otro lado plantean serios problemas en la gestión de las claves, ya que el usuario que cifra la información debe confiar su clave a todos aquellos individuos que deban acceder a la información.

Los algoritmos de cifrado simétrico más comunes son:

Algoritmo	Bloque (Bits)	Clave (Bits)	Comentarios
Lucifer	128	128	Antecesor del DES
DES	64	56	EL más usado e inseguro
Loki	64	64	
RC2	64	Variable	
CAST	64	64	

Blowfish	64	Variable	
IDEA	64	128	Muy bueno, lo usa PGP
Skipjack	64	80	
RJNDAEL	128	128 o más	Nuevo estándar AES (Advanced Encryption Algorithm)

Tabla 1.2 Algoritmos de Cifrado Simétrico.

Fuente: ECCert
Autor: ECCert

Cifrado asimétrico: También llamados Cifrado de Clave Pública. Cada usuario dispone de un par de claves, una clave pública y una clave privada, de forma que todo aquello que sea cifrado con la clave pública sólo podrá ser descifrado con la privada y viceversa. En estos algoritmos de clave pública la clave privada generalmente será de 1024 o 2048 bits y la pública suele ser de 128 bits. Ya que estos algoritmos de cifrado suelen usar claves de tamaños elevados, no se usan para el cifrado de mensajes enteros, solo para generar las firmas digitales. Para realizar la firma solo tendremos que realizar una Función Hash y luego cifrarla con la clave privada. Una función hash consiste en un algoritmo criptográfico muy rápido, de uso público e irreversibles, esto último significa que es una función resumen de un solo sentido, como un checksum del contenido del mensaje. Así, si el mensaje se modifica por el camino, al descifrar el hash y recalcularlo este nunca será el mismo que el del mensaje original, esto nos asegura la integridad del mensaje.

Si bien computacionalmente no es tan eficiente como el cifrado simétrico, soluciona el problema de distribución de las claves.

Los algoritmos de clave asimétrica más comunes son:

- Diffie Hellman (DF): fue el primero, desarrollado en 1976.
- RSA: es el mas usado, desarrollado en 1978.
- ElGamal: el más moderno (1985).

Algoritmos de funciones resumen (Hash) más comunes:

- MD5: Se usa, entre otros usos para las contraseñas de Unix (junto con DES). Genera hashes de 128 bits.
- SHA-1: similar al MD5 pero con resúmenes de 160 bits, lo que le hace más seguro.
- HMAC: combina los anteriores con claves asimétricas, lo que le dota de mayor seguridad.

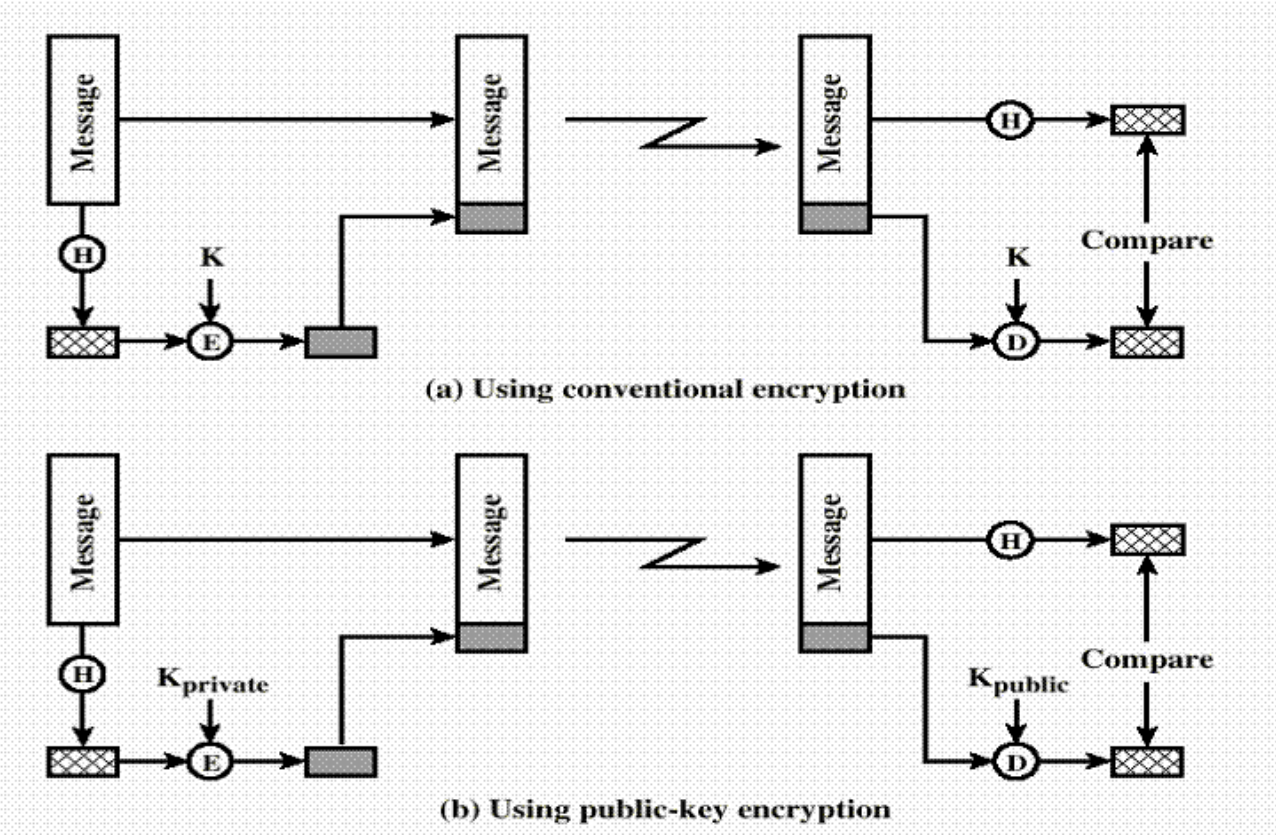

Figura 1-1 Enfoque típico de uso de firma digital en ambos tipos de cifrado

Fuente: Information Security Second Edition
Autor: Stallings William.

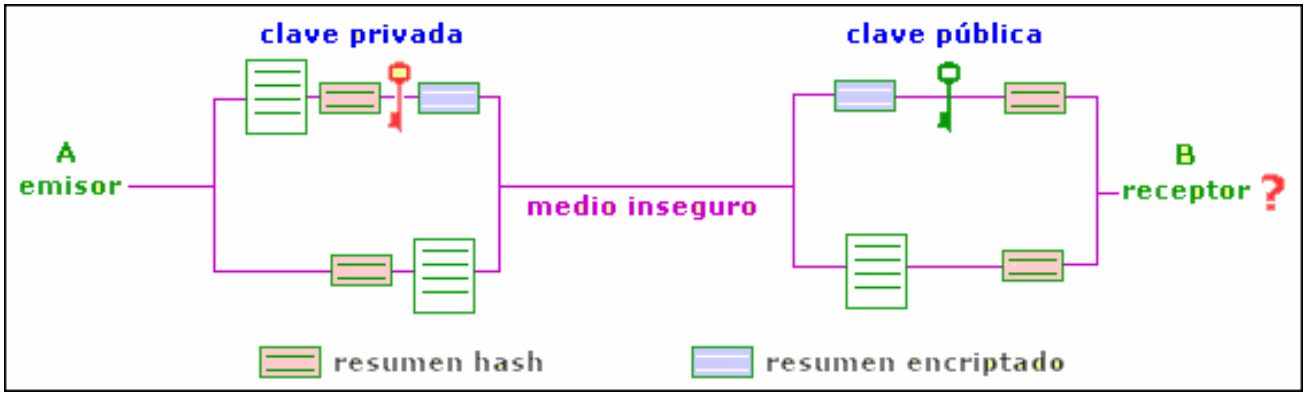

Figura 1-2 Firma digital con resumen Hash.

Fuente: http://www.htmlweb.net/seguridad/ssl/ssl_3.html
Autor: Luciano Moreno, del departamento de diseño web de BJS Software.

.

Sistemas de cifrado Híbridos: Como se pudo notar los algoritmos de cifrado simétrico o clave privada permiten cifrar grandes volúmenes de datos sin cargar en exceso el procesador. En cambio con los de cifrado simétrico el cifrar grandes volúmenes de datos con la clave privada es en la práctica una tarea lenta y tediosa.

Por eso siempre usaremos algoritmos de clave simétrica para cifrar la información, pero aquí el problema es que el intercambio de claves se produce a través de un canal inseguro. Si se compromete la clave se rompe el sistema. Pero la principal peculiaridad de los algoritmos de clave asimétrica es que pueden transmitir una pequeña información de manera segura. Así pues, se puede cifrar la clave con un algoritmo de clave pública. Esto es un criptosistema híbrido, y es la clave del IKE, o lo que es igual, la clave de que se puedan realizar comunicaciones seguras a través de Internet.

En general, el propósito de una firma digital es proveer una forma para que una entidad pueda vincular su identidad con una pieza de información. El proceso de firmar involucra transformar el mensaje y alguna información secreta conocida por la entidad en una sección llamada *firma*. Es una primitiva criptográfica que es fundamental para la autenticación, autorización y no-repudio. Una firma digital cambia de acuerdo al contenido del mensaje; se puede tener una combinación de firma digital y encriptación.

La firma digital es una herramienta basada en las técnicas de cifrado asimétrico, con la que se consigue garantizar la autenticidad e integridad de un mensaje. Supongamos que el usuario A(ntonio) desea enviarle un mensaje a B(eatriz); ambos tienen un par clave pública/privada. Antonio conoce la clave pública de Beatriz y ésta conoce la clave pública de Antonio. Si Antonio envía el mensaje acompañado de una copia del mensaje (en realidad es una función de éste) cifrada con su clave privada, cuando Beatriz reciba el mensaje - utilizando para descifrar el mensaje cifrado la clave pública de Antonio – podrá comprobar que sólo éste pudo enviar el mensaje (autenticidad y no-repudio) ya que sólo él conoce su clave privada. Además queda garantizado que el mensaje no ha sido alterado, ya que la copia en claro, o función de ésta, y la copia cifrada deben coincidir (integridad).

De esta forma, cualquiera que disponga de la clave pública de Antonio puede verificar cualquier texto firmado por él, ya que sólo él puede haberlo firmado con su clave privada.

Certificado Digital: Para solucionar el problema de la Autentificación en las transacciones por Internet se buscó algún sistema identificador único de una entidad o persona. Ya existían los sistemas criptográficos de clave asimétrica, mediante los cuales una persona disponía de dos claves, una pública, al alcance de todos, y otra privada, sólo conocida por el propietario. Cuando se desea enviar un

mensaje confidencial a otra persona, basta cifrarlo con su clave pública, y así se puede estar seguro de que sólo el destinatario correcto podrá leer el mensaje en claro.

El problema es estar seguro de que efectivamente la clave pública que nos envían sea de la persona correcta, y no de un suplantador. Entonces se pensó en implementar una especie de documento de identidad electrónica que identificará sin dudas a su emisor.

La solución a este problema la trajo la aparición de los Certificados Digitales o Certificados Electrónicos, documentos electrónicos basados en la criptografía de clave pública y en el sistema de firmas. La misión principal de un Certificado Digital es garantizar con toda confianza el vínculo existente entre una persona, entidad o servidor web con una pareja de claves correspondientes a un sistema criptográfico de clave pública.

Un Certificado Digital es un documento electrónico que contiene datos identificadores de una persona o entidad (empresa, servidor web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Autoridad Certificadora. Las principales Autoridades Certificadoras actuales son Verisign (filial de RSA Data Security Inc.) y Thawte.

El certificado Digital vincula indisolublemente a una persona o entidad con una llave pública, y mediante el sistema de firma digital se asegura que el certificado que recibimos es realmente de la persona que consta en el mismo. El sistema de firma digital liga un documento digital con una clave de cifrado.

El procedimiento de firma digital lo que hace es obtener un resumen de un documento o de un texto aleatorio y cifrarlo con llave privada del propietario del certificado. Cuando nos llega un certificado, y su firma digital asociada, tan sólo se debe obtener el resumen del mismo, descifrar la firma con la llave pública del remitente y comprobar que ambos resúmenes coinciden, lo que nos hace estar totalmente seguros de la autenticidad del certificado. Se firma un resumen del documento y no el documento mismo para evitar ataques contra el sistema de cifrado RSA (por ejemplo, encriptar un documento especialmente concebido por un pirata, con lo que éste podría llegar a obtener la llave privada) y para no hacer el proceso demasiado lento.

Los procesos de validación de certificados, obtención de resúmenes, descifrados y comprobación de coincidencia se realizan por el software adecuado del navegador web o programa de seguridad particular de forma transparente al usuario, por lo que éste será informado sólo en el caso de que el certificado no sea válido.

PKI: Básicamente una Infraestructura de Clave Publica (PKI[8]) es el conjunto de hardware y software necesarios para la creación, administración, distribución y revocación de certificados digitales. Todo esto nos permitirá el poder enviar datos de manera segura a través de Internet.

Modelos de PKI: Existen dos modelos de PKI, las PKI centrales y las jerárquicas.

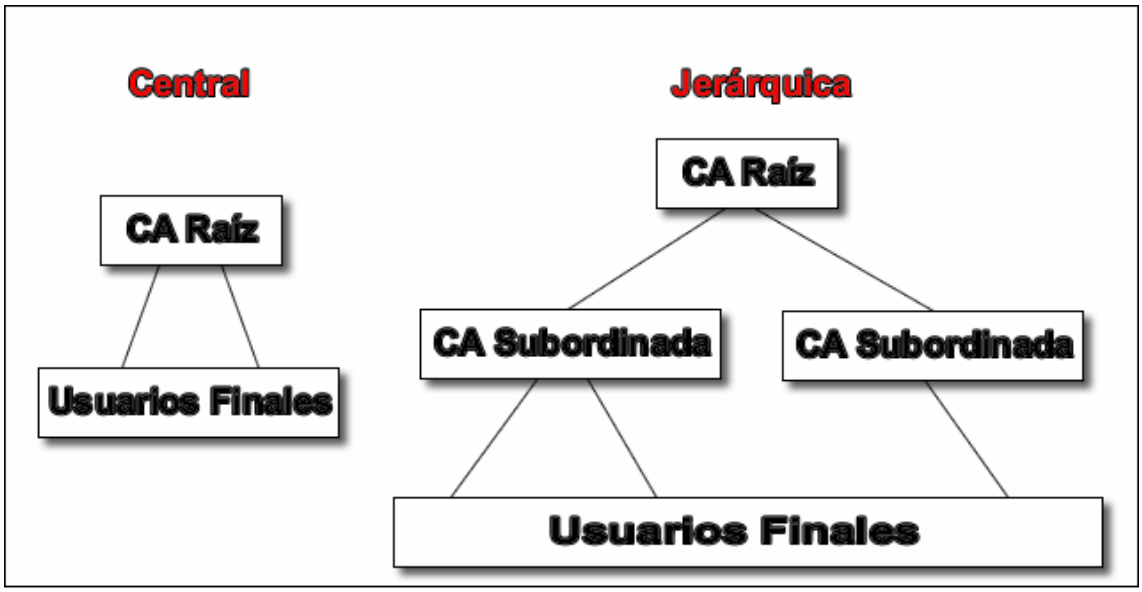

Figura 1-3 Modelos de PKI.

Fuente: es.comp.hackers
Autor: ECCert.

En las PKI Centrales los certificados son firmados por el CA Raíz, y todos los certificados son comprobados con la clave pública de esa CA.

[8] PKI: Public Key Infraestructure.

En las PKI Jerárquicas, el CA Raíz firma los certificados para las CA subordinadas y éstas a su vez se los firman a los usuarios finales. Para verificar un certificado, éste debe ser validado por las CA's en sentido ascendente hasta llegar a la CA raíz.

En los sistemas Microsoft, el esquema de PKI central se denomina: CA Standalone, y el Jerárquico se denomina: Integrado en Directorio Activo.

Como se vio anteriormente, un certificado básicamente es una clave pública empaquetada en un determinado formato. La pregunta que surge es ¿por qué se hace esto?, la razón es para que la gestión de las claves públicas sea automatizada en grado sumo. Eso es una PKI. Las máquinas expiden el certificado, calculan las claves públicas, introducen los atributos en base a los datos de la petición del usuario, y comprueban su validez mientras se navega por Internet.

La operación de cara al usuario es la siguiente. Tomando el modelo de PKI central por ser más sencillo:

- Un usuario realiza una petición de un certificado (CSR[9]) a una Autoridad de Certificación CA, para lo que tiene que

[9] CSR: Certificate Signing Request. Es un certificado sin firma que se utiliza para petición a una Autoridad de Certificación, la cual la firma con su clave privada. Una vez que el CSR es firmado, se convierte en un Certificado Digital.

entregarle sus datos personales y puede entregarle una clave pública autogenerada o que se la genere el CA.

- La CA genera el certificado del usuario con sus datos como atributos y luego la firma con su propia clave privada (la de la CA). Este es el punto clave del tema, la CA firma digitalmente el certificado del usuario con su clave privada.
- Por último, dependiendo de como se haya hecho la petición, el certificado es entregado al usuario. Si la petición se hizo a través del explorador es muy posible que una vez que lo reciba, lo instale automáticamente. En el caso de certificados para servidores estos se guardaran en formatos específicos para incluirlos en el arranque del servidor y así poder usarlo.

En esto hay un concepto que es el de la confianza, o sea que si una CA dice que una persona es quien dice ser, habrá que creerle a la CA.

Ahora, hay que entender para qué se quiere empaquetar una clave pública en un certificado y no usarla directamente. La diferencia es que una clave que nos hemos generado, por ejemplo con PGP[10] siempre nos identificará a nosotros mismos, mientras que en una

[10] PGP: Pretty Good Privacy. Es una aplicación que utiliza encriptación híbrida, que combina encriptación convencional con encriptación asimétrica (Clave pública/privada).

clave pública incluida en un certificado siempre será la CA quien nos identifique ante los demás.

De esta manera, en vez de tener 2000 usuarios, cada uno con un par de claves propias y personales y teniendo que firmarse los unos a los otros para poder tener las claves públicas de los demás, sólo se necesita una CA y un navegador y se podrá tener a los 2000 usuarios usando todas las ventajas de las claves asimétricas y además todo ello gestionado por el administrador de la CA.

Así pues, se puede definir un certificado como un documento que, dentro de una infraestructura de clave pública (PKI) nos va a permitir usar todas las ventajas de los algoritmos de clave asimétrica, teniendo en cuenta de que ésto implica la gestión de accesos, uso de cifrado, autenticación e integridad.

TIPOS DE CERTIFICADOS: Dependiendo del uso que se vaya a dar al certificado y de qué persona o entidad lo solicita, las Autoridades Certificadoras han dividido los certificados en varios tipos. Del tipo de certificado a emitir van a depender las medidas de comprobación de los datos y el precio del mismo.

Los certificados, según las comprobaciones de los datos que se realizan, se dividen en cuatro clases:

- **Certificados de Clase 1: C**orresponde a los certificados más fáciles de obtener e involucran pocas verificaciones de los datos que figuran en él: sólo el nombre y la dirección de correo electrónico del titular confirmado mediante el envío de información vital.

- **Certificados de Clase 2:** En los que la Autoridad Certificadora comprueba además la dirección postal, el permiso de conducir, el número de la Seguridad Social y la fecha de nacimiento.

- **Certificados de Clase 3:** En la que se añaden a las comprobaciones de la Clase 2, la verificación de crédito de la persona o empresa, además la persona debe aparecer en persona o con algún sistema de confirmación de datos y proporcionar documentos notariados.

- **Certificados de Clase 4:** *Q*ue a todas las comprobaciones anteriores suma la verificación del cargo o la posición de una persona dentro de una organización.

Desde el punto de vista de la finalidad, los certificados electrónicos se dividen en:

- **Certificados SSL para cliente:** Usados para identificar y autenticar a clientes ante servidores en comunicaciones mediante el protocolo Secure Socket Layer[11], y se expiden normalmente a una persona física, bien un particular, bien un empleado de una empresa.

- **Certificados SSL para servidor: U**sados para identificar a un servidor ante un cliente en comunicaciones mediante el protocolo Secure Socket Layer, y se expiden generalmente a nombre de la empresa propietaria del servidor seguro o del servicio que éste va a ofrecer, vinculando también el dominio por el que se debe acceder al servidor. La presencia de éste certificado es condición imprescindible para establecer comunicaciones seguras SSL.

- **Certificados S/MIME:** Usados para servicios de correo electrónico firmado y cifrado, que se expiden generalmente a una persona física. El mensaje lo firma digitalmente el remitente, lo que proporciona Autenticación, Integridad y No Repudio. También se puede cifrar el mensaje con la llave pública del destinatario, lo que añade Confidencialidad al envío.

[11] SSL: Secure Socket Layer. Es un protocolo para transmitir nuestra información a través del Internet de manera encriptada.

- **Certificados de firma de objetos:** Usados para identificar al autor de ficheros o porciones de código en cualquier lenguaje de programación que se deba ejecutar en red (Java, JavaScript, CGI, etc). Cuando un código de éste tipo puede resultar peligroso para el sistema del usuario, el navegador lanza un aviso de alerta, en el que figurará si existe certificado que avale al código, con lo que el usuario puede elegir si confía en el autor, dejando que se ejecute el código, o si por el contrario no confía en él, con lo que el código será rechazado.

- **Certificados para AC:** que identifican a las propias Autoridades Certificadoras, y es usado por el software cliente para determinar si pueden confiar en un certificado cualquiera, accediendo al certificado de la AC y comprobando que ésta es de confianza.

Toda persona o entidad que desee obtener un certificado debe pagar una cuota a las Autoridades de Certificación, cuota que irá en función de la clase del certificado y del uso que se le vaya a dar al mismo (ambas están relacionadas). A mayor nivel de comprobación de datos (clase mayor), más costará el certificado.

X.509: Es el formato estándar de los Certificados Digitales, siendo X.509 v3 el recomendado por la Unión Internacional de Comunicaciones (ITU) y el que está en vigor en la actualidad. Según este formato se puede tener un conjunto de servidores distribuidos que mantengan una base de datos sobre los usuarios. En resumen, cada certificado contiene la clave pública de un usuario y es firmada con la clave privada de la Autoridad de Certificación. Es usado en S/MIME, IP Security, SSL/TLS, SET, entre otros, que son protocolos para transmisión de información segura.

A continuación se muestra el aspecto de los certificados X.509:

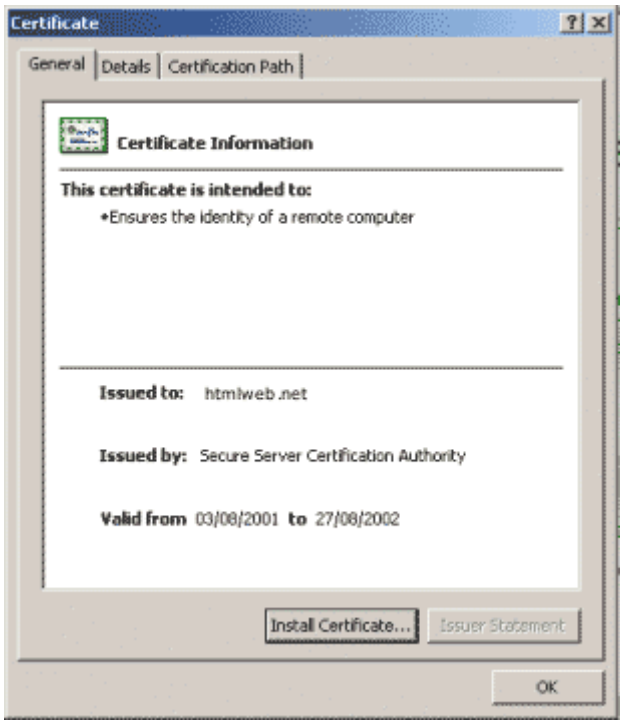

Figura 1-4 Vista General de un Certificado Digital X.509 v3.

Fuente: www.htmlweb.net/seguridad/ssl/ssl_3.html
Autor: Desconocido.

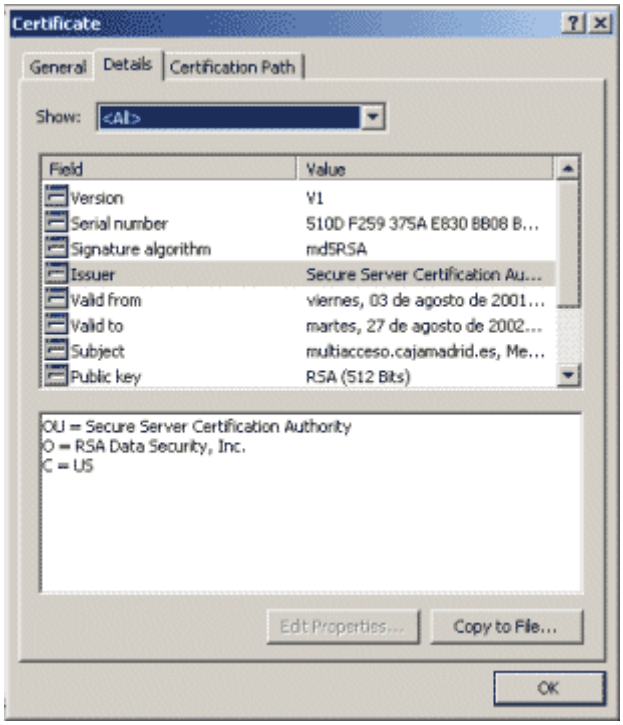

Figura 1-5 Vista Detalle de un Certificado Digital X.509.

Fuente: www.htmlweb.net/seguridad/ssl/ssl_3.html
Autor: ECCert.

Los datos que figuran generalmente en un certificado son:

- **Versión:** Versión del estándar X.509, generalmente la 3, que es la más actual.
- **Número de serie:** Número identificador del certificado, único para cada certificado expedido por una AC determinada.
- **Algoritmo de firma:** Algoritmo criptográfico usado para la firma digital.
- **Autoridad Certificadora:** Datos sobre la autoridad que expide el certificado.

- **Fechas de inicio y de fin de validez del certificado:** Definen el periodo de validez del mismo, que generalmente es de un año.
- **Propietario:** Persona o entidad vinculada al certificado. Dentro de este apartado se usan una serie de abreviaturas para establecer datos de identidad. Un ejemplo sería:

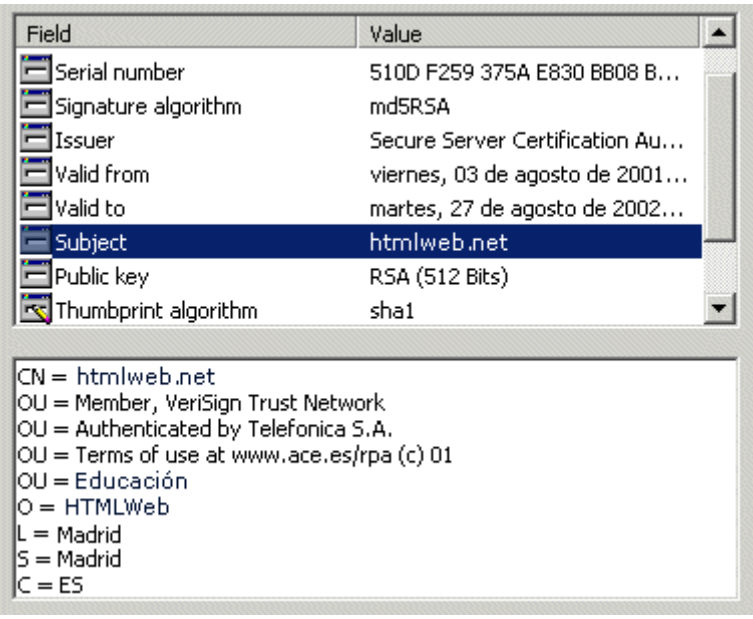

Figura 1-6 Datos que contiene el Propietario o Subject de un Certificado Digital X509 v3.

Fuente: www.htmlweb.net/seguridad/ssl/ssl_3.html
Autor: ECCert.

Abreviatura	Descripción
CN	Nombre Común del Usuario
OU	Unidad Organizacional
O	Organización
L	Ciudad o Localidad
S	Estado (provincia)
C	País
E	Correo Electrónico
UID	ID de usuario

Tabla 1.3 Datos que contiene el Propietario o Subject de un Certificado Digital X509 v3.

Fuente: ECCert

Autor: ECCert

- **Llave pública:** Representación de la llave pública vinculada a la persona o entidad (en hexadecimal), junto con el algoritmo criptográfico para el que es aplicable.
- **Algoritmo** usado para la misma para obtener la firma digital de la Autoridad Certificadora.
- **Firma de la Autoridad Certificadora**, que asegura la autenticidad del mismo.
- **Información adicional como tipo de certificado** que sirve básicamente para controlar más aun las operaciones de los certificados.

Como ya vimos, un certificado tiene Fecha de Inicio y Fecha de Fin, las cuales definen la validez del mismo. Así mismo un Certificado Digital podría ser anulado por cualquier motivo, por ejemplo, compromiso o pérdida de la clave privada. Al proceso de anulación de un Certificado Digital se lo conoce como Revocación.

Un CRL[12] es una lista de certificados que han sido revocados por la Autoridad de Certificación. El CRL puede ser comparado como una lista negra que contiene los certificados que ya no son válidos.

[12] CRL: Certificate Revocation List.

Por ultimo, mostramos el siguiente gráfico que presenta las diferencias entre las versiones 1, 2 y 3 de los formatos X.509, y el contenido un una Lista de Revocación de Certificados:

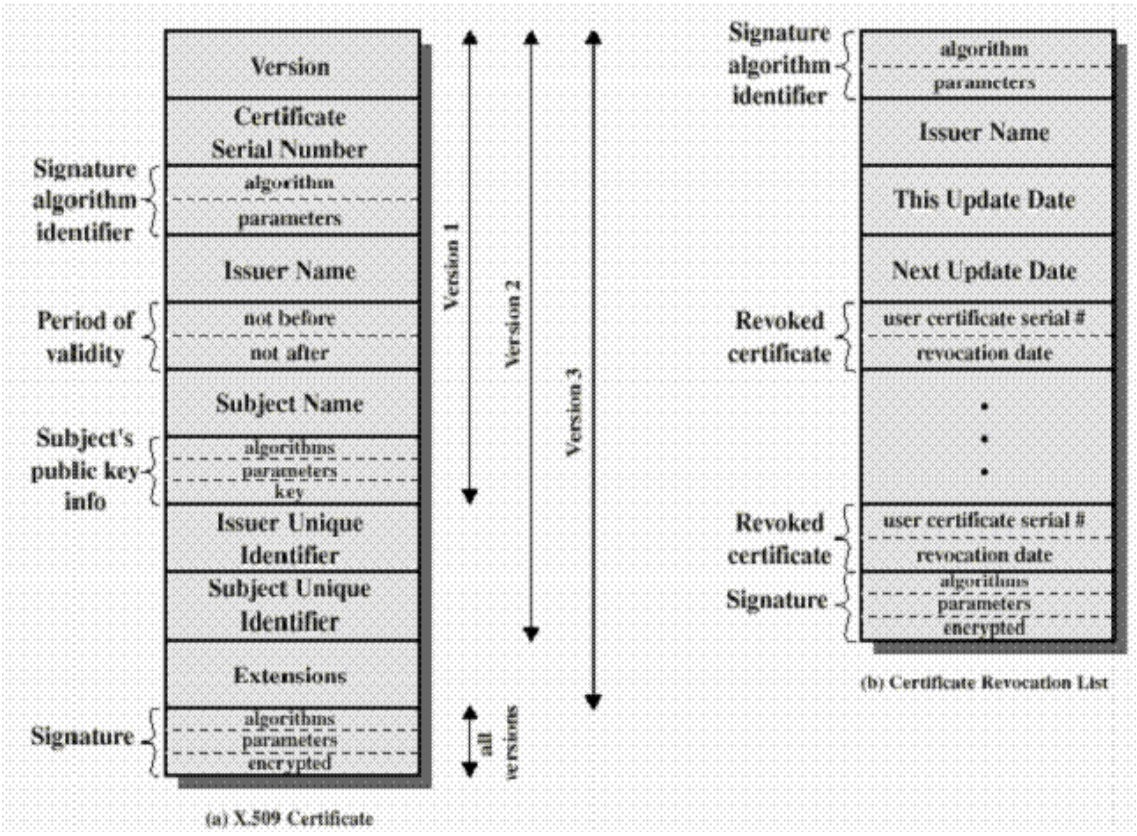

Figura 1-7 Formatos X.509 y CRL.

Fuente: www.its.bth.se/staff/hjo/
Autor: Henric Jonson.

Autoridades de Certificación y Registro: Una Autoridad de Certificación tiene como misión garantizar la validez de los certificados emitidos por ella, emitiendo y revocando los certificados de forma que en todo momento sea posible conocer el estado de un certificado.

El problema que se plantea ahora es: si la Autoridad Certificadora avala los datos del certificado ¿Quién avala a la Autoridad Certificadora?. Para solventar esto se han creado una serie de entidades autorizadas a emitir certificados, de tal forma que éstas a su vez son avaladas por otras entidades de mayor confianza., hasta llegar a la cabeza de esta organización, que se autofirma su certificado.

La organización de las CA’s se basa en una jerarquía en la cual en el nivel superior se encuentra una CA que avala, mediante sus correspondientes certificados, a las CA’s que dependen jerárquicamente de ella. Esta CA "principal" debe ser una entidad que ofrezca las suficientes garantías.

Cada certificado emitido por una AC debe estar firmado por una AC de mayor grado en el esquema jerárquico de autoridades certificadoras, formándose así una cadena de certificados, en los que unas AC se avalan a otras hasta llegar a la AC superior, que se avala a sí misma, en la cabeza de la jerarquía figuran unas pocas entidades de reconocido prestigio y confianza, como Verisign. La jerarquía de firmas y la cadena con ella formada están contempladas en el estándar X.509 v3, que indica la forma correcta de realizar estas cadenas de certificaciones.

A la hora de obtener un certificado se requiere un proceso de autentificación de los datos que en él van a figurar. En ocasiones, la verificación de datos corresponde a unas entidades creadas al efecto, que se denominan **Autoridades de Registro**.

Su labor es la actuar como avaladores ante la CA de los usuarios que solicitan el certificado, encargándose también de tramitar los mismos. Un ejemplo de entidades destinadas a asumir el papel de Autoridades de Registro pueden ser los bancos, permitiendo con ello que los certificados estén asociados a cuentas bancarias y no a personas físicas, con lo que hace posible la compra anónima por Internet, al no aparecer en ningún momento el nombre del cliente que va a efectuar el pedido.

Las Autoridades de Registro actúan en nombre y por cuenta de la Autoridad de Certificación correspondiente, y deben superar un proceso de homologación antes para garantizar su fiabilidad. Sus principales misiones son validar solicitudes de certificado en base a determinados procedimientos de identificación según el tipo de certificado, solicitar luego el correspondiente certificado a la Autoridad Certificadora y entregar el mismo, una vez obtenido, al usuario final del mismo, usando para ello un disquete u otro soporte adecuado.

Toda Autoridad de Registro debe tener a disposición de los solicitantes un documento, denominado **Prácticas de Registro**, que

especifique claramente los procedimientos operativos y de garantía de seguridad que exige y facilita.

Es importante recalcar que un esquema de certificados digitales emitidos por CA’s que garantizan la validez de las firmas digitales, debe estar respaldado por un marco legal en el cual se reconozca su entidad jurídica.

1.3 Objetivos

Los objetivos a alcanzar con este proyecto son:

- Implementar la infraestructura básica que provea las funcionalidades que debe proveer una Autoridad de Certificación.
- Implementar una esquema de red seguro para una Autoridad de Certificación.
- Implementar una aplicación segura que pueda ser usada por una Autoridad de Certificación para la generación y Administración de Certificados Digitales.
- Proporcionar esquemas de diseño de software y redes, así mismo, procedimientos utiles, que permiten incrementar la seguridad en cualquier sistema conectado o no al Internet.

1.4 Metodología

La Metodología usada para el desarrollo del proyecto es el de Paradigma Orientado a Objetos usando una base de datos relacional siguiendo primeramente el modelo de prototipos para elaborar las primeras interfaces y obtener requerimientos. Una vez obtenidas éstas se aplicó el modelo RAD[13] para el desarrollo de la aplicación.

Para la documentación se ha usado UML (Lenguaje de Modelamiento Unificado), específicamente diagramas de caso de uso, diagramas de clase, diagramas de secuencia y diagramas de despliegue. Para la documentación de la base de datos se usó diagramas entidad – relación.

1.5 Contribución

El presente proyecto pretende incrementar el interés en el tema de seguridad de la información, por medio de la implementación básica de una Autoridad de Certificación, la cual se constituye en el concepto que reúne todos los componentes menores en lo que a seguridad de información se refiere, y más específicamente, transacciones seguras.

[13] RAD: Rapid Application Development. Es una metodología de elaboración de aplicaciones que utiliza primero el prototipo en un ciclo iterativo para el desarrollo rápido de aplicaciones y de alta calidad, e incorpora el uso de componentes y generadores de código para agilitar el proceso.

El proyecto pretende crear un precedente para incrementar el conocimiento acerca de este tema dentro de la ESPOL y apoyar futuras investigaciones y nuevos desarrollos en lo que respecta a la creación, implementación y administración de una Autoridad de Certificación para Ecuador.

1.6 Soluciones que implementan seguridad mediante Certificados Digitales

Existen muchas aplicaciones de los Certificados Digitales. Además, existen muchos protocolos que implementan seguridad utilizando Certificados. A continuación mencionamos los más conocidos.

Firmas Digitales: No se refiere al término común de firmas que utilizan los Certificados Digitales, y que hemos mencionado en secciones anteriores. Pero el concepto es el mismo. Se refiere a la capacidad de utilizar PKI para firmar documentos, por ejemplo, en formatos PDF[14]. Utilizando nuestro certificado digital y mediante un aplicativo seguro, podemos firmar un documento, similar al esquema de firmas físicas con un bolígrafo, pero que permite incrementar la seguridad mediante las validaciones de la firma digital hecha por el esquema de la Autoridad de Certificación.

[14] PDF: Portable Document Format. Es un formato creado por la compañía Adobe que provee un estándar para el almacenamiento y edición de documentos publicables e imprimibles.

VPN (Virtual Private Network): Es una red privada que se extiende en base a un proceso de encapsulación y encriptación de los paquetes de datos a distintos puntos remotos mediante el uso de unas infraestructuras públicas de transporte. Visto de la manera más simple, una VPN es un túnel de encriptación entre dos redes privadas por intermedio de una red pública (por ejemplo: Internet).

El túnel en sí mismo cumple con dos funciones: separar datos sensitivos de la masa de información pública disponible en la red, y asegurar los datos de intrusos de la red. Típicamente, una VPN consiste de 2 partes: el túnel y la capa de encriptación.

La tecnología VPN está basada en la idea de “túnel” (ver siguiente figura). La formación de un túnel para redes comprende el establecimiento y mantenimiento de una conexión lógica de redes (que puede contener puntos intermedios). Sobre esta conexión, los paquetes construidos sobre en un formato de protocolo VPN específico son encapsulados dentro de alguna otra base o protocolo de portadora, y entonces transmitidos entre VPN cliente y servidor, y finalmente desencapsulado en el lado receptor.

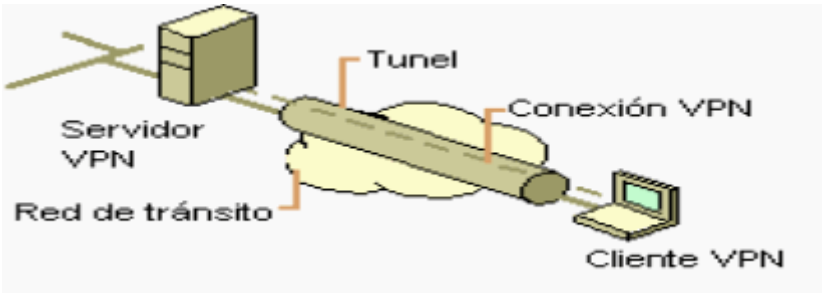

Figura 1-8 Formación de túnel VPN.

Fuente: www.monografias.com/trabajos11/repri/repri.shtml
Autor: Desconocido.

El rasgo más importante de una VPN, es su habilidad de usar redes públicas como el Internet en lugar de confiar en líneas privadas arrendadas. Las tecnologías de VPN implementan redes de tipo acceso restringido que utilizan los mismos cableados y equipos de una red pública, sin sacrificar rasgos de la seguridad básica.

Para VPN's basadas en Internet, los paquetes son encapsulados dentro de paquetes IP. Los protocolos de VPN también soportan autenticación y encriptación para mantener el túnel seguro para que de esta forma los datos sean ilegibles para los extraños. La autenticación se puede implementar mediante intercambio de claves (más sencillo y con mejor desempeño) o mediante Certificados Digitales (más confiable).

SSL (Secure Socket Layer): Como vimos al principio, toda transacción segura por la red debe contemplar los aspectos de

Autenticidad, Integridad, Confidencialidad y No Repudio. Son varios los sistemas y tecnologías que se han desarrollado para intentar implementar estos aspectos en las transacciones electrónicas, siendo sin duda SSL el más conocido y usado en la actualidad.

SSL permite la Confidencialidad y la Autentificación en las transacciones por Internet, siendo usado principalmente en aquellas transacciones en la que se intercambian datos sensibles, como números de tarjetas de crédito o contraseñas de acceso a sistemas privados. SSL es una de las formas base para la implementación de soluciones PKI (Infraestructuras de Clave Pública).

Secure Socket Layer es un sistema de protocolos de carácter general diseñado en 1994 por la empresa Nestcape Communications Corporation, y está basado en la aplicación conjunta de Criptografía Simétrica, Criptografía Asimétrica (de llave pública), certificados digitales y firmas digitales para conseguir un canal o medio seguro de comunicación a través de Internet.

De los sistemas criptográficos simétricos, motor principal de la encriptación de datos transferidos en la comunicación, se aprovecha la rapidez de operación, mientras que los sistemas asimétricos se usan para el intercambio seguro de las claves simétricas, consiguiendo con ello resolver el problema de la Confidencialidad en la transmisión de datos.

SSL implementa un protocolo de negociación para establecer una comunicación segura a nivel de socket (nombre de máquina más puerto), de forma transparente al usuario y a las aplicaciones que lo usan.

Actualmente es el estándar de comunicación segura en los navegadores web más importantes (protocolo HTTP), como Nestcape Navigator e Internet Explorer.

La identidad del servidor web seguro (y a veces también del usuario cliente) se consigue mediante el Certificado Digital correspondiente, del que se comprueba su validez antes de iniciar el intercambio de datos sensibles (Autenticación), mientras que de la seguridad de Integridad de los datos intercambiados se encarga la Firma Digital mediante funciones hash y la comprobación de resúmenes de todos los datos enviados y recibidos.

Desde el punto de vista de su implementación en los modelos de referencia OSI y TCP/IP, SSL se introduce como una especie de nivel o capa adicional, situada entre la capa de Aplicación y la capa de Transporte, sustituyendo los sockets del sistema operativo, lo que hace que sea independiente de la aplicación que lo utilice, y se implementa generalmente en el puerto 443. (Los puertos son las

interfaces que hay entre las aplicaciones y la pila de protocolos TCP/IP del sistema operativo).

Figura 1-9 Capa SSL.

Fuente: www.htmlweb.net/seguridad/ssl/ssl_5.html
Autor: Desconocido.

SSL proporciona servicios de seguridad a la pila de protocolos, encriptando los datos salientes de la capa de Aplicación antes de que estos sean segmentados en la capa de Transporte y encapsulados y enviados por las capas inferiores. Es más, también puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 2^{14} bytes, volviéndolos a reensamblarlos en el receptor.

La versión más actual de SSL es la 3.0. que usa los algoritmos simétricos de encriptación DES, TRIPLE DES, RC2, RC4 e IDEA, el asimétrico RSA, la función hash MD5 y el algoritmo de firma SHA-1.

TLS (Transport Layer Security): Para intentar corregir las deficiencias en SSL v3 se buscó un nuevo protocolo que permitiera

transacciones seguras por Internet, sobre todo teniendo en cuenta que SSL es propiedad de la empresa Nestcape. El resultado de esta búsqueda fue el protocolo TLS, que permite una compatibilidad total con SSL siendo un protocolo público, estandarizado por el IETF[15].

TLS busca integrar en un esquema tipo SSL al sistema operativo, a nivel de la capa TCP/IP, para que el efecto "túnel" que se implementó con SSL sea realmente transparente a las aplicaciones que se están ejecutando. Parte de las mismas bases que SSL, pero se diferencia de él en algunos aspectos fundamentales, entre ellos:

- Las claves de sesión se calculan de forma diferente.
- A la hora de intercambiar las claves, TLS no soporta el algoritmo simétrico Fortezza, que sí es soportado por SSL. Esto es debido a la búsqueda de un código público, ya que Fortezza es de propiedad privada.

A pesar de mejorar SSL y de ser público, TLS no está teniendo la aceptación que se esperaba (por lo menos por ahora).

Protocolo S-HTTP: El protocolo Secure HTTP fue desarrollado por Enterprise Integration Technologies, EIT, y al igual que SSL permite tanto el cifrado de documentos como la autenticación mediante firma y certificados digitales, pero se diferencia de SSL en que se

[15] IETF: Internet Engineering Task Force. Organización que define, entre otras cosas, los protocolos que operan en el Internet.

implementa a nivel de aplicación. Se puede identificar rápidamente a una página web servida con este protocolo porque la extensión de la misma pasa a ser .shtml en vez de .html como las páginas normales.

El mecanismo de conexión mediante S-HTTP, cuya última versión es la 1.1, comprende una serie de pasos parecidos a los usados en SSL, en los que cliente y servidor se intercambian una serie de datos formateados que incluyen los algoritmos criptográficos, longitudes de clave y algoritmos de compresión a usar durante la comunicación segura.

En cuanto a estos algoritmos, lo usados normalmente son RSA para intercambio de claves simétricas, MD2, MD5 o NIST-SHS como funciones hash de resumen, DES, IDEA, RC4 o CDMF como algoritmos simétricos y PEM o PKCS-7 como algoritmos de encapsulamiento.

A diferencia de SSL, el protocolo S-HTTP está integrado con HTTP, actuando a nivel de aplicación, como ya hemos dicho, negociándose los servicios de seguridad a través de cabeceras y atributos de página, por lo que los servicios S-HTTP están sólo disponibles para el protocolo HTTP. Recordemos que SSL puede ser usado por otros protocolos diferentes de HTTP, pues se integra a nivel de sockets.

SET: SSL adolece de defectos a la hora de implementar las condiciones básicas de una transacción segura. Estas carencias hicieron que diferentes empresas y organismos buscaran un nuevo sistema que permitiera realizar operaciones sensibles por Internet de forma segura, con el objeto de estimular la confianza de los consumidores en el comercio electrónico.

En febrero de 1996 un grupo de empresas del sector financiero, informático y de seguridad (Visa International, MasterCard, Microsoft, Nestcape, IBM, RSA, etc.) anunciaron el desarrollo de una nueva tecnología común destinada a proteger las compras a través de redes abiertas como Internet basadas en el uso de tarjetas de crédito. Esta nueva tecnología se conoce con el nombre de Secure Electronic Transactions (Transacciones Electrónicas Seguras), SET, y ha sido creada exclusivamente para la realización de comercio electrónico usando tarjetas de crédito.

SET se basa en el uso de certificados digitales para asegurar la perfecta identificación de todas aquellas partes que intervienen en una transacción en línea basada en el uso de tarjetas de pago, y en el uso de sistemas criptográficos de clave pública para proteger el envío de los datos sensibles en su viaje entre los diferentes servidores que participan en el proceso. Con ello se persigue mantener el carácter estrictamente confidencial de los datos, garantizar la integridad de los mismos y autenticar la legitimidad de las entidades o personas que

participan en la transacción, creando así un protocolo estándar abierto para la industria que sirva de base a la expansión del comercio electrónico por Internet.

Como características principales de SET podemos destacar:

- Es un estándar abierto y multiplataforma, en el que se especifican protocolos, formatos de mensaje, certificados, etc., sin limitación alguna respecto al lenguaje de programación, sistema operativo o tipo de máquina usados.
- Su principal objetivo es la transferencia segura de números de tarjetas de crédito.
- Utiliza codificación estándar (ASN.1 y DER).
- Es independiente del medio de comunicación utilizado. Fue diseñado para su uso en Internet, pero permite la conexión a través de cualquier tipo de red siempre que se definan los interfaces adecuados. Además, el protocolo SET se puede transportar directamente mediante TCP, mediante correo electrónico basado en SMTP o MIME y mediante HTTP en páginas web.
- Utiliza estándares criptográficos reconocidos y ampliamente usados (PKCS, Certificados X.509, etc.).
- El formato de los mensajes usados está basado en el estándar PKCS-7, al igual que SSL y S-MIME.
- Se basa en el uso de la Criptografía de Clave Pública.

- Realiza una Autenticación de todas las partes participantes en la transacción usando certificados digitales.

Vimos que en el proceso SSL sólo intervienen dos entidades: el Comprador (Cardholder) y el Vendedor (Merchant). SET incluye otras entidades adicionales necesarias para la transacción:

- La Pasarela de Pago (Gateway Payment), que permite la comunicación directa a través de Internet entre el comerciante y las Redes Bancarias, con lo que el papel del vendedor queda limitado a un mero intermediario entre el cliente y su banco. Puede ser una entidad independiente o el mismo banco del comerciante.
- El Banco o entidad financiera (Issuer) que ha emitido la tarjeta de crédito que va a usar el cliente en el proceso de pago.
- El Banco del comerciante (Adquirer), en el que éste tiene su cuenta.

Además de estas entidades principales existen otras dos relacionadas con ellas:

- La empresa propietaria de la marca de la tarjeta de crédito, como Visa, MasterCard, American Express, etc., que avalan las tarjetas.

- Autoridades de Certificación, que emiten los Certificados Digitales usados como medio de autenticación de las entidades que intervienen directamente en la operación. Pueden ser entidades independientes autorizadas, bancos o los mismos propietarios de la marca de la tarjeta.

Además existen empresas propietarias que ofrecen sus propios productos para desarrollar soluciones de seguriad para empresas basadas en Certificados Digitales y otros recursos, como un ejemplo de ello se puede citar a Novell Nsure o Verisign Inc.

1.7 Realidad en el Ecuador.

Firma Digital: Si la definición de “firma” de por sí envuelve muchas complicaciones, la de “firma digital” todavía más. Basta visualizarlo el ejemplo de los dantescos debates del Congreso de los Estados Unidos cuando quisieron dar una definición federal de firma digital; cada uno de los estados la definía de forma diversa y le daba diferentes efectos jurídicos. Sin embargo, la definición a la que se llegó parece ser la más acertada, conciliadora y precisa de todas. El E–Signatures Act la define como “*an electronic sound, symbol, or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record*”, que se puede interpretar como “un sonido, símbolo o proceso

electrónico, adjuntado o asociado lógicamente a un contrato u otro tipo de archivo y ejecutado o adoptado por una persona con la intención de firmar tal archivo".

Ecuador carecía de una definición legal de firma digital o electrónica, y bien podía recoger la transcrita porque no es restrictiva ni la limitaba a una serie de códigos electrónicos, como en otras legislaciones, y, además, da más valor a la intención del sujeto de firmar ("*intent to sign*"). Se podía abarcar dentro de la definición de firma digital el campo más amplio posible de medios con los que una parte intenta dar autenticidad a un documento, porque, en definitiva, ese es el fin natural de la firma. No obstante, no se pueden dar los mismos efectos jurídicos a todos los tipos de firmas digitales. Un ejemplo de aquello es que la legislación española distingue las firmas electrónicas avanzadas de las no avanzadas o débiles, que no hacen prueba tan plena.

La Ley De Comercio Electrónico, en el Artículo 13 define a la firma digital de la siguiente forma: "Son los datos en forma electrónica consignados en un mensaje de datos, adjuntados o lógicamente asociados al mismo, y que puedan ser utilizados para identificar al titular de la firma en relación con el mensaje de datos, e indicar que el titular de la firma aprueba y reconoce la información contenida en el mensaje de datos". En este país, el reconocimiento de la firma electrónica en la Ley antes mencionada abre sus puertas a la "Era de

la Tecnología", con temas nuevos que revolucionan el mundo cibernético.

Como catalizadores para la introducción de este tema en la legislación ecuatoriana están los diferentes proyectos que han surgido en el país y han hecho tomar conciencia sobre la importancia de implementar un esquema de seguridad. Uno de estos es el desarrollo de un proyecto de Correo Seguro, dentro del cual se encuentran dos puntos principales:

- Declaración de los Operadores de Comercio Exterior (OCE's) vía correo electrónico, encriptación de datos en tránsito.
- 2da Fase: Firma Digital - Aún se esperan las modificaciones al Reglamento de Comercio Electrónico.

A pesar de no poder aplicar la norma por falta de Reglamento, es oportuno destacar, que la ley equipara la validez de la firma manuscrita con la firma electrónica. De allí, que es firme y se lo puede presentar dentro de juicio, siempre y cuando esté basado en un certificado reconocido y haya sido producida mediante un dispositivo seguro de creación de firma.

Con relación a la impugnación del certificado o de firma electrónica, el Artículo 54 Inciso segundo de la Ley de Comercio Electrónico Ecuatoriano, dispone que el "juez o Tribunal, a petición de parte,

ordenará a la entidad de certificación de información correspondiente, remitir a ese despacho los certificados de firma electrónica y documentos en los que se basó la solicitud del firmante, debidamente certificados.

Por su parte, el efecto legal que se persigue está íntimamente vinculado con la confianza que tenga la Autoridad de Certificación.

Calvo Sotelo, autor español, afirma que la "firma electrónica tiene en relación con un documento electrónico, el mismo valor jurídico que la firma manuscrita. Por ello, es obligatorio su admisión como prueba en juicio, lo cual debe ser valorada conforme a los criterios de apreciación judicial establecidos en las normas procesales (es decir, si aquel contra quien se imputa un documento firmado electrónicamente alega error o falsedad, intervienen los peritos y, a la vista de sus dictámenes y de las alegaciones de las partes, decide".

Al respecto, el Presidente de la Corporación Ecuatoriana de Comercio Electrónico (CORPECE) Ingeniero Carlos Vera, afirma que la posibilidad de falsificación de una firma electrónica es nula " porque tiene incluida una serie de seguridades y algoritmos imposibles de descifrar", bajo esa perspectiva la firma electrónica viene a constituir un mecanismo seguro para las transacciones económicas o para realizar alguna diligencia.

Autoridades de Certificación: En Ecuador no existe una entidad calificada dentro del país como Autoridad de Certificación, pero Telconet está en la fase final para convertirse en Autoridad de Registro autorizada en el país, ya tiene el equipamiento necesario y está a la espera de la revisión del Reglamento de Comercio Electrónico y de sus respectivas modificaciones y de los trámites legales.

Conclusiones: En cuanto a lo que certificados digitales se refiere, se puede preveer un futuro promisorio sobre este tema en el país, dicho optimismo se fundamenta en los avances ocurridos, entre dichos avances se encuentran:

- Se destacan algunas iniciativas privadas que están usando firmas digitales particularmente.
- Como se mencionó anteriormente se espera que próximamente se hagan las últimas modificaciones al Reglamento de Comercio Electrónico.
- Ya existe una empresa lista para actuar como Autoridad de Registro, y se prevee que surgan más en el futuro.
- Ya están especificadas las siguientes sanciones por delitos informáticos (como reformas al código penal), aunque se puede observar que aún faltan muchos más por sancionar:

- Violentación de claves o sistemas de seguridad: 6 meses a 1 año de prisión, $500,000 multa.
- Divulgación de información protegida: 3 a 6 años de prisión, $2 a $10 (miles) de multa.
- Destrucción de información: 6 meses a 3 años de cárcel.

La sana crítica aplicada a la valoración de la prueba informática es quizá el mejor sistema en la mayoría de los casos, excepto en aquellos donde la ley exige expresamente la firma quirografaria, eso sí, siempre y cuando nuestros jueces tengan, efectivamente, una “sana crítica”. Una ley puede aclarar el valor probatorio de determinada información, obviamente si está redactada con esa “sana crítica”, pero ¿nuestros jueces que apenas manejan un ordenador tendrán esa “sana crítica” para aplicarla?

El desarrollo del comercio electrónico y la firma digital, juegan un papel determinante en la recuperación de la confianza y seguridad de los usuarios, que sienten en las comunicaciones electrónicas una apertura al mundo actual.

En el Ecuador, la "era digital", está empezando a despuntar, pero lamentablemente, no todos tienen acceso a la información. Así, apenas el 2% de la sociedad ecuatoriana utiliza Internet en forma directa, por lo que todavía está lejano el B2B, *business to business* (negocios entre empresas) B2C, *business to consumer* (negocios

entre empresas y consumidores) o el C2C (Consumer to consumer) que es el comercio minorista.

Es importante anotar, que la firma digital, es un instrumento que permite la adaptación a este nuevo paradigma socio-económico-cultural, porque posibilita la expansión del comercio dentro de esta nueva economía digital globalizada y, a su vez, en el ámbito administrativo o gubernamental, optimiza la eficiencia a un bajo costo, con intervención y participación de los ciudadanos.

Finalmente, el Internet exige que todos los países unan esfuerzos e impulsemos leyes que garanticen y estimulen al individuo estar acorde con las nuevas tecnologías, para que al comunicarse a través de Internet, lo hagan con toda confianza y así puedan realizar transacciones exitosas.

1.8 Perfil del Proyecto

Este documento se compone de 3 partes principales, la primera consiste de los capítulos 1 y 2 en los cuales se realiza una introducción sobre lo que significa Seguridad de Información y transacciones seguras sobre Internet.

La segunda parte consiste en la definición y análisis del proyecto, su diseño y finalmente la implementación realizada.

Por último la tercera parte consiste del capítulo 6 en el cual se asientan las conclusiones del proyecto y las observaciones recopiladas durante el desarrollo del mismo.

CAPITULO 2

2 DISEÑO E IMPLEMENTACIÓN DE LA SEGURIDAD FÍSICA PARA UNA AUTORIDAD DE CERTIFICACIÓN.

"Los intentos de acceso no autorizados no es lo más grave que puede ocurrir. Lo realmente grave es que alguien logre un acceso no autorizado y no sea inmediatamente detectado: le estamos dando el tiempo precioso para que se garantice nuevos accesos, conozca el sistema, lo controle y pueda cometer el fraude. NO ES GRAVE UN INTENTO, O UN ACCESO NO AUTORIZADO. ES REALMENTE GRAVE NO ENTERARNOS[16]".

[16] Fuente: CYBSEC S.A. Artículos – Mitos y Realidades en la Seguridad Informática. Fuente: http://www.cybsec.com

En este capítulo mostramos el diseño de dos arquitecturas de red, una denominada Ideal, y una red que hemos implementado en el presente proyecto.

La red Ideal contiene muchas seguridades y equipos que proporcionan redundancia de los servicios, alta disponibilidad, alta confiabilidad.

La red Implementada, proporciona una sugerencia económica para una Autoridad de Certificación, y a la vez las bases para cualquier empresa que desee presentar sus servicios en Internet.

2.1 Diseño Ideal de una Red Segura.

Nuestra Autoridad de Certificación debe mantener una alta disponibilidad debido a que el servicio que ofrece a nuestros clientes enmarca el uso de nuestro producto, los Certificados Digitales, en cualquier momento y en cualquier lugar, a través del Internet, cuando uno de nuestros clientes desea realizar transacciones electrónicas, o administrar su certificado digital.

Dado esto, la arquitectura de red debe ser convergente, rápida, en especial segura, y debe mantener una redundancia tal que permita mantener una alta disponibilidad de nuestros servicios.

Para garantizar todo esto, hemos diseñado una arquitectura de red segura. Lastimosamente por falta de recursos no se la ha podido implementar, es por esto que la hemos denominado: Arquitectura Ideal de una Red Segura para una Autoridad de Certificación. Y el esquema lo mostramos a continuación:

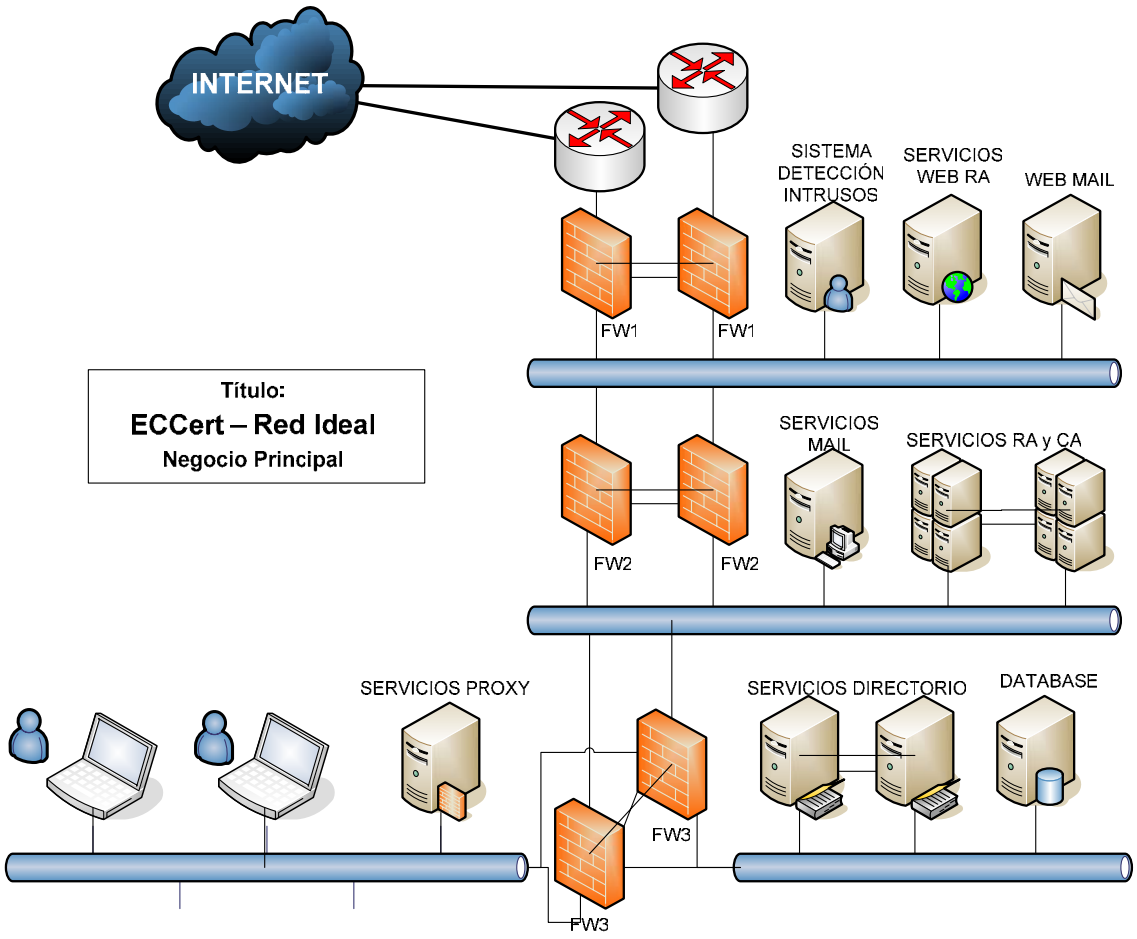

Figura 2-1 Red Ideal para una Autoridad de Certificación

Fuente: ECCert
Autor: ECCert

Como se muestra en el gráfico, se ha usado una serie de equipos y servidores para cumplir la meta de seguridad, los cuales los describimos a continuación:

Muros de fuego (FW1, FW2, FW3): Un Firewall o Muro de Fuego es una combinación de hardware y software, localizados como puerta de enlace de una red, que protege los recursos de una red privada de usuarios de otras redes. Básicamente un muro de fuego filtra todo el tráfico de la red y determina si deja pasar o no hacia el destino.

Un Muro de Fuego puede servir como punto de entrada único a su sitio, normalmente llamado punto de estrangulamiento. A medida que se reciben las peticiones de conexión, el muro de fuego las va evaluando. Sólo se procesan las peticiones de conexión de los clientes autorizados, el resto de las peticiones son descartadas.

Pero ésta es una definición demasiado limitada, los muros de fuego actuales realizan todo tipo de tareas, como por ejemplo:

- Filtro y análisis de paquetes. Los muros de fuego pueden analizar paquetes entrantes de múltiples protocolos. Basándose en ese análisis, los muros de fuego pueden realizar e evaluaciones condicionales (“Si se encuentra este tipo de paquete, haré esto”).
- Bloqueo de protocolo y contenido. Los muros de fuego le permiten proteger contenidos. Puede explotar esta capacidad para bloquear Java, Javascript, VBScript, ActiveX y otras

cosas en el muro de fuego. De hecho, incluso puede crear normas para bloquear firmas de ataques particulares.

- Autentificación y encriptación de usuario, conexión y sesión. Muchos muros de fuego utilizan varios algoritmos y sistemas de autentificación (DES, Triple DES, SSL, IPSEC, SHA, MD5, BlowFish, IDEA, etc.) para verificar la identidad de sus usuarios, comprobar la integridad de la sesión y proteger los datos en tránsito de los rastreos.

En el sentido más esotérico, en su comienzo, un muro de fuego es un concepto, más que un producto. Es la suma total de todas las normas que quiera aplicar a su red. Generalmente, proporcionará a su muro de fuego reglas que reflejen la normativa de acceso de su propia organización. Este esquema depende de los servicios que se ofrecen y mantienen en la empresa. En la sección 2.2 se muestra un ejemplo de las reglas de un muro de fuego, enfocadas a una Autoridad de Certificación.

Nuestra configuración sugiere el uso de dos firewalls, que dividen a la red en tres segmentos, cada uno más protegido que el anterior. Dado que la información más delicada para una Autoridad de Certificación son los Certificados Digitales, la Base de Datos y el Servicio de almacenamiento de Certificados se encuentran detrás de los dos Firewall. Con esto disminuimos la probabilidad de que un intruso penetre hasta la red más interna.

Sistemas de Detección de Intrusos: Los Sistemas Detectores de Intrusos (IDS[17]) analizan el cliente, la red o las redes y, de acuerdo a patrones específicos, indican intentos maliciosos o sospechosos.

Los IDS's en clientes (o basados en clientes) básicamente analizan logs, patrones en el tráfico de la red, o ambas cosas. Los IDS's que analizan redes (o basados en red) utilizan el adaptador de red en modo promiscuo lo que significa que permite escuchar y analizar todo el tráfico en tiempo real tal como este viaja a través de la red.

Por la importancia que tiene la información que viaja a través de las redes, hemos seleccionado un Sistema Detector de Intrusos basado en Red, cuyas fortalezas mostramos a continuación:

- **Costo de Administración.** Los IDS basados en Red permiten un despliegue estratégico a puntos de acceso crítico para ver el tráfico destinado a numerosos sistemas que necesitan ser protegidos. Para esto no se requiere software que sea cargado y administrado en una variedad de clientes (IDS basado en cliente). Siendo que son pocos los puntos de detección requeridos, produce un costo de administración más efectivo para nuestra empresa.
- **Análisis de Paquetes.** Los IDS basados en Red examinan la cabecera de todos los paquetes por signos de actividad

[17] IDS: Intrusion Detection System.

maliciosa y sospechosa. Esto sirve para detectar la mayoría de los ataques de Denegación de Servicio (DOS[18]), por medio del análisis de las cabeceras a lo que viajan a través de la red.

- **Borrado de Evidencias.** Los IDS basados en Red utilizan el tráfico de la red para la detección en tiempo real y un intruso no puede remover sus evidencias, lo cual es un problema ya que en los IDS basados en clientes, los atacantes conocen en donde se encuentran la mayoría de los logs de auditoria y remueven su rastro luego del ataque.
- **Detección y Respuesta en Tiempo Real.** Los IDS basados en Red detectan ataques simultáneamente cuando estos ocurren y proveen una rápida respuesta y notificación.
- **Independencia del Sistema Operativo.** Los IDS basados en Red no son dependientes del sistema operativo del cliente, como lo son los IDS basados en cliente. Esto es debido a que analizan el tráfico de la red.

Asimismo mostramos las desventajas de implementar un IDS basado en Red:

- Pueden tener dificultades procesando todos los paquetes en una red grande o con mucho tráfico y pueden fallar en reconocer ataques lanzados durante períodos de tráfico alto. Una solución a esto es utilizar un IDS basado en Red que este

[18] DOS: Denial of Service. Es un ataque que es específicamente diseñado para evitar el normal funcionamiento de un sistema, generalmente inundando de tráfico a la red.

implementado completamente en hardware, lo cual los hace mucho más rápidos.

- Los IDS basados en red no analizan la información cifrada. Este problema se incrementa cuado la organización utiliza cifrado en el propio nivel de la red (IPSec[19]) entre computadores clientes, pero se puede resolver con políticas de seguridad leves.

Los IDS basados en red no saben si el ataque tuvo o no éxito, lo único que pueden saber es que el ataque fue lanzado. Esto significa que después de que un IDS basado en red detecte un ataque, los administradores deben manualmente investigar cada computador cliente atacado para determinar si el intento de penetración tuvo o no éxito. En este caso, lo que si puede hacer un IDS en caso de detectar un ataque es negar el acceso al Intruso. Esto generalmente lo hace integrándose al muro de fuego, enviándole una señal para que este deniegue el acceso al intruso.

La Autoridad de Certificación incluye un IDS en la primera red, donde se encuentran los servicios web y mail. Hemos ubicado al detector de intrusos en este lugar debido a que es aquí donde los intrusos primero trataran de entrar.

[19] IPSec: Internet Protocol Security. Es un protocolo de la IETF que permite intercambio seguro de paquetes en la capa IP.

Servidores: En lo que corresponde a los Servicios, hay que tomar en cuenta el restringirlos a únicamente lo necesario. Es decir, por ejemplo, si es un Servidor Web, solo debe permitir el acceso al puerto para el servicio Web. En la siguiente sección (Sección 2.2) se muestran las configuraciones básicas para los servicios de una Autoridad de Certificación.

Redundancia: Para garantizar la alta disponibilidad, sugerimos un esquema de redundancia de servicios. De ser posible, debería haber redundancia en todos los equipos. Nuestra sugerencia es que esto debe hacerse para los firewalls, bases de datos de certificados, y servidor de aplicaciones, debido a que son los más criticos.

Marcas y Costos: Actualmente existen muchos proveedores de servidores y equipos de redes. Para equipos de redes, Cisco System y CheckPoint Inc. son dos buenas opciones, debido a su trayectoria en nuestro mercado. Para servidores, tenemos a HP e IBM como empresas de buen nombre. Además existe la opción de implementarlos usando Software Open Source. Pero en definitiva, nuestra sugerencia es usar equipos especializados para cada función, es decir, la mejor opción para un firewall e IDS seria uno implementado en hardware de marca Cisco o CheckPoint, debido a que son más rápidos, tienen el soporte de empresas de renombre y permiten actualizaciones. Para servidores y servios, utilizar HP o IBM.

2.2 Diseño Implementado en el Proyecto.

El diseño implementado en el proyecto es basado en una solución económica, que no por ser económica deja de ser muy segura. Nos acompañamos del Open Source[20] en la mayoría de servidores y servicios. Denotamos la utilización del Sistema Operativo Linux Red Hat 9 y Fedora Core 3 para todos los equipos y servicios.

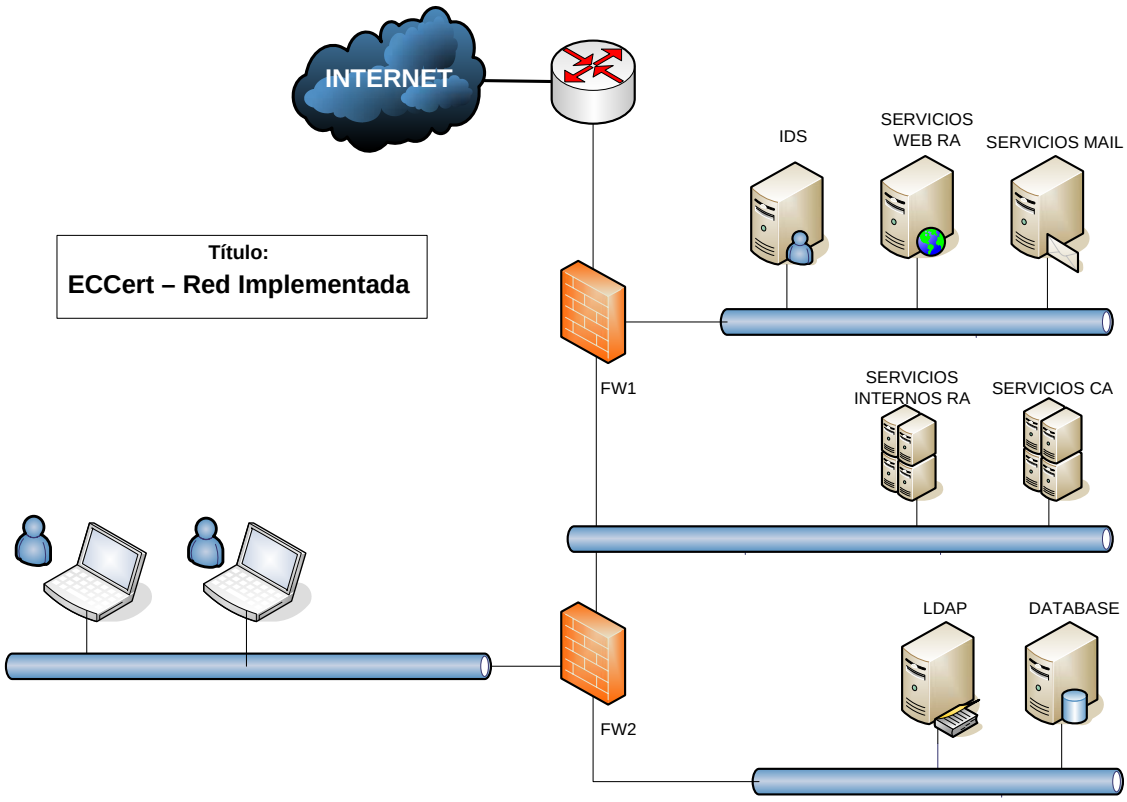

Figura 2-2 Red Implementada en el proyecto ECCert.

Fuente: ECCert
Autor: ECCert

[20] Open Souce: Software cuyo código fuente es disponible gratuitamente.

Reglas del Firewall Externo de ECCert:

	Source	Destination	Service	Action	Time	Optic	Comment
0	ADMINISTRATOR	fw1	ECHO REQUEST	Accept	Any		Permisos para que el administrador haga ping al firewall
1	ADMINISTRATOR	IDS WEBSERVER	SSH ECHO REQUEST	Accept	Any		Permiso para que el administrador acceda al web server y al ids
2	ADMINISTRATOR	IDS	HTTP	Accept	Any		Permiso para que el administrador acceda al sitio web del IDS
3	Any	fw1	Any	Deny	Any		Cualquier otro intento de conexion al firewall es denegado
4	Any	Any	AUTH	Reject	Any		
5	Any	WEBSERVER	HTTP HTTPS	Accept	Any		Despues de hacer un DNAT habilitar el forward para peticiones al WEBSERVER
6	APP USERSNET	WEBSERVER	SMTP DNS	Accept	Any		Permitir que el APPSERVER y la red de usuarios utilice el servicio de correo
7	Any	WEBSERVER	Any	Deny	Any		Denegar el resto de peticiones al WEBSERVER
8	WEBSERVER	APP	OC4J	Accept	Any		Peticiones del webserver al APPSERVER son permitidas
9	DMZNET	APPNET	Any	Deny	Any		Cualquier otro acceso de la DMZ a la red APP es denegada
10	WEBSERVER	Any	SMTP DNS DNS	Accept	Any		El mail server necesita el servicio DNS y SMTP de algun servidor de internet
11	USERSNET	Any	ECHO REQUEST HTTP	Accept	Any		Permitir todos los servicios estipulados de la red de usuarios a Internet

Figura 2-3 Reglas del Firewall Externo de ECCert.

Fuente: ECCert
Autor: ECCert

Reglas NAT del Firewall Externo de ECCert:

	Original Src	Original Dst	Original Srv	Translated Src	Translated Dst	Translated Srv	Comment
0	Any	INTERNET	HTTP HTTPS	Original	WEBSERVER	HTTPS	Todas las peticiones HTTP y HTTPS al firewall redireccionarlas al HTTPS Webserver
1	USERSNET	WEBSERVER	Any	Original	Original	Original	
2	USERSNET	Any	Any	INTERNET	Original	Original	Todas las peticiones de los usuarios a Internet redireccionarlos y cambiarles la ip fuente por la del firewall

Figura 2-4 Reglas NAT del Firewall Externo de ECCert.

Fuente: ECCert
Autor: ECCert

Reglas del Firewall Interno de ECCert

	Source	Destination	Service	Action	Time	Options	Comment
0	ADMINISTRATOR	fw2	ECHO REQUEST	Accept	Any		Permisos para que el administrador haga ping al firewall
1	ADMINISTRATOR	APP DB WEBSERVER IDS	SSH ECHO REQUEST	Accept	Any		Permiso para que el administrador acceda al appserver, a la base de datos, al web server y al ids
2	ADMINISTRATOR	IDS	HTTP	Accept	Any		Permiso para que el administrador acceda al sitio web del IDS
3	Any	fw2	Any	Deny	Any		Cualquier otro intento de conexion al fw es denegado
4	ADMINISTRATOR	fw1	SSH	Accept	Any		Permitir accesos ssh del administrador al firewall1
5	ADMINISTRATOR	fw1	ECHO REQUEST	Accept	Any		Permisos para que el administrador haga ping al firewall1
6	USERSNET	INTERNET	HTTP HTTPS	Accept	Any		Permitir pasar las peticiones http y https a la tarjeta externa del gateway
7	USERSNET	WEBSERVER	SMTP DNS	Accept	Any		Permitir que los usuarios de la red interna utilicen el servicio de correo
8	Any	WEBSERVER	Any	Deny	Any		Cualquier otra peticion al WEBSERVER es denegada
9	APP	DB	ORACLE	Accept	Any		Aceptar las peticiones a la base de datos, de parte del APPSERVER
10	APP	DB	LDAP	Accept	Any		Aceptar las peticiones al LDAPSERVER, de parte del APPSERVER
11	Any	Any	Any	Deny	Any		Denegar el resto

Figura 2-5 Reglas del Firewall Interno de ECCert.

Fuente: ECCert
Autor: ECCert

CAPITULO 3

3 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN SEGURA PARA UNA AUTORIDAD DE CERTIFICACIÓN.

"El 85% de los usuarios del Internet examinados por VeriSign Inc reportaron que una carencia de seguridad hizo que ellos se sientan inconformes de enviar su número de tarjeta de crédito a través del Internet. Los comerciantes que pueden ganar la fiabilidad de esos clientes ganarán la lealtad y la oportunidad de expandir su negocio[21]".

[21] Fuente: VeriSign Inc White Paper – Guide to Securing Your Web Site For Business. http://www.verisign.com

La intención de este capítulo es explicar el diseño de la aplicación segura para Internet, basado en el uso de patrones de seguridad y diseño, y de estándares de desarrollo de Base de Datos y Programación, conjuntamente con el uso de Nuevas Tecnologías de modelado e Implementación de sistemas y con las herramientas de Desarrollo Actuales y más robustas.

Para tener una visión sistémica del proyecto, y para facilitar el entendimiento de esta primera parte, mostramos la siguiente figura, que explica a la Autoridad de Certificación a nivel mundial con sus componentes, como son las sucursales y Autoridades de Registro:

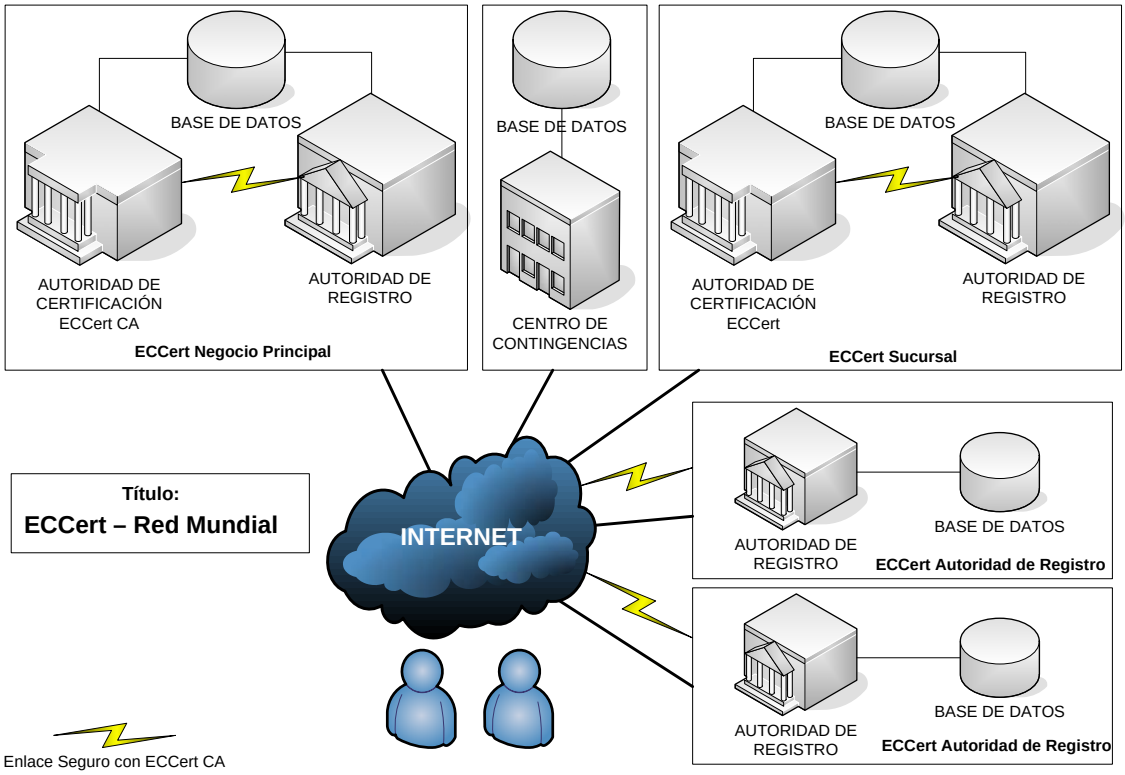

Figura 3-1 Distribución de ECCert a nivel mundial

Fuente: ECCert
Autor: ECCert

3.1 Requisitos de Seguridad

La Autoridad de Certificación debe cumplir varios requisitos de la Seguridad de la Información a nivel de la aplicación. Todos los requisitos son muy importantes y van de la mano unos con otros, produciéndose así un conjunto de Servicios de Seguridad que se aplican e integran con nuestro diseño de red segura, y que son descritos a continuación:

Tecnología de Encriptación: La encriptación[22] es la base de la Integridad[23] de datos y Confidencialidad[24] necesaria para el comercio electrónico. Los clientes envían información sensitiva y compran bienes y servicios a través del Internet solo cuando ellos confían que su información personal esta segura.

Autentificación: Antes que los clientes establezcan un canal confiable, usando encriptación, deben estar seguros que la otra parte es efectivamente quien dice ser. Este es el problema de la autentificación. Para que un cliente haga uso de los servicios críticos primero debe decir quien es y nuestra Autoridad de Certificación debe aprobar su acceso mediante la comprobación de las credenciales del cliente.

[22] Encriptación: Es el proceso de transformar información para hacerla no entendible para todos pero entendible para el receptor previsto.
[23] Integridad: Característica que hace que la información no se pueda alterar por personas no autorizadas.
[24] Confidencialidad: O Privacidad, se refiere a que la información solo sea conocida por personas autorizadas.

Disponibilidad: Es la capacidad de que la Información este siempre presente para ser procesada por las personas autorizadas. Esto requiere que la misma se mantenga correctamente almacenada con el hardware y el software funcionando perfectamente y que se respeten los formatos para su recuperación en forma satisfactoria.

Revocación: Dado a que los Certificados Digitales permiten el acceso a los Servicios de Seguridad, nace el problema de que sea comprometido alguno de sus componentes, como la clave privada, por ejemplo. La revocación es un requisito que se refiere a la retracción o anulación de certificación o autorización.

Validación: Nuestra Autoridad de Certificación provee validación de límites de tiempo de autorización para uso o manipulación de información o recursos de nuestros clientes.

Control de Acceso: El acceso a los recursos y servicios de nuestra Autoridad de Certificación es restringido sólo para las entidades autorizadas, siendo estos las Autoridades de Registro u otras sucursales de nuestra Autoridad de Certificación.

Confirmación: Se refiere al acto de realizar una confirmación de que los servicios han sido provistos, y de mostrar un error en el caso de existirlo.

3.2 Selección de la Plataforma

En la actualidad existen muchas tecnologías. Nuestra selección se basa en las Tecnologías del Futuro o Nuevas Tecnologías, tratando en lo posible de utilizar los estándares y herramientas de desarrollo más seguras y de código abierto. La intención es establecer un balance, considerando lo más posible los temas de seguridad, con las plataformas de desarrollo actuales, y metodologías que van de la mano con ellas, como son el Lenguaje de Modelamiento Unificado (UML[25]) y los Patrones de Diseño.

3.2.1 Plataforma de Programación

En esta parte abordaremos la comparación de las características principales y de las ventajas que cada tecnología web nos ofrece para el desarrollo de ciertas aplicaciones. También se presentarán los resultados obtenidos que constan en la documentación de proyectos publicados en Internet.

Debido a que el foco principal de nuestro proyecto es el desarrollo del Núcleo de una Autoridad de Certificación, hemos considerado plataformas web tradicionales. Aunque actualmente están

[25] UML: Unified Modeling Language. Lenguaje de Modelamiento Unificado, es un lenguaje gráfico que sirve para modelar, diseñar, estructurar, visualizar, especificar, construir y documentar software

emergiendo nuevas, basadas en xml y nuevas tecnologías, como por ejemplo Oracle ADF[26] UIX[27], Java Server Faces y Oracle ADF Faces, pero que su mayor desventaja es que todavía son muy nuevas y están en evolución, y todavía tomará un poco de tiempo ver implementaciones satisfactorias, mayor acogida, y por supuesto libres de fallas de seguridad.

Características principales de las tecnologías web: El objetivo de esta sección es realizar una comparación general de las tecnologías ASP, JSP y PHP que nos permiten la creación de páginas dinámicas que corren del lado del servidor. En la siguiente tabla mostramos las características principales de cada tecnología que se obtuvieron después de una investigación:

	Java Server Pages (JSP)	Active Server Pages (ASP)	PHP
Servidores Web	Apache, Netscape, Xitami y Microsoft IIS.	Microsoft IIS o Microsoft Personal Web Server.	Apache, IIS, Netscape, etc.
Plataforma	Independiente.	Dependiente de la plataforma Windows	Independiente
Componentes principales	Reuso de componentes como JavaBeans, Enterprise Java Beans y Tag Libraries.	Basado principalmente en la arquitectura COM de Win32.	Basado en el motor de interpretación creado por Zend ([28]).
Scripting	Se utiliza el lenguaje de programación Java.	Se utiliza VBScript o Jscript.	Utiliza una sintaxis similar a C++ y Java.

[26] ADF – Application Development Framework – Framework de Oracle para el desarrollo de Aplicaciones Empresariales.
[27] UIX – User Interface XML – Framework de Oracle para presentación, basado en tecnologías XML.
[28] *Zend es la empresa que desarrolló el PHP4 (principalmente el Zend optimizer).*

Seguridad	Trabaja con el modelo de seguridad Java.	Funciona con el modelo de seguridad de Windows NT.	Depende de la manera de instalar si es por modo CGI o como módulo del servidor Web.
Acceso a bases de datos	Acceso por medio de JDBC.	Acceso por medio de los objetos ADO.	Funciones incorporadas para los dife-entes DBMS que PHP soporta.
Manejo de tags personalizados	Se pueden utilizar librerías de tags.	No se pueden utilizar tags personalizados.	Soporta el uso de librerías de tags.

Tabla 3.1 Características principales de las principales tecnologías Web

Fuente: http://www.osmosislatina.com/index.htm
Autor: Desconocido

Gráficas comparativas de las tecnologías usadas: A continuación se muestran unas gráficas comparativas en lo que se refiere al tiempo de carga desde una página sencilla hasta una página con diferentes accesos a la base de datos. En la documentación obtenida de dichas pruebas se especifica que se creó un script que permitió medir el tiempo desde el inicio hasta el final de la carga de una página.

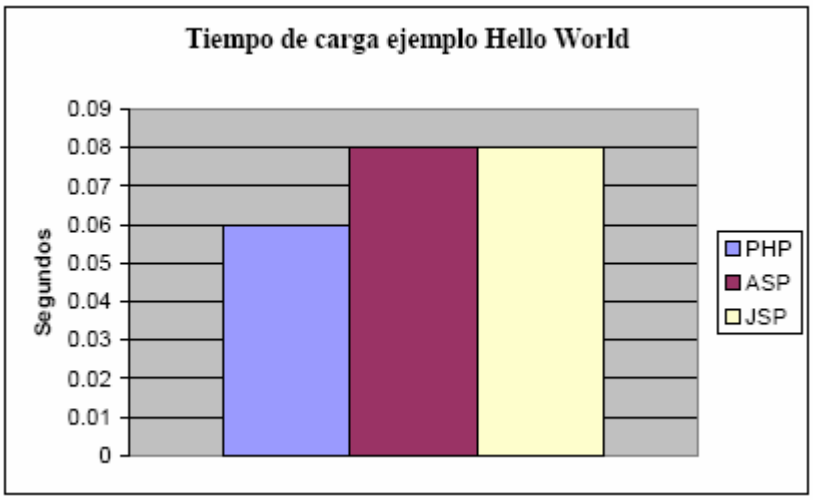

Figura 3-2 Carga de una página sencilla

Fuente: www.pue.udlap.mx/~tesis/lis/benitez_m_m/capitulo6.pdf
Autor: Miguel Benítez Méndez

Como podemos observar en la gráfica, el tiempo de carga para una página simple es en milisegundos, donde la tecnología PHP lleva una pequeña ventaja sobre las otras tecnologías.

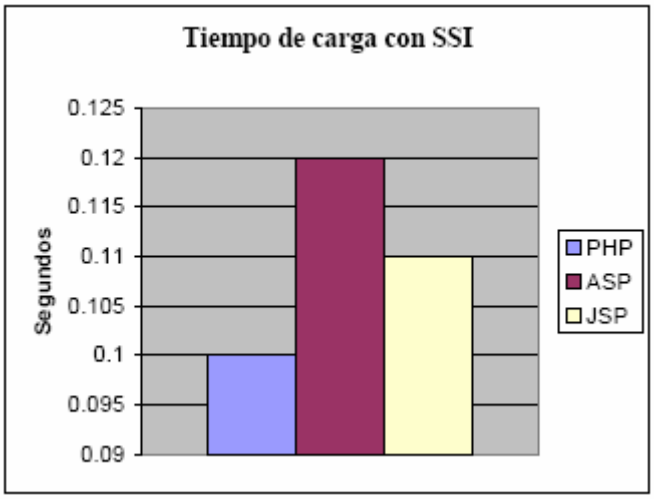

Figura 3-3 Carga de una página con SSI

Fuente: www.pue.udlap.mx/~tesis/lis/benitez_m_m/capitulo6.pdf
Autor: Miguel Benítez Méndez

Para esta gráfica se realizaron pruebas donde se incluían SSI (Server Side Includes) que son líneas de código que nos permiten incluir

archivos externos dentro de una página. La ventaja en la carga de una página sigue siendo de la tecnología PHP.

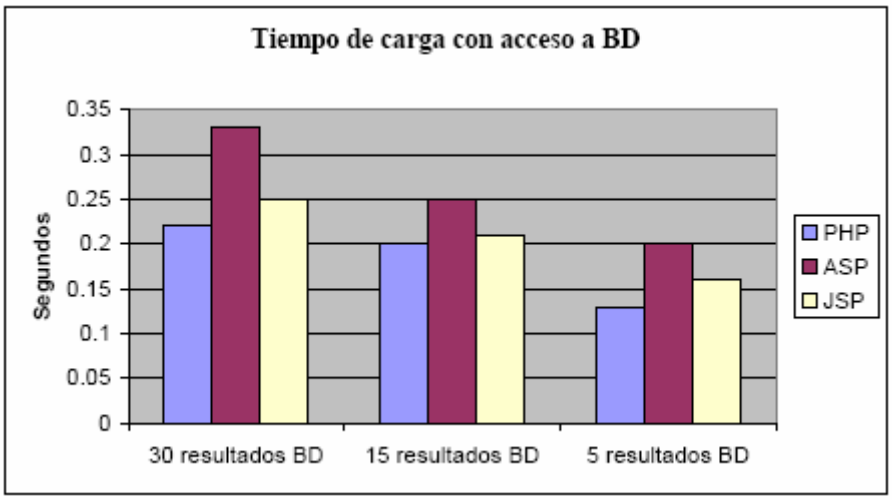

Figura 3-4 Carga de una página con diferentes resultados de una base de datos.
Fuente: www.pue.udlap.mx/~tesis/lis/benitez_m_m/capitulo6.pdf
Autor: Miguel Benítez Méndez

La última gráfica nos presenta el tiempo de carga con un diferente número de resultados obtenidos desde una base de datos. Hay que tomar en cuenta que para PHP y JSP se utilizó un DBMS diferente al de ASP por lo que los resultados son diferentes. Sin embargo la carga de todas las páginas en las 3 tecnologías es muy corta y la diferencia es en estos casos imperceptible para el usuario.

Análisis del Acceso a los Batos: En el caso de ASP el acceso a fuentes de datos se realiza por medio de un ODBC (Open Data Base Connection), que es una interfaz donde se establece el nombre de una fuente de datos para que varias aplicaciones puedan acceder a esta información. Esta conexión se realiza por medio de un driver

proporcionado por Microsoft y la mayoría de los programas que generan estas fuentes de datos son desarrollados por ellos por lo que para la realización de una aplicación que necesite extraer datos de estos tipos de archivos lo más conveniente es utilizar ASP.

Para PHP, el manejo de datos se realiza por medio de funciones que forman parte del sistema, pero el mejor desempeño para esta tecnología se obtiene al utilizar MySQL. La ventaja de utilizar las funciones nativas de PHP para el manejo de bases de datos con MySQL es que se suprime la capa de acceso a bases de datos ODBC que se encarga de la traducción de comandos SQL genéricos a la sintaxis propia del gestor (por ejemplo Microsoft Access).

En JSP el acceso a datos se realiza por medio de JDBC, que nos permite establecer mediante un driver la conexión a diferentes manejadores de bases de datos, no solo se está restringido a un tipo de aplicaciones en particular y se puede utilizar tanto en JSP, como en otras aplicaciones creadas en JAVA.

\

Ventajas de JSP: Con la información presentada hasta ahora la decisión de qué plataforma usar se inclina hacia PHP o JSP, siendo las que mejor desempeño y cualidades poseen, sin embargo JSP muestra otras características:

- **Reusabilidad entre plataformas:** Los componentes JSP son reusables en distintas plataformas (UNIX, Windows).
- **La ventaja Java:** La tecnología JSP usa Java como lenguaje de Script mientras que ASP usa VBScript o Jscript. Java es un lenguaje más potente y escalable que los lenguajes de Script. Las páginas JSP son compilados en Servlets por lo que actúan como una puerta a todos los servicios Java de Servidor y librerías Java para aplicaciones http. Java hace el trabajo del desarrollador más fácil, por ejemplo, ayuda a proteger el sistema contra las "caídas" mientras que las aplicaciones ASP sobre sistemas NT son más susceptibles a sufrirlas, también ayuda en el manejo de la memoria protegiendo contra fallos de memoria y el duro trabajo de buscar los fallos de pérdida de punteros de memoria que pueden hacer más lento el funcionamiento de una aplicación.
- **Mantenimiento:** Las aplicaciones que usan JSP tiene un mantenimiento más fácil que las que usan ASP y PHP. Los lenguajes de Script y PHP están bien para pequeñas aplicaciones, pero no encajan bien para aplicaciones grandes. Java es un lenguaje estructurado y es más fácil de construir y soporta mantenimientos grandes como aplicaciones modulares. La tecnología JSP hace mayor énfasis en los componentes que en los Scripts, esto hace que sea más fácil revisar el contenido sin que afecte a la lógica o revisar la lógica sin cambiar el contenido.

- **Integración con la arquitectura EJB**[29] que encapsula la lógica de negocio, por ejemplo: acceso a BD, seguridad, integridad transaccional y aislamiento de la aplicación.
- Debido a que la tecnología JSP es abierta y multiplataforma, los servidores web, plataformas y otros componentes pueden ser fácilmente actualizados o cambiados sin que afecte a las aplicaciones basadas en la tecnología JSP.

Selección: Basándose en las características anteriores se observa la ventaja que JSP tiene sobre ASP y PHP, por ello se concluye que es conveniente buscar una solución que involucre el manejo de JSP combinado con Servlets, Java Beans y/o EJB's.

3.2.2 Plataforma de Administración de Certificados Digitales.

A nivel de implementación de Certificados Digitales, se ha seleccionado las herramientas Open Source para la generación y almacenamiento de certificados. Las cuales son descritas a continuación:

Generación y Administración de Certificados Digitales: Se ha utilizado OpenSSL, que es un conjunto de herramientas

[29] EJB: Enterprise Java Beans. Es un conjunto de componentes Java, desarrollados por SUN, que permiten desarrollar aplicaciones distribuidas y multicapas.

Open Source multiplataforma para SSL/TSL[30]. Es basado en SSLeay, que fue desarrollada por Eric A. Young para implementar seguridad mediante SSL/TSL.

OpenSSL permite, mediante comandos, generar claves, Certificate Signing Request, Certificados Digitales, y además permite revocar certificados y administrar listas de revocación (CRL's).

Almacenamiento de Certificados Digitales: Hemos utilizado OpenLDAP, que es una implementación Open Source de LDAP.

LDAP, de sus siglas en Ingles Lightweight Directory Access Protocol, es un conjunto de protocolos que permiten acceder a información guardada en directorios, denominados generalmente Servicios de Directorio.

Para nuestra Autoridad de Certificación almacenamos los Certificados Digitales en un Servicio de Directorio. Esto permite mayor rapidez de lectura, que es la operación más común en el esquema PKI. Además, contrario a una Base de Datos, que generalmente es Relacional, LDAP tiene un

[30] SSL/TSL: Secure Socket Layer/Transport Socket Layer. Son protocolos de transporte seguros.

esquema jerárquico, lo que permite rapidez de acceso debido a que es muy similar al formato X.509 de los Certificados Digitales.

3.2.3 Patrones de Diseño

Según Christopher Alexander[31], un patrón de diseño es “Una solución a un problema en un contexto”. Cada patrón describe un problema que ocurre una y otra vez en un ambiente y entonces describe el núcleo de la solución de ese problema y a su vez el camino para usar esa solución millones de veces, sin la necesidad de recorrer dos veces el mismo camino.

Existen muchos problemas, y muchos patrones para solucionarlos. Para el presente proyecto, hemos considerados algunos de ellos, tomando en cuenta la regla 80/20, que menciona que más del 80% de los problemas se pueden solucionar con menos del 20% de las herramientas disponibles.

[31] Christopher Alexander, es un arquitecto que postuló los lineamientos de los patrones de diseño, basado en la objetividad de los sistemas arquitectónicos.

Nuestra arquitectura consiste en un modelo n-capas, las que describimos a continuación:

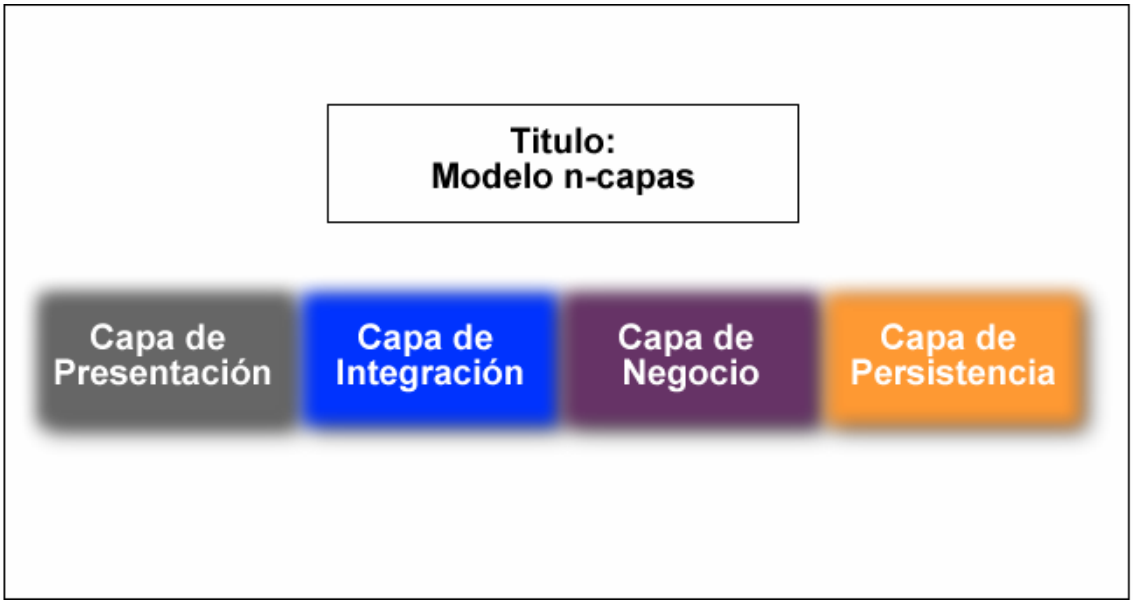

Figura 3-5 Modelo N-Capas de ECCert

Fuente: ECCert
Autor: ECCert

Capa de Presentación: Existen varios usuarios que se conectan a nuestros servicios a través del browser. Estos usuarios pueden ser Autoridades de Registro, Sucursales de nuestra Autoridad de Certificación, Clientes, Administradores y Empleados Internos con diferentes roles.

Capa de Negocio: Administra los servicios de la Autoridad de Certificación, como son la Generación y Manejo de

Certificados Digitales, envío de correo electrónico de negocio, almacenamiento y recuperación de Información.

Capa de Persistencia: Mantiene la persistencia de datos de clientes, Certificados Digitales y todo lo relacionado a los Servicios de la Autoridad de Certificación.

Capa de Integración: Es un conjunto de Clases y Servicios Web que permiten que la Capa de Presentación interactúe con la Capa de Negocio.

Solo para ser descritos, hemos dividido los patrones de diseño ubicándolos en relación a la capa en la que se encuentran, y específicamente en la Capa de Presentación, de Integración y la Capa de Negocio. Esto no quiere decir que no se use un patrón en dos o más capas, pero se lo describe en la capa donde tiene mayor presencia.

A nivel de la capa de presentación hemos considerado los siguientes patrones:

Patrón Cadena de Responsabilidad: “Permite acoplar al que envía el requerimiento con el que lo recibe, dándole a más de un objeto la responsabilidad de manejar el requerimiento.

Encadena los objetos que recibe y pasa el requerimiento a través de la cadena hasta que un objeto lo maneje".

El objetivo de este patrón es mantener un control de responsabilidad delegado, permitiendo que las clases estén organizadas en relación a la función que cumplen.

La Autoridad de Certificación utiliza este patrón para:

- Mantener un control sobre los requerimientos de los usuarios, centralizando las peticiones en un controlador, y delegando las responsabilidades a clases de soporte y control.

Patrón Singleton: "Asegurarse de que una clase tenga una sola Instancia, y proveer un punto de acceso global a ella".

Singleton tiene como principal objetivo el optimizar la memoria, reduciendo a una el número de instancias de una clase que no requiere tener más de una instancia.

Nuestra Autoridad de Certificación utiliza el patrón de diseño SIngleton para:

- Optimizar accesos a archivos de recursos y configuraciones, creando una sola instancia si no existe, y llamándola cuando se la necesite durante el tiempo de vida de la instancia del Servidor de Aplicaciones.

Patrón Front Controller: "Proveer un controlador centralizado para gestionar las peticiones de la aplicación".

El objetivo de este patrón es mantener un controlador frontal que reciba todas las peticiones entrantes de los clientes, remitiendo a su vez cada petición al gestor de peticiones, manteniendo la lógica de control centralizada.

La Aplicación para la Autoridad de Certificación utiliza este patrón de la siguiente manera:

- Captura centralizadamente los requerimientos de los usuarios, llevando el control en un archivo XML[32].
- Controlar la seguridad por roles, de acuerdo a niveles de acceso, descrito en el archivo XML.

Patrón View Helper: "Encapsula los trozos de lógica correspondiente a la presentación y acceso a datos y

[32] XML: eXtensible Markup Language. Es hoy por hoy el formato estándar de intercambio de datos.

componentes que necesita una vista[33], haciendo que la vista sea mucho mas simple, reutilizable y mantenible".

El objetivo de este patrón es simplificar la vista, colocando en ella sólo los códigos que le corresponde a presentación, preocupándose por el formateo y accesos a los datos.

En nuestra Autoridad de Certificación, este patrón es usado de la siguiente manera:

- Existen varios View Helpers que acceden a la lógica del negocio, y que son accedidos desde las páginas solo para obtener los resultados formateados, con el objetivo de solo imprimirlos.
- Los View Helpers interactuan entre si para reutilizar métodos y optimizar accesos.

A nivel de la capa de negocio, hemos considerado los siguientes patrones:

Patrón Façade: "Provee una Interfaz unificada para un conjunto de Interfaces en un subsistema. Define la Interface de más alto nivel que hace al subsistema fácil de usar"[34].

[33] Vista: Es la parte del sistema que el usuario ve. En nuestro caso, son las páginas que presenta el Browser.

El patrón de diseño Façade es llamado así debido a que provee una nueva frontera (una fachada) en frente del sistema original.

Nuestra Autoridad de Certificación utiliza este patrón de diseño para cumplir los siguientes objetivos:

- Simplificar el uso de los servicios de Autoridad de Certificación para las autoridades de registro, o para los clientes autorizados, mediante un lenguaje de representación de datos, XML[35], y clases para procesar mensajes.
- Encapsular y/o esconder todas las capacidades y servicios de la Autoridad de Certificación.
- Mantenimiento futuro de las funcionalidades de la Autoridad de Certificación, añadiendo o quitando servicios y publicándolos para que sean usados.

Patrón Business Delegate: "Encapsula el acceso a un servicio del negocio. Permite esconder detalles de la

[34] Fuente: Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Reading, Mass.: Addison-Wesley, 1995, p. 185.
[35] XML: eXtensible Markup Language. Es hoy por hoy el formato estándar de intercambio de datos.

implementación de los servicios del negocio, como búsquedas y mecanismos de acceso"[36]

Business Delegate permite delegar la lógica de solicitud, conexión y petición hacia el núcleo del negocio.

La Autoridad de Certificación implementa este patrón de diseño de la siguiente manera:

- Las llamadas hacia los Servicios de la Autoridad de Certificación, que son publicados mediante un Servicio Web, se realizan delegando los requerimientos hacia un conjunto de clases encargadas de armar, enviar e interpretar archivos XML.
- Las clases encargadas de la comunicación, crean una lógica de transacciones, luego son enviadas al servicio web de la Autoridad de Certificación, la cual interpreta y procesa las transacciones, y a su vez envía la respuesta. Business Delegate interpreta la respuesta y devuelve un XML respuesta.
- Con esto se evita las llamadas embebidas en los clientes, ocultando de esta manera la lógica del negocio.

[36] Fuente: Core J2EE Patterns. Deekpak Alur, Joht Crupi, Dan Malks. Pag. 303

Patrón Service Locator: "Permite implementar y encapsular servicios y búsqueda de componentes. Esconde detalles de implementación de los mecanismos de búsqueda y encapsula las dependencias relacionadas"[37]

La Autoridad de Certificación usa este patrón para:

- Ocultar o encapsular la búsqueda de servicios del API, y provee una interface simple a los clientes.

Patrón Composite Entity: "Permite implementar persistencia de los Objetos del Negocio mediante Entity Beans. Agrega un conjunto de Objetos de Negocio relacionados en una implementación de Entity Beans agrupados lógicamente".

Nuestra Autoridad de Certificación utiliza este patrón para:

- Evitar el mapeo del Modelo de Objetos directamente con el Modelo de Entidades.
- Evitar el mapeo del Modelo de Relaciones directamente con el Modelo de Entidades.

[37] Fuente: Core J2EE Patterns. Deekpak Alur, Joht Crupi, Dan Malks. Pag. 316

A nivel de la capa de integración, hemos considerado el siguiente patrón:

Patrón Web Service Broker: "Permite exponer y acceder a uno o más servicios usando XML y protocolos web"[38]

Web Service Broker es un servicio ligero expuesto como un Web Service(Servicio Web). Permite coordinar interacciones entre uno o más servicios, respuestas y puede demarcar y compensar transacciones.

Puede implementarse mediante SOAP o XML-RPC, o con cualquier tecnología de llamadas a Servicios mediante protocolos web.

Nuestra Autoridad de Certificación utiliza este patrón para:

- Exponer los servicios del negocio a las Autoridades de Certificación o Registro afiliadas.

Convergencia de los Patrones de Diseño: Los patrones de diseño se complementan entre si, evolucionando en lo que llamamos el Framework o Marco de Trabajo de la Aplicación,

[38] Fuente: Core J2EE Patterns. Deekpak Alur, Joht Crupi, Dan Malks. Pag. 558

produciéndonos el Modelo o Patrón de Desarrollo General, denominado MVC.

MVC, o Modelo Vista Controlador, consiste en la separación de las capas de la aplicación en tres entidades: El Modelo, la Vista y el Controlador.

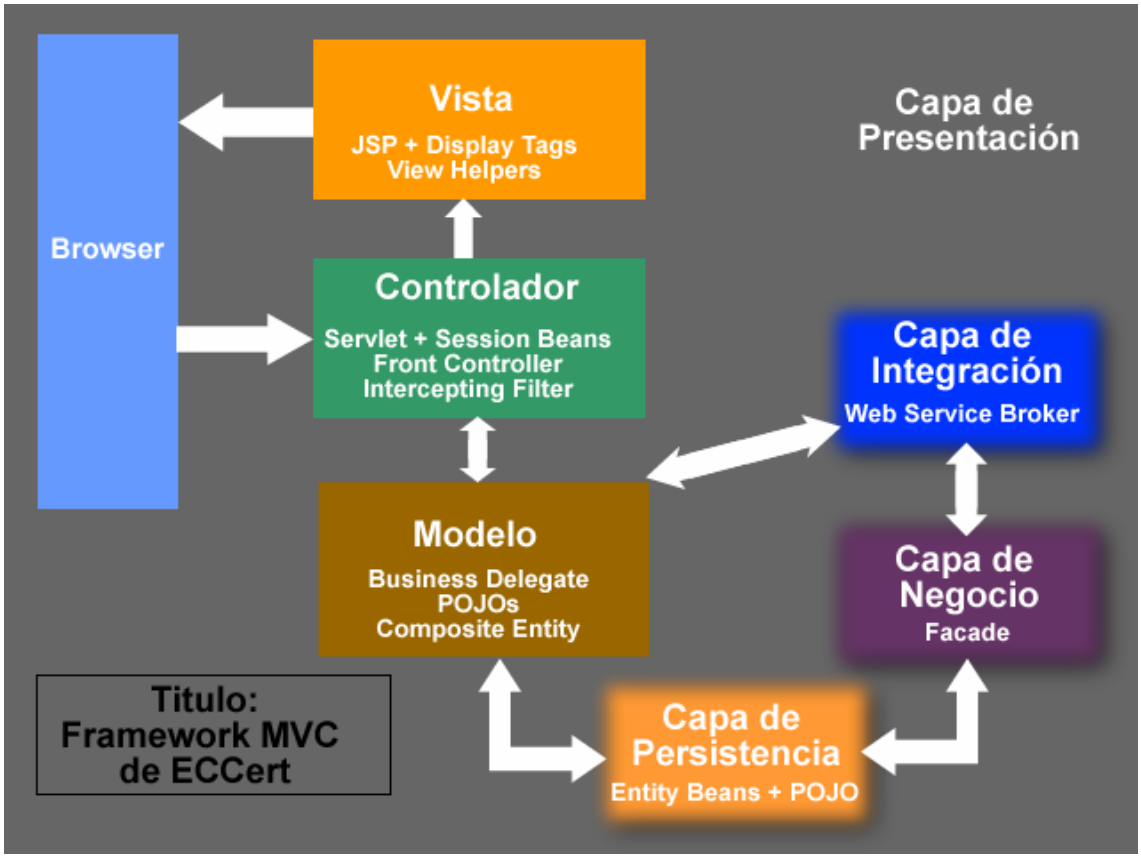

Figura 3-6 Framework MVC utilizado en ECCert

Fuente: ECCert
Autor: ECCert

Modelo: Se refiere a la parte del proyecto que se encarga de administrar la persistencia y los accesos a la capa del negocio. El modelo contiene el soporte para acceso a la Base

de Datos, y contiene las clases POJO[39]'s necesarias para interactuar con los Servicios de la Autoridad de Certificación, como son la creación, Validaciones de Certificados y por supuesto, el almacenamiento de los Certificados Digitales. Además el modelo contiene la lógica de formateo de datos y validaciones necesarias para presentación de Información.

Vista: Se refiere netamente a las páginas de Internet que el usuario visualiza. Contiene código de presentación de datos, y en nuestro caso, contiene la lógica de validaciones de Ingreso de Datos hecha por los Usuarios.

Controlador: Contiene un mapeo de requerimientos solicitados por los usuarios, centralizando el administrador de eventos, y disparando procesos a realizarse por el modelo, y devolviendo las respuestas a la Vista. Además se preocupa de controlar el acceso mediante roles de usuarios.

Trabajar con el Framework MVC nos proporciona las siguientes ventajas:

- Separar la capa de presentación de las otras capas.
- Provee un punto central de control.
- Facilita la implementación de Patrones de diseño.

[39] POJO: Plain Old Java Object. Se refiere a clases Java.

- Facilita las pruebas modulares y el mantenimiento.
- Provee estabilidad.
- Existe soporte de la comunidad desarrolladora.
- Facilita la validación de ingresos.
- Facilita la internacionalización.

La Capa de Persistencia cumple los estándares internacionales de almacenamiento de datos, y es analizada en la siguiente sección (Sección 3.2.4).

3.2.4 Base de Datos

El objetivo de esta sección es realizar una comparación general de las diferentes bases de datos que nos permiten la creación de páginas dinámicas que corren del lado del servidor. Para ello se proveen a continuación breves descripciones de las bases de datos más conocidas.

ORACLE: Es el mayor y más usado Sistema Manejador de Base de Dato Relacional (RDBMS) en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios.

Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código.

Junto con SQL Server, lidera el mercado NT. Puede funcionar en una gran cantidad de sistemas operativos y diversidad de Hardware, prácticamente en todas las familias de UNIX, MVS, VM, Siemens ICL, y Novell Netware.

Además el funcionamiento está optimizado para ajustarse a las peculiaridades de cada sistema operativo.

La idea de Oracle es de potenciar los grupos de trabajo distribuido y se necesita un acceso a los datos de dicho grupo. En este sentido se posibilitan consulta y actualizaciones distribuidas permitiendo compartir datos a través de múltiples servidores, con una consistencia en un *commit* de dos fases.

Otro cometido de Oracle es detectar problemas que puedan ir surgiendo e informar de los mismos.

En lo referente a Internet las aplicaciones web pueden acceder a los datos almacenados en la base de datos de Oracle así como presentar documentos HTML generados dinámicamente a partir del modelo de una consulta.

Oracle soporta paralelismo dentro de una consulta lo que proporciona un desempeño notable en su ejecución.

También soporta procesos de transacciones On-Line y DataWareHousing.

Atendiendo a las características de manejabilidad escalabilidad rendimiento y soporte entre plataformas tenemos una arquitectura de servidor con ejecución multihilo y rendimiento de multiprocesadores simétricos (SMP).

Las bases de datos Oracle pueden crecer hasta límites que en la práctica pueden considerarse inalcanzables y gracias a un rediseño respecto a las versiones se han eliminado ciertos cuellos de botella.

Las características más relevantes en lo referente a la escalabilidad residen en las tablas particionadas.

Los Protocolos de Red soportados por Oracle son los siguientes: Net 8, TCP/IP, IPX/SPX, Pipes con nombre, DECNET, DCE, NDS y LU 6.2 (APPC).

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos.

Es un producto vendido a escala mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general.

En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc.

Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Como mejoras de SQL Server en ORACLE se pueden mencionar:

- Oracle posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Estos porque

más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.

- Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.
- Oracle provee salvar con seguridad de error lo visto en el monitor y la información de acceso y uso.
- Oracle soporta un verdadero ambiente cliente servidor. Este establece un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.

SQL SERVER: Las características de este producto son:

- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server.
- Gestión: Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.

- Diseñada desde su inicio para trabajar en entornos Internet e Intranet: Microsoft SQL Server es capaz de integrar los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones "tradicionales". Es más, cada aplicación que desarrollemos para ser empleada en entornos de red local puede ser utilizada de forma transparente -en parte o en su totalidad- desde entornos Internet, Intranet o Extranet.
- Plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transacción Server y con las mejores herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.
- La Base de Soluciones Integradas: La Integración total con BackOffice permite resolver todas las necesidades de infraestructura de la empresa con un sólo paquete.
- Potente y Escalable: Microsoft SQL Server es la única base de datos cuyo rendimiento sobre Internet está publicado, ofreciendo registros espectaculares.

- Mínimo coste de Propiedad: La sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor coste.

Otras características de esta base de datos son:

- Datos distribuidos y replicación.
- Data Warehousing y amplio soporte de datos.
- Integración Internet y correo electrónico.
- Gestión y administración centralizada de bases de datos.
- Disponibilidad, fiabilidad y tolerancia a fallos.
- Mejoras en programabilidad y lenguaje.

MySQL: Su principal objetivo de diseño fue la velocidad. Se sacrificaron algunas características esenciales de sistemas más "serios" con este fin.

Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria.

Como ventajas de esta base de datos se encuentran:

- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.

- Mejores utilidades de administración (backup, recuperación de errores, etc).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".

MySQL tiene los siguientes inconvenientes:

- No soporta transacciones, "roll-backs" ni subselects.
- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

PostgreSQL: Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

Ventajas:

- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.

- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

Inconvenientes:

- Consume mucho más recursos y carga más el sistema.
- Límite del tamaño de cada fila de las tablas a 8k (hasta la versión 7.1).
- Es de 2 a 3 veces más lenta que MySQL.
- Tiene menos funciones en PHP.

Selección de la Base de Datos Relacional: Dado a lo crítico que es mantener la información de nuestros clientes de la forma más segura posible, seleccionamos Oracle Data Base 10G.

Oracle Data Base es actualmente la Base de Datos más segura del mercado.

3.2.5 Herramientas de Desarrollo

Actualmente existen un sinnúmero de Herramientas e IDE's[40]. Y día a día aparecen muchas más. Como hemos visto, la base en la que esta implementada nuestra Autoridad de Certificación es Java, XML y Oracle. Además una característica importante a recatar es que es Multiplataforma, y que ha sido desarrollada en Linux y Windows. El núcleo del desarrollo ha sido bajo Linux, utilizando los servicios de Código Abierto para la generación y Almacenamiento de Certificados Digitales, como son OpenSSL y OpenLDAP respectivamente, y descritos en las secciones anteriores.

Hemos resumido las herramientas utilizadas para el desarrollo del presente proyecto en la siguiente tabla, considerando solo una descripción de las funcionalidades usadas en el proyecto, más no todas las funcionalidades de las herramientas:

Herramienta	Descripción de Uso	Empresa	Plataforma
JDeveloper 10G	IDE para construir aplicaciones y servicios Web en Java, XML y SQL	Oracle	Linux, Windows
Eclipse	IDE para construir aplicaciones en Java	Eclipse Fundation	Linux, Windows
SQL Plus	Herramienta para interactuar	Oracle	Linux,

[40] IDE: Integrated Development Environment. Ambiente de Desarrollo Integrado. Es una aplicación que contiene un conjunto de herramientas y facilidades para ayudar a los desarrolladores a diseñar e Implementar software.

	con la Base de Datos		Windows
PL SQL Developer	IDE para interactuar con la Base de Datos	Allround Automations	Windows
Tora	IDE para interactuar con la Base de Datos	Quest Software	Linux
Enterprise Manager	Herramienta para administrar la Base de Datos	Oracle	Linux, Windows
Studio MX	IDE para desarrollo Web	Macromedia	Windows

Tabla 3.2 Herramientas de Desarrollo utilizadas en ECCert

Fuente: ECCert
Autor: ECCert

3.2.6 Servidor de Aplicaciones

La decisión en este tema se encuentra algo limitada debido a que, de los líderes del mercado actuales en este ámbito (BEA, Oracle, IBM WebSphere, Sybase Enterprise Application Server y Netscape Application Server) es Oracle con la que hemos tenido más experiencia de desarrollo y la que provee mayor seguridad en el mercado. Además, un punto a favor del Servidor de Aplicaciones de Oracle es que se puede utilizar el componente para aplicaciones en Java que es el OC4J[41] (Contenedor de Oracle para Java), el cuál es de adquisición

[41] El OC4J es el componente núcleo de ejecución J2EE del Servidor de Aplicaciones Oracle.

más sencilla y es el que hemos usado para el desarrollo del presente proyecto. También se tomó en consideración para la selección de este servidor de aplicaciones el dominio de Oracle en bases de datos empresariales y el mercado de servidores de aplicaciones, lo cual le provee de una reputación muy importante a ser considerada.

La flexibilidad de Oracle se extiende también al Servidor de Aplicaciones Oracle dentro del soporte de plataformas las cuales incluyen Windows NT y virtualmente cualquier otra variante de Unix, incluido Linux.

Un amplio rango de lenguajes de aplicación es soportado en el nivel de capa media por el Servidor de Aplicaciones Oracle, empezando con el obligatorio Java e incluyendo Perl, C, COBOL, y PL/SQL. Los objetos CORBA son soportados también en este nivel, gracias a una herramienta de desarrollo: JCORBA. (En el futuro, Oracle también soportará así mismo objetos DCOM.) En el mismo nivel, la conectividad ODBC es soportada, lo cual permite a cualquier componente acceder una base de datos ODBC y entonces retornar los resultados formateados. Se pueden crear APIs para comunicarse entre estos componentes, inclusive sobre diferentes servidores en un cluster. Enterprise Java Beans (EJB) son soportados, no obstante sobre una base limitada –

debido a que EJB es todavía un estándar emergente, la línea oficial es que Oracle probablemente estará cambiando y posiblemente expandiendo el soporte EJB en el futuro.

En el lado transaccional, el Servidor de Aplicaciones Oracle soporta los mayores estándares de transacción, incluyendo TX, XA, y CORBA OTS. El Protocolo Inter-ORB puede ser usado para acceso, así como el estándar HTTP. En términos de seguridad, el Servidor de Aplicaciones Oracle soporta certificados X.509 y Secure Socket Layers (SSL) 3.0, así como también restricción por dominio e IP. Si se está usando Netscape Directory Server, el Servidor de Aplicaciones Oracle soporta LDAP (Lightweight Directory Access Protocol).

El servidor de aplicaciones Oracle es un servidor de aplicaciones de fuerza industrial excelente para el servicio de aplicaciones y manejo de transacciones. Está bien recomendado para cualquier empresa, el Servidor de Aplicaciones Oracle es una gran herramienta para cualquier compañía (su nicho de mercado) que tenga necesidad de un servicio de aplicaciones seguro y confiable.

El servidor de aplicaciones de Oracle es un servidor de aplicaciones que provee toda la funcionalidad que se necesita para negocios por Internet, además provee de un ambiente

J2EE de alto desempeño debido a su menor uso de memoria, que además lo hace altamente escalable, confiable y fácil de usar.

El servidor de aplicaciones de Oracle está optimizado para las bases de datos Oracle, por lo que hace las aplicaciones más confiables y fáciles de manejar. Por lo tanto se puede hacer uso de los conocimientos en Oracle para tomar ventaja de las capacidades únicas en este sentido de este servidor de aplicaciones.

A continuación se mencionan algunas características y detalles acerca del Servidor de Aplicaciones Oracle.

Características	Beneficios
Servidor de aplicaciones completo, integrado y cumple con los estándares	Reduce complejidad, simplifica el middleware y previene limitaciones de vendedor
Servidor J2EE ligero y de alto desempeño	Construye y despliega rápidamente aplicaciones J2EE.
Arquitectura abierta lista para integrarse	Permite conectarse a fuentes dispersas de información y optimizar procesos de negocio.
Seguridad integrada y aprovisionamiento de usuario	Asegura la más alta seguridad, aprovisionamiento de usuario manejada centralizadamente e interoperación con

	sistemas no-Oracle.
Reportes y consultas ad-hoc	Analizar datos, desarrollar reportes sobre métricas clave, resultando en mejor toma de decisiones.
Construcción de software portal	Acceder información y aplicaciones con un portal empresarial.
Información a cualquier hora, desde cualquier parte	Incrementar la flexibilidad de la fuerza laboral, proveer de conectividad instantánea a las aplicaciones desde un escritorio, laptop o dispositivo móvil.

Tabla 3.3 Características y beneficios del Servidor de Aplicaciones Oracle.

Fuente: ECCert
Autor: ECCert

El Servidor de Aplicaciones Oracle proporciona soporte completo para J2EE, portales de empresa, puesta en caché de alta velocidad, inteligencia del negocio, desarrollo rápido de aplicaciones, integración de la aplicación y el negocio, capacidades wireless, servicios Web, entre otras, todas pre-integradas en un solo producto. Otros proveedores proporcionan estas funcionalidades en aplicaciones separadas, como se demuestra a continuación:

Características	Oracle Application Server 10*g*	IBM	BEA	Microsoft
Servidor Java	Oracle Application	WebSphere Application	WebLogic Server	No

	Server 10g	Server		
Caché Web	Incluído	WS Edge Server	No	No
Integración de la Aplicación	Incluído	WS CrossWorlds	WL Integration	Biz Talk Server
Integración B2B	Incluído	WS Business Integrator	WL Integration	Biz Talk Server
Portal Empresarial	Incluído	WS Portal Server	WL Portal	SharePoint Portal Server
Personalización	Incluído	WS Personalization	WL Personalization Server	Commerce Server
Conexión Inalámbrica	Incluído	WS Everyplace Server	No	Mobile Information Server
Consultas & Reportes Adhoc	Incluído	No	No	No
Directorio	Incluído	Tivoli SecureWay	No	Active Directory

Tabla 3.4 Comprobación de las soluciones integradas del Servidor de Aplicaciones Oracle.

Fuente: ECCert
Autor: ECCert

Por las razones anteriormente expuestas se eligió el Servidor de Aplicaciones de Oracle.

3.3 Diseño de la Aplicación

El modelado de una aplicación es una parte imprescindible al momento de diseñar una aplicación. También es importante el lenguaje y las herramientas con el que se lo realiza. Hemos seleccionado el Lenguaje de Modelamiento Unificado UML para esta

parte, siendo este uno de los lenguajes más utilizados actualmente al momento del diseño de aplicaciones informáticas.

Nuestra herramienta de modelado es “Smart Development Environment for JDeveloper”, de la compañía Visual Paradigm[42]. Esta herramienta se integra en el IDE, y facilita la sincronización entre el modelado y la implementación de las aplicaciones.

Estereotipos UML: Para apoyar los diagramas UML hemos usado estereotipos para indicar diferentes tipos de objetos y roles en los diagramas UML. La siguiente tabla muestra una lista de los estereotipos utilizados y su significado:

Estereotipo	Significado
EJB	Representa un componente Enterprise JavaBean
SessionEJB	Representa un Session Bean como un todo, sin especificar su interface remota, home o implementación.
EntityEJB	Representa un Entity Bean como un todo, sin especificar su interface remota, home o implementación.
View	Representa una Vista (que es la representación de información que el cliente ve).
JSP	Representa un JavaServer Page.
Servlet	Representa un Java Servlet.
Singleton	Representa una clase que implementa el patrón de diseño

[42] Visual Paradigm: Fuente: http://www.visual-paradigm.com

	Singleton.
Facade	Representa una clase que implementa el patrón de diseño Façade.
Subsistema	Se refiere a un conjunto de clases y recursos que realizan una funcionalidad.
Include	Es una relación de casos de uso en la cual el caso de uso padre utiliza a los casos de uso hijos.
extend	Es una relación de casos de uso en la cual un caso de uso deriva de otro.

Tabla 3.5 Estereotipos UML utilizados para modelar el proyecto.

Fuente: ECCert
Autor: ECCert

Procesos de Adquisición de Certificados: El siguiente gráfico muestra el proceso de adquisición de Certificados Digitales de prueba, cuya duración es de 1 mes:

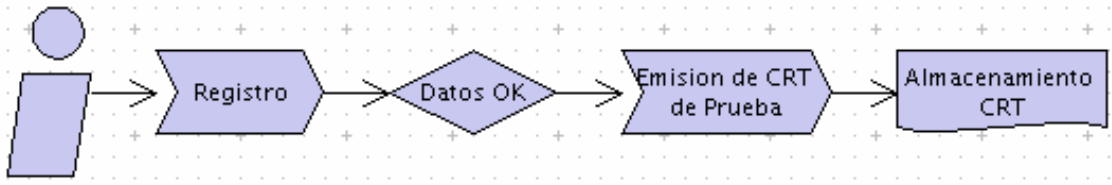

Figura 3-7 Proceso de adquisición de Certificados de Prueba.

Fuente: ECCert
Autor: ECCert

Así mismo, el siguiente gráfico presenta el proceso de compra de un Certificado Digital con duración de uno o dos años:

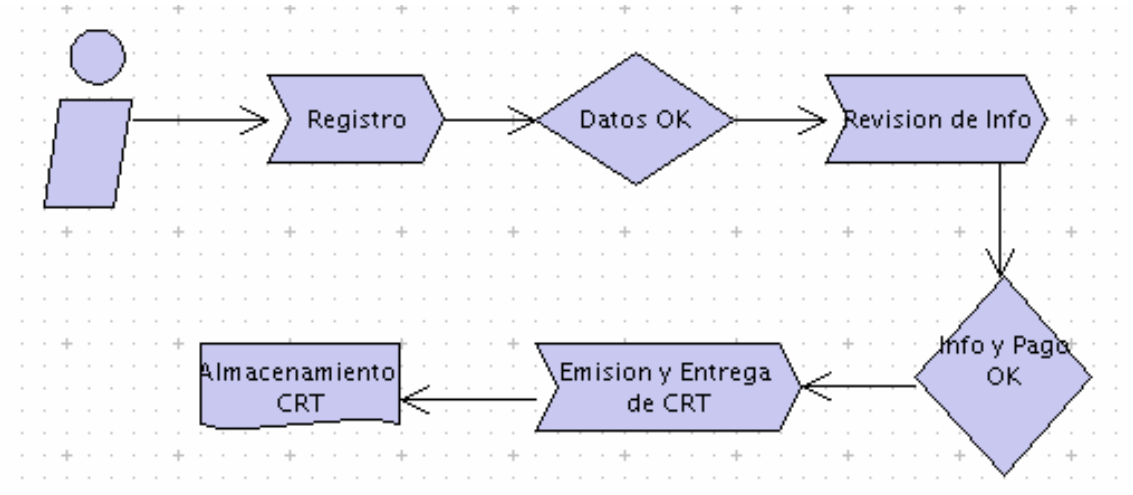

Figura 3-8 Proceso de adquisición de Certificados Reales.

Fuente: ECCert
Autor: ECCert

El modelado del Sistema esta representado por los artefactos de UML y el modelado de datos con Diagramas Entidad Relación. Las siguientes secciones muestran el modelado de datos y del sistema.

3.3.1 Diagramas de Casos de Uso.

Casos de Uso: Existen tres Actores en nuestra Autoridad de Certificación:

- El cliente SOAP[43] de los Servicios Web del núcleo de negocio de ECCert: Puede ser una Autoridad de Registro, una Autoridad de Certificación filial o cualquier entidad autorizada a utilizar los servicios de la Autoridad de Certificación.

[43] SOAP: Simple Object Access Protocol. Es un protocolo ligero para el intercambio de información estructurada en un ambiente descentralizado y distribuido, usando tecnología XML.

- Clientes de ECCert: Son todas las empresas o personas que desean adquirir un Certificado Digital para Servidor.
- Empleados: Son las personas encargadas de verificar los datos y la forma de pago de los clientes, aprobando o rechazando las ordenes de compra (solicitudes de emisión de Certificados Digitales).

Dado los Actores de nuestra Autoridad de Certificación, a continuación listamos los Casos de Uso divididos por actores:

Casos de Uso de Cliente SOAP de Servicios Web:

Caso de Uso:	**Invocar los Servicios de ECCert CA.**
Actor:	Cliente SOAP.
Propósito:	Procesar las transacciones solicitadas de una Autoridad de Registro, Certificación o cliente autorizado.
Resumen:	El cliente arma un XML de solicitud de transacciones, cuyo formato es expuesto por el núcleo de ECCert y es enviado mediante SOAP hacia ECCert, el cual lo procesa y devuelve un XML resultado.
Tipo:	Primario.

Tabla 3.6 Caso de Uso: Invocar los Servicios de ECCert CA.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	**Invocar los Servicios SSL**

Actor:	Cliente SOAP.
Propósito:	Procesar las transacciones solicitadas de una Autoridad de Registro, Certificación o cliente autorizado, referente al procesamiento SSL.
Resumen:	El cliente arma un XML de solicitud de transacciones, cuyo formato es expuesto por el núcleo de ECCert y es enviado mediante SOAP hacia ECCert, el cual rutea hacia el manejador de transacciones SSL para que lo procese y devuelve un XML resultado, el cual es devuelto a su vez al cliente.
Tipo:	Primario, depende del tipo de transacción solicitada del cliente.

Tabla 3.7 Caso de Uso: Invocar los Servicios SSL.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	**Invocar los Servicios LDAP**
Actor:	Cliente SOAP.
Propósito:	Procesar las transacciones solicitadas de una Autoridad de Registro, Certificación o cliente autorizado, referente al almacenamiento de Certificados en LDAP.
Resumen:	El cliente arma un XML de solicitud de transacciones, cuyo formato es expuesto por el núcleo de ECCert y es enviado mediante SOAP hacia ECCert, el cual rutea hacia el manejador de directorio LDAP para que lo procese y devuelve un XML resultado, el cual es devuelto a su vez al cliente.
Tipo:	Primario, depende del tipo de transacción solicitada del cliente.

Tabla 3.8 Caso de Uso: Invocar los Servicios SSL.

Fuente: ECCert
Autor: ECCert

Casos de Uso de Clientes de ECCert:

Caso de Uso:	Ingreso al Sistema.
Actor:	Cliente.
Propósito:	Acceder a la información personalizada del cliente.
Resumen:	El cliente ingresa su usuario y clave configurada al hacer su primera compra en ECCert, luego se le presenta su portal con las opciones y certificados adquiridos y en proceso de compra.
Tipo:	Primario.

Tabla 3.9 Caso de Uso: Ingreso al Sistema.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Emisión de Certificado
Actor:	Cliente.
Propósito:	Emitir un Certificado Digital.
Resumen:	El cliente ingresa los datos necesarios para que ECCert verifique su autenticidad, y además ingresa la información de pago y el tipo de certificado a adquirir. Al terminar el proceso, un Certificado Digital es expedido al cliente.
Tipo:	Primario

Tabla 3.10 Caso de Uso: Emisión de Certificado.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Búsqueda de Certificado
Actor:	Cliente, usuario de Internet.
Propósito:	Chequear la existencia y el estado de un Certificado Digital.
Resumen:	Una persona en el Internet busca un Common Name (nombre del servidor al que se le entrego el certificado), y selecciona el estado en

	el que se debe encontrar, los cuales pueden ser: Activo, Revocado, Expirado, o Todos. Al terminar la transacción, se muestra una lista de los certificados que coincidan con el criterio de búsqueda.
Tipo:	Primario.

Tabla 3.11 Caso de Uso: Búsqueda de Certificado.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Revocación de Certificado.
Actor:	Cliente.
Propósito:	Anular un certificado.
Resumen:	El cliente ingresa con su usuario y clave al sitio de ECCert, consulta sus certificados adquiridos y selecciona la opción de Revocar. Al terminar la transacción, el Certificado Digital queda anulado, la base de datos de certificados se actualiza, y se genera la lista de certificados revocados.
Tipo:	Primario.

Tabla 3.12 Caso de Uso: Revocación de Certificado.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Renovación de Certificado.
Actor:	Cliente.
Propósito:	Expandir el tiempo de uso de un Certificado Digital.
Resumen:	El cliente ingresa con su usuario y clave al sitio de ECCert, consulta sus certificados adquiridos y selecciona la opción de Renovar. Al terminar la transacción, el Certificado Digital queda automáticamente renovado, con una duración de un año adicional.

Tipo:	Primario

Tabla 3.13 Caso de Uso: Renovación de Certificado.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	**Reemplazo de Certificado.**
Actor:	Cliente.
Propósito:	Reemplazar un certificado con otro.
Resumen:	El cliente ingresa con su usuario y clave al sitio de ECCert, consulta sus certificados adquiridos y selecciona la opción de Reemplazar. Al terminar la transacción, el Certificado Digital queda anulado, y se solicita al cliente el ingreso de una nueva petición de certificado y se ingresa la orden de compra.
Tipo:	Primario

Tabla 3.14 Caso de Uso: Reemplazo de Certificado.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	**Consulta de Certificados Adquiridos.**
Actor:	Cliente.
Propósito:	Listar los certificados de un cliente.
Resumen:	El cliente ingresa su usuario y clave. Selecciona la opción de ver sus certificados. Al terminar la transacción, se muestra una lista de todos los Certificados Digitales adquiridos por el cliente.
Tipo:	Primario.

Tabla 3.15 Caso de Uso: Consulta de Certificados Adquiridos.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Consulta de órdenes ingresadas
Actor:	Cliente.
Propósito:	Listar las órdenes para adquisición de Certificados.
Resumen:	El cliente ingresa con su usuario y clave a ECCert. Luego selecciona la opción de ver sus órdenes. Al terminar la transacción, se muestra una lista de todas las órdenes de compra solicitadas por el cliente, con el respectivo estado en la que se encuentran.
Tipo:	Opcional

Tabla 3.16 Caso de Uso: Consulta de Órdenes Ingresadas.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Actualizar Datos de Clientes.
Actor:	Cliente.
Propósito:	Mantener una base de datos real de clientes.
Resumen:	El cliente ingresa a ECCert con su usuario y clave. Selecciona la opción de ver sus datos, y se muestra la información de datos personales, de ubicación y de pagos. Al terminar la transacción el cliente acepta los cambios hechos de sus datos personales y de contacto.
Tipo:	Primario

Tabla 3.17 Caso de Uso: Actualizar datos de Clientes.

Fuente: ECCert
Autor: ECCert

Casos de Uso de Empleados de ECCert:

Caso de Uso:	Ingreso al Sistema Interno.

Actor:	Empleado.
Propósito:	Acceder a la información de órdenes ingresadas por los clientes, para su respectivo procesamiento.
Resumen:	El empleado ingresa su usuario y clave asignada por ECCert. AL terminar la transacción se le presenta su portal con las opciones respectivas dependiendo del rol que tenga (Verificador de Datos o Verificador de Pagos).
Tipo:	Primario.

Tabla 3.18 Caso de Uso: Ingreso al Sistema Interno.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	**Consultar Clientes**
Actor:	Empleado.
Propósito:	Listar los clientes y sus respectivas órdenes.
Resumen:	El empleado de ECCert selecciona la opción de consulta de clientes. Al terminar la transacción se muestra un listado de los clientes de ECCert con la opción de poder ver sus ordenes ingresadas.
Tipo:	Primario

Tabla 3.19 Caso de Uso: Consultar Clientes.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	**Verificar Datos de Orden.**
Actor:	Empleado.
Propósito:	Mostrar información de cliente, referente a datos personales, ubicaciones y de empresa.
Resumen:	El empleado con el rol de “Verificador de Datos” selecciona la opción

	de ver órdenes ingresadas y se muestra un listado con datos de la empresa solicitante. Luego existen opciones de consulta de datos personales y ubicaciones. Al terminar la transacción, el empleado visualiza lo necesario para confirmar los datos ingresados y para luego aprobar la orden.
Tipo:	Primario.

Tabla 3.20 Caso de Uso: Verificar datos de Orden.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Verificar Datos de Pago.
Actor:	Empleado.
Propósito:	Mostrar información de cliente, referente a datos de forma de pago.
Resumen:	El empleado con el rol de "Verificador de Pagos" selecciona la opción de ver órdenes ingresadas y se muestra un listado con datos de la empresa solicitante. Luego existen opciones de consulta de datos de forma de pago. Al terminar la transacción, el empleado visualiza lo necesario para confirmar los datos ingresados y para luego aprobar la orden.
Tipo:	Primario.

Tabla 3.21 Caso de Uso: Verificar datos de Pago.

Fuente: ECCert
Autor: ECCert

Caso de Uso:	Aprobar Orden.
Actor:	Empleado.
Propósito:	Aprobar una orden, pasándola al siguiente paso en el proceso de compra.

Resumen:	El empleado, chequea los datos y aprueba la orden de compra. Al Terminar la transacción, la orden es colocada automáticamente en el siguiente paso del proceso de compra.
Tipo:	Primario

Tabla 3.22 Caso de Uso: Aprobar Orden.

Fuente: ECCert
Autor: ECCert

Diagramas de Casos de Uso: A continuación se presentan los diagramas de Casos de Uso descritos en la sección anterior.

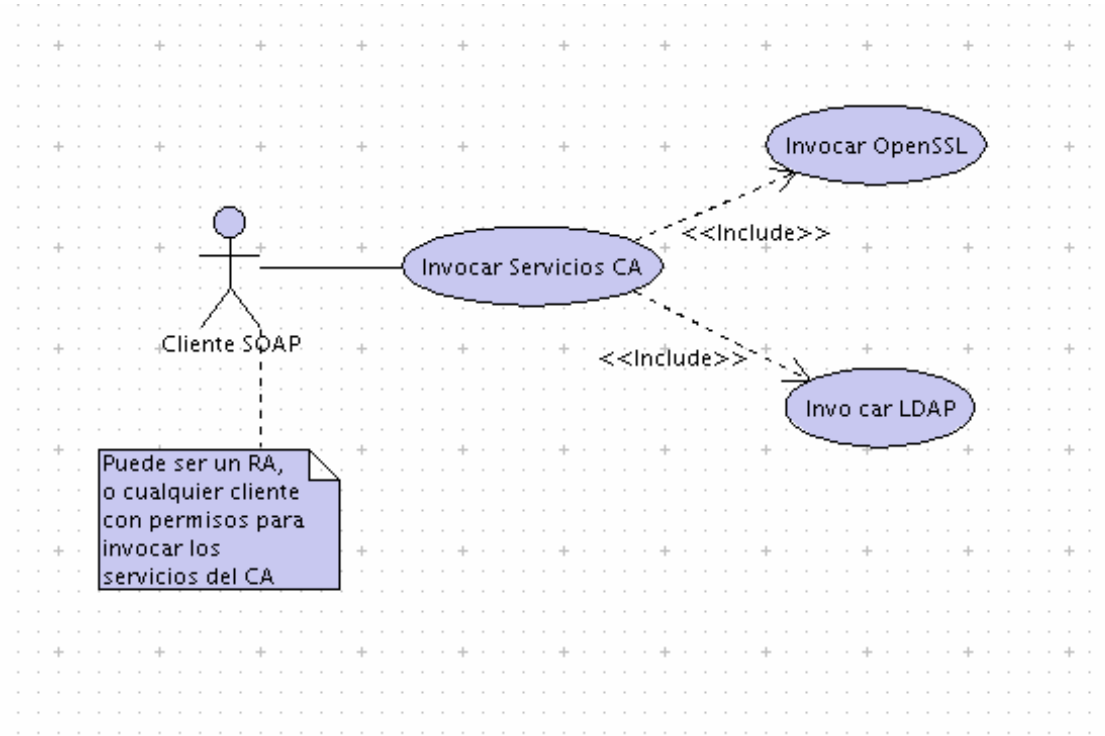

Figura 3-9 Diagrama Caso de Uso del Núcleo de Negocio de ECCert

Fuente: ECCert
Autor: ECCert

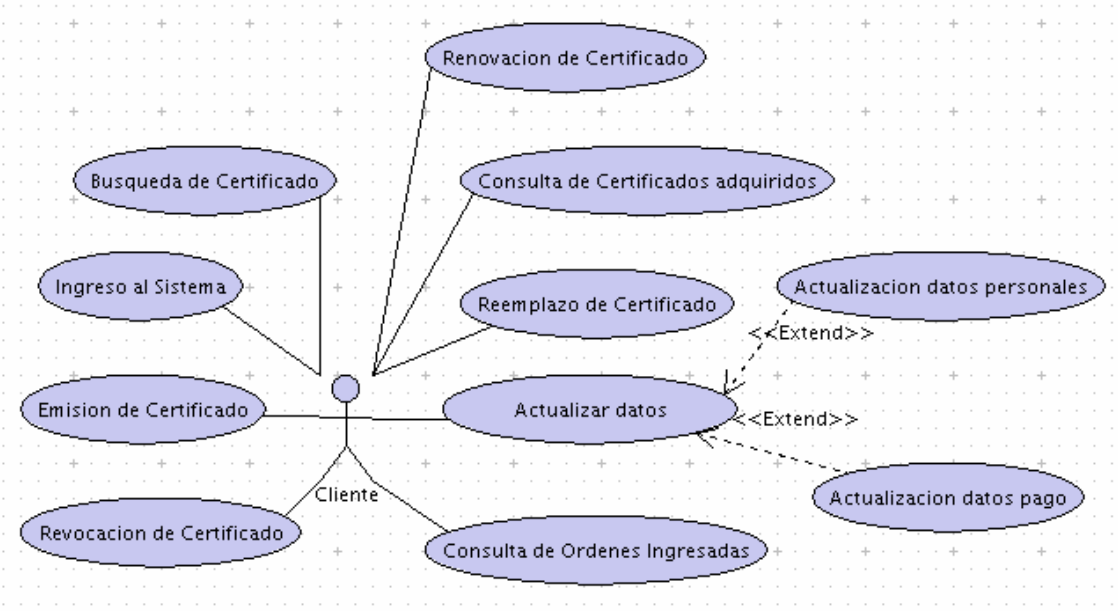

Figura 3-10 Diagrama Caso de Uso de Clientes de ECCert

Fuente: ECCert
Autor: ECCert

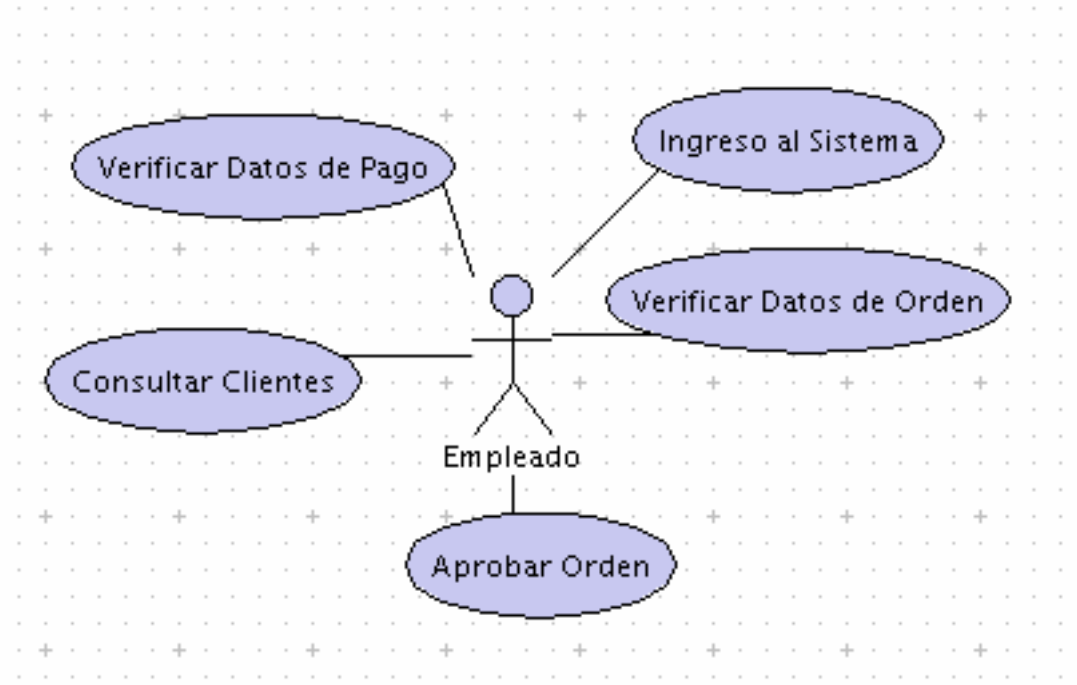

Figura 3-11 Diagrama Caso de Uso de Empleados de ECCert

Fuente: ECCert
Autor: ECCert

3.3.2 Modelo Entidad – Relación

Para presentar con mayor claridad el modelado de datos, hemos dividido los diagramas Entidad – Relación en varias áreas:

Modelo E-R de Clientes y Clientes de Prueba:

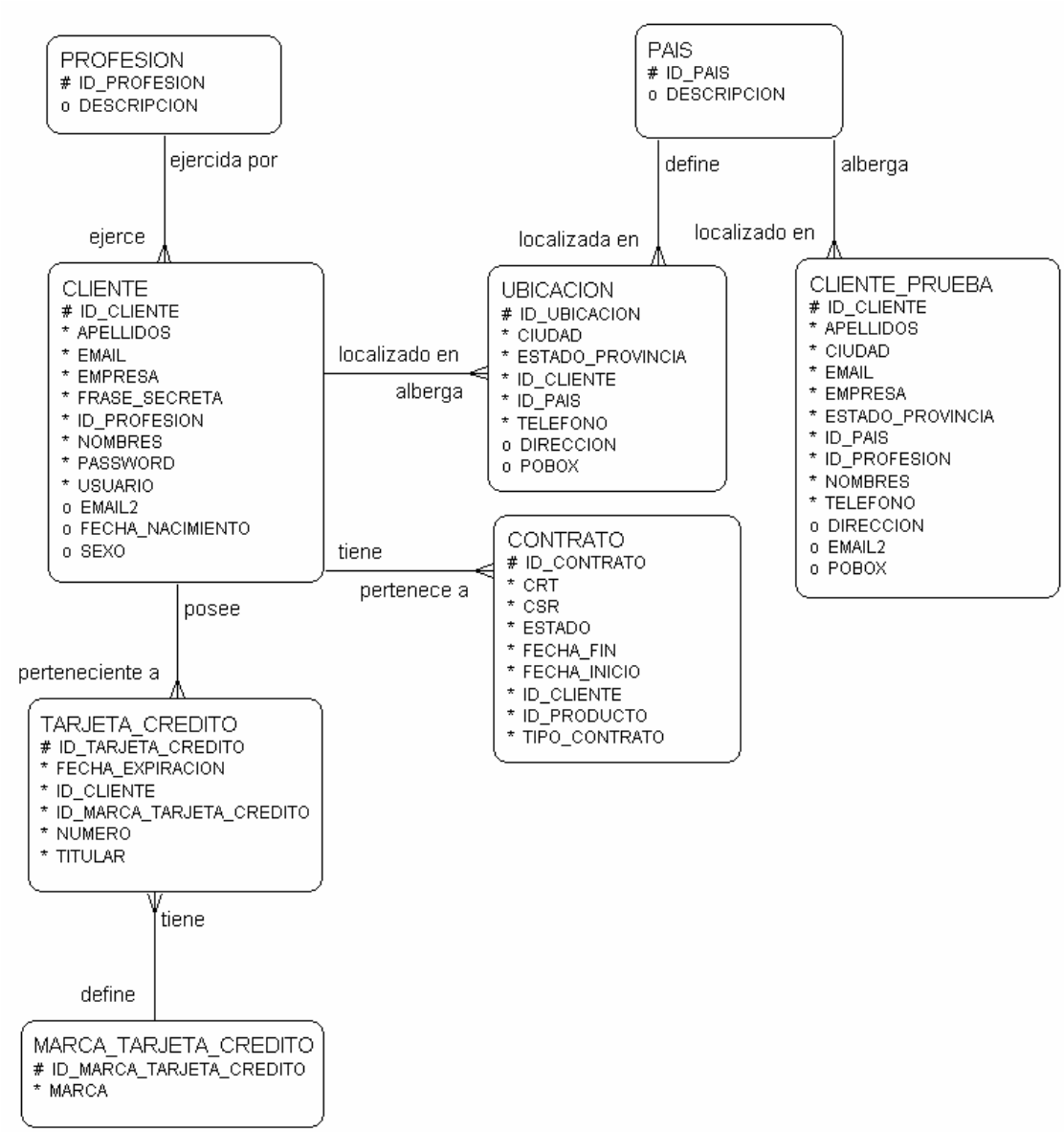

Figura 3-12 Modelo E-R Clientes

Fuente: ECCert
Autor: ECCert

Model E-R de Ordenes de Compra, Contratos y Pagos:

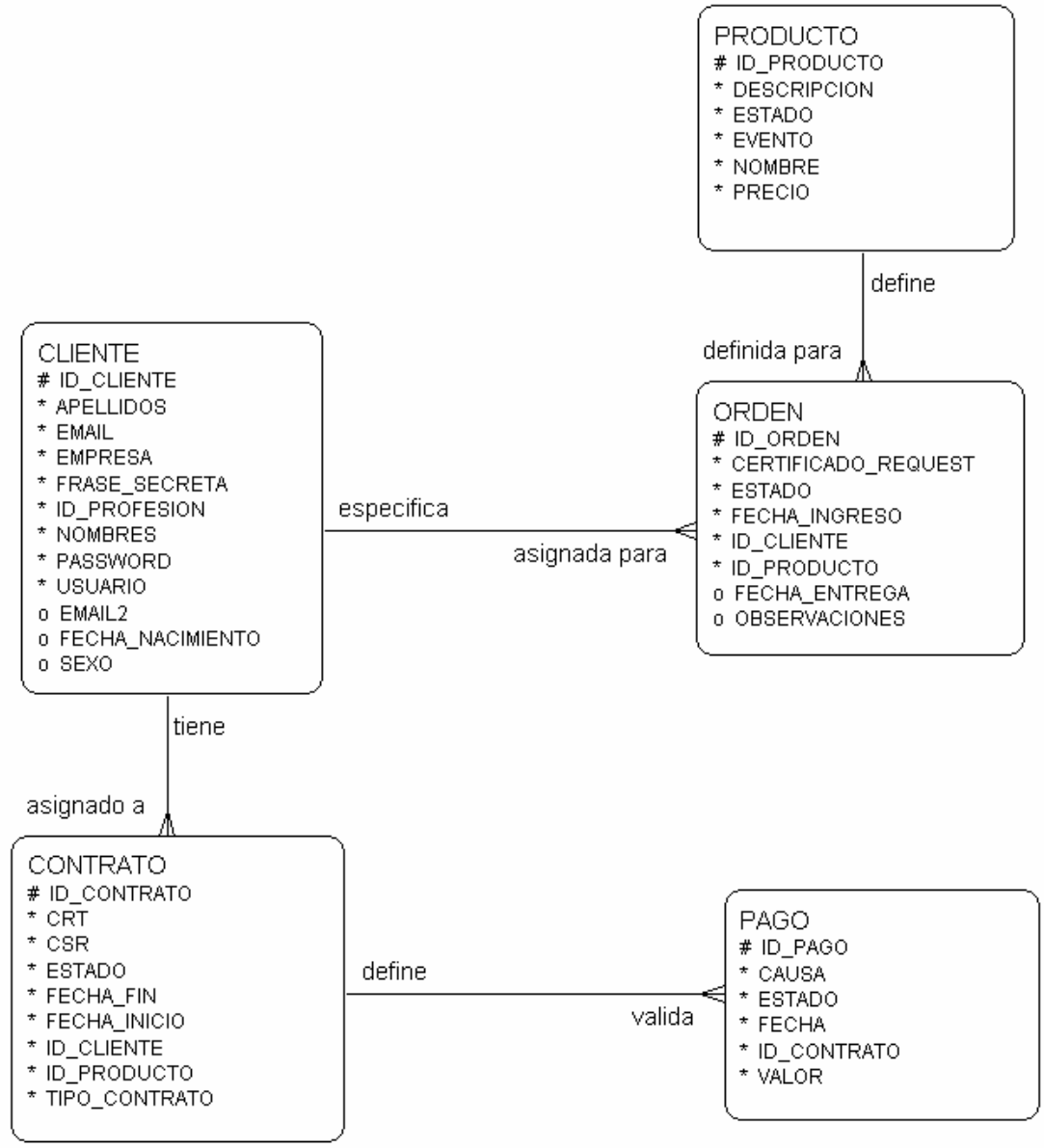

Figura 3-13 Modelo E-R Ordenes, Contratos, Pagos

Fuente: ECCert
Autor: ECCert

Modelo E-R de Productos y Servicios:

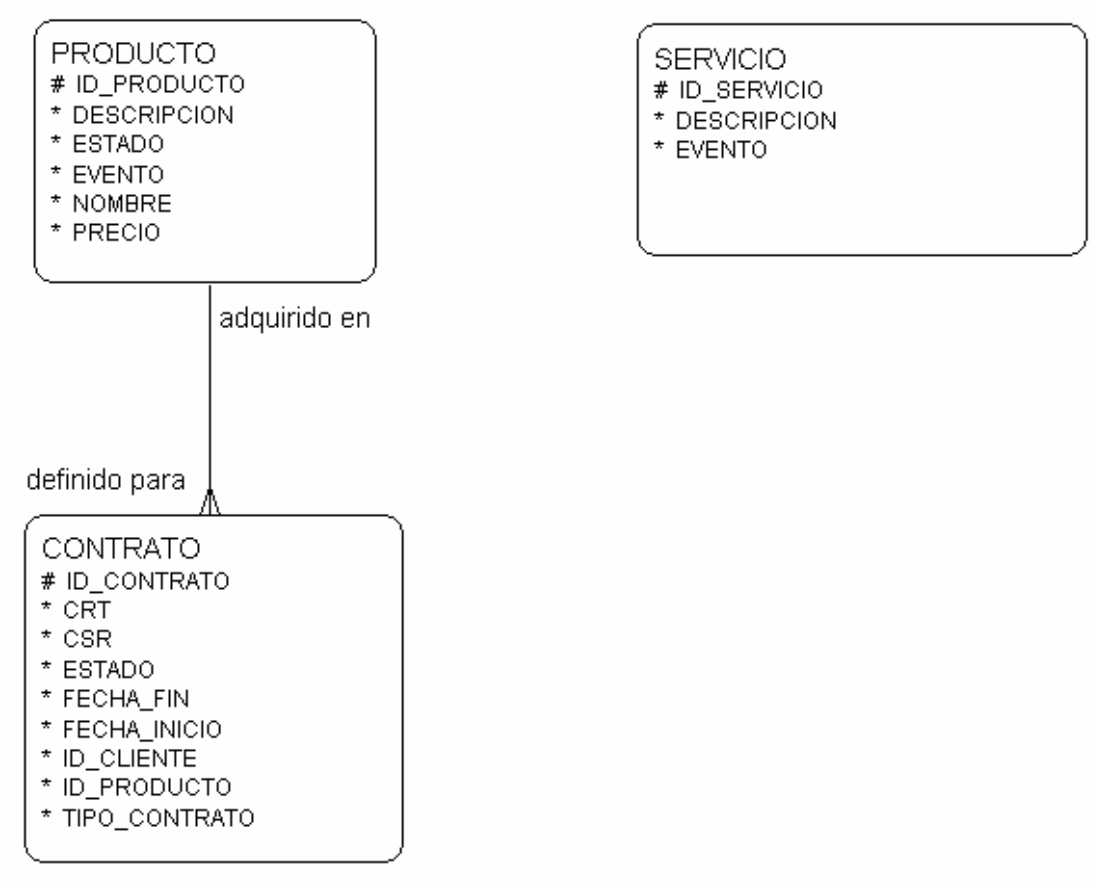

Figura 3-14 Modelo E-R Productos y Servicios de ECCert

Fuente: ECCert
Autor: ECCert

Modelo E-R de Administración de Usuarios y Roles:

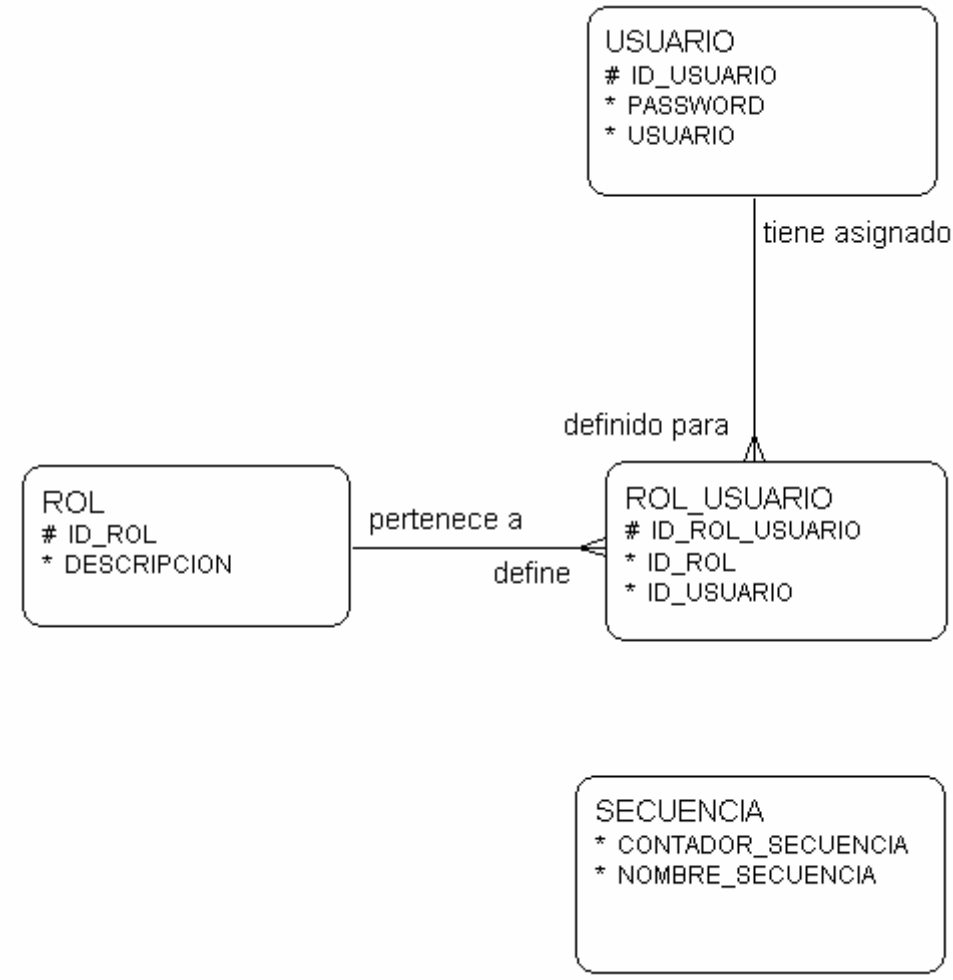

Figura 3-15 Modelo E-R Usuarios y Roles de ECCert

Fuente: ECCert
Autor: ECCert

A continuación mostramos la descripción de los campos de las tablas:

Nombre de tabla: **CLIENTE**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_cliente		Number	5		✔
nombres		Varchar2	30		
apellidos		Varchar2	30		
usuario		Varchar2	16		
id_profesion		Number	5		
email		Varchar2	30		
email2	✔	Varchar2	30		
empresa		Varchar2	30		
sexo	✔	Varchar2	1		
fecha_nacimiento		Date			
password		Varchar2	16		
frase_secreta		Varchar2	50		

Nombre de tabla**: PROFESION**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_profesion		Number	5		✔
descripcion	✔	Varchar2	30		

Nombre de tabla**: CLIENTE_PRUEBA**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_cliente		Number	5		✔
nombres		Varchar2	30		
apellidos		Varchar2	30		
id_profesion		Number	5		
email		Varchar2	30		
email2	✔	Varchar2	30		

empresa		Varchar2	30		
telefono		Varchar2	15		
id_pais		Number	3		
estado_provincia		Varchar2	30		
ciudad		Varchar2	30		
pobox	✔	Varchar2	10		
direccion	✔	Varchar2	255		

Nombre de tabla**: PAIS**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
id_pais		Number	5		✔
descripcion	✔	Varchar2	15		

Nombre de tabla**: CONTRATO**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
id_contrato		Number	5		✔
id_cliente		Varchar2	5		
fecha_inicio		Date			
fecha_fin		Date			
tipo_contrato		Varchar2	1		
estado		Varchar2	2		
id_producto		Number	5		
csr		Varchar2	4000		
crt		Varchar2	4000		

Nombre de tabla**: PRODUCTO**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
id_producto		Number	5		✔

descripcion		Varchar2	255		
precio		Number	8	2	
estado		Varchar2	1		
evento		Varchar2	40		
nombre		Varchar2	40		

Nombre de tabla**: MARCA_TARJETA_CREDITO**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_marca_tarjeta_credito		Number	5		✔
marca		Varchar2	20		

Nombre de tabla**: ORDEN**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_orden		Number	5		✔
id_cliente		Number	5		
fecha_ingreso		Date			
fecha_entrega	✔	Date			
estado		Varchar2	2		
observaciones		Varchar2	255		
id_producto		Number	3		
certificado_request		Varchar2	4000		

Nombre de tabla**: PAGO**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_orden		Number	5		✔
id_contrato		Number	5		
valor		Number	8	2	
fecha		Date			
causa		Varchar2	2		

estado		Varchar2	1		

Nombre de tabla**: TARJETA_CREDITO**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_tarjeta_credito		Number	5		✔
id_marca_tarjeta_credito		Number	5		
id_cliente		Number	5		
numero		Varchar2	15		
titular		Varchar2	15		
fecha_expiracion		Date			

Nombre de tabla**: UBICACION**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_ubicacion		Number	5		✔
id_cliente		Number	5		
telefono		Varchar2	15		
direccion	✔	Varchar2	255		
pobox	✔	Varchar2	10		
id_pais		Number	3		
estado_provincia		Varchar2	30		
ciudad		Varchar2	30		

Nombre de tabla**: ROL**

Nombre	Opcional	Formato	Longitud Máxima	Decimales	Clave Primaria
id_rol		Number	5		✔
descripcion		Varchar2	30		

Nombre de tabla: **ROL_USUARIO**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
id_rol_usuario		Number	5		✔
id_usuario		Number	3		
id_rol		Number	3		

Nombre de tabla: **USUARIO**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
id_ usuario		Number	5		✔
usuario		Varchar2	30		
password		Varchar2	30		

Nombre de tabla: **SECUENCIA**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
nombre_ secuencia		Varchar2	20		
contador_secuencia		Number	3		

Nombre de tabla: **SERVICIO**					
Nombre	**Opcional**	**Formato**	**Longitud Máxima**	**Decimales**	**Clave Primaria**
id_servicio		Number	20		✔
descripcion		Varchar2	255		
evento		Varchar2	30		

3.3.3 Diagramas de Interacción de Objetos

Para representar claramente la Interacción de Objetos, hemos seleccionado los diagramas de Secuencia, mostrados a continuación:

Diagrama de Secuencia del Controlador de la Aplicación:

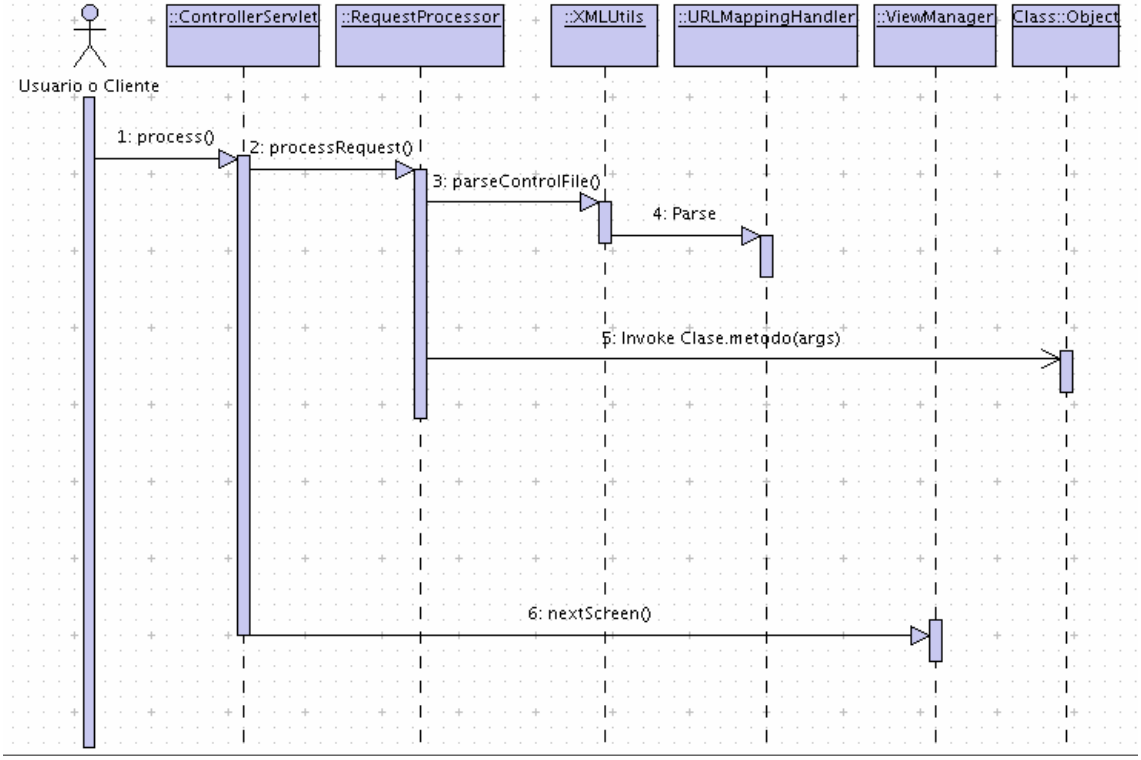

Figura 3-16 Diagrama de Secuencia del Controlador de ECCert.

Fuente: ECCert
Autor: ECCert

Diagrama de Secuencia de Emisión de Certificado Digital:

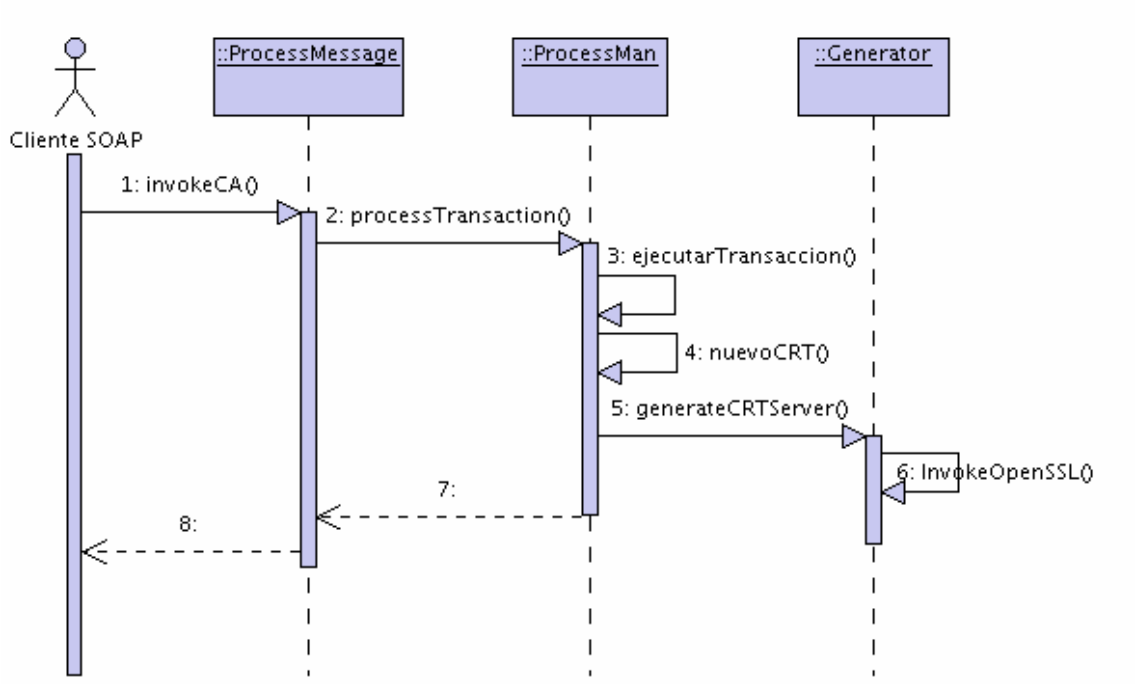

Figura 3-17 Diagrama de Secuencia de emisión de Certificado Digital.

Fuente: ECCert
Autor: ECCert

Diagrama de Secuencia de manejo XML de Transacciones del Núcleo de ECCert:

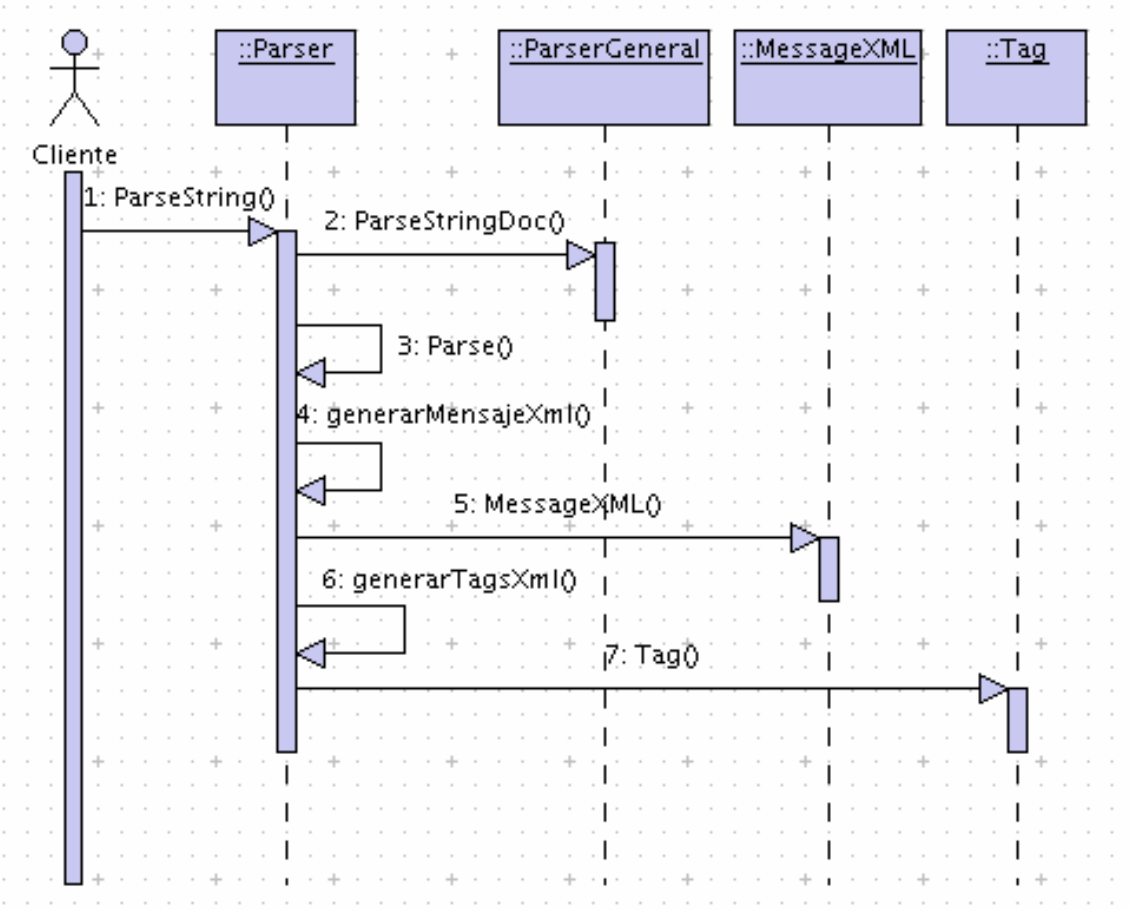

Figura 3-18 Diagrama de Secuencia de manejo XML de Transacciones del Núcleo de ECCert

Fuente: ECCert
Autor: ECCert

Diagrama de Secuencia de revocación de Certificado Digital:

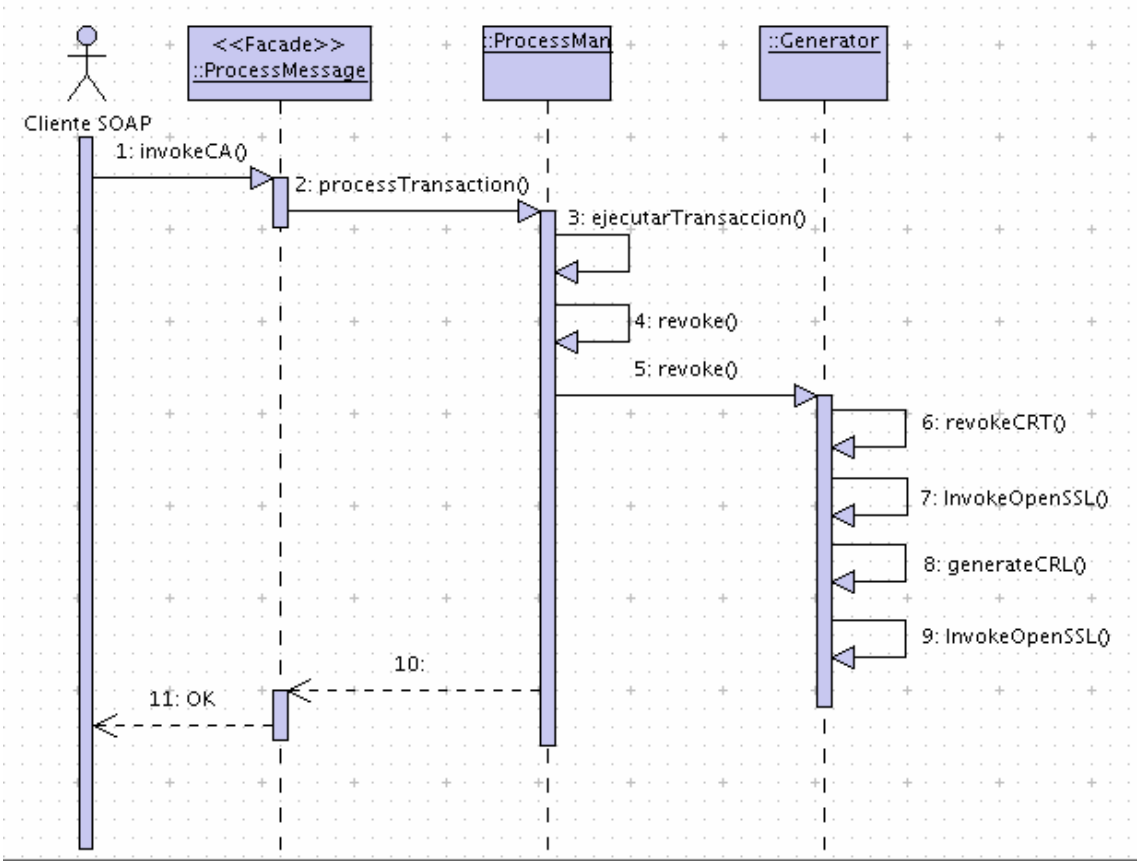

Figura 3-19 Diagrama de Secuencia de Revocación de Certificado Digital.

Fuente: ECCert
Autor: ECCert

Diagrama de Secuencia de renovación de Certificado Digital:

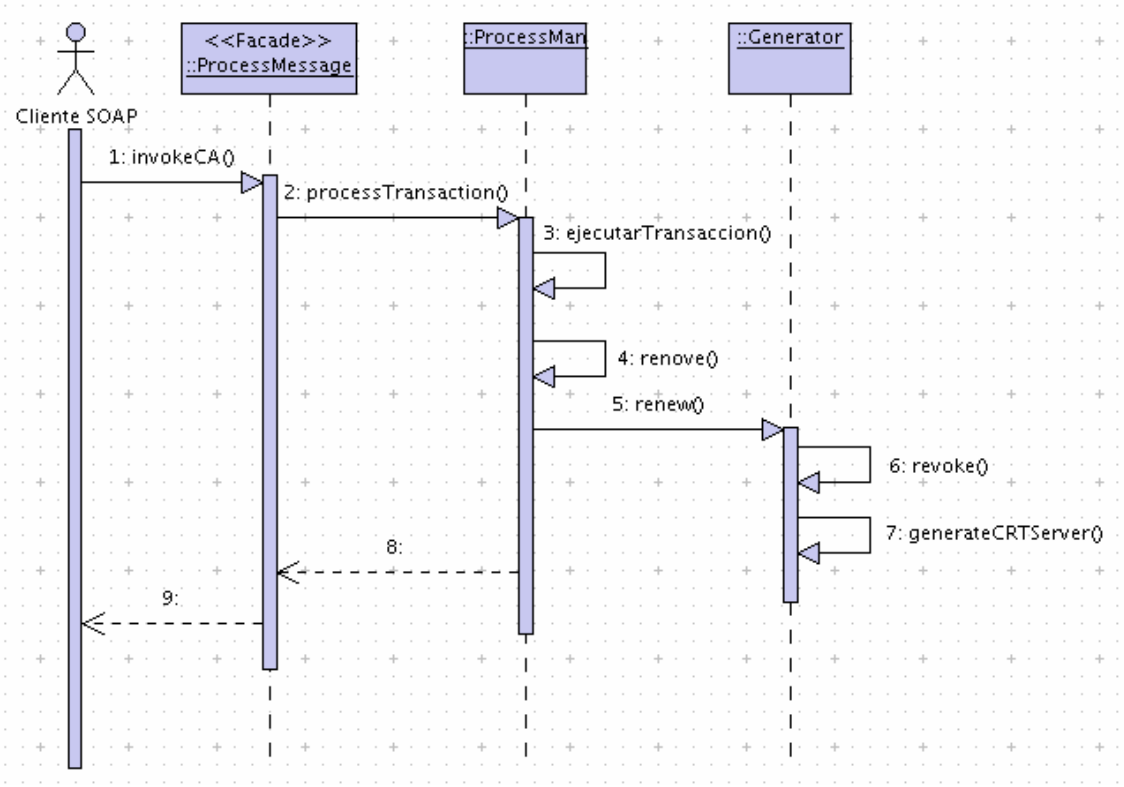

Figura 3-20 Diagrama de Secuencia de Renovación de Certificado Digital.

Fuente: ECCert
Autor: ECCert

Diagrama de Secuencia de reemplazo de Certificado Digital:

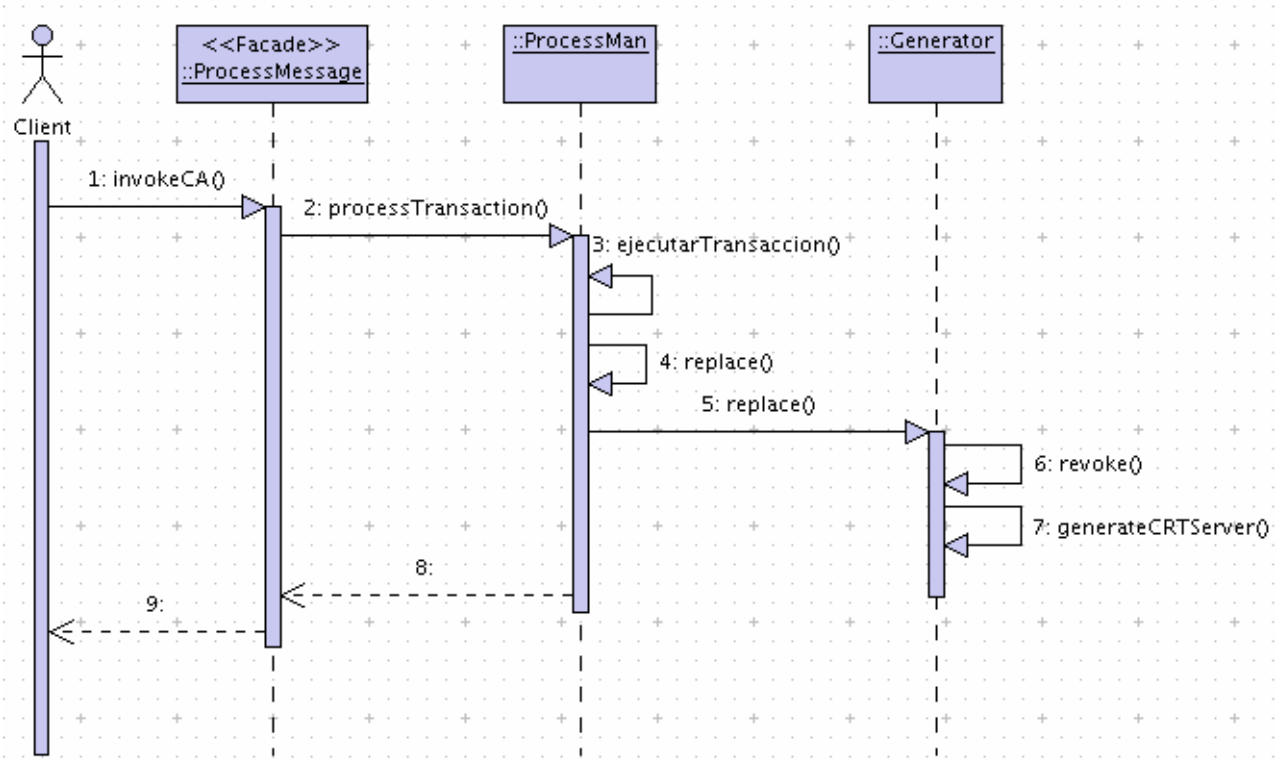

Figura 3-21 Diagrama de Secuencia de Reemplazo de Certificado Digital.

Fuente: ECCert
Autor: ECCert

Diagrama de Secuencia de búsqueda de Certificado Digital:

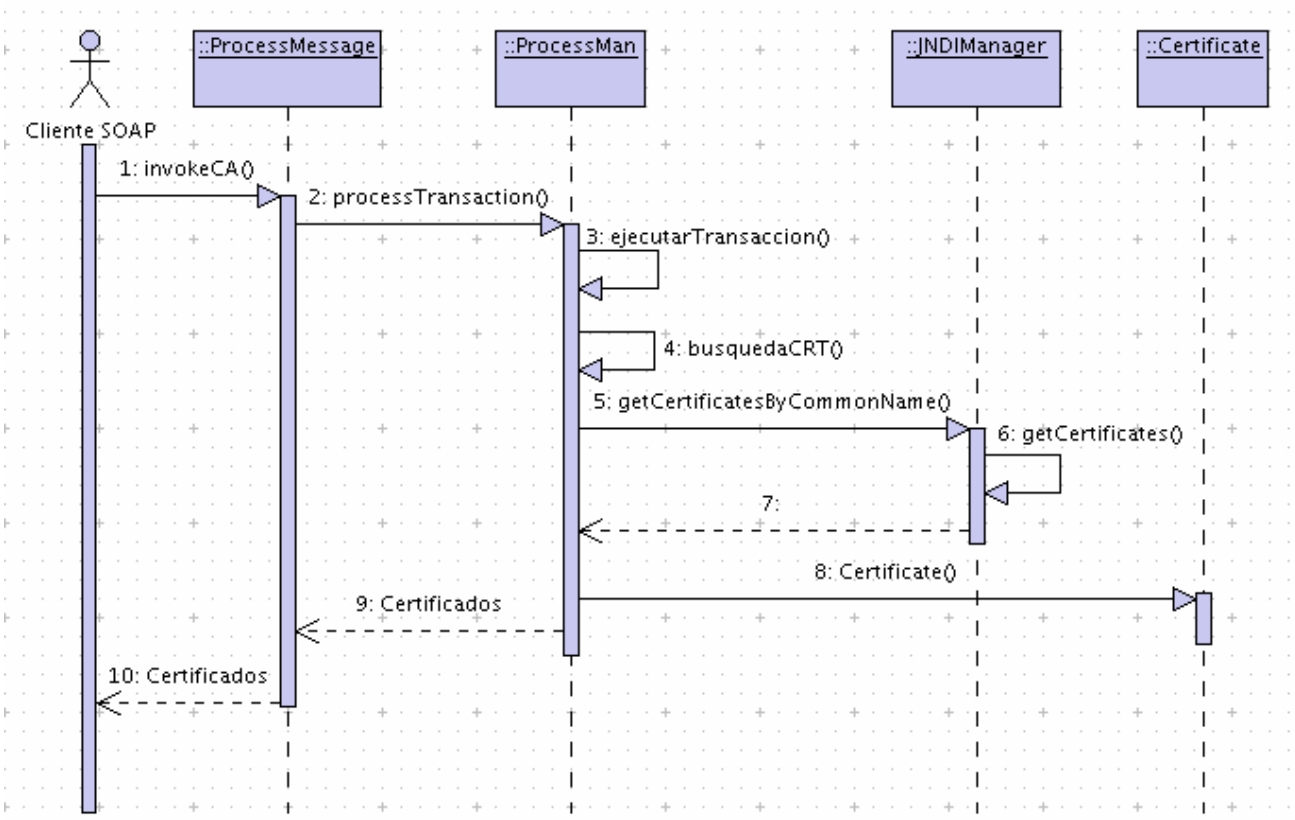

Figura 3-22 Diagrama de Secuencia de Búsqueda de Certificados Digitales.

Fuente: ECCert
Autor: ECCert

3.3.4 Diagrama de Clases

Diagrama de Clases del Controlador de la Aplicación:

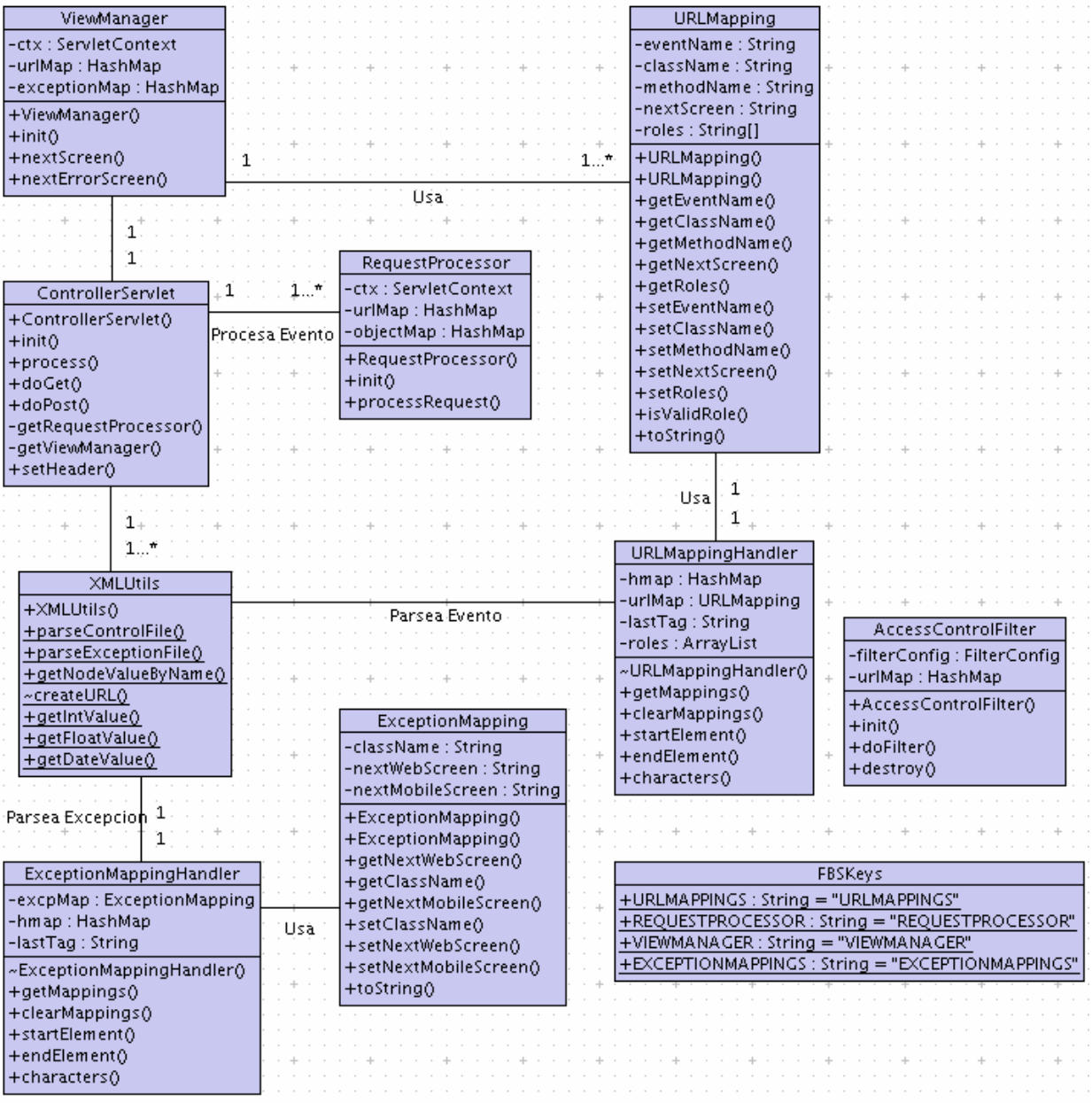

Figura 3-23 Diagrama de Clases del Controlador de la Aplicación.

Fuente: ECCert
Autor: ECCert

Diagrama de Clases del Procesamiento de Transacciones del Núcleo de ECCert:

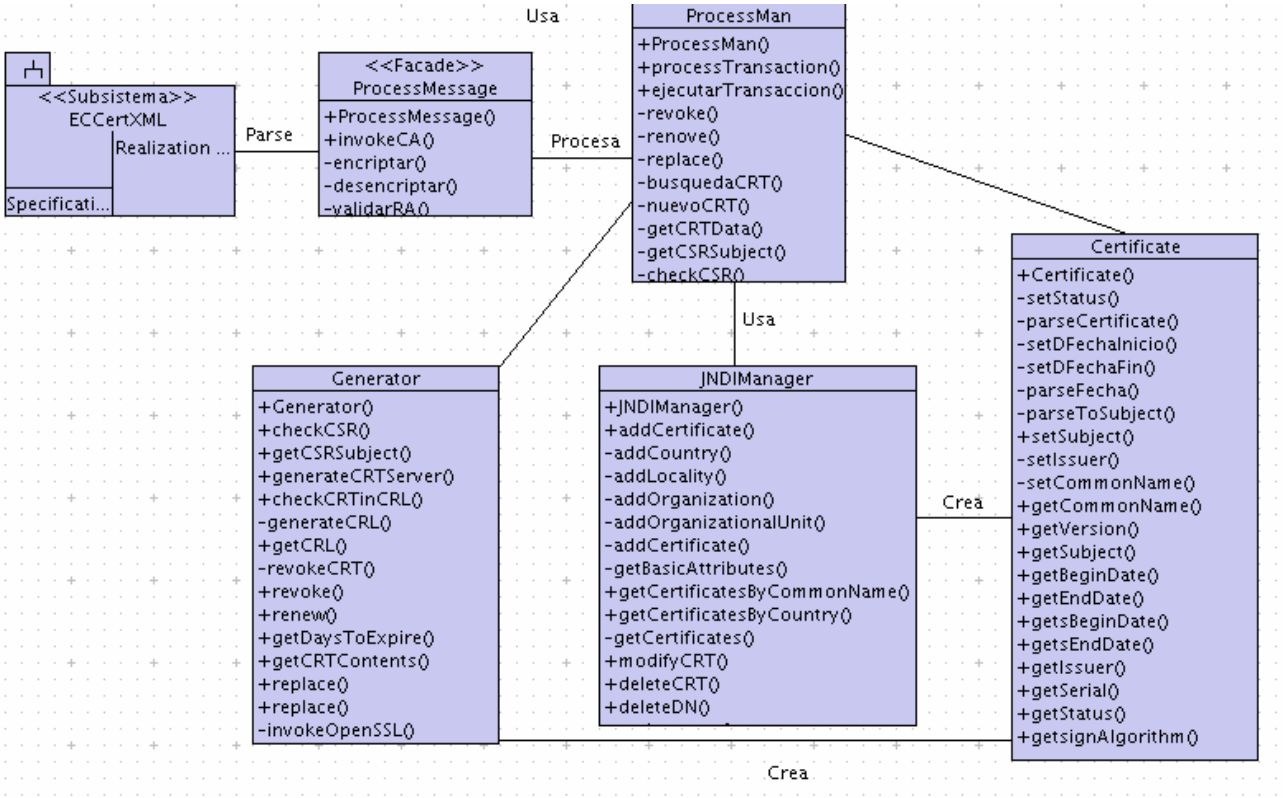

Figura 3-24 Diagrama de Clases del Procesamiento de Transacciones del Núcleo de ECCert.

Fuente: ECCert
Autor: ECCert

Diagrama de Clases del manejo XML para Transacciones del Núcleo de ECCert:

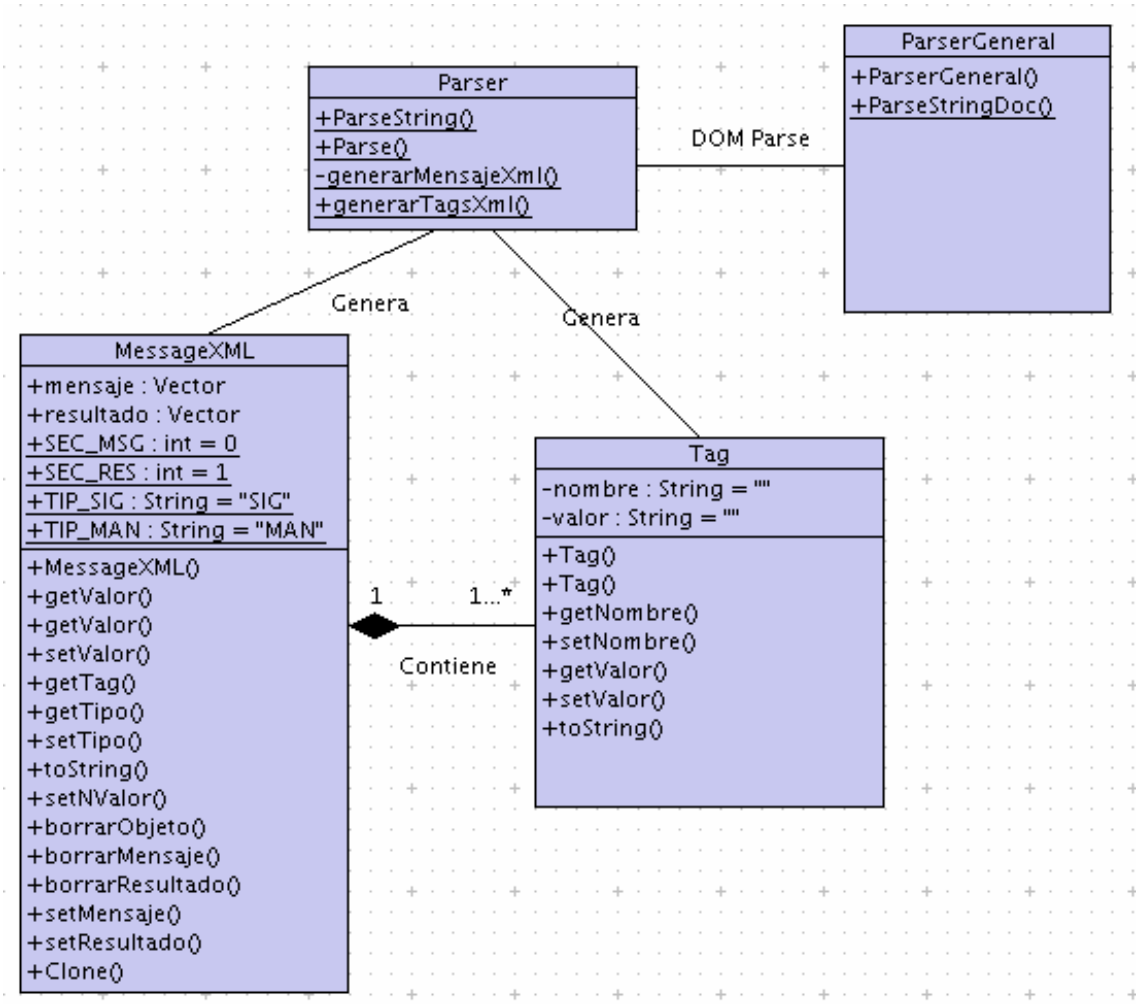

Figura 3-25 Diagrama de Clases del manejo XML para Transacciones del Núcleo de ECCert.

Fuente: ECCert
Autor: ECCert

Diagrama de Clases de emisión de Certificados Digitales:

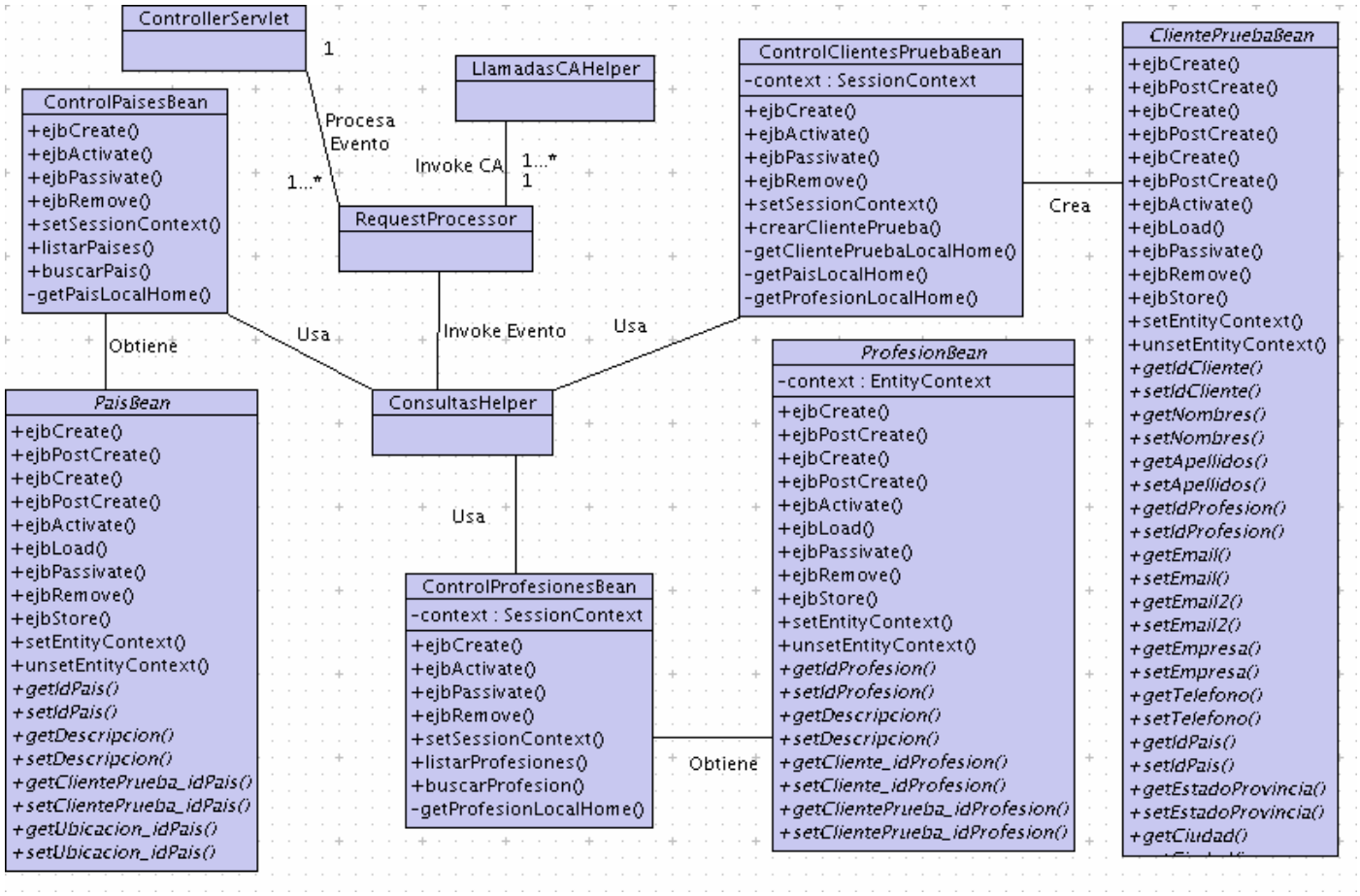

Figura 3-26 Diagrama de Clases de emisión de Certificados Digitales.

Fuente: ECCert
Autor: ECCert

CAPITULO 4

4 IMPLEMENTACIÓN DEL PROYECTO.

Casi todas las organizaciones adquieren códigos de aplicaciones, muchas personas los escriben, y todo el mundo los usa. Pero defectos siguen siendo encontrados en aplicaciones, incluso después de casi cincuenta años de experiencia en programación. Peor, el mismo tipo de defectos aparecen repetidamente otra vez. Esta falla a aprender desde no solo nuestros errores sino también desde la generación de nuestros padres crearon también más vulnerabilidades para ataques potenciales. No es sorprendente que los atacantes contra aplicaciones estén mejorando[44].

[44] The Ten Most Critical Web Application Security Vulnerabilities. Fuente: http://www.owasp.org

A lo largo de este capítulo usted tendrá una presentación del Sitio Web implementado para nuestra Autoridad de Certificación, en donde podrá visualizar la implementación de los procesos. Antes de detallar las funcionalidades de negocio, presentamos la página inicial de ECCert, mostrando los productos y servicios ofrecidos:

Figura 4-1 Página de Inicio de ECCert, mostrando los Productos.

Fuente: ECCert
Autor: ECCert

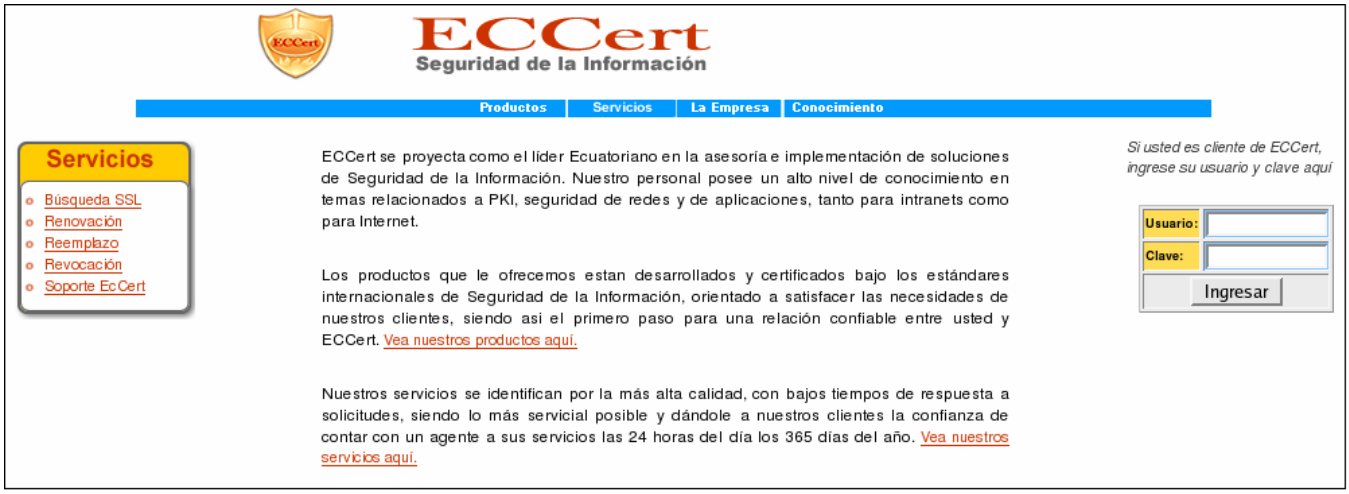

Figura 4-2 Página de Inicio de ECCert, mostrando los Servicios.

Fuente: ECCert
Autor: ECCert

4.1 Autentificación y Roles de Usuarios

Para ingresar a ECCert, un cliente debe configurar un usuario y clave. Estas son definidas al momento de registro de datos. Una vez que ingreso, el cliente tiene las opciones de consulta de certificados, estado de órdenes y datos personales y de pago.

La autentificación del cliente le permite acceder y cambiar su información personal, y es la única manera de que él administre y realice operaciones sobre su Certificado Digital, como la revocación, renovación o reemplazo.

Además, una vez que el usuario accede al sitio de ECCert, tiene la facilidad de adquirir un nuevo certificado basado en su información personal y de pago, agilitando el proceso de emisión de certificados para el cliente.

Para los empleados de ECCert, se manejan dos roles: El verificador de datos y el verificador de pagos. Estos roles están definidos en su usuario, y es validado mediante autentificación en la aplicación ECCert Interna (solo para usuarios de la empresa).

El verificador de datos es el encargado de chequear la información de contacto, datos del cliente y en primordialmente el Common Name. El verificador de pagos es el encargado de contactar a la emisora de la

tarjeta de crédito con la que el cliente cancela el valor del producto adquirido.

Figura 4-3 Ingreso a ECCert, luego de autenticación de clientes.

Fuente: ECCert
Autor: ECCert

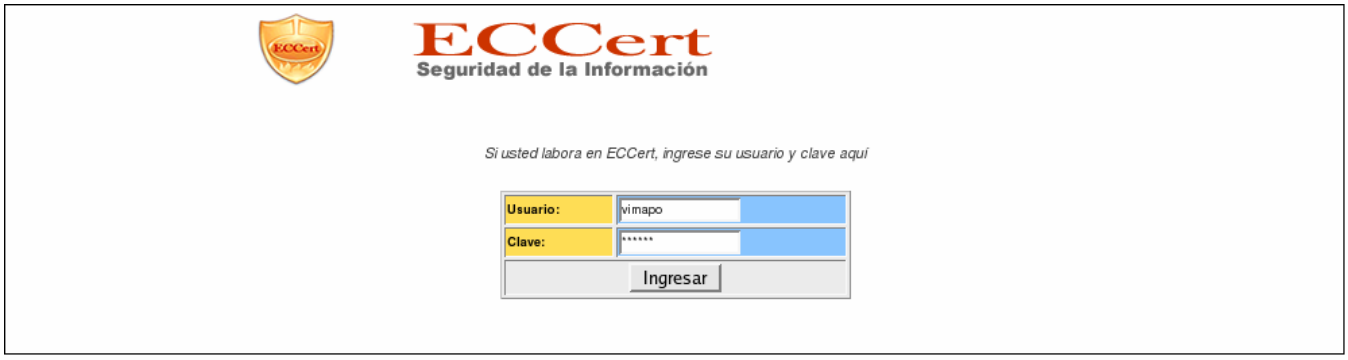

Figura 4-4 Ingreso de empleados de ECCert.

Fuente: ECCert
Autor: ECCert

4.2 Emisión de Certificados Digitales de Prueba

Con el objetivo de que los usuarios del Internet realicen pruebas y se familiaricen con el uso de Certificados Digitales, ECCert emite Certificados de Prueba. Estos tienen la duración de un mes, y no es necesario ser cliente para obtenerlo.

El único requisito es llenar datos básicos de contacto, esto es para mantener una base de datos de los usuarios que han probado nuestro producto.

Una vez completo el proceso, ECCErt crea automáticamente un Certificado Digital de Prueba, y este es enviado por correo electrónico.

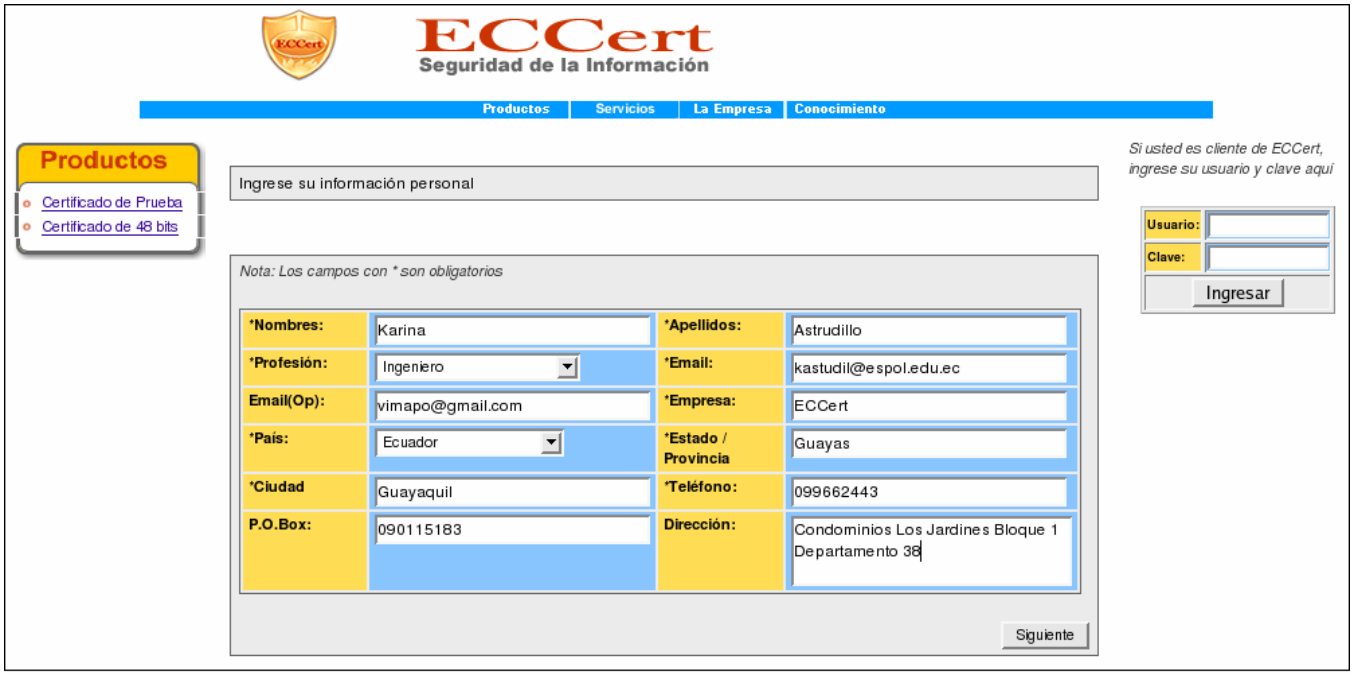

Figura 4-5 Información General, previa la emisión de Certificado de Prueba.

Fuente: ECCert
Autor: ECCert

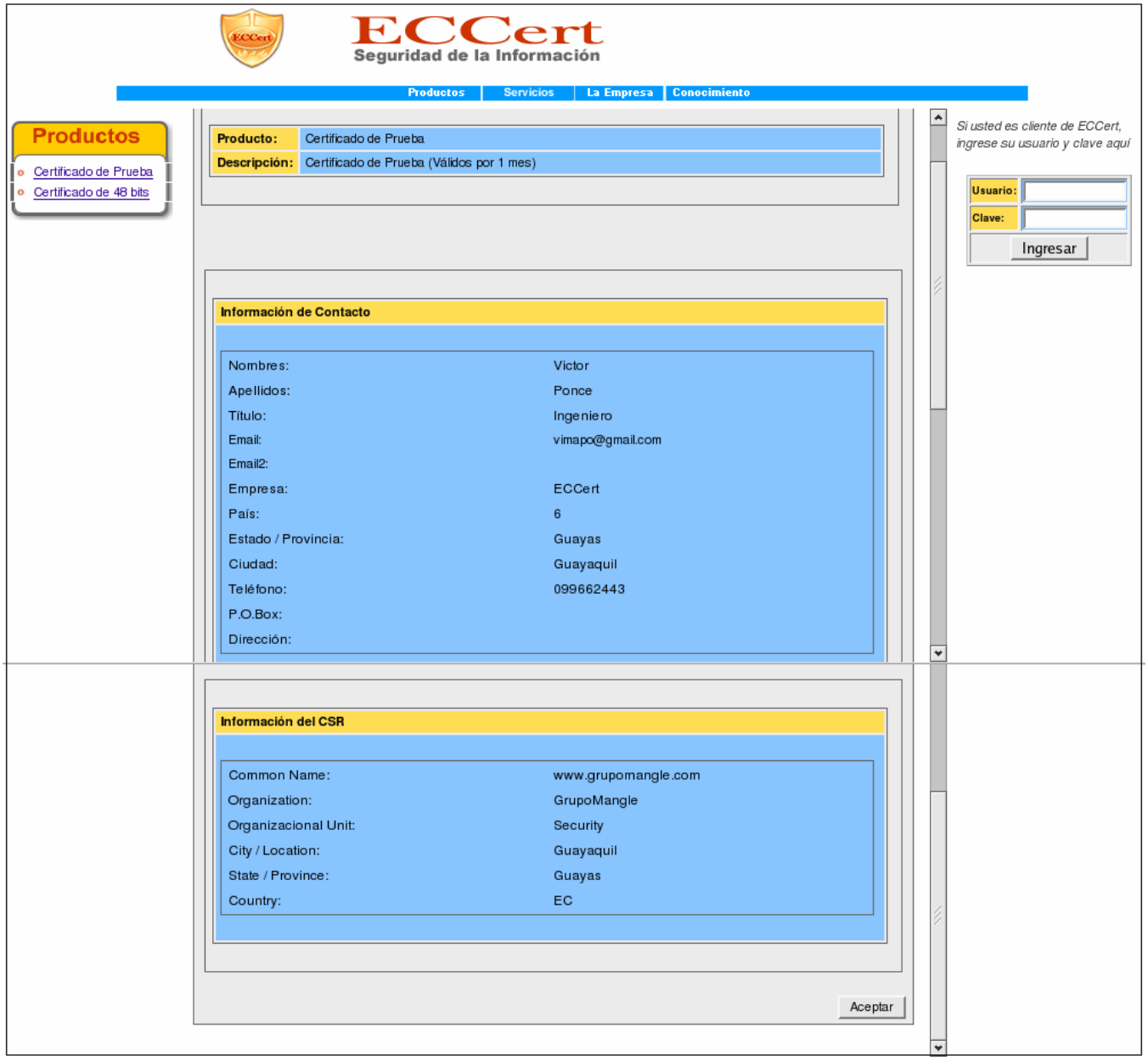

Figura 4-6 Resumen de Información ingresada, en emisión de Certificados de Prueba.

Fuente: ECCert
Autor: ECCert

Figura 4-7 Fin del proceso de emisión de Certificado de Prueba.

Fuente: ECCert
Autor: ECCert

4.3 Venta de Certificados Digitales

El cliente interesado en la compra de un Certificado Digital de ECCert debe seguir los siguientes pasos:

- Registro de Información General.
- Ingreso de Información de Ubicaciones.
- Ingreso de Tarjetas de Crédito para pagos.
- Ingreso del CSR (Certificate Signing Request).
- Confirmación del producto requerido.
- Aceptar la emisión, una vez presentado el resumen de la orden.

A manera de “asistente de compra”, ECCert facilita el proceso de adquisición, solicitando en cada paso los datos requeridos. Si el cliente ya posee un usuario y clave, ECCert omite los pasos de registro.

Las siguientes figuras muestran la secuencia de pantallas del proceso de venta de un Certificado Digital:

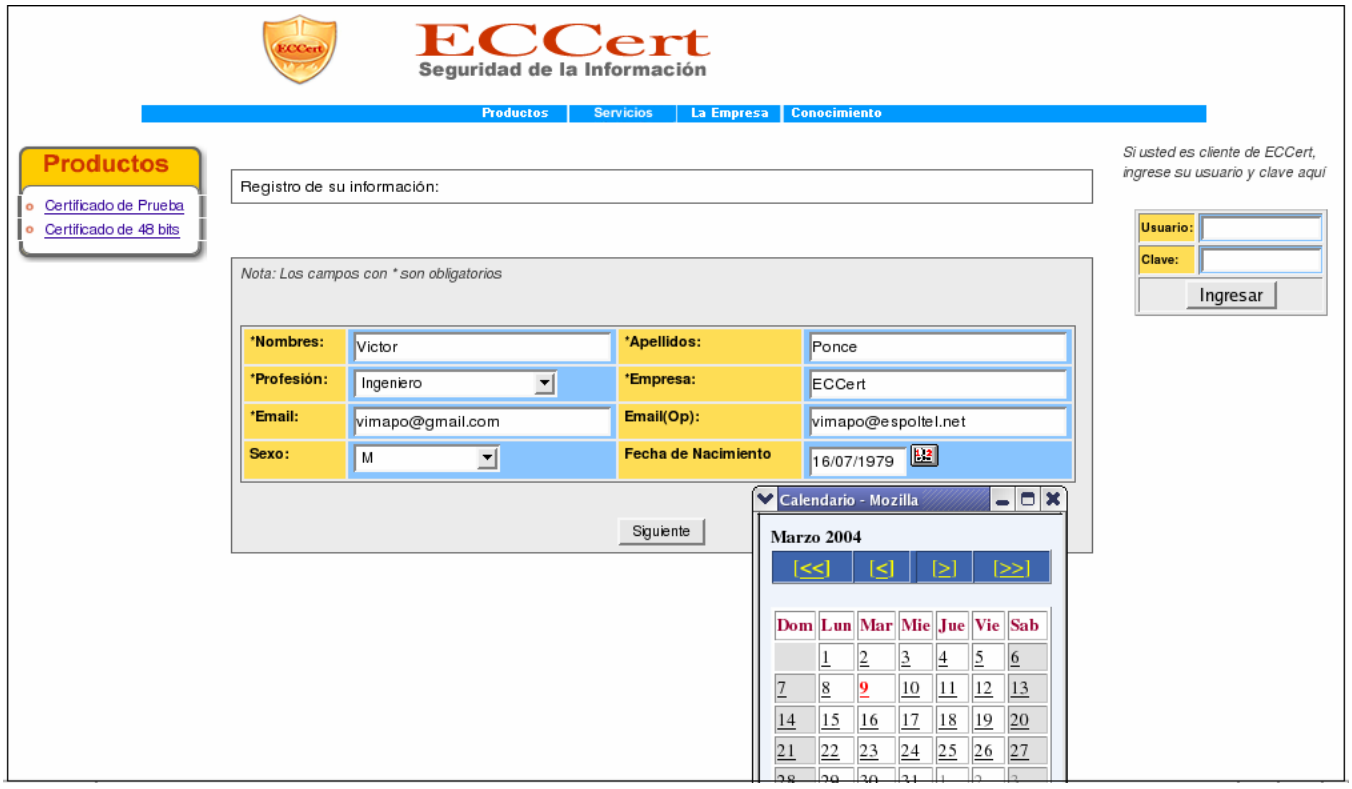

Figura 4-8 Información General de registro de un Usuario nuevo.

Fuente: ECCert
Autor: ECCert

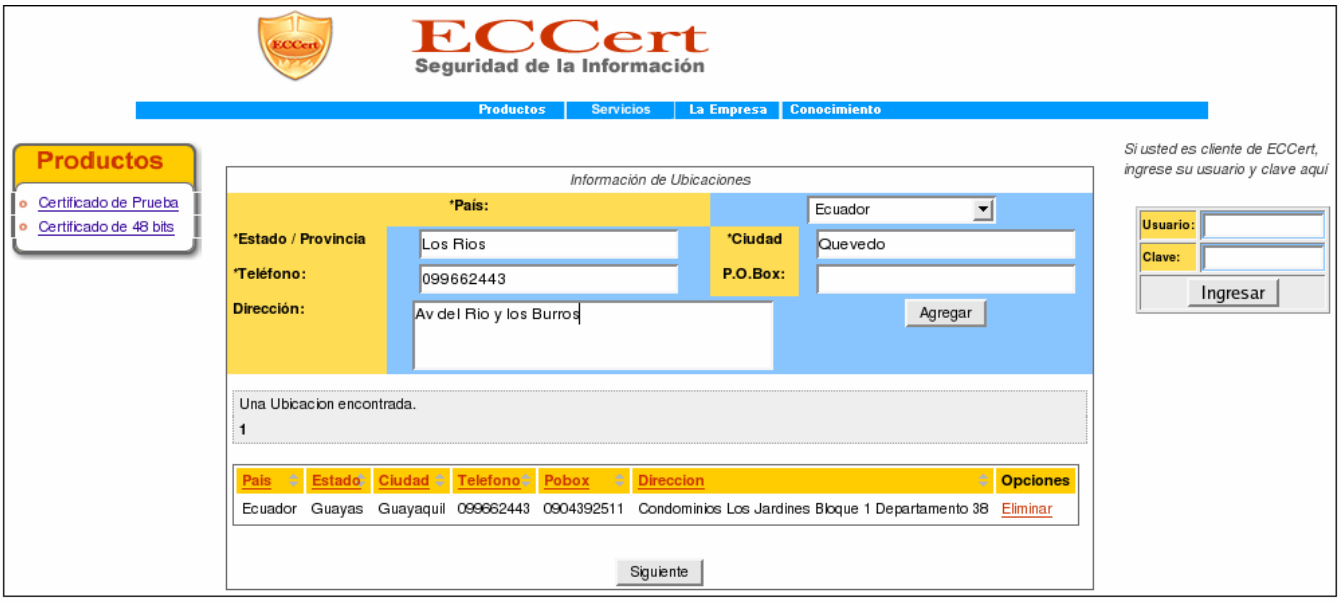

Figura 4-9 Información de Ubicaciones de un Usuario nuevo.

Fuente: ECCert
Autor: ECCert

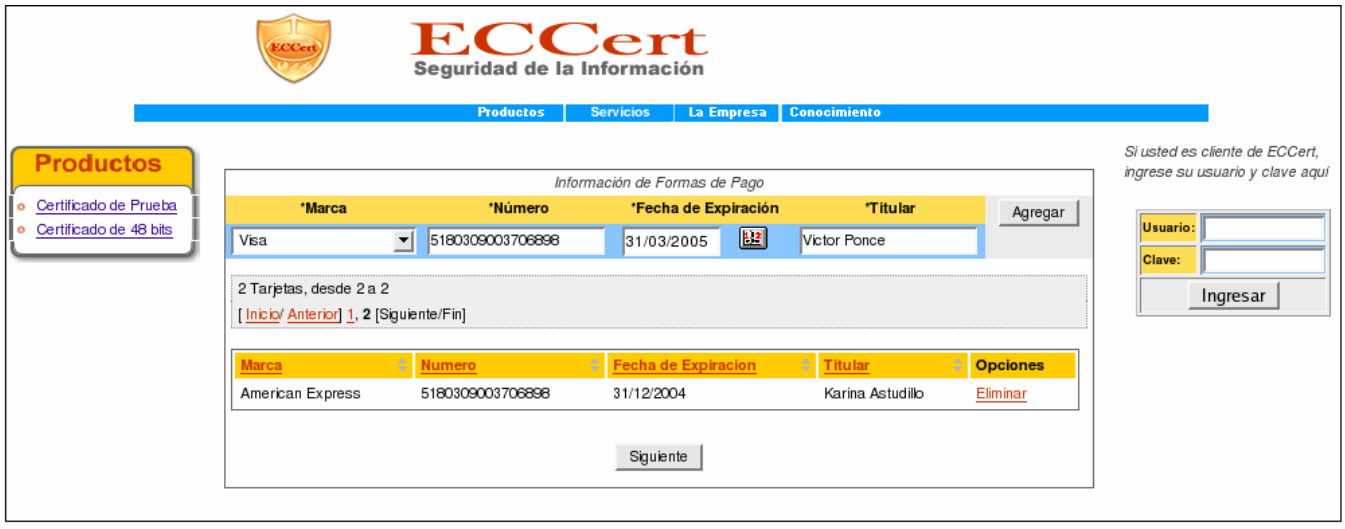

Figura 4-10 Información de Tarjetas de Crédito de un Usuario nuevo.

Fuente: ECCert
Autor: ECCert

Figura 4-11 Ingreso del Certificate Signing Request, para solicitud de un Certificado.

Fuente: ECCert
Autor: ECCert

Figura 4-12 Confirmación del producto solicitado y precio.

Fuente: ECCert
Autor: ECCert

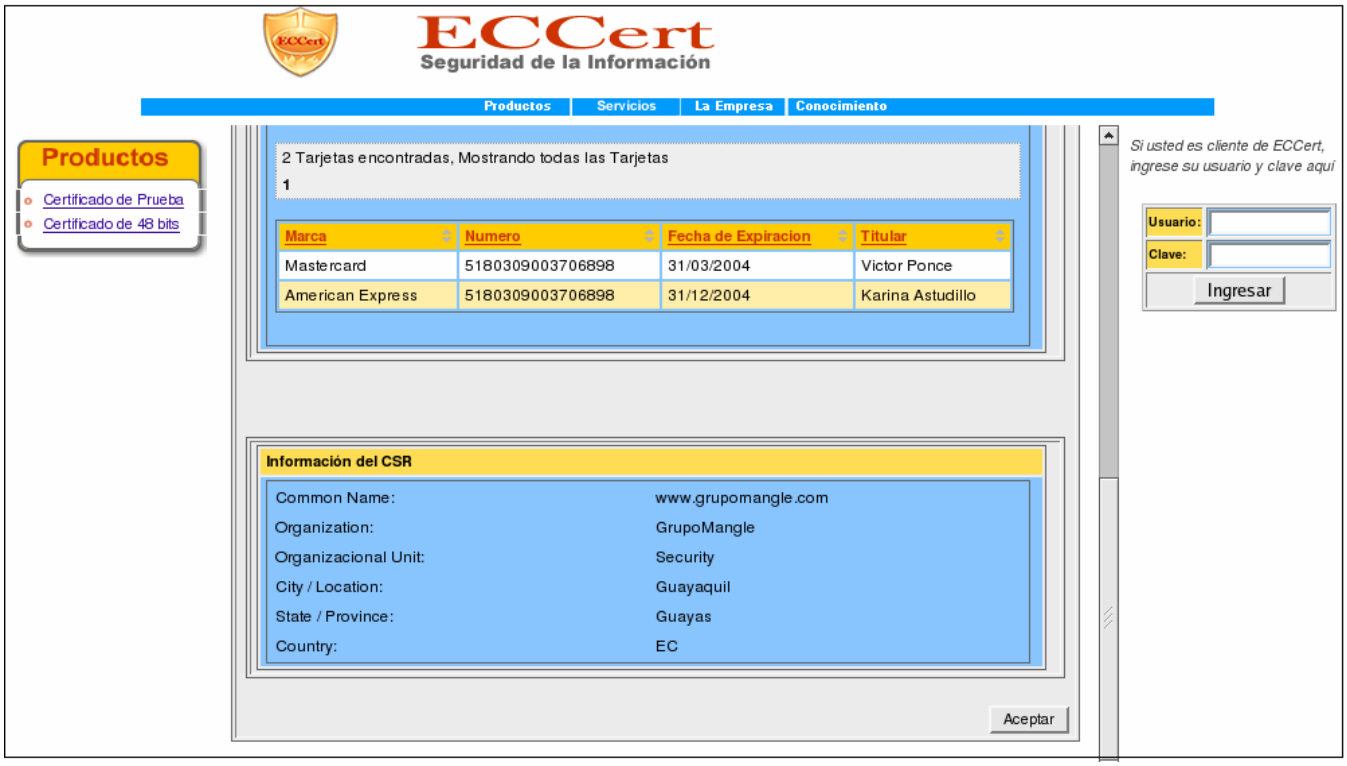

Figura 4-13 Resumen de los datos ingresados y la orden solicitada.

Fuente: ECCert
Autor: ECCert

Figura 4-14 Fin del proceso de ingreso de solicitud de emisión de Certificado.

Fuente: ECCert
Autor: ECCert

4.4 Estado de la Orden de Compra

Una vez ingresada una orden, es pasada al proceso de validación de datos y forma de pago. El cliente puede acceder con su usuario y clave a ECCert, y consultar el estado de su orden para ver en que estado se encuentra.

Los estados pueden ser:

- Verificando Datos: El empleado encargado de verificar los datos esta validando la existencia del cliente, así como de su Common Name.
- Verificando Pago: El empleado encargado de verificar la forma de pago esta realizando el cobro del rubro de la compra.
- Emitido: La orden ha sido aprobada y ha pasado al proceso de emisión y envío del certificado al cliente.

Figura 4-15 Consulta del estado de la Orden de emisión de Certificado.

Fuente: ECCert
Autor: ECCert

4.5 Verificación de Datos, Pagos, Aprobación y Rechazo de Ordenes.

Luego de que la orden es ingresada por el cliente, son los empleados de ECCert los encargados de la verificación de datos y de forma de pago. Ellos tienen la posibilidad de consultar solo información necesaria del cliente.

Una vez verificada, la orden puede ser aprobada o rechazada. Si es aprobada, la orden pasa al siguiente proceso.

Cuando el verificador de datos aprueba la orden, esta pasa al verificador de forma de pago, y una vez este la apruebe, el certificado es emitido automáticamente y enviado por correo electrónico al cliente.

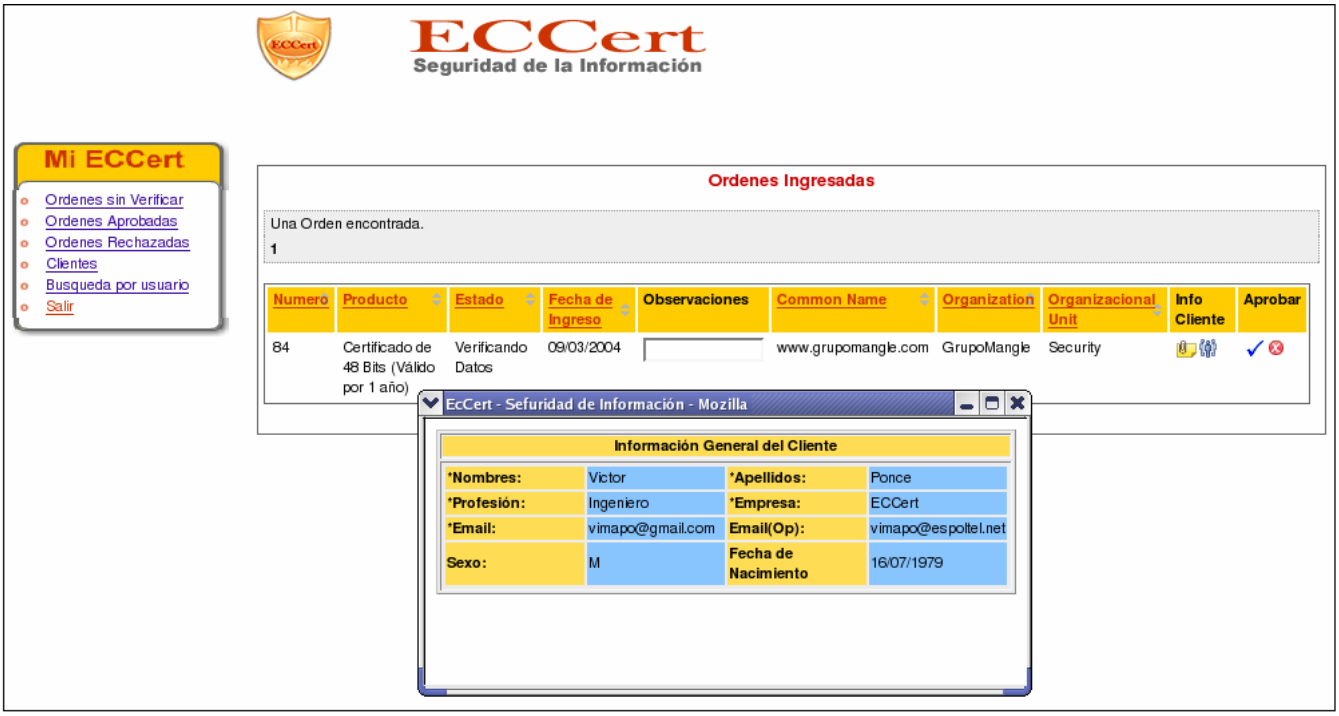

Figura 4-16 Visualización de datos del cliente de una orden ingresada.

Fuente: ECCert
Autor: ECCert

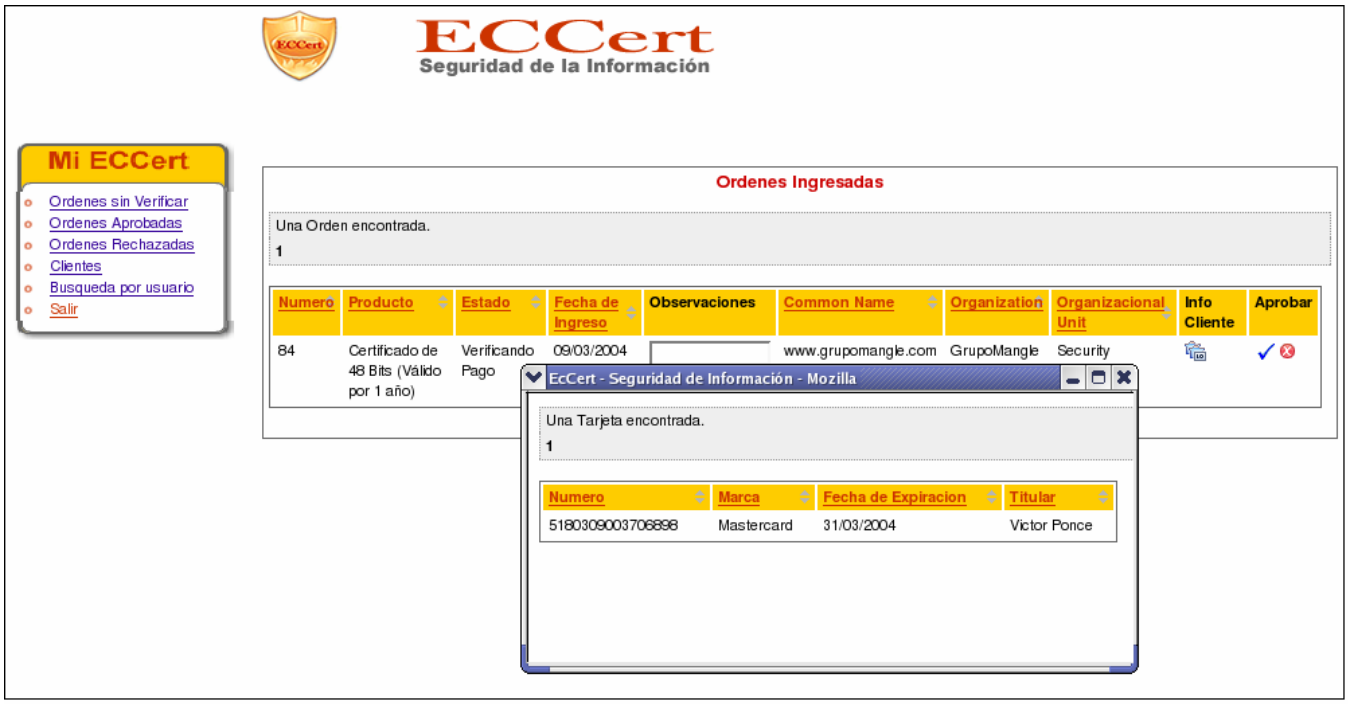

Figura 4-17 Visualización de información de pago de un cliente.

Fuente: ECCert
Autor: ECCert

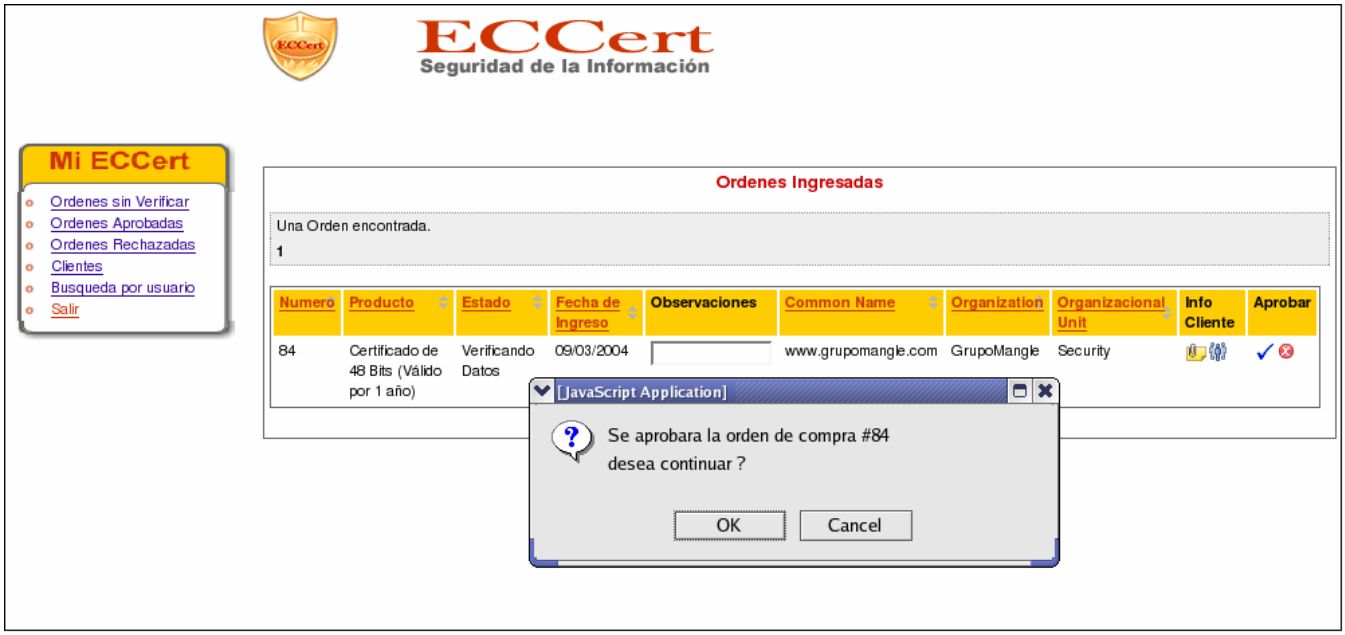

Figura 4-18 Aprobación de una orden para emisión de Certificado Digital.

Fuente: ECCert
Autor: ECCert

Figura 4-19 Rechazo de una orden.

Fuente: ECCert
Autor: ECCert

4.6 Consulta de Certificados Digitales

Esta opción permite al cliente, una vez autentificado, consultar los certificados adquiridos en nuestra Autoridad de Certificación. Muestra los datos más relevantes como el Common Name, Duración y estado del Certificado.

Además presenta opciones con las que el cliente puede administrar su certificado digital, por ejemplo para revocarlo en caso de que haya existido algún inconveniente o renovarlo si es que desea continuar usándolo una vez concluida la duración.

La siguiente figura muestra el listado de certificados de un cliente:

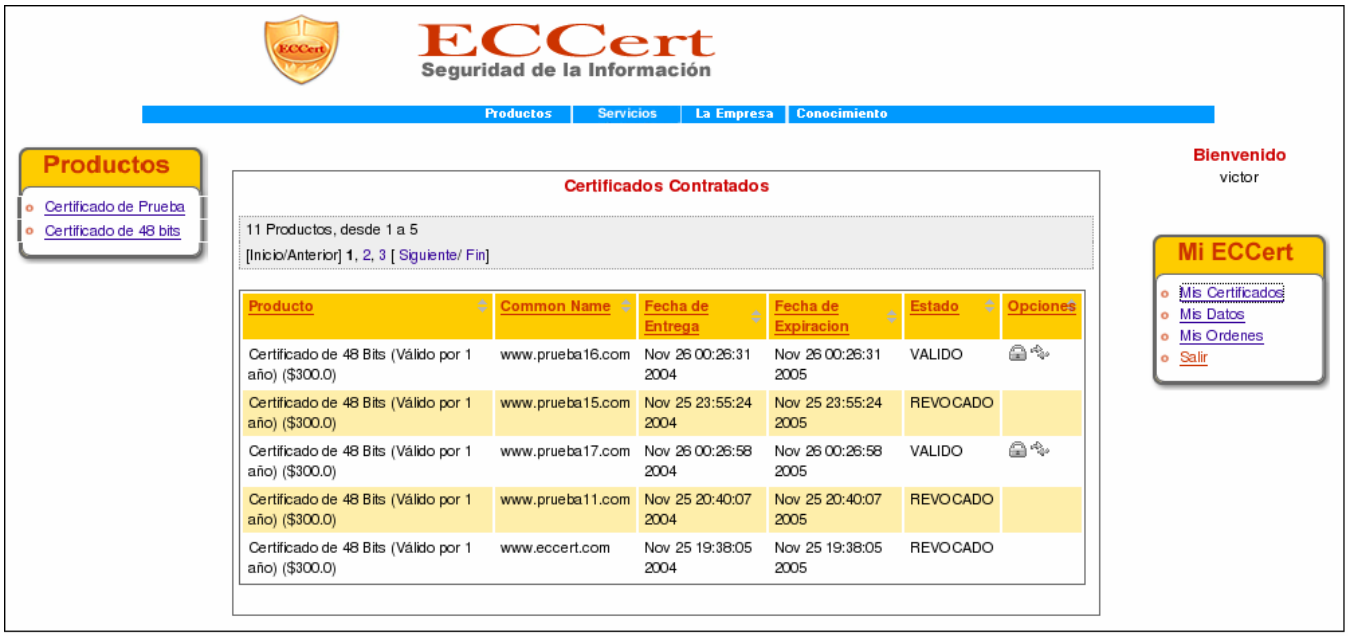

Figura 4-20 Listado de Certificados de un Cliente.

Fuente: ECCert
Autor: ECCert

4.7 Revocación de Certificados Digitales

La revocación solo puede ser realizada por un cliente autentificado. El cliente debe consultar su lista de certificados, y en opciones seleccionar el ícono de “Revocar”. Luego debe confirmar la acción.

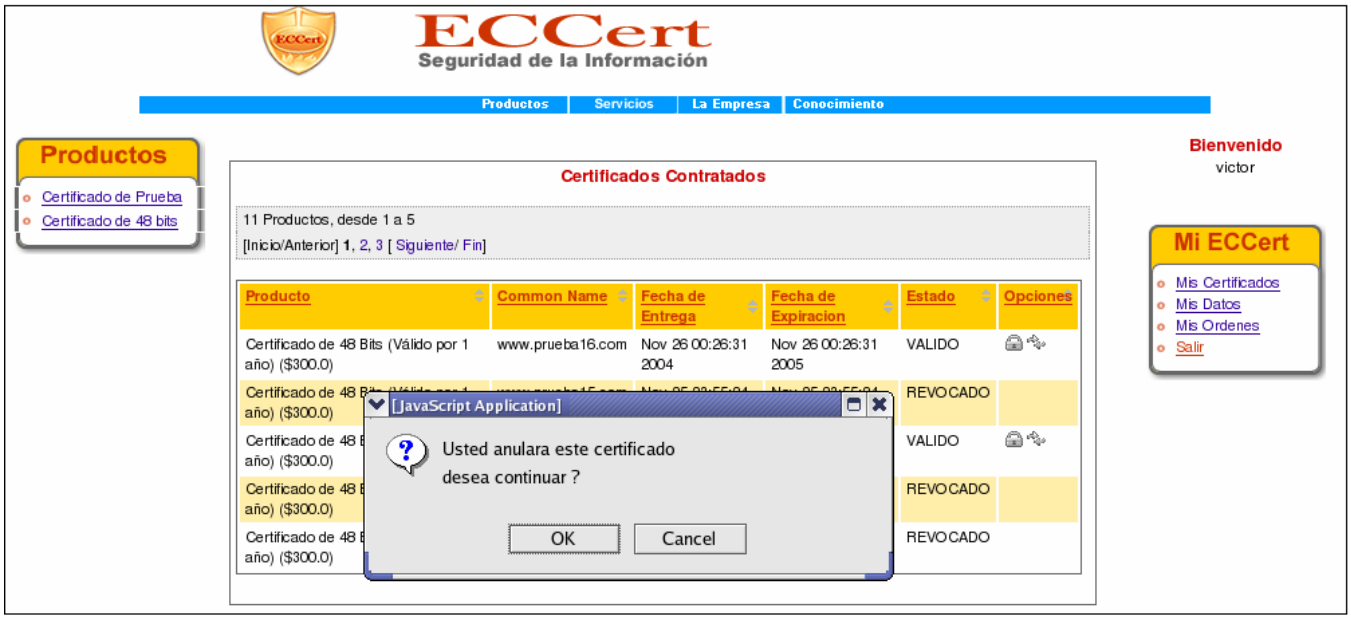

Figura 4-21 Revocación de un Certificado Digital.

Fuente: ECCert
Autor: ECCert

4.8 Renovación de Certificados Digitales.

La renovación solo puede ser realizada por un cliente autentificado. El cliente debe consultar su lista de certificados, y en opciones seleccionar el ícono de “Renovar”. Luego debe confirmar la acción.

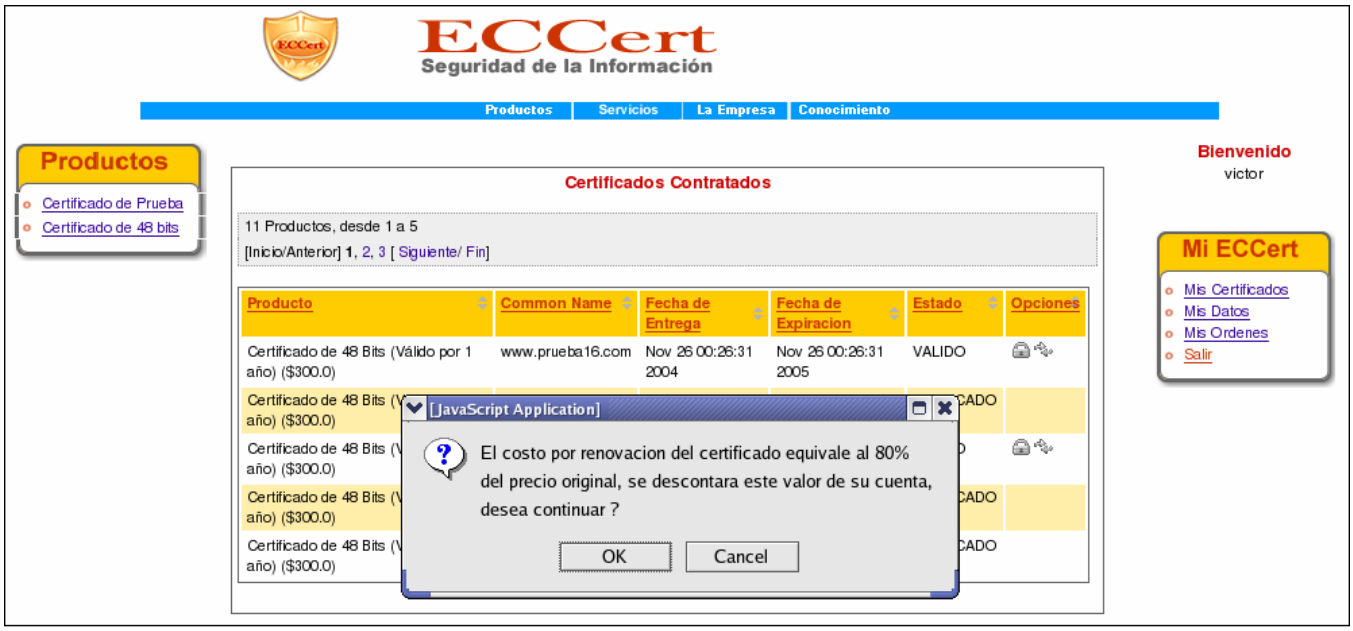

Figura 4-22 Renovación de un Certificado Digital.

Fuente: ECCert
Autor: ECCert

4.9 Reemplazo de Certificados Digitales

El proceso de reemplazo de un Certificado Digital involucra los siguientes pasos:

- Revocar el Certificado que se desea reemplazar.
- Ingresar nuevamente una petición de certificado, para luego sea procesada como una venta de certificado normal.

4.10 Búsqueda de Certificados Digitales

Cualquier usuario del Internet puede verificar la existencia de un Certificado Digital emitido por ECCert, mediante la búsqueda de

Certificados. Ésta se realiza por el Common Name, y con criterios como la validez, duración o estado.

Los resultados muestran la duración y estado, datos que son importantes para la confianza en el Certificado Digital.

Figura 4-23 Búsqueda de Certificados Digitales.

Fuente: ECCert
Autor: ECCert

Figura 4-24 Resultados de Búsqueda de Certificados Digitales.

Fuente: ECCert
Autor: ECCert

CAPÍTULO 5

5 SEGURIDADES ADICIONALES

“La proliferación de vías alternas dentro de una organización, los ataques a nivel de la capa de aplicación, y gusanos devastadores advierte la conclusión de que las defensas locales deben ser complementadas por una gama completa de medidas internas de seguridad. Cumplir esta necesidad requerirá inevitablemente implementar una combinación de diferentes tipos de controles de seguridad, aunque esperamos que los productos que se adapten mejor a los desafíos únicos de la seguridad interna comiencen a emerger en el 2004[45]”.

[45] Chek Point White Paper – Securing Internal Networks: The Final Frontier. http://www.checkpoint.com

Hasta ahora hemos descrito a una Autoridad de Certificación desde el punto de vista de la Red y de la Aplicación. Pero existen otras dimensiones a considerar, como es la Organización y Sociedad que la rodea, o los desafíos tecnológicos a los que nos afrontamos, en donde se enmarcan claramente los virus y los intrusos humanos. Y por supuesto, las condiciones de la naturaleza, que de alguna u otra forma tenemos que considerar.

5.1 Centro de Contingencia.

Para garantizar la alta disponibilidad de los servicios de nuestra Autoridad de Certificación, hemos diseñado un centro de contingencias, cuyo objetivo es funcionar en caso de que exista algún inconveniente con la sucursal principal.

El siguiente gráfico muestra nuestro esquema, que considera solamente los servicios críticos con el objetivo de ahorrar recursos, debido a que la contingencia es eventual y de poca duración:

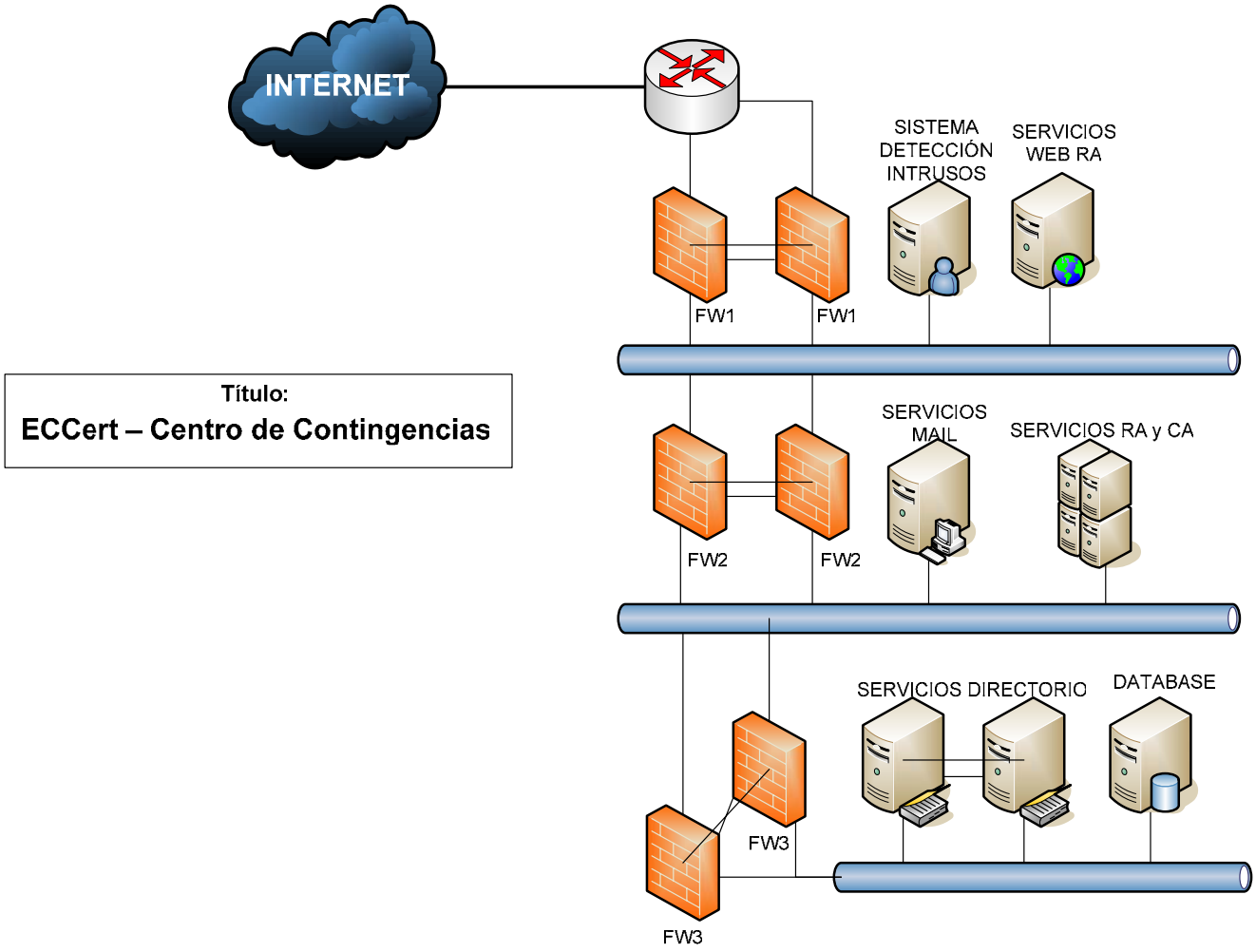

Figura 5-1 Centro de Contingencias de ECCert.

Fuente: ECCert
Autor: ECCert

5.2 Seguridad en Sistemas Operativos.

El Sistema operativo es normalmente solo una porción del total de software que corre en un sistema particular. Pero el Sistema Operativo controla el acceso a los recursos del sistema. La seguridad de los Sistemas Operativos es solo una pequeña parte del problema total de la seguridad en los sistemas de computación, pero éste viene incrementándose en gran medida.

Hay muchas razones para que la seguridad de los Sistemas Operativos reciba especial atención hoy en día.

La evolución de los sistemas de computación, ha sido en las últimas décadas de una magnitud asombrosa. Las computadoras se han tornado más accesibles, también se tiene un aumento en los riesgos vinculados con la seguridad. Pero hay una cosa que se ha mantenido constante a través de todo este tiempo, y es que los sistemas digitales se han vuelto cada vez más complejos. Los microprocesadores se han vuelto más complejos. Los sistemas operativos se han vuelto más complejos. Los ordenadores se han vuelto más complejos. Las redes se han vuelto más complejas. Las redes individuales se han combinado y han aumentado todavía más su complejidad. Ejemplo claro de ello es Internet, la gran red de computadoras, a medida que aumenta su complejidad va tornándose más insegura.

Si se tiene en cuenta que todo software no está libre fallos, entonces un software complejo es probable que falle y un porcentaje de estos fallos afecte a la seguridad.

También es importante mencionar que los sistemas complejos son necesariamente modulares, ya que de otra manera no se podría manejar su complejidad. Pero el aumento de la modularidad significa

que la seguridad disminuye porque falla a menudo donde dos módulos se comunican.

La única manera razonable de probar la seguridad de un sistema es realizar evaluaciones de seguridad en él. Sin embargo, cuanto más complejo es el sistema, más dura se vuelve la evaluación de su seguridad. Un sistema más complejo tendrá más errores relacionados con la seguridad en su análisis, diseño y programación. Y desgraciadamente, el número de errores y la dificultad de evaluación no crecen de acuerdo con la complejidad, si no mucho más rápido. Cuanto más complejo es un sistema, más difícil es de entender. Hay toda clase de puntos de vulnerabilidad -interfase entre usuario y máquina, interacciones del sistema- esto crece exponencialmente cuando no se puede mantener el sistema completo en la cabeza. Cuanto más complejo es un sistema, más duro es hacer este tipo de análisis. Todo es más complicado: su análisis, su diseño, su programación y su uso.

Los sistemas operativos no escapan a esta realidad y se tornan cada vez más complejos. Un ejemplo es Microsoft Windows, que cuando se publicó en 1992 (Versión 3.1) tenía alrededor de 3 millones de líneas de código; Windows 95 alcanzó a los 15 millones y Windows 98 tiene 18 millones; Windows NT lanzado en 1992 tenía 4 millones de líneas de código; Windows NT 4.0 tiene 16.5 millones; Windows 2000 tiene entre 35 y 80 millones de líneas. Como punto de

comparación tenemos a Solaris que mantuvo su código fuente en aproximadamente 7 a 8 millones de líneas y Linux (Incluso con la suma del entorno gráfico X Windows y de Apache) que todavía se mantiene por debajo de los 5 millones de líneas.

En el pasado la seguridad física fue suficiente para resguardar un computadora contra ataques de intrusos, actualmente controles sofisticados deben instrumentarse para prevenir intentos de ingreso desde terminales remotas y sobre otras redes de comunicación.

Por último, cabe destacar que el nivel de seguridad apropiado para un sistema en particular depende del valor de los recursos que se aseguran.

Dentro de la protección de sistemas operativos se puede dividir en los siguientes puntos para tener una mejor división del trabajo a considerar.

Seguridad Interna y Externa: La seguridad interna está relacionada a los controles incorporados al hardware y al Sistema Operativo para asegurar los recursos del sistema. La seguridad externa está compuesta por la seguridad física y la seguridad operacional. La seguridad física incluye la protección contra desastres (como inundaciones, incendios, etc.) y protección contra intrusos.

Seguridad Operacional: La seguridad operacional consiste en varias políticas y procedimientos implementados por el administrador del sistema de computación. Mediante la autorización se determina qué acceso se permite y a qué entidad. Como punto crítico se destaca la selección del personal y la asignación del mismo. Generalmente se dividen responsabilidades, de esta manera un operario no debe conocer la totalidad del sistema para cumplir con esas responsabilidades. Se deben instrumentar diversos controles, y el personal debe saber de la existencia de dichos controles, pero desconocer cuáles son, para reducir la probabilidad de que intrusos puedan evadirlos.

Protección. Metas de la protección: Existen varios mecanismos que pueden usarse para asegurar los archivos, segmentos de memoria, CPU, y otros recursos administrados por el Sistema Operativo. Por ejemplo, el direccionamiento de memoria asegura que unos procesos puedan ejecutarse solo dentro de sus propios espacios de dirección. El timer asegura que los procesos no obtengan el control de la CPU en forma indefinida. La protección se refiere a los mecanismos para controlar el acceso de programas, procesos, o usuarios a los recursos definidos por un sistema de computación. Seguridad es la serie de problemas relativos a asegurar la integridad del sistema y sus datos. Hay importantes razones para proveer protección. La más obvia es la necesidad de prevenirse de violaciones intencionales de acceso por

un usuario. Otras de importancia son, la necesidad de asegurar que cada componente de un programa, use solo los recursos del sistema de acuerdo con las políticas fijadas para el uso de esos recursos. Un recurso desprotegido no puede defenderse contra el uso no autorizado o de un usuario incompetente. Los sistemas orientados a la protección proveen maneras de distinguir entre uso autorizado y desautorizado.

Mecanismos y Políticas: El rol de la protección es proveer un mecanismo para el fortalecimiento de las políticas que gobiernan el uso de recursos. Tales políticas se pueden establecer de varias maneras, algunas en el diseño del sistema y otras son formuladas por el administrador del sistema. Otras pueden ser definidas por los usuarios individuales para proteger sus propios archivos y programas. Las políticas son diversas, dependen de la aplicación y pueden estar sujetas a cambios a lo largo del tiempo. Un principio importante es la separación de políticas de los mecanismos. *'Los mecanismos determinan cómo algo se hará. Las políticas deciden que se hará'.* La separación es importante para la flexibilidad del sistema.

Vigilancia: La vigilancia se compone de la verificación y la auditoria del sistema, y la identificación de usuarios. En la vigilancia se utilizan sistemas muy sofisticados, a tal punto, que a veces pueden surgir problemas en la autentificación generando un rechazo al usuario legítimo.

Monitoreo de amenazas: Una manera de reducir los riesgos de seguridad es tener rutinas de control en el sistema operativo para permitir o no el acceso a un usuario. Estas rutinas interactúan con los programas de usuario y con los archivos del sistema. De esta manera, cuando un usuario desea realizar una operación con un archivo, las rutinas determinan si se niega o no el acceso y en caso de que el mismo fuera permitido devuelven los resultados del proceso. Además las rutinas de control permiten detectar los intentos de penetración al sistema y advertir en consecuencia.

Amplificación: Los programas de vigilancia interactúan con los programas de usuario y los archivos del sistema. A veces estos programas (los primeros) requieren de más derechos de acceso de los que posee el usuario para realizar una operación determinada. Esto se conoce como amplificación.

Protección por contraseña: Existen tres clases principalmente de elementos que permiten establecer la identidad de un usuario:

- Algo sobre las personas. Esto incluye huellas digitales, reconocimiento de voz, fotografía y firmas.
- Algo poseído por la persona. Esto incluye distintivos, tarjetas de identificación y llaves.

- Algo conocido por el usuario. Esto incluye contraseñas, combinación de cerraduras. El esquema de autentificación más común es la simple protección por contraseña. El usuario elige una palabra que se le viene a la memoria, y la escribe de inmediato para ganar admisión al sistema de computación.

Muchos sistemas no muestran la contraseña tal como ha sido ingresada (mostrar asteriscos en lugar de letras).

La protección por contraseña es un esquema débil. En el sentido de que los usuarios tienden a elegir contraseñas fáciles de recordar. Entonces alguien que conoce al usuario podría intentar ingresar al sistema usando nombres de gente que la persona conoce. Esto puede resultar en una violación de la seguridad por los intentos repetitivos de ingreso.

Algunos sistemas usan contraseñas cortas lo que facilita la conformación rápida de la lista de todas las posibles combinaciones. Los sistemas actuales utilizan contraseñas largas para frenar tales intentos de penetración.

Auditoria: La auditoria normalmente es realizada en sistemas manuales “después del hecho”. Los auditores son llamados periódicamente para examinar las transacciones recientes de una organización y para determinar si ha ocurrido actividad fraudulenta.

El registro de auditoria es un registro permanente de acontecimientos de importancia que ocurren en el sistema de computación. Se produce automáticamente cada vez que ocurren los eventos y es almacenado en un área protegida del sistema. Las auditorias periódicas prestan atención regularmente a problemas de seguridad; las auditorias al azar ayudan a detectar intrusos.

Fallas famosas de seguridad: Al igual que la industria del transporte tiene el Titanic y el Hindenburg, los expertos en seguridad de las computadoras tienen algo que quisieran poder olvidar. En esta sección se nombrarán algunos interesantes problemas de seguridad que surgieron en cuatro sistemas operativos: Unas, Multics, Tenex, y el OS/360.

La utilería LPR de Unix imprime un archivo en la impresora de líneas tiene una opción para eliminar el archivo después de ser impreso. En las primeras versiones de Unix, era posible que cualquiera utilizara LPR para imprimir, para que después el sistema eliminara el archivo con la contraseña.

Otra forma de penetrar a Unix era mediante el enlace de un archivo llamado *core* en el directorio de trabajo al archivo con la contraseña. El intruso forzaba entonces un vaciado de un programa SETUID, que escribía el sistema en el archivo core; es decir, en la parte superior del archivo con la contraseña. De esta forma, un usuario podía

reemplazar el archivo con contraseña por otro que contuviera unas cuantas cadenas de su elección (por Ej., argumentos de un comando).

Otro ligero error de Unix es relativo al comando *mkdir foo.*

Mkdir, que era un programa setuid poseído por la raíz, crea primero el nodo-i para el directorio foo con la llamada al sistema mknod y después cambia el propietario de foo de su identificación de usuario efectiva (es decir, desde la raíz) hasta su identificación real.cuando el sistema era lento, a veces era posible eliminar de manera rápida el nodo-i del directorio y hacer un enlace con el archivo de contraseña bajo el nombre foo después de ejecutar el mknod pero antes de *chown.* Cuando mkdir ejecutaba chown, hacía que el usuario fuera el propietario del archivo de contraseña. Si se colocaban los comandos necesarios en un guión del shell, podrían estar intentando una y otra vez hasta que funcionara el truco.

El problema de seguridad de Multics se relacionaba con el hecho de que los diseñadores de sistema siempre pensaron en Multics como un sistema de tiempo compartido, cuyas facilidades para el procesamiento por lotes surgieron como una idea tardía, para pacificar ciertas viejas intransigencias de los lotes. La seguridad en el tiempo compartido era excelente; la seguridad en los lotes era inexistente. Cualquier persona podía introducir un trabajo por lotes

que leyera un paquete de naipes en el directorio de cualquier otro usuario.

Para robar los archivos de alguien, lo único que había que obtener era una copia del código fuente de editor, modificarlo para robar archivos (pero que siguiera funcionando en forma perfecta como editor), compilarlo y leerlo dentro del directorio bin de la víctima. La siguiente vez que la víctima llamaba al editor, obtenía la versión del intruso, la cual editaba muy bien, pero que también robaba todos sus archivos. La idea e modificar un programa normal para que hiciera cosas adversas además de su función usual y arreglar las cosas para que la víctima utilizara la versión modificada se conoce ahora como el ataque del Caballo de Troya.

El sistema operativo Tenex era muy popular en las computadoras DEC-10. ya no se utiliza mucho pero su nombre quedará grabado para siempre en los anales de la seguridad de las computadoras, debido al siguiente error del diseño. Tenex permitía la paginación. Para que los usuarios supervisaran el comportamiento de sus programas, era posible indicar al sistema que llamara una función del usuario si ocurría un fallo de página.

Tenex también utilizaba contraseñas para la protección de archivos. Para tener acceso a un archivo, el programa debía presentar la contraseña adecuada. El sistema operativo verificaba la contraseña

carácter por carácter y se detenía en cuanto notara que la contraseña estuviera equivocada. Para penetrar a Tenex, un intruso colocaría una contraseña de la forma que se muestra en la figura siguiente, con el primer carácter al final de la página y el resto al principio de la siguiente.

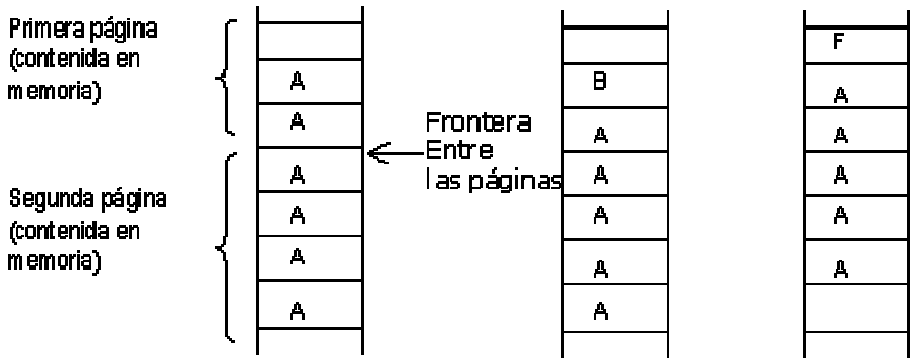

Figura 5-2 Violación de Seguridad en Tenex.

Fuente:
http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGSO200_archivos/fallas%20famosas.htm
Autor: Desconocido

El siguiente paso era garantizar la segunda página no estuviera en la memoria; por ejemplo, para hacer referencia a un número grande de páginas, con lo que seguramente la segunda página saldría de la memoria para tener espacio para las demás. Ahora, el programa intentaría abrir el archivo de la víctima, mediante la contraseña alineada con cuidado. Si el primer carácter de la verdadera contraseña era distinto de A, el sistema detenía la verificación en el primer carácter e informaría que la contraseña no era válida. Sin embargo, si la verdadera contraseña comenzaba con A, el sistema continuaba la lectura y obtenía un fallo de página, de la que era informado el intruso.

Si la contraseña no comenzaba con A, el intruso cambiaba la contraseña por la de la figura anterior y repetía todo el proceso para ver si comenzaba con B. Sólo requería 128 intentos recorrer todo el conjunto de caracteres ASCII y determinar así el primer carácter.

La última falla se refiere a OS/360. La descripción siguiente es un poco simplificada, pero contiene las esencias de las fallas. Este sistema, era posible iniciar una lectura de cinta y continuar después el cálculo mientras que la unidad de cinta transferiría datos al espacio el usuario. El truco aquí consistía en iniciar con cuidado una lectura de cinta y después hacer una llamada al sistema que necesitará una estructura de datos del usuario; por ejemplo, un archivo que leer y su contraseña.

El sistema operativo verificaba en primer lugar que la contraseña fuera la correcta para el archivo dado. Después regresaba y leía el nuevo nombre para el verdadero acceso. Por desgracia, justo antes de que el sistema buscara el nombre del archivo por segunda vez, se escribía encima el nombre del archivo en la unidad de cinta. El sistema leía entonces el nuevo archivo, para el cual no se presentaba contraseña alguna. La sincronización necesitó algo de práctica, pero eso no fue difícil. Además, si hay algo que hacen bien las computadoras, es repetir la misma operación una y otra vez.

Ataques genéricos a la seguridad: Las fallas descriptas arriba han sido arregladas, pero el común de los sistemas operativos continúa siendo tan permeable como una coladera. La forma usual de probar la seguridad de un sistema es contratar un grupo de expertos, conocidos como el equipo tigre o equipo de penetración, para ver si pueden penetrar en él. Algunos intentan lo mismo con estudiantes graduados. Durante varios años, estos equipos de penetración han descubierto varias áreas en las que es probable que los sistemas tengan debilidades.

Al diseñar sistemas se debe asegurar que pueden resistir ataques de estos tipos.

- Solicitar páginas de memoria, espacio en disco y solo léalas. Muchos sistemas no las borran antes de asignarlas y podrían contener interesante información escrita por el anterior usuario.
- Intentar llamadas al sistema inválidas, o bien llamadas válidas con parámetros inválidos, o incluso llamadas válidas con parámetros válidos pero no razonables. Muchos sistemas pueden ser confundidos con facilidad.
- Iniciar la conexión al sistema y oprima entonces DEL, RUBOUT, O BREAK a la mitad de la secuencia de acceso. En ciertos sistemas el programa de verificación de la contraseña quedará eliminado y se considerará un acceso exitoso.

- Intentar modificar las complejas estructuras del sistema operativo que está en el espacio del usuario. En muchos sistemas, para abrir un archivo, el programa construye una enorme estructura de datos, la cual contiene el nombre del archivo y muchos otros parámetros, lo cual se transfiere al sistema. Al leer o escribir en un archivo, el sistema puede actualizar a veces la propia estructura. La modificación de estos campos puede causar estragos a la seguridad.
- Engañar al usuario, escribiendo un programa que haga aparecer *"login"*: En la pantalla y que después desaparezca. Muchos usuarios irán hacia la terminal y le indicarán su nombre y su contraseña de acceso, la cual registrará el programa para su registro de contraseñas.
- Buscar manuales que digan "no lleve a cabo X" he intente tantas variaciones de X como sea posible.
- Convencer a un programador del sistema para que modifique el sistema, con el fin de evitar ciertas verificaciones vitales de seguridad para cualquier usuario con su nombre de acceso. Este ataque se conoce como una trampa (trap door).
- Si todo esto falla, el atacante debe encontrar a la secretaria del director del centro de cómputos y engañarla, o sobornarla. Es probable que la secretaria tenga un fácil acceso a todo tipo de información maravillosa y que por lo general se le pague poco. No hay que subestimar los problemas que puede causar el personal.

5.3 Antivirus y Firewall de Clientes.

Cuando software malintencionado llega a un equipo host, los sistemas de defensa se deben centrar en proteger el sistema y los datos que contiene, y detener la propagación de la infección. Estas defensas no son menos importantes que las físicas y de red del entorno. Se deben diseñar partiendo de la base de que el software malintencionado ha sido capaz de superar todas las anteriores capas de defensa para llegar al host. Este método es la mejor forma de lograr el máximo nivel de protección.

Pasos de la protección antivirus del cliente: Son varios los métodos y tecnologías que se pueden utilizar para las configuraciones antivirus de los clientes. A continuación se proporcionan detalles que Microsoft recomienda tener en cuenta:

- Paso 1 - Reducir el ámbito de ataque: Quitar o deshabilitar todas las aplicaciones o servicios innecesarios para minimizar las vías desde las que un atacante podría aprovechar.
- Paso 2 - Aplicar actualizaciones de seguridad.
- Paso 3 - Habilitar un servidor de seguridad basado en host: Estos servidores de seguridad filtran todos los datos que intentan entrar o salir de un determinado equipo host.

- Paso 4 - Instalar un software antivirus y mantenerlo actualizado.
- Paso 5 – Realizar pruebas con un escáner de vulnerabilidades: Una vez configurado un sistema, se debe comprobar periódicamente para garantizar que no aparecen puntos débiles para la seguridad. Para ayudarle con este proceso, determinadas aplicaciones actúan de escáner y buscan los puntos débiles de los que software malintencionado e intrusos se pueden intentar aprovechar. La mejor de estas herramientas actualiza las propias rutinas de exploración para defender el sistema contra los puntos débiles más recientes.
- Paso 6 - Usar directivas de privilegio mínimo: Otra zona de las defensas del cliente que no se debe pasar por alto es la de los privilegios asignados a los usuarios en funcionamiento normal. Microsoft recomienda adoptar una directiva que ofrezca la menor cantidad de privilegios posible, a fin de reducir al mínimo el impacto del software malintencionado que se aprovecha de los privilegios al ejecutarse. Dicha directiva resulta especialmente importante para los usuarios que suelen contar con privilegios administrativos locales. Contemple la posibilidad de quitar esos privilegios para las operaciones diarias y, en su lugar, utilizar el comando RunAs para abrir las herramientas de administración pertinentes cuando resulte necesario.

- Paso 7 - Restringir aplicaciones no autorizadas.

Configuración antivirus en aplicaciones clientes: A continuación se ofrecen directrices para configurar determinadas aplicaciones clientes que pueden ser objetivos del software malintencionado:

- Clientes de correo electrónico: Si el software malintencionado consigue traspasar las defensas antivirus en la red y el servidor de correo electrónico, deben existir algunos valores que se puedan configurar para ofrecer mayor protección al cliente de correo electrónico. Por lo general, la posibilidad que tiene un usuario de abrir directamente documentos adjuntos desde el mensaje de correo electrónico supone una de las principales formas de propagación de software malintencionado en el cliente. Si puede, limite esta capacidad en los sistemas de correo electrónico de la organización. De no ser posible, algunos clientes permiten configurar pasos adicionales que los usuarios deberán realizar antes de abrir un archivo adjunto. Por ejemplo, en Microsoft Outlook® y Outlook Express, se puede realizar lo siguiente:

  - Utilizar las zonas de seguridad de Internet Explorer para deshabilitar contenido activo en los mensajes de correo electrónico en HTML.

- Habilitar una opción de forma que los usuarios sólo puedan leer los mensajes en texto.
- Impedir que los programas envíen mensajes de correo electrónico sin especificar la aprobación del usuario.
- Bloquear archivos adjuntos en los mensajes no seguros.

- Aplicaciones de escritorio: A medida que las aplicaciones de escritorio se consolidan, también se convierten en objetivos del software malintencionado. Los virus de macro utilizan los archivos creados en el procesador de texto, hoja de cálculo u otra aplicación habilitada por macros para replicarse. Siempre que sea posible, debe tomar medidas que aseguren que se ha habilitado la configuración de seguridad más apropiada en todas las aplicaciones del entorno que trabajan con estos archivos.

- Aplicaciones de mensajería instantánea: El fenómeno de la mensajería instantánea ha servido para mejorar la comunicación entre usuarios de todo el mundo. Por desgracia, también tiene el potencial de permitir que software malintencionado se introduzca en el sistema. Aunque los mensajes de texto no presentan una amenaza directa de software malintencionado, la mayoría de los clientes de mensajería instantánea ofrecen la posibilidad de transferencia de archivos, para mejorar la comunicación entre los usuarios.

Al permitir esta transferencia, se presenta ante los posibles ataques por software malintencionado una ruta directa a la red de la organización. Los servidores seguros de la red pueden bloquear este tipo de transferencias de archivos con tan solo filtrar los puertos utilizados para esta comunicación. Por ejemplo, los clientes con Microsoft Windows y MSN® Messenger utilizan un intervalo de puertos TCP entre 6891 y 6900 para la transferencia de archivos, por lo que, si el servidor de seguridad perimetral los bloquea, no se producirá la transferencia de archivos a través de la mensajería instantánea. No obstante, los equipos cliente móviles sólo estarán protegidos mientras se encuentren en la red de la organización. Por todo ello, se aconseja configurar en los clientes el servidor de seguridad basado en host para bloquear estos puertos y proteger los clientes móviles de la organización cuando se encuentren fuera de las defensas de la red. Si la organización no puede bloquear estos puertos porque otras aplicaciones necesarias los requieren o porque la transferencia de archivos es precisa, se debe asegurar de que se comprueban todos los archivos antes la transferencia. Si las estaciones de trabajo clientes no utilizan un escáner antivirus en tiempo real, hay que configurar la aplicación de mensajería instantánea para que los archivos transferidos pasen automáticamente a una aplicación antivirus y se comprueben en cuanto se hayan recibido. Por ejemplo, se

puede configurar MSN Messenger para comprobar automáticamente los archivos transferidos.

- Exploradores Web: Antes de descargar o ejecutar código de Internet, se debe asegurar de que proviene de una fuente conocida y confiable. Los usuarios no pueden confiar únicamente en el aspecto o la dirección del sitio, ya que tanto las páginas como las direcciones Web se pueden falsificar. Existen varias técnicas y tecnologías que se han desarrollado para ayudar al explorador Web de un usuario a determinar la confiabilidad del sitio Web que se están examinando. Por ejemplo, Microsoft Internet Explorer utiliza la tecnología Microsoft Authenticode® para comprobar la identidad del código descargado. Esta tecnología comprueba que el código tiene un certificado válido, que la identidad del fabricante de software coincide con el certificado y que éste sigue siendo válido. Si se superan todas estas pruebas, se habrán reducido las posibilidades de que un atacante haya introducido código malintencionado en el sistema. La mayoría de los principales exploradores Web admiten la posibilidad de restringir el nivel de acceso automatizado disponible a código que se ejecuta desde un servidor Web. Internet Explorer utiliza zonas de seguridad que reducen la posibilidad de que el contenido Web realice operaciones que puedan resultar perjudiciales en el cliente. Estas zonas de seguridad se basan en la ubicación

(zona) del contenido Web. Por ejemplo, si confía en que todo lo descargado en la intranet de la organización es seguro, puede establecer la configuración de seguridad de los clientes de la zona de intranet en un nivel bajo, de modo que los usuarios puedan descargar contenido de la intranet con pocas o ninguna restricción. Sin embargo, si el origen de la descarga se encuentra en la zona de Internet o la zona de los sitios restringidos, debe establecer la configuración de seguridad del cliente en un nivel medio o elevado. De este modo, los exploradores de los clientes enviarán al usuario información sobre el certificado del contenido antes de descargarlo o impedirán que lo haga.

- Aplicaciones de igual a igual: Con la llegada de las aplicaciones de igual a igual (P2P) para Internet, se ha favorecido más que nunca la tarea de encontrar e intercambiar archivos con otros individuos. Por desgracia, esta situación ha llevado a diversos ataques de software malintencionado que intentan utilizar dichas aplicaciones para replicar archivos en los equipos de otros usuarios. Gusanos como W32.HLLW.Sanker han apuntado a las aplicaciones P2P, como Kazaa, para realizar las mencionadas replicaciones. Existen muchos más ejemplos de software malintencionado que intentan utilizar esas aplicaciones, como Morpheus y Grokster. Los problemas de seguridad que afectan a las

aplicaciones P2P tienen poco que ver con los propios programas clientes. Por el contrario, guardan más relación con la capacidad de estas aplicaciones de ofrecer rutas directas de un equipo a otro a través de las cuales se puede transmitir contenido sin las comprobaciones de seguridad correspondientes. En la medida de lo posible, se recomienda limitar la cantidad de clientes de la organización que utilizan estas aplicaciones. Se pueden utilizar las directivas de restricción de software tratadas anteriormente en este capítulo para bloquear a los usuarios que las utilicen. De no ser posible en el entorno, asegúrese de que las directivas antivirus tienen en cuenta que los clientes corren mayor riesgo debido a estas aplicaciones.

Beneficios de soluciones firewall en clientes: Los firewalls cliente proveen de una capa extra de protección que ayuda a alcanzar los objetivos corporativos para la seguridad al nivel del cliente. Los siguientes beneficios están disponibles para aquellas compañías que implementan una solución firewall en los clientes:

- Esfuerzos de seguridad incrementados: La protección de muro de fuego en el cliente incrementa el manejo de seguridad habilitando a los administradores poder definir y cerrar reglas sobre máquinas cliente individualmente. Esto previene que los usuarios finales esquiven las medidas de seguridad.

- Protección interna contra amenazas: Los muros de fuego de perímetro carecen de la habilidad de establecer políticas de comunicación por aplicación, y son por ello incapaces de identificar intentos inesperados de comunicación en red sobre una máquina cliente. Usando un firewall cliente, estos intentos de comunicación de red inesperados pueden ser identificados y bloqueados. Esto detiene amenazas desde adentro evitando que se esparzan en la red corporativa y minimiza su impacto en las operaciones del negocio, incluso si la amenaza es capaz de penetrar el muro de fuego de perímetro.

- Protección contra aplicaciones oportunistas: La habilidad de las soluciones de muro de fuego clientes para establecer y reforzar las políticas de comunicación sobre una base por aplicación permite a los administradores de red prohibir que aplicaciones desconocidas e indeseables como “spyware” o “scumware” se comuniquen, haciéndolas inefectivas. Esto provee una herramienta que ayuda a mantener la productividad del staff y previene uso no autorizado de los equipos y de recursos del ancho de banda.

La necesidad por un firewall cliente: Las empresas despliegan muros de fuego de perímetro para controlar el tráfico entrando y saliendo de una red corporativa, por ello proveen de una primera línea de defensa contra ataques externos. Algunos administradores de

seguridad creen que el muro de fuego de perímetro provee suficiente protección para toda la empresa.

Sin embargo, los ambientes de redes abiertas de hoy en día son un riesgo para ataques no identificados y maliciosamente perpetrados por aquellos con acceso legítimo a la red (por ejemplo, trabajadores remotos y consultores). De hecho, de acuerdo al informe de Seguridad y Crimen Computacional del 2002 de la CSI[46]/FBI[47], una tercera parte de los ataques de red son perpetrados por aquellos dentro del firewall tradicional[48]. Una de las principales razones por las que los muros de fuego de perímetro no ofrecen suficiente protección es que el perímetro de red ya no se puede diferenciar tan fácilmente. Además, como un firewall de perímetro sirve como un punto único de entrada y salida para todo el tráfico de la red, su administración apropiada es crítica para asegurar su correcto funcionamiento, el personal de seguridad está saturado con peticiones y requerimientos diarios, lo que resulta en configuraciones y administración pasadas por alto o no priorizadas, así que estos asuntos administrativos

---

[46] **Computer Security Institute** (CSI) es una organización mundial dedicada específicamente a servir y entrenar a los profesionales de la seguridad tanto de información, computación y redes. Desde 1974, la CSI ha estado proveyendo de educación y avocando agresivamente la importancia de proteger los activos de la información.

[47] **FBI** (Federal Bureau of Investigations), Oficina Federal de Investigaciones, cuyos objetivos primarios son proteger a los Estados Unidos de ataques terroristas, de operaciones de inteligencia extranjera, y de cyberataques y de alta tecnología. http://www.fbi.gov/aboutus.htm

[48] Richard Power, Computer Security Institute, "Computer Security Issues and Trends," 2002 CSI/FBI Computer Crime and Security Survey. http://www.gocsi.com/forms/fbi/pdf.html.

pueden costar mucho para aquellas compañías que asignan la seguridad de factores de mucho valor a los firewall de perímetro.

De acuerdo a Mark Bouchard, Director Senior del Programa de Seguridad, para META Group[49], los muros de fuego personales son esenciales para la seguridad para controlar muchos accesos remotos e implementaciones de telecomunicación, particularmente cuando información crítica está siendo almacenada en un computador remoto. También pueden ser útiles dentro de la empresa, ayudando a asegurarse contra amenazas internas y proveer contención cuando un ataque ocurre.

5.4 Detección de Intrusos en Clientes.

Un sistema de detección de intrusos basado en máquina (*host-based IDS*) es un mecanismo que permite detectar ataques o intrusiones contra la máquina sobre la que se ejecuta.

Tradicionalmente, los modelos de detección basados en máquina han consistido por una parte en la utilización de herramientas automáticas de análisis de *logs* generados por diferentes aplicaciones o por el propio *kernel* del sistema operativo, prestando siempre especial atención a los registros relativos a demonios de red, como un servidor *web* o el propio inetd, y por otra - quizás no tan habitual como la

[49] **META Group** es un proveedor de investigación de tecnología de la información, servicios consultivos, y consultoría estratégica. http://www.metagroup.com/corporate/index.htm

anterior - en el uso de verificadores de integridad de determinados ficheros vitales para el sistema, como el de contraseñas; no obstante, desde hace unos años un tercer esquema de detección se está implantando con cierta fuerza: se trata de los sistemas de detección, *honeypots* o tarros de miel.

El análisis de *logs* generados por el sistema (entendiendo como tales no sólo a los relativos al núcleo, sino también a aquellos de aplicaciones nativas de cada Unix, como syslogd) varía entre diferentes clones de Unix por una sencilla razón: cada uno de ellos guarda la información con un cierto formato, y en determinados ficheros, aunque todos - o casi todos - sean capaces de registrar los mismos datos, que son aquellos que pueden ser indicativos de un ataque. La mayor parte de las versiones Unix son capaces de registrar con una granularidad lo suficientemente fina casi todas las actividades que se llevan a cabo en el sistema, en especial aquellas que pueden suponer - aunque sea remotamente - una vulneración de su seguridad; sin embargo, el problema radica en que pocos administradores se preocupan de revisar con un mínimo de atención esos *logs*, por lo que muchos ataques contra la máquina, tanto externos como internos, y tanto fallidos como exitosos, pasan finalmente desapercibidos. Aquí es donde entran en juego las herramientas automáticas de análisis, como swatch o logcheck; a grandes rasgos, realizan la misma actividad que podría ejecutar un *shellscript* convenientemente planificado que

incluyera entre sus líneas algunos grep de registros sospechosos en los archivos de *log*.

A qué entradas de estos ficheros se debe estar atentos? Evidentemente, esto depende de cada sistema y de lo que sea `normal' en él, aunque suelen existir registros que en cualquier máquina denotan una actividad cuanto menos sospechosa. Esto incluye ejecuciones fallidas o exitosas de la orden su, peticiones no habituales al servicio SMTP (como vrfy o expn), conexiones a diferentes puertos rechazadas por *TCP Wrappers*, intentos de acceso remotos como súper usuario, etc.; si en la propia máquina hay instalado un cortafuegos independiente del corporativo, o cualquier otro *software* de seguridad - uno que quizás es especialmente recomendable es *PortSentry* -, también conviene atender los *logs* generados por los mismos, que habitualmente se registran en los ficheros normales de auditoría del sistema (syslog, messages...) y que suelen contener información que con una probabilidad elevada denotan un ataque real.

Por otra parte, la verificación de integridad de archivos se puede realizar a diferentes niveles, cada uno de los cuales ofrece un mayor o menor grado de seguridad. Por ejemplo, un administrador puede programar y planificar un sencillo *shellscript* para que se ejecute periódicamente y compruebe el propietario y el tamaño de ciertos ficheros como /etc/passwd o /etc/shadow; evidentemente, este esquema es

extremadamente débil, ya que si un usuario se limita a cambiar en el archivo correspondiente su UID de 100 a 000, este modelo no descubriría el ataque a pesar de su gravedad. Por tanto, parece obvio que se necesita un esquema de detección mucho más robusto, que compruebe aparte de la integridad de la información registrada en el inodo asociado a cada fichero (fecha de última modificación, propietario, grupo propietario...) la integridad de la información contenida en dicho archivo; y esto se consigue muy fácilmente utilizando funciones resumen sobre cada uno de los ficheros a monitorizar, funciones capaces de generar un *hash* único para cada contenido de los archivos. De esta forma, cualquier modificación en su contenido generará un resumen diferente, que al ser comparado con el original dará la voz de alarma; esta es la forma de trabajar de *Tripwire*, el más conocido y utilizado de todos los verificadores de integridad disponibles para entornos Unix.

Sea cual sea el modelo de verificación, en cualquiera de ellos se debe llevar a cabo inicialmente un paso común: generar una base de datos de referencia contra la que posteriormente se comparará la información de cada archivo. Por ejemplo, si alguien se limita a comprobar el tamaño de ciertos ficheros se debe, nada más configurar el sistema, registrar todos los nombres y tamaños de los ficheros deseados, para después comparar la información que periódicamente registraremos en nuestra máquina con la que hemos almacenado en dicha base de datos; si existen diferencias, podemos

encontrarnos ante un indicio de ataque. Lo mismo sucederá si registramos funciones resumen: se debe generar un *hash* inicial de cada archivo contra el que comparar después la información obtenida en la máquina.

Independientemente de los contenidos que deseemos registrar en esa base de datos inicial, siempre se debe tener presente una cosa: si un pirata consigue modificarla de forma no autorizada, habrá burlado por completo al sistema de verificación. Así, es vital mantener su integridad; incluso es recomendable utilizar medios de sólo lectura, como un CD-ROM, o incluso unidades extraíbles - discos o disquetes - que habitualmente no estarán disponibles en el sistema, y sólo se utilizarán cuando tengamos que comprobar la integridad de los archivos de la máquina.

Por último, aunque su utilización no esté tan extendida como la de los analizadores de *logs* o la de los verificadores de integridad, es necesario hablar, dentro de la categoría de los sistemas de detección de intrusos basados en máquina, de los sistemas de decepción o *honeypots*. Básicamente, estos `tarros de miel' son sistemas completos o parte de los mismos (aplicaciones, servicios, subentornos...) diseñados para recibir ciertos tipos de ataques; cuando sufren uno, los *honeypots* detectan la actividad hostil y aplican una estrategia de respuesta. Dicha estrategia puede consistir desde un simple correo electrónico al responsable de la seguridad de

la máquina hasta un bloqueo automático de la dirección atacante; incluso muchos de los sistemas - la mayoría - son capaces de simular vulnerabilidades conocidas de forma que el atacante piense que ha tenido éxito y prosiga con su actividad, mientras es monitorizado por el detector de intrusos.

El concepto teórico que está detrás de los tarros de miel es el denominado `conocimiento negativo': proporcionar deliberadamente a un intruso información falsa - pero que él considerará real - de nuestros sistemas, con diferentes fines: desde poder monitorizar durante más tiempo sus actividades, hasta despistarlo, pasando por supuesto por la simple diversión.

Evidentemente, para lograr engañar a un pirata medianamente experimentado la simulación de vulnerabilidades ha de ser muy `real', dedicando a tal efecto incluso sistemas completos (denominados 'máquinas de sacrificio'), pero con atacantes de nivel medio o bajo dicho engaño es muchísimo más sencillo: en muchos casos basta simular la existencia de un troyano como *BackOrifice* mediante *FakeBO* (http://cvs.linux.hr/fakebo/) para que el pirata determine que realmente estamos infectados e intente utilizar ese camino en su ataque contra el sistema.

Algunos sistemas de decepción no simulan vulnerabilidades tal y como habitualmente se suele entender este concepto; dicho de otra

forma, su objetivo no es engañar a un atacante durante mucho tiempo, proporcionándole un subentorno en el sistema que aparente de forma muy realista ser algo vulnerable, sino que su `decepción' es bastante más elemental: se limitan a presentar un aspecto (quizás deberíamos llamarle *interfaz*) que parece vulnerable, pero que cualquier aprendiz de pirata puede descubrir sin problemas que no es más que un engaño.

¿Dónde está entonces la función de estos sistemas, denominados en ocasiones `detectores de pruebas'? Por norma, su tarea es recopilar información del atacante y del ataque en sí; por ejemplo, un programa que se encargue de escuchar en el puerto 31337 de un sistema - donde lo hace el troyano real - y, cada vez que alguien acceda a él, guardar la hora, la dirección origen, y los datos enviados por el atacante. En realidad, no es una simulación que pueda engañar ni al pirata más novato, pero se ha logrado el objetivo de cualquier sistema de detección de intrusos: registrar el ataque; además, también se puede considerar un *honeypot*, ya que simula un entorno vulnerable, aunque sólo logre engañar al atacante durante unos segundos. De cualquier forma, es necesario indicar que en algunas publicaciones se diferencia a los sistemas de decepción ‘completos' de estos mecanismos de engaño simple, aunque a todos se les englobe dentro del conjunto denominado *honeypots*.

Estas son las ideas más generales de los sistemas de detección de

intrusos basados en *host*; aunque hoy en día, los basados en red, son con diferencia los más utilizados, todos son igualmente necesarios si se desea crear un esquema de detección con la máxima efectividad. Se trata de niveles de protección diferentes pero que tienen un mismo objetivo: alertar de actividades sospechosas y, en algunos casos, proporcionar una respuesta automática a las mismas.

5.5 Políticas de Seguridad.

Las Políticas de Seguridad es un conjunto de normas que se deben seguir, propuestas por los Administradores y Responsables de la Seguridad de nuestra Autoridad de Certificación, con el objetivo de estandarizar y disminuir al mínimo los riesgos de sufrir perturbaciones en los sistemas y servicios ofrecidos.

5.5.1 Administración de Usuarios y Roles.

Tanto el componente interno como externo de la Autoridad de Certificación requieren autentificación, creando la necesidad del control de usuarios mediante roles.

Existen tres roles:

Rol Usuario General: Se refiere a los clientes en general, quienes usan los servicios mediante previa autentificación.

Rol Verificador de Datos: Estos usuarios son los encargados de verificar los datos ingresados por los clientes, manteniendo una bandeja de trabajo asignado en relación a las solicitudes de Certificados hechas por los Clientes,

Rol Verificador de Pagos: Estos usuarios están encargados de revisar fondos de las tarjetas de crédito, o de las formas de pago dispuestas por la empresa, y aprobar las órdenes para que queden listas para la emisión de Certificados Digitales.

Todos estos roles se implementan mediante autentificación de usuario y clave, y las políticas a cumplir son las siguientes:

- El usuario debe contener máximo 16 caracteres alfanuméricos.
- La clave debe contener máximo 16 caracteres alfanuméricos.
- La clave debe estar codificada en la base de datos.
- La clave del Usuario General debe ser cambiada frecuentemente, no es responsabilidad nuestra hacerlo, el cliente debe cambiarla a su criterio, pero le sugerimos que lo haga una vez cada tres meses.
- La clave del Rol Verificador de Datos y de Pago debe ser cambiada cada tres meses.

5.5.2 Respaldos.

Los respaldos permiten mantener una reserva de la Información delicada, tanto de la Autoridad de Certificación, como de la Información de nuestros clientes. Es por esto que se tomarán las siguientes consideraciones con respecto a los respaldos:

- Se debe crear un procedimiento de recuperación en caso de algún percance. Este procedimiento debe cubrir todo lo necesario para poner lo más rápido posible en función los servicios más críticos de la Autoridad de Certificación.
- El procedimiento de recuperación debe ser actualizado frecuentemente, modificando versiones de las herramientas a utilizar, y optimizando los procesos involucrados.
- Cada Autoridad de Certificación, sucursal y Autoridad de Registro, como los Centros de Contingencias deben mantener sus respaldos.
- Los respaldos deben realizarse todos los días a la media noche.
- Se debe verificar obligatoriamente la integridad de los respaldos.

- Los respaldos deben ser guardados en un lugar seguro, con llave. La llave la debe tener el Administrador de Seguridad, con una copia en Gerencia General.
- Una vez cumplido el tiempo de vida de un respaldo, se lo debe desechar. Previo al desecho, se debe borrar su contenido.

5.5.3 Actualización de Software.

El software es una de las partes más importantes de los sistemas computacionales. Y cada día se descubren errores y nacen nuevas versiones, actualizaciones o parches que solucionan alguna falla de seguridad o mejoran el rendimiento del software. Aquí nace la obligación de actualizar el software, para lo cual se deben cumplir las siguientes políticas:

- Los servicios críticos deben mantenerse con las últimas versiones y parches. Dentro de los servicios críticos se encuentran: SendMail, OpenSSL, OpenLDAP, Iptables, Snort y Httpd.
- Las versiones de La Base de Datos y el Servidor de Aplicaciones debe ser actualizado en relación al costo. Pero lo esencial es mantener actualizado estos servicios con los parches respectivos, publicados en el sitio del proveedor.

- El software en las máquinas de los usuarios internos de igual manera debe ser actualizado y/o parchado con las últimas versiones.

5.5.4 Claves de Acceso.

Las claves son un aspecto muy importante en la Seguridad de la Información. Son la protección de las cuentas de usuario en los sistemas. Las siguientes políticas tienen como propósito establecer un estándar para la creación de claves robustas, protección de dichas claves y frecuencias de cambio:

- Todas las claves a nivel de sistema, como la del usuario "root"[50] de Linux, o del comando "enable"[51] de los ruteadores y switches, deben cambiarse cada tres meses.
- Todas las claves a nivel de usuario, como la del servidor email, servidor de aplicaciones o usuario de base de datos, deben ser cambiadas cada seis meses.
- Las claves no deben estar insertados en código fuente ni en mensajes de correo electrónico ni en ningún medio electrónico de comunicación, a menos de que este codificado.

[50] root: Usuario administrador en los sistemas Unix y Linux.
[51] Enable: commando de los equipos de redes que permite ingresar al modo para configurar funcionalidades.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Esta sección describe las conclusiones planteadas a partir de los conocimientos obtenidos durante el desarrollo del proyecto y sus respectivas pruebas.

Con respecto a la implementación de la red se puede mencionar la siguiente:

No existe una red completamente segura. La Seguridad de la Información se encarga de reducir al mínimo posible los riesgos de que los servicios básicos de un negocio no estén disponibles o mal funcionen, esto junto con el objetivo de mantener costos en una escala aceptable para el medio, forman el enfoque principal bajo el cual se ha propuesto el esquema de red diseñado, cumpliéndose con el objetivo planteado.

Con respecto a la aplicación se pueden enumerar las siguientes:

Existen librerías de acceso gratuito (por ejemplo openSSL, usada en este proyecto) bajo las cuales se puede desarrollar toda una infraestructura PKI, o desarrollar una aplicación muy segura.

La separación entre la aplicación que accede a Internet y la que se utiliza para operaciones internas facilita el control de información y reduce los riesgos de que los datos sensibles puedan ser accedidos directamente desde el exterior. Además que la arquitectura utilizada, mediante la implementación de patrones de diseño, permite la reutilización de los componentes básicos del negocio tales como las que se refieren al acceso a las tablas o la generación de certificados. Asimismo facilita el mantenimiento ya que dichas interfaces pueden ser modificadas sin necesidad de modificar el código de los componentes reutilizables antes mencionados, manteniendo así la seguridad en nuestros sistemas.

El planteamiento de las funciones de monitoreo y administración de las órdenes y/o solicitudes de los clientes permite una mayor eficacia en cuanto a la rapidez en lo que a servicio al cliente se refiere.

El planteamiento de usuarios internos y roles relacionados con éstos permite alcanzar el objetivo de limitar el campo de acción dentro de la empresa a las funciones específicas para el personal reduciendo las los riesgos de errores o eventos maliciosos casa adentro.

El costo de implementar una Autoridad de Certificación según el esquema propuesto es factible en el medio, la limitante viene dada por las referencias de escalas internaciones que una empresa de este tipo necesita para ganarse la confianza de los clientes.

Se debe llegar a tener ley y reglamento de comercio electrónico bien elaborados para que sustenten las operaciones de negocios mediante el Internet proveyendo la seguridad a los usuarios de que se estará protegido ante cualquier violación o amenaza que se presente.

Ante la inevitable introducción del país en la globalización, las empresas ecuatorianas necesitarán implementar esquemas seguros para realizar transacciones con otras alrededor del mundo, necesitándose entonces de al menos una autoridad de registro local avalada por una autoridad de certificación confiable reconocida mundialmente para proporcionar los elementos principales para la infraestructura básica para estas operaciones. Todo esto conlleva finalmente a la culturización general acerca de la importancia del tema.

Recomendaciones

Incentivar el desarrollo de proyectos, tesis y tópicos de graduación acerca de la Seguridad de Información y la utilización de software de código abierto (Open Source) para denotar la importancia del tema y así mejorar la calidad de los sistemas computacionales en el medio.

Motivar a la comunidad del país en temas de Seguridad, por ejemplo, la ESPOL podría crear un Laboratorio de Seguridad, en el cual se implementen configuraciones de redes seguras y en la cual se puedan realizar pruebas de penetración.

Glosario

Applet: A J2EE component that typically executes in a Web browser but can execute in a variety of other applications or devices that support the applet programming model.

Application client: Componente J2EE del cliente que se ejecuta en su propia Máquina Virtual. Los Application clients tienen acceso para algunas APIs de la Plataforma J2EE.

Archivo EAR: Enterprise Archive file. Es un JAR que contiene una aplicación Empresarial J2EE.

Archivo WAR: Web Archive file. Es un JAR que contiene una aplicación Web J2EE.

Archivo JAR: Java Archive file. Es un archivo comprimido que contiene un conjunto de archivos y clases en Java.

Ataque: Intento de traspasar el control de seguridad de un sistema

Autentificación: Es el proceso de verificar la identidad de un usuario, equipo u otra entidad en un sistema computacional, usualmente como un pre requisito para permitir acceso para recursos en un sistema.

Autoridad de Certificación: Es una empresas totalmente confiables que se dedican al negocio de generar y administrar Certificados Digitales

Autoridad de Registro: Es una entidad de la Autoridad de Certificación que actúa como avaladoras de los usuarios que solicitan un Certificado Digital, encargándose también de tramitar los mismos.

Autorización: Es el proceso por el cual el acceso para un método o recurso es determinado. La autorización depende sobre la determinación si el principal asociado con un requerimiento a través de la autentificación esta en un rol de seguridad dado.

Backdoor: Es un programa depositado por los intrusos para posteriormente poder acceder al sistema sin dejar huella alguna sobre los controles de seguridad.

Browser: Navegador de Internet.

Buffer Overflow: En términos simples, es cuando un programa (generalmente un servidor o “demonio”) recibe una entrada mayor a la que espera, sobrescribiendo, por tanto, áreas críticas de memoria. Esto genera que se ejecute cierto código, generalmente proporcionando al usuario acceso al sistema como root.

Bug: Una propiedad del software o hardware que causa un mal funcionamiento.

Caballo de Troya: Programa aparentemente útil el cual contiene código adicional escondido.

Certificado Digital: Es un documento electrónico que contiene datos identificadores de una persona o entidad (empresa, servidor web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Autoridad Certificadora.

Clave Privada: En la Criptografía con Claves Privada/Pública es la clave que solo el emisor del mensaje conoce para cifrar o descifrar un mensaje.

Clave Pública: En la Criptografía con Claves Privada/Pública es la clave que todos conocen para cifrar o descifrar un mensaje.

Contenedor: Es una entidad que provee administración del ciclo de vida, seguridad, despliegue y servicios de tiempo de ejecución a los componentes J2EE. Cada tipo de contenedor (EJB, Web, Appet y Aplication Client) también provee servicios de componentes específicos.

Contenedor Applet: Un contenedor que incluye soporte para el modelo de programación de applet.

Contenedor Application client: Un contenedor que soporta los componentes Application Client.

Contenedor EJB: Es un contenedor que implementa los componentes EJB, en base a la arquitectura J2EE. La arquitectura especifica un ambiente de tiempo de ejecución para Enterprise Beans que incluye seguridad, concurrencia, ciclo de vida, administración, transacciones, despliegue, nombramiento y otros servicios. Un Contenedor EJB es provisto por un Servidor EJB o J2EE.

Context root: Es el nombre que da el mapeo al documento principal de una aplicación web (o Document Root).

Cracker: Similar al anterior pero con la diferencia que éste saca algún tipo de beneficio del sistema que explora.

Credenciales: Es la información que describe los atributos de seguridad de un Principal.

Criptografía con Claves Privada/Pública: Ver Public Key Cryptography.

CRL: Certificate Revocation List. Es una lista de Certificados Digitales que han sido revocados por la Autoridad de Certificación.

CSS: Cascading style sheet. Es una hoja de estilos usada por los documentos HTML y XML para añadir un estilo a todos los elementos en un tag particular.

CSR: Certificate Signing Request. Es un certificado sin firma que se utiliza para petición a una Autoridad de Certificación, la cual la firma con su clave privada. Una vez que el CSR es firmado, se convierte en un Certificado Digital.

Deployment: Es el proceso mediante el cual el software es instalado en dentro de un ambiente operacional.

Deployment descriptor: Es un archivo XML proveído por cada módulo y aplicación J2EE que describe como ellos van a ser desplegados.

DES: Data Encryption Standard. Algoritmo de Encriptación Simétrico, que utiliza la misma clave tanto para encriptar como para desencriptar.

Denial of Service (DoS): Acciones que impiden a cualquier SIA funcionar de acuerdo con su propósito.

Detección de Intrusos: Sistemas que conglomeran un conjunto de técnicas cuyo propósito es detectar las intrusiones en una computadora o un sistema.

Document Object Model (DOM): Es un API para acceder y manipular Documentos XML como estructura tipo árbol. DOM provee interfaces de

plataforma y lenguaje neutral que habilita a los programas y scripts acceso dinámico y modificación de contenido y estructura en documentos XML.

Document root: O Documento principal. Es el directorio de más alto nivel de una aplicación Web. Document root es donde son almacenados las páginas JSP, clases y archivos de lado del cliente, y recursos Web estáticos.

EJB Home object: Es un objeto que provee operaciones del ciclo de vida(crear, remover, encontrar) para un Enterprise Bean. La clase para un EJB Home object es generada por las herramientas de despliegue del contenedor. El EJB Home object implementa la interface Home del Enterprise Bean. El cliente hace una referencia al EJB Home object para realizar operaciones del ciclo de vida sobre el objeto EJB. El cliente utiliza JNDI para localizar un EJB Home object.

EJB Object: Es un objeto cuya clase implementa la Interface remota del Enterprise Bean. Un cliente nunca referencia una instancia de un Enterprise Bean directamente, el cliente siempre referencia un EJB Object. La clase de un EJB Object es generada por las herramientas de despliegue del contenedor.

Enterprise Bean: Es un componente J2EE que implementa una tarea del negocio o una entidad del negocio y esta hospedada por un contenedor EJB, puede ser un Entity Bean, un Session Bean, o un Message-Driven Bean.

Enterprise JavaBeans (EJB): Es una arquitectura de componentes para el desarrollo y despliegue de aplicaciones orientada a objetos, distribuidas y de nivel empresarial. Las aplicaciones escritas usando la arquitectura Enterprise JavaBeans son escalables, transaccionales y seguras.

Entity Bean: Es un Enterprise Bean que representa la persistencia de datos mantenidos en una base de datos. Un Entity Bean puede manejar su propia persistencia o puede delegar esta función a su contenedor. Un Entity Bean es identificado por una clave primaria. Si el contenedor en el cual el Entity Bean es hospedado se cae, el Entity Bean, su clave primaria, y cualquier referencia remota sobrevive a la caída.

Firewall: Un sistema de combinaciones de sistemas que fija los límites entre dos o más redes y restringe la entrada y salida de la información.

Finder Method: Es un método definido en la Interface Home y es invocado por un cliente para localizer un Entity Bean.

Hacker: Una persona que disfruta explorando los detalles de las computadoras y de cómo extender sus capacidades.

Home Interface: Una o dos interfaces para un Enterprise Bean. La interface Home define cero o muchos métodos para manejar un Enterprise Bean. La Interface Home de un Session Bean define los métodos create y remove, así

como la Interface Home de un Entity Bean define los métodos create, Finder, y remove.

HTML: Hypertext Markup Language. Lenguaje para escribir documentos en Internet. HTML habilita el uso embebido de imágenes, sonido, flujo de video, formularios, referencias de URLs y formateo básico de texto.

HTTP: Hypertext Transfer Protocol. Es el Protocolo de Internet usado para obtener objetos hipertexto desde hosts remotos. Los mensajes HTTP consisten en requerimientos desde clientes hacia servidores y respuesta desde servidores a clientes.

HTTPS: HTTP sobre el protocolo SSL.

Intrusión: Conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso.

Intruso: Aquella persona que con una variedad de acciones intenta comprometer un recurso, puede ser por hardware o software.

JAR: Java archive. Es un formato de archivo independiente de la plataforma que permite que muchos archivos sean agregados en un solo archivo.

Java 2 Platform, Enterprise Edition (J2EE): Es un ambiente para desarrollo y despliegue de aplicaciones empresariales. La plataforma J2EE consiste de un

conjunto de servicios, APIs y protocolos que proveen la funcionalidad para desarrollo multi hilo y aplicaciones basadas en Web.

Java Naming and Directory Interface (JNDI): Es una API que prove funcionalidad de nombramiento y directorios.

JavaBeans component: Es una clase de Java que puede ser manipulada por herramientas y compuesto dentro de aplicaciones. Un componente JavaBeans puede adherirse a ciertas propiedades y eventos de las convenciones de la Interface.

JavaMail: Es un API para envío y recepción de email.

JavaServer Pages (JSP): Es una tecnología Web extensible que usa datos estáticos, elementos JSP, y objetos Java de lado del servidor para generar contenido dinámico en clientes. Típicamente los datos estáticos son elementos HTML o XML, y en muchos casos el cliente es un Navegador o Browser.

JavaServer Pages Standard Tag Library (JSTL): A tag library that encapsulates core functionality common to many JSP applications. JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization and locale-specific formatting tags, SQL tags, and functions.

JDBC: Es un API para conectividad, independiente de la base de datos, entre la plataforma J2EE y un amplio rango de Data Sources (Fuente de Datos).

JSP tag library: Es una colección de tags personalizados descritos mediante un descriptor tag library y clases Java.

Life Cycle (J2EE component): Ciclo de Vida. Se refiere a la existencia de los components J2EE del Framework. Cada tipo de componente tiene definido eventos que marcan su transición dentro de estados en los cuales tiene disponibilidad variable para uso.

Lógica del Negocio: Es el código que implementa la funcionalidad de una aplicación.

Método del Negocio: Es un método que implementa una lógica del negocio o las reglas de la aplicación.

Modo Promiscuo: Normalmente interfaz ethernet que permite leer toda la información sin importar su destino, aplicable a un segmento de red.

Persistenacia manejada por el bean: Es el mecanismo por el que la transferencia de datos entre las variables y el manejador de recursos de un Entity Bean es manejado por el Entity Bean.

Persistencia manejada por el Contenedor: Es el mecanismo mediante el cual la transferencia de datos entre las variables y el manejador de recursos de un Entity Bean es manejado por el contenedor del Bean.

PGP: Pretty Good Privacy. Es un esquema de encriptación asimétrica (Clave pública/privada).

Public Key Cryptography. Criptografía de Clave Pública es una ciencia matemática usada para proveer confidencialidad y autenticidad en el intercambio de información usando algoritmos criptográficos que trabajan con claves públicas y privadas.

Rootkit: Es un conjunto de programas que permiten a un intruso implementar "puertas traseras" (backdoor's) para asegurar su regreso al sistema atacado, y al mismo tiempo esconderse del resto de los usuarios del sistema, en particular del administrador.

RSA - Ron Rives, Adi Shamir and Len Adleman at MIT, in 1977.

Servlet. Programa en Java que se ejecuta como parte de un servicio de red, típicamente un servidor del HTTP y responde a las peticiones de clientes.

Sistema de Detección de Intrusos: mecanismo cuyo objetivo es detectar, identificar y responder ante una intrusión.

Sniffer: Es un programa que permite “escuchar furtivamente” en redes de medios de comunicación compartidos (tales como Ethernet). Se ejecuta en una máquina que está conectada a la red y captura el tráfico de todo el segmento de red.

SSL: Secure Socket Layer. Es un protocolo para transmitir nuestra información a través del Internet de manera encriptada.

UML: Es un lenguaje gráfico que sirve para modelar, diseñar, estructurar, visualizar, especificar, construir y documentar software. UML proporciona un vocabulario común para toda la cadena de producción, desde quien recaba los requisitos de los usuarios, hasta el último programador responsable del mantenimiento. Es un lenguaje estándar para crear los planos de un sistema de forma completa y no ambigua. Fue creado por el Object Management Group, OMG, un consorcio internacional sin ánimo de lucro, que asienta estándares en el área de computación distribuida orientada a objetos, y actualmente revisa y actualiza periódicamente las especificaciones del lenguaje, para adaptarlo a las necesidades que surgen. El prestigio de este consorcio es un aval más para UML, considerando que cuenta con socios tan conocidos como la NASA, la Agencia Europea del Espacio ESA, el Instituto Europeo de Bioinformática EBI, Boeing, Borland, Motorla y el W3C, por mencionar algunos[52].

Vulnerabilidad: Hardware, firmware o software que deja a un SIA abierto para su uso potencial.

[52] Sacado del VII Congreso Nacional de Informática para la Salud. Tutorial de UML.

Bibliografía.

Libros:

- J2EE Core Design Patterns.
- Information Security. William Stallings.
- Oracle JDBC. Oracle.
- Red Hat Bible.
- The Open Source PKI Book. http://ospkibook.sourceforge.net/docs/OSPKI-2.4.7/OSPKI-html/ospki-book.htm
- Design Patterns Explained: A New Perspective on Object-Oriented Design. InformIT. http://www.informit.com

Documentos:

- RFC 2511 - Internet X.509 Certificate Request Message Format http://www.faqs.org/rfcs/rfc2511.html
- Tesis Argentina de Seguridad Informática.
- Clases de Ingeniería de Hardware, USM, Cañas.
- Symantec WHITE PA P E R: The need for a client firewall inside the network perimeter
- Symantec BLENDED THREATS OVERVIEW: Protecting the Client Tier from Blended Tretas.

- Verisign White Papers.
- Check Point White Papers.

Enlaces:

- Verisign Inc. http://www.verisign.com
- Eclipse. http://eclipse.org/
- Oracle. http://www.oracle.com
- Osmosis Latina. http://www.osmosislatina.com/index.htm
- http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGSO200_archivos/ataques.htm
- http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGSO200_archivos/fallas%20famosas.htm
- http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGSO200_archivos/Principios%20del%20diseno%20para%20la%20seguridad.html
- http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGSO00.htm
- http://www.microsoft.com/spain/technet/recursos/articulos/avdind_3.mspx
- http://www.symantec.com/region/mx/enterprisesecurity/content/framework/LAM_2284.html
- http://www.rediris.es/cert/doc/unixsec/node26.html
- http://www.maestrosdelweb.com/editorial/snort/