205

IF

# UNIVERSIDAD NACIONAL DEL CALLAO

MAR 2015

**FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS**

# INFORME FINAL DE INVESTIGACIÓN

"Certificados Digitales para aumentar la seguridad de documentos electrónicos en la UNAC FIIS mediante PKI (Infraestructura de Clave Pública)"

**Presentado por:**

**Gerber F. Incacari Sancho, M.Sc.**

**(Resolución Rectoral N° 799-2012-R)**

**Periodo: 01/08/2012 a 31/07/2014**

**Callao 2014 Perú**

## 1. INDICE

## 2. RESUMEN

Los protocolos sobre los que se ha construido Internet (TCP/IP) ofrecen muy poco o ninguna tipo de seguridad. Mientras un paquete viaja por varias redes hasta alcanzar su destino, éste se puede leer e incluso modificar fácilmente, lo que supone un problema cuando la información que se transmite es especialmente sensible, como por ejemplo: datos personales, números de tarjeta de crédito, información corporativa confidencial y con propiedad intelectual, etc.

Hay que ser sumamente cuidadoso con las aplicaciones que manejan este tipo de datos, debido a las numerosas implicaciones que pueden surgir debido a problemas de seguridad, en distintos ámbitos, incluido el jurídico. Este último puede comprobarse con la aparición progresiva de leyes en Venezuela como la Ley Sobre Mensajes de Datos y Firmas Electrónicas, y Ley Especial Contra los Delitos Informáticos. Ref www.asambleanacional.gov.ve/ns2/leves.aspl

Una de las técnicas más utilizadas en seguridad informática son los certificados digitales que ofrecen algunas soluciones a problemas como: la lectura no autorizada de correo electrónico, suplantación de personalidad, suplantación de servidores, acceso a datos confidenciales

Palabras clave

Certificados digitales, firma digital, documentos electrónicos, PKI

# 3. INTRODUCCION

## 3.1. PLANTEAMIENTO Y DEFINICION DEL PROBLEMA

Tras la masiva utilización de Internet, como medio de comunicación de información, y el nacimiento del comercio electrónico, como aplicación de la actividad comercial a nivel mundial, surge la necesidad de asegurar las conexiones que se realizan a través de la red.

En estos momentos surge la demanda en muchas organizaciones de permitir a sus usuarios acceder a determinada información de una manera sencilla y permanente, pero el motor de la demanda ha sido la popularidad de Internet y sus propias características que permiten esa nueva perspectiva en la gestión de la información.

## 3.2. ENUNCIADO DEL PROBLEMA

Por otra parte en el país se encuentra en sus inicios la implementación del PKI del estado peruano, pero que sin embargo no provee certificados digitales para seguridad de documentos electrónicos por lo que: ¿la implementación Certificados Digitales mediante PKI (Infraestructura de Clave Pública) ayudará a aumentar la seguridad de documentos electrónicos en la UNAC-FIIS?

## 3.3. OBJETIVOS DE LA INVESTIGACION

Mejorar la seguridad de los documentos electrónicos mediante el cifrado y firma por certificados digitales

- Desarrollar le marco teórico de la autoridad certificadora y de la entidad de registro.
- Desarrollar un modelo de Autoridad Certificadora (AC): La autoridad certificadora es el componente clave de una infraestructura de claves públicas y es la encargada de realizar la emisión y administración de los certificados durante todo el ciclo de vida de los mismos.
- Desarrollar un modelo de Autoridad de Registro (AR): Que es la responsable del registro y la autenticación inicial de suscritores, que son los usuarios a quienes se les expide un certificado después de que les ha sido aprobada una solicitud de registro.

Configurar un sistema de administración de certificados y distribución, que establece el tratamiento que recibirán los certificados generados, desde el procedimiento de generación hasta su revocación.

- Implementar el Software PKI para la Autoridad Certificadora y la Autoridad de Registro.
- Realizar pruebas de cifrado y firma de documentos electrónicos de los estudiantes de la FIIS-UNAC utilizando certificados digitales generados por la PKI.ales de la PKI(Infraestructura de clave pública).

## 3.4. JUSTIFICACION

Conseguir certificados digitales y hacer uso de ella en otros países es relativamente simple, un ejemplo de ello es el gobierno Español quien fue uno de los primeros en toda Europa en implementar y regularlos. En el Perú la implementación del PKI se encuentra en la fase de implementación y que dichos certificados tiene valor legal, es decir tendrán el mismo valor como se firmara un documento manuscrito al de la firma digital.

El cifrado y la firma de documentos mediante certificados digitales permiten conservar, clasificar y almacenar información en espacios menores, en las instituciones públicas la información en papel debe conservarse por aproximadamente 10 años, aun cuando la información digital requiere de menor espacio físico y permite una conservación infinitamente superior a la física, únicamente condicionado al almacenamiento en medio de memoria secundaria.

El tratamiento por medios informáticos permite la sustitución del soporte en papel del documento por un nuevo soporte contenido en un medio electrónico.

La presente investigación es importante porque generará confianza en los documentos electrónicos al estar firmados y cifrados digitalmente lo que garantiza la confidencialidad al mismo tiempo la integridad del documento electrónico.

# 4. MARCO TEORICO

## 4.1. SEGURIDAD INFORMATICA

Según K. Charlie [2]. Podemos entender como seguridad una característica que determina que cualquier sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. Para la mayoría de los expertos el concepto de seguridad en la informática es supuesto, porque no existe un sistema totalmente o 100% seguro. Para que un sistema se pueda definir como seguro debemos de brindar tres características al mismo tiempo.

## 4.2. CONFIDENCIALIDAD

Se refiere a que se debe mantener inaccesible a todos los usuarios que no estén autorizados a los datos e información, cuando se requiera. Por ejemplo; se estable una comunicación por correo electrónico entre el decano de la Facultad de Ingeniería y el director de una escuela de la misma facultad y donde se le informa la decisión que tomaron por una determinada situación de carácter urgente, y el único que la puede conocer es el director de la escuela, se debe mantener inaccesible a esta comunicación a una tercera persona que se pueda aprovechar o utilizar para fines deshonestos.

## 4.3. INTEGRIDAD

Se refiere a la protección que debe tener la información, datos, sistemas y otros activos informáticos contra cambios o alteraciones en su estructura o contenido ya sean intencionales o causales. Por ejemplo; siguiendo el ejemplo anterior, una vez que s e envía la información por el correo electrónico, el mismo no debe sufrir alteraciones en su contenido que pueda ocasionar una toma de decisión diferente a la planteada en la comunicación que genere malestares. Se quiere que llegue a su destino la información tal cual se generó en la fuente.

## 4.4. DISPONIBILIDAD

Se refiere a la capacidad que tenga el sistema para mantener los datos e información el mayor tiempo y lugar posible accesible a todos los usuarios autorizados o pertinentes. Por ejemplo; el sistema de inscripción de la facultad de ingeniería, debe permitir a sus estudiantes inscribirse en cualquier sitio que tenga acceso a Internet y no que tengan que asistir a un laboratorio o sitio de la facultad para poder realizar la inscripción, logrando un compromiso con los aspectos de confidencialidad e integridad.

## 4.5. AUTENTICACION Y AUTORIZACION

Se refiere a la capacidad que tenga el sistema de probar que un usuario o agente es el que dice ser, con la finalidad de permitirle su acceso. Por ejemplo; que el sistema de administración de notas de los bachilleres de la facultad de ingeniería, a través de cierto mecanismo como por ejemplo una tarjeta inteligente, pueda identificar y permitir el acceso a dicho sistema solo a la(s) persona(s) responsable(s) y autorizada(s) en manipular la información de ese sistema que son delicadas.

## 4.6. NO REPUDIO

Se refiere a mecanismo que permita verificar que una persona fue la que envió una determinada información, y que esa persona no pueda negarse de que envió dicha información.

En la actualidad hay muchos sistemas, actividades, servicios, etc, que utilizan Internet para realizar operaciones, y no cumple con estos aspectos de seguridad mencionados, por ejemplo, el más utilizado, el correo electrónico, que es relativamente vulnerable falsificar información con este medio, y la información que se envía fácilmente esta expuesta a tercera personas.

Todo lo que se ha nombrado sobre seguridad informática tiene el objetivo de producir confianza; es decir, la percepción de seguridad que tiene el usuario de los sistemas e información digital con los cuales interactúa.

Para alcanzar los niveles aceptables que conforman diferentes aspectos de seguridad informática, se requiere de la construcción de mecanismos como por ejemplo la Infraestructura de Clave Pública (PKI), que utiliza muchas de las nuevas tecnologías usadas en la construcción de soluciones para el comercio electrónico.

## 4.7. INTRODUCCION AL CIFRADO

a. Según A. Nash [1]. Es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas (algoritmos matemáticos). El cifrado tiene como finalidad, garantizar el secreto en la comunicación entre dos entidades (personas, organizaciones, etc.), asegurar que la información sea enviada por el remitente que realmente dice ser y que el contenido del mensaje enviado, no haya sido modificado en su tránsito

b. Información original que debe protegerse se denomina texto en claro. El cifrado es el proceso de convertir el texto claro en un texto ilegible, denominado texto cifrado o criptograma

c. Estos algoritmos matemáticos o de cifrado utilizan claves para cifrar y descifrar los texto en claros, estas claves son similares a una llave física que se usan para cerrar o abrir una puerta, las claves tienen un tamaño en bits que está determinado de acuerdo al tipo de algoritmo que se esté utilizando.

## 4.8. TIPOS DE CIFRADO

### 4.8.1. CIFRADO SIMETRICO

Según A. Nash [1]. Algoritmo de cifrado que usa una misma clave para cifrar y para descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez ambas tienen acceso a esta clave, el remitente cifra el mensaje que quiere proteger (texto en claro) usando la clave (clave simétrica), lo envía al destinatario, y éste lo descifra con la misma clave.

Entre las características importantes del cifrado simétrico tenemos:

- El cifrado simétrico utiliza la misma clave para cifrar y descifrar. El cifrado simétrico es rápido. El cifrado simétrico es seguro
- El texto cifrado que resulta del cifrado simétrico es compacto.
- El cifrado simétrico requiere una administración compleja de claves.
- El cifrado simétrico no se ajusta a las firmas digitales o a la aceptación.

### 4.8.2. CIFRADO ASIMETRICO

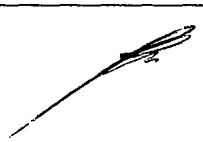
Según A. Nash [1]. Algoritmo de cifrado que usa un par de claves para cifrar y descifrar el mensaje. Las dos claves pertenecen a la persona que ha enviado el mensaje. Una clave es pública y se puede entregar a cualquier persona. La otra clave es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. El remitente usa la clave pública del destinatario para cifrar el mensaje, y una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje, entre las características importantes del cifrado asimétrico tenemos:

- El cifrado asimétrico utiliza una clave (publica/privada) para cifrar y la otra clave (publica/privada) para descifrar
- El cifrado asimétrico es relativamente lento.
- El cifrado asimétrico es seguro.
- El cifrado asimétrico expande el texto cifrado.
- El cifrado asimétrico no tiene los problemas complejos de distribución de claves.

## 4.9. FIRMAS DIGITALES

Según A. Nash [1]. La firma digital es el resultado de aplicar cierto algoritmo matemático, denominado función hash, a su contenido. Esta función asocia un valor dentro de un conjunto finito (generalmente los números naturales) a su entrada. Cuando la entrada es un





documento, el resultado de la función (reseña) es un número que identifica casi unívocamente al texto. Si se adjunta este número al texto de manera cifrada con algoritmo asimétrico usando la clave privada del que firma, el destinatario puede aplicar de nuevo la función hash y comprobar su resultado (reseña) con el que ha recibido, si ambos son iguales, tiene la seguridad de que el texto no fue modificado una vez que fue firmado y que lo envió la persona dueña de la clave privada que firmo el texto.

## 4.10. CERTIFICADOS DIGITALES

Según A. Nash [1]. Un Certificado Digital es un documento digital firmado digitalmente por un tercero confiable (una Autoridad de Certificación) el cual garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública. Si el Certificado es auténtico y confiamos en la Autoridad Certificadora (AC). Entonces, podemos confiar en que el usuario identificado en el Certificado Digital posee la clave pública que se señala en dicho certificado. Así pues, si un sujeto firma un documento y anexa su certificado digital, cualquiera que conozca la clave pública de la AC podrá autenticar el documento

- El estándar, internacionalmente aceptado, para Certificados Digitales, es el denominado X.509, en su versión 3.
- Contiene datos del sujeto, como su nombre, dirección, correo electrónico, etc (Ver figura 10).
- Con la versión 3 de X.509, sucesora de la versión 2, no hace falta aplicar restricciones sobre la estructura del certificado gracias a la definición de las extensiones de certificados. Se permite que una organización pueda definir sus propias extensiones para contener información específica dentro de su entorno de operación. Este tipo de certificados es el que usa el protocolo de comercio electrónico SET.
- X.509 y X.500 fueron originalmente diseñados a mediados de los años 80, antes del enorme crecimiento de usuarios en Internet. Es por esto por lo que se

diseñaron para operar en un ambiente donde sólo los computadores se interconectaban intermitentemente entre ellos. Por eso en las versiones

- 1 y 2 de X.509 se utilizan CRLs muy simples que no solucionan el problema de la granularidad de tiempo

La versión 3 introduce cambios significativos en el estándar. El cambio fundamental es el hacer el formato de los certificados y los CRLs extensible. Ahora los que implementen X.509 pueden definir el contenido de los certificados como crean conveniente. Además se han definido extensiones estándares para proveer una funcionalidad mejorada.

## 4.11.COMPONENTES DE LA INFRAESTRUCTURA PKI

## 4.12. AUTORIDAD DE CERTIFICACION RAIZ

La Autoridad de Certificación de primer nivel. Esta AC sólo emite certificados para sí misma y sus AC Subordinadas. Únicamente estará en funcionamiento durante la realización de las operaciones para las que se establece. Sus datos más relevantes son:

## 4.13. AUTORIDAD DE CERTIFICACION SUBORDINADAS

Las Autoridades de Certificación subordinadas de "AC Raíz". Su función es la emisión de certificados para los titulares de DNIe.

- **Emisión de certificados**

  La solicitud de certificados digitales se realiza en las Autoridades de Registro, que son las encargadas de verificar la identidad de los solicitantes, de acuerdo al protocolo establecido en la Política de Certificación. Para ello es necesario que el solicitante acuda a la Autoridad de Registro correspondiente con su DNI o tarjeta de residencia, junto a su carnet inteligente. La Autoridad de Registro verificará su identidad, solicitando, a continuación, el certificado. Se imprimirá un resguardo en papel, que en el que se recogen los datos del certificado solicitado.

- **Renovación de certificado**

La renovación de un certificado digital se da cuando este a vencido su periodo de validez(2 años), para ello se tiene que recurrir a la autoridad de registro para que nuevamente valide la información.

- **Revocación de certificados no válidos**

  ***Efectos de la Revocación.***

  Los efectos de la revocación de un Certificado Digital son:

  - La expiración instantánea de su vigencia
  - La eliminación de su capacidad para identificar al firmante en cualquier Mensaje de Datos que se genere con posterioridad a la fecha de revocación.

- **Causas de Revocación.**

  Un Certificado Digital emitido por la Autoridad Certificadora puede ser revocado a solicitud específica del Titular cuando a su criterio amerite la terminación anticipada de la vigencia por las siguientes causas:

  - Por olvido o extravío de la contraseña de la clave privada.
  - Por robo o extravío de la propia clave privada.
  - Por la sospecha de la utilización por terceros de su clave privada.
  - Por el cambio de alguno de los datos contenidos en el Certificado Digital.

  También, puede ser revocado por las siguientes causas:

  - Por fallecimiento del Titular.
  - Por resolución judicial.

## 4.14. AUTORIDAD DE REGISTRO

La Autoridad de Registro (RA) es la entidad responsable por la comunicación entre el usuario y la autoridad certificadora (CA). Está vinculada a una CA y tiene por objetivo recibir, validar, verificar y gestionar las solicitudes de emisión o revocación de los

certificados digitales, cumpliendo con lo establecido en la "política de certificación" y en concordancia con las políticas y procedimientos definidos por la CA correspondiente.

### 4.14.1. Actividades ejecutadas por el RA

La RA debe establecer los procedimientos y guías para asegurar el cumplimiento de la política de certificados de la jerarquía nacional y de este documento, además de tomar las acciones que prevengan alguna deficiencia de la RA, incluyendo la terminación o suspensión de sus deberes.

Las áreas y actividades ejecutadas por la RA incluyen, entre otras:

- Verificar y validar los documentos de identidad
- Registrar y enrolar a los suscriptores
- Entregar certificados digitales
- Gestionar la aceptación del certificado por parte del suscriptor
- Gestionar revocaciones de certificados
- Registrar los eventos en las bitácoras
- Controlar y supervisar a los agentes de registro
- Almacenar y custodiar la documentación
- Controlar los reportes de incidentes

### 4.14.2. Recepción de solicitudes Recepción de pagos

La autoridad de registro procesa todos los pedidos de solicitud de certificado requeridos por la Autoridad Certificadora.

Para el procesamiento de la solicitud del certificado se efectúan las siguientes actividades:

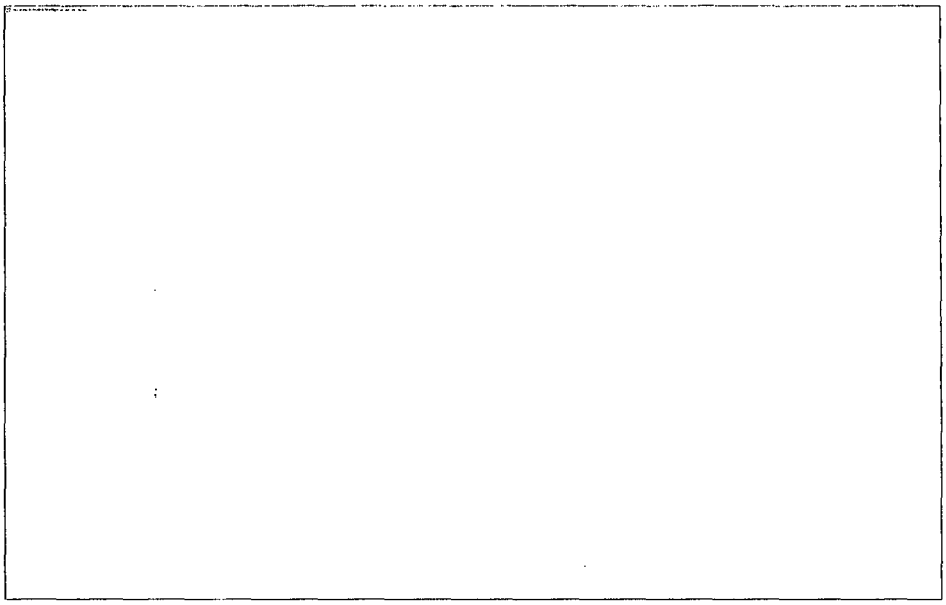
- Realización de las funciones de identificación y autenticación
- Aprobación o rechazo de la solicitud de emisión de un certificado
- Tiempo para el procesamiento de la solicitud del certificado

### 4.14.3. Entrega de solicitud a la autoridad de certificación

El personal encargado de registrar la solicitud de certificado digital a la CA deberá regístralo y firmarlo digitalmente, para que luego se genere el certificado para el cliente.

### 4.14.4. Diagrama de actividades para generar un certificado

Se siguen los siguientes procedimientos para la generación de un certificado digital.



Para el diagrama anterior los certificados se crean primero con la generación de llaves, en donde el sistema de Carlos o la autoridad de Certificación debe generar un par de llaves que se usarán para firmar y leer firmas. El par de llaves se genera comúnmente o por el sistema de Carlos o por una tarjeta inteligente o una caja especial de hardware que hace las operaciones de criptografía que Carlos usa con su sistema.

Una vez generada la llave se verifica la identidad del solicitante. El CA entonces crea un certificado. El certificado es creado por el CA que firma una recopilación de mensaje de la información del certificado, que incluye la llave pública del solicitante. Después de que el par de llaves se ha generado y se ha completado el certificado que da testimonio de la titularidad del par de llaves, ellos se cargan en la aplicación que usará el certificado y llave. No hay método estándar para hacer esto, tanto la autoridad de certificación debe tener los

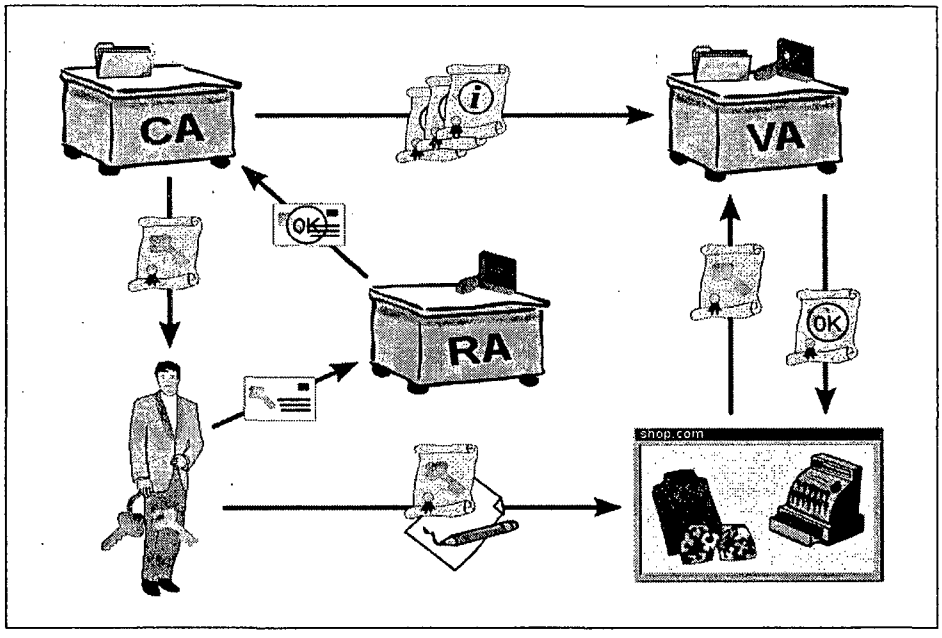
módulos de aplicación para cargar los certificados en todos las aplicaciones soportadas. Frecuentemente, la aplicación para el certificado y la carga real del certificado y el par de llaves se separa por un período de tiempo, donde la verificación adicional se ha hecho.

## 4.15. CONFIGURACION DE UNA AUTORIDAD DE CERTIFICACION RAIZ

### 4.15.1. Determinar diagrama de despliegue

Los componentes principales de la infraestructura PKI se desplegaron de la siguiente manera (ver gráfico N° 1).

Gráfico N° 1: Diagrama de despliegue de PKI




### 4.15.2. Determinar requisitos de seguridad lógica

La seguridad de un certificado también se conoce como nivel de seguridad. Puede decirse que es la medición de la fuerza que vincula al sujeto del certificado con el certificado en sí. Refleja el nivel de confianza que puede tener en que la persona (o el dispositivo) que utiliza el certificado es realmente la misma que el sujeto nombrado en el certificado. El nivel de seguridad es la medición de dos elementos principales:

El rigor del proceso de registro e inscripción de certificados. Por ejemplo, ¿tuvo la persona que acudir personalmente y presentar un documento de identificación con fotografía para obtener su certificado o bastó con una dirección de correo electrónico?

La manera en que se almacena la clave privada. Cuanto más difícil sea copiar o comprometer de otra forma la clave, mayor será la seguridad de que siga en posesión exclusiva del propietario original, el sujeto del certificado.

Los dos están fuertemente vinculados, ya que no existe ninguna razón para invertir en medidas costosas de protección de claves privadas si nunca ha estado verdaderamente seguro de la identidad del propietario de la clave privada. De forma similar, un arduo proceso de registro que implique exhaustivas comprobaciones de antecedentes y pruebas de ADN sirve de poco si, posteriormente, la clave privada se almacena de una forma que no es lo suficientemente segura.

Conseguir una seguridad superior para un certificado cuesta dinero y, con frecuencia, no es necesario para muchos de los usos de los certificados. Si la seguridad que desea de un certificado es que pertenezca a un usuario de dominio autorizado, las credenciales del dominio son totalmente aceptadas como evidencia de registro para inscribir un certificado.

Debe documentar el significado de los niveles de seguridad que utiliza en las directivas de certificados y declaraciones de prácticas.

### 4.15.3. Determinar arquitectura de hardware necesario

Las operaciones criptográficas suelen ser tareas que consumen muchísimos ciclos de CPU (debido a la necesidad de generar números primos de gran tamaño), mediante un procesador aparte especializado, de manera que libera al servidor de ese tipo de tareas.

- Memory 2GB Memory (2x1GB), 1066MHz Single Ranked UDIMMs for 1 Processor, Adv ECC con crecimiento a 144GB
- Procesador Primario Intel Xeon E5520, 2.26Ghz, 8M Cache, 5.86 GT/s QPI,

Turbo, HT

- Onboard NIC Type (Fabric A)Onboard Broadcom 5709 Quad Port 1GbE NIC with TOE
- Operating System Media KitsNo Operating System Media Kit
- Configuración de Arreglos de Discos SAS drives using the CERC6 Daughtercard with drives in a RAID 5 Array
- Hard Drive 73GB 15K RPM Serial-Attach SCSI 2.5" Hot Plug Hard Drive

# 5. MATERIALES Y METODOS

## 5.1. POBLACION Y MUESTRA

### 5.1.1. POBLACION

Para la población se considerará a todos los estudiantes del primer ciclo al v ciclo de la Facultad de Ingeniería Industrial y de Sistemas de la Universidad Nacional del Callao, N=500 estudiantes (escuela de Ingeniería de Sistemas e ingeniería Industrial)

### MUESTRA

El tipo de muestreo que se utilizará para la investigación será el muestreo aleatorio simple, estimándose un tamaño de muestra de n=100 estudiantes.

## 5.2. TECNICAS DESCRIPTIVAS PARA LA CONTRASTACION

Para apoyar la interacción de grupo se considerará varios aspectos de colaboración. Estos incluyen información compartida, comunicación entre miembros del grupo, y coordinación de las actividades cooperativas.

## 5.3. DETERMINACION DE REQUISITOS DE SOFTWARE

### 5.3.1. Sistema operativo

El sistema operativo a utilizarse es el Sistema Operativo Red Hat Enterprise Linux v6.

### 5.3.2. Requisito de software

El software necesario son los siguientes:

- EJBCA v4.0.14, Software PKI
- JDK 1.6 OpenJDK, Librería JAVA
- JBoss Application Server 5.1.x, servidor Web
- Apache Ant 1.7.1

### 5.3.3. Requisitos de hardware

Las operaciones criptográficas suelen ser tareas que consumen muchísimos ciclos de CPU

(debido a la necesidad de generar números primos de gran tamaño), mediante un procesador aparte especializado, de manera que libera al servidor de ese tipo de tareas. En general, la efectividad de los módulos HSM, es mayor en criptografía de clave pública (o asimétrica), que en criptografía simétrica.

### 5.3.4. Sobre el HSM

El HSM utiliza mecanismos que detectan los intentos de manipulación, en tal caso, produce el borrado automático e inmediato de toda la información sensible contenida en el HSM, de tal manera que son inviable para recuperar información secreta

La protección contra una amenaza esta basada en una combinación de al menos dos mecanismos de seguridad independientes. El fallo de un solo mecanismo de seguridad no compromete la seguridad del HSM.

El HSM también incluye características para la detección de manipulación de los resultados del dispositivo.

El HSM también incluye características tales que el acceso a los resultados del dispositivo evidencien una visible manipulación que tiene una alta probabilidad de ser detectados

No existe una forma posible para determinar cualquier información sensible monitoreando las emisiones electro magnéticas, el consumo de energía, o cualquier otra característica interna o externa .

No existe una forma posible para determinar cualquier información sensible monitoreando las emisiones electro magnéticas, el consumo de energía, o cualquier otra característica interna o externa.

EL diseño del HSM lo protege contra la sustitución del HSM de tal forma que no es posible construir un duplicado a partir de componentes disponibles comercialmente. Por ejemplo, el gabinete de un HSM no suele estar disponible.

La información o funciones sensibles solo son utilizadas en las áreas protegidas del HSM.

El manejo de la información y funciones con información sensible son protegidas a partir

de la modificación o sustitución, y adicionalmente son protegidas con claves secretas y privadas.

Si el dispositivo permite el acceso a las áreas internas de seguridad que contiene componentes sensibles (p.e. para el servicio o mantenimiento), el acceso inmediato a datos sensibles tales como PINS o datos criptográficos son impedidos por el diseño de las áreas internas (p.e. encapsular los componentes en un gabinete anti-sabotaje o sensible a la manipulación), o tener un mecanismo de modo tal que al acceder a las áreas internas se produzca el inmediato borrado de los datos sensibles.

Una política de seguridad a disposición del proveedor para el uso correcto del HSM, incluyendo la información sobre las responsabilidades en la gestión de claves, responsabilidades administrativas, funcionalidades del dispositivo, identificación y requerimientos del entorno. La política de seguridad debe definir las funciones compatibles por el HSM e indicar los servicios disponibles para cada función en una forma tabular determinada.

### 5.3.5. INSTACION DE SOFTWARE PKI

Se desarrolló el procedimiento seguido para la instalación de la herramienta y de los componentes que necesita para que funcione correctamente, también se aborda la configuración del paquete de registro de *logs* que se utilizó para la gestión de los mismos, y por último, se listan los diferentes errores encontrados en las repetitivas instalaciones que se hicieron acompañados de las soluciones que los resolvieron. La determinación de una secuencia de pasos para lograr una correcta instalación del software de certificación EJBCA.

#### 5.3.5.1. Proceso de instalación

En este punto se describirá el proceso de instalación de EJBCA, desde la creación de nuevas variables de entorno, de usuarios y de repositorios de datos, la instalación de ciertos componentes y herramientas necesarias hasta la modificación de archivos

específicos, estos pasos son necesarios para lograr un correcto funcionamiento de la herramienta EJBCA (versión 3.5.2). La instalación que se detalla a continuación se realizó en el servidor remoto *ovh1.firmaprofesional.com,* que tiene el sistema operativo GNU/Linux de distribución *Ubuntu.*

Antes de nada es recomendable ejecutar en una consola y como usuario privilegiado los siguientes comandos para verificar que se tienen instalados algunos programas útiles:

*apt-get install autoconf lynx zip unzip tofrodos ldap-utils apt-get install db4.2-util libldap2-dev libssl-dev libnet-ldap-perl apt-get install libapache2-mod-jk tdsodbc sqsh junit libapr1 slapd*

#### 5.3.5.2. Instalación de Java JDK 1.5.0

Dado que EJBCA está basada en la tecnología J2EE es necesario instalar el JDK *(Java Devolopment Kit)* respectivo, en este caso se utilizó la versión 1.5.0 porque con esta versión se habían realizado pruebas anteriores.

El primer paso es obtener el paquete directamente de la página de Sun, *http://java.sun.com/downloads/*, y luego instalarlo, o se puede utilizar el comando *apt- get install.*

apt-get install sun-java5-jdk

Lo normal es que luego de esto el JDK se haya instalado correctamente pero puede darse el caso de que al ejecutar este comando se obtenga un error como el siguiente:

*E: No se pudo encontrar el paquete sun-java5-jdk*

Esto se debe a que la descarga de paquetes del repositorio "*multiverse*" no está habilitada en el archivo */etc/apt/sources.list,* para habilitarla se le debe descomentar la siguiente línea:

*#deb-src http://es.archive.ubuntu.com/ubuntu gutsy universe*

Y al final de la misma, añadir la palabra *multiverse.* Debería quedar de la siguiente manera:

*deb-src http://es.archive.ubuntu.com/ubuntu gutsy universe multiverse*

Luego de haber realizado correctamente la instalación del JDK y de haber resuelto las posibles dependencias que hayan podido aparecer, *apt-get install* resuelve los problemas de dependencias por sí mismo, y en caso de tener instalada otra versión de Java se debe establecer que versión del JRE se quiere utilizar. Por ejemplo, Ubuntu 7.10 tiene la versión 6 instalada por defecto y esta versión usa el paquete *java-6-sun.* El sistema operativo permite tener más de una versión instalada pero solo utilizar una en un determinado momento, en este caso se quiere que el sistema utilice la versión 1.5.0 *(java-1.5.0-sun*), esto se hace con el comando *update-java-alternatives.*

*update-java-alternatives -s java-1.5.0-sun*

El siguiente paso es añadir al archivo */etc/environment* la variable de entorno *JAVA_HOME*, para esto se le debe añadir la línea siguiente:

*JAVA_HOME="/usr/lib/jvm/java-1.5.0-sun"*

De ahora en adelante se hará referencia a la ubicación del JRE 1.5.0 utilizando *java_home.*

#### 5.3.5.3. Las librerías Bouncy Castle

La versión 1.4 de Java introdujo el manejo de criptografía y basó la implementación de las clases que la proporcionan en una arquitectura llamada *"Proveedores de Seguridad"*, paro lo que añadió dos extensiones: la JCA *(Java Cryptography Architecture)* y la JCE *(Java Cryptography Extension).* La primera proporciona la arquitectura para realizar las operaciones criptográficas y la segunda extiende las funcionalidades de la JCA añadiendo proveedores de seguridad. El proveedor de seguridad por defecto de Java es uno proporcionado por Sun, el cual no soporta demasiados algoritmos criptográficos debido a las restricciones legales establecidas por algunos países respecto a la exportación e

importación de criptografía.

*BouncyCastle* es un proveedor de seguridad desarrollado por el grupo de programadores *Legion of BouncyCastle,* y es el utilizado por EJBCA ya que es de gran calidad, completamente gratuito y su licencia permite utilizarlo en cualquier tipo de aplicación.

Un ejemplo de un problema que se puede tener si se usa solo el proveedor de seguridad proporcionado por Sun es que no se podrán manipular los archivos del tipo PKCS12 de EJBCA con el *keytool* de Sun. El estándar PKCS12 define un formato para almacenar claves privadas acompañadas de sus respectivos certificados digitales y normalmente el *keytool* de Sun puede solo leer los archivos PKCS12 pero no escribir sobre ellos, este problema lo soluciona la instalación de las librerías *BouncyCastle.*

Es importante resaltar que EJBCA no es compatible con otro JCE ya que solo *BouncyCastle* contiene clases que permiten no solo el uso de certificados sino también la generación de los mismos.

Para instalar las *BouncyCastle JCE* se tiene que descargar de la página *http://www.bouncycastle.org/latest_releases.html* la última versión de las librerías *bcprov-jdk15-XXX.jar* y *bcmail-jdk15-XXX.jar,* llamadas *"BouncyCastle provider"* y *"BouncyCastle SMIME/CMS"* respectivamente, y copiarlas en *java_home /jre/lib/ext/que* es donde se guardan las extensiones. Cuando se realizó esta instalación la última versión era la 138 *(XXX=138).*

Por último se debe modificar el archivo *java_home/jre/lib/security/java.security,* aumentando una línea en la definición de los *security providers* (proveedores de seguridad). Esta parte del archivo tiene esta apariencia:

```
#
#       List of providers and their preference orders (see above):
#
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.crypto.provider.SunJCE
```

*security.provider.5=sun.security.jgss.SunProvider*
*security.provider.6=com.sun.security.sasl.Provider*

La línea a aumentar define al nuevo proveedor *BouncyCastle* y lo coloca en el segundo orden de preferencia, esta línea es:

*security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider*

El archivo finalmente debe quedar de la siguiente manera:

```
#
#       List of providers and their preference orders (see above):
#
security.provider.1=sun.security.provider.Sun
security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProvider
security.provider.3=sun.security.rsa.SunRsaSign
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
```

Es importante que el primer proveedor de seguridad sea Sun y que el segundo sea BouncyCastle.

#### 5.3.5.4. Las JCE Unlimited Strength Jurisdiction Policy Files

EJBCA es una de las aplicaciones reconocidas como exentas de las restricciones criptográficas existentes en algunos países, debido a que utiliza *"strong criptography"* y *"keystore passwords"* de longitud mayor a 7 caracteres necesita tener instalado el JCE

*Unlimited Strength Jurisdiction Policy Files para funcionar correctamente. Este JCE se puede descargar de la página http://java.sun.com/javase/downloads/index_jdk5.jsp.*

Se deberá descargar el correspondiente a la versión 1.5.0 de Java, el *Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0.* El archivo que se obtendrá será el *jce_policy-1_5_0.zip.*

Al descomprimirlo se obtendrán los siguientes archivos:

*unzip jce_policy-1_5_0.zip creating: jce/*

inflating: jce/COPYRIGHT.html inflating: jce/README.txt inflating:

*jce/US_export_policy.jar inflating: jce/local_policy.jar*

Finalmente, de estos archivos se tendrán que copiar los archivos JAR *(local_policy.jar* y *US_export_policy.jar)* a la carpeta *java_home/jre/lib/security/*, los otros archivos se pueden borrar tranquilamente.

#### 5.3.5.5. Instalación del servidor de aplicaciones JBoss

EJBCA necesita de un servidor de aplicaciones para ejecutarse, aparte la instalación por defecto y más sencilla de la herramienta está hecha sobre el servidor de aplicaciones JBoss, y es el que se usará. Es recomendable descargar la última versión estable de la página

*http://labs.jboss.com/jbossas/downloads/*.

En este caso la última versión estable fue la 4.2.2. El archivo a instalar está disponible en los formatos ZIP y TAR-GZ, por comodidad se escogió el de formato ZIP llamado *jboss-4.2.2.GA.zip*. Se recomienda descomprimir este archivo directamente a */usr/local/* que es el directorio en el que se guarda el software que se desea conservar en una actualización del sistema. Esto se puede hacer fácilmente con el siguiente comando:

*unzip jboss-4.2.2.GA.zip -d /usr/local/*

Luego, se tendrá que añadir al archivo */etc/environment* la variable de entorno *JBOSS_HOME* que define la ubicación exacta de JBoss, esto es importante porque EJBCA utiliza esta variable para saber en donde se encuentra el servidor de aplicaciones. La línea a añadir es la siguiente:

JBOSS_HOME="/usr/local/jboss-4.2.2.GA/"

Una vez realizado lo anterior JBoss estará listo para usarse, los archivos que controlan la ejecución del servidor se encuentran en el directorio */usr/local/jboss-4.2.2.GA/bin/*, se

utilizan de la siguiente manera:

**Para iniciar el servidor de aplicaciones JBoss**
*./usr/local/jboss-4.2.2.GA/bin/run.sh*

**Para pararlo (también se puede utilizar CTRL+C)**
*./usr/local/jboss-4.2.2.GA/bin/shutdown.sh –S*

De ahora en adelante se hará referencia a la ubicación de JBoss 4.2.2 utilizando *jboss_home.*

#### 5.3.5.6. Instalación de MySQL

El servidor de aplicaciones JBoss tiene su propia base de datos *in-mermory* llamada *"Hypersonic database"*, que sea *in-mermory"* significa que mientras más se utilice ocupará más memoria lo cual es un problema si se piensan emitir muchos certificados. Otro problema de esta base de datos es que no soporta todos los comandos SQL. Por estos motivos si se piensa construir un entorno de producción es recomendable instalar otra base de datos.

En este caso, como se está construyendo un entorno de producción, se utilizará el gestor de base de datos relacional MySQL. Para este propósito se necesitan instalar los paquetes *mysql-server* y *mysql-client,* los cuales a su vez utilizan los paquetes *mysql-server-5.0* y *mysql-client-5.0* que son los binarios de los primeros.

Si se quiere se puede instalar también una interfaz gráfica para manejar con más comodidad las bases de datos que se creen, una alternativa es utilizar MySQL *Administrator* conjuntamente con *MySQL Query Browser,* la primera servirá para administrar la base de datos y la segunda para ejecutar sentencias SQL. Para esto se necesitan instalar los paquetes *mysql-admin* y *mysql-query-browser,* los cuales a su vez dependen de los paquetes *mysql-server-common, mysql-client-common* y *mysql_common.*

La instalación de *mysql-server* creará por defecto un usuario *root* para el que pedirá que se ingrese una contraseña. Es recomendable que con este usuario se cree otro usuario que tenga asignados todos los privilegios y que sea el que realice todas las tareas de

administración necesarias relativas a EJbCa. Para esto habrá que iniciar sesión en MySQL con el usuario *root* y usar el comando *GRANT*, al usuario creado se le llamó *ejbca* y se le identificó con la contraseña *ejbca*, a continuación se detallan los pasos:

**Se inicia sesión como root**
*mysql -u root -p*

**Se crea el usuario ejbca identificado con contraseña ejbca**
*mysql> GRANT ALL ON *.* TO ejbca IDENTIFIED BY 'ejbca' WITH GRANT OPTION; mysql> exit;*

En la configuración por defecto de MySQL, la herramienta escucha las peticiones en el puerto *3306* y en la dirección *0.0.0.0*, que representa a todas las interfaces. Esta configuración se puede ver y modificar en el archivo */etc/mysql/my.cnf.*

Antes de proseguir con la instalación se deberá crear una base de datos en MySQL, en ella se crearán todas las tablas y, dentro de ellas, los datos que se generen al instalar y utilizar EJBCA posteriormente. A la base de datos creada se le llamó *ejbca_3_5_2*, a continuación se detallan los pasos:

**Se inicia sesión como ejbca**
*mysql -u ejbca -p*
**Se crea la base de datos ejbca_3_5_2**
*mysql> create database ejbca_3_5_2 mysql> exit;*
**Se inicia sesión como ejbca directamente en la tabla creada**
*mysql -u ejbca -p ejbca_3_5_2*

#### 5.3.5.7. El driver JDBC

Para que MySQL funcione correctamente con Java será necesario instalar un controlador JDBC *(Java Database Connectivity)* el cual es un API que permite la conectividad entre el lenguaje JAVA y cualquier base de datos SQL existente. En este caso se usará el *Connector/J* que es el controlador JDBC oficial para MySQL. Este controlador es el que provee la conectividad con las aplicaciones desarrolladas en Java, EJBCA es una de ellas.

Lo primero será descargar dicho controlador de *www.mysql.com*, la versión del *Connector/J* disponible al momento de realizar esta instalación fue la 5.1, en todo caso se recomienda

descargar la última versión. Se puede descargar esta versión directamente desde

*http://dev.mysql.com/downloads/connector/j/5.1.html.*

El archivo a instalar está disponible en los formatos ZIP y TAR-GZ, por comodidad se escogió el de formato ZIP llamado *mysql-connector-java-5.1.5.zip*. Luego de descomprimirlo se deberá copiar solo el archivo *mysql-connector-java-5.1.0-bin.jar* dentro del directorio *jboss_home/server/default/lib/*. Este directorio es en donde se guardan las librerías que utilizará cualquier configuración del JBoss.

#### 5.3.5.8. Instalación de EJBCA

Finalmente se instalará EJBCA, antes habrá que descargarlo siguiendo el enlace correspondiente de la página *http://www.ejbca.com/download.htm*. En la página de descargas se encontrarán todas las versiones disponibles de EJBCA, es recomendable descargar la última versión. En este caso se descargó la 3.5.2, que tiene disponibles los paquetes *ejbca_3_5_2.zip* y *extra_3_5_2.zip*, el primero contiene la herramienta EJBCA y el segundo contiene el API de una RA externa que se puede implementar dentro de EJBCA, aquí solo se tratará la instalación de EJBCA. Para esto se tendrá que descargar el archivo *ejbca_3_5_2.zip* y, como se hizo con JBoss, descomprimirlo en */usr/local/*. Esto se hace con el comando:

*unzip ejbca_3_5_2.zip -d /usr/local/*

De ahora en adelante nos referiremos al directorio */usr/local/ejbca_3_5_2/* como *ejbca_home*.

La instalación de EJBCA creará, por defecto, una autoridad certificadora, un usuario superadministrador y un certificado de servidor SSL. Estos tres elementos son necesarios, uno para empezar a emitir certificados y los otros dos para poder acceder a la admin web.

En los archivos de configuración de la herramienta se puede cambiar la configuración de estos elementos, como por ejemplo el contenido de sus certificados electrónicos.

**Modificación de los archivos de configuración**

Después de haber descomprimido la herramienta se procederá a modificar los archivos de configuración del directorio *ejbca_home/conf/*, este directorio es importante porque contiene la mayoría de los archivos de configuración de las funcionalidades que ofrece EJBCA. Estos archivos podrán ser personalizados según las necesidades de cada instalación, en un principio cada uno tendrá la extensión **.properties.sample* pero luego de modificarlos ésta se deberá cambiar a *.properties. Para mantener un orden es recomendable primero copiar el archivo a otro con el nombre cambiado para recién luego modificarlo.

Las modificaciones consisten principalmente en cambiar los valores por defecto de las variables definidas dentro de estos archivos, estas variables contienen datos de la configuración de EJBCA. El cambiar el nombre de los archivos que se modifiquen es importante porque la herramienta conoce los valores por defecto de las variables y la forma en que se le avisa que se han modificado estos valores es cambiando de nombre a los archivos, si esto no se hace la herramienta ignorará las modificaciones y tomará los valores por defecto.

Los archivos a modificar son *database.propiertes.sample, ejbca.properties.sample* y *web.properties.sample* porque definen algunos parámetros de instalación importantes, como el gestor de base de datos que se utilizará, las propiedades de la CA que se creará por defecto o las propiedades del servidor HTTP. El primer paso será hacer una copia de cada archivo, los nuevos archivos tendrán por nombres *database.propiertes, ejbca.properties* y *web.properties* respectivamente.



En el archivo *database.properties* se descomentarán las líneas que definen los valores de las variables *datasource* y *database,* como el nombre y prefijo del JNDI *(Java naming and directory Interface)* de la fuente de datos, el gestor de base de datos a utilizar y los datos de acceso a la base de datos que utilizará EJBCA. A continuación se muestran las variables que se modificaron en este archivo con los valores que se le dieron.

**El nombre y prefijo del JNDI a usar**

*datasource.jndi-name=EjbcaDS*
*datasource.jndi-name-prefix=java:/*

**El gestor de base de datos a usar**

*database.name=mysql*
*datasource.mapping=mySQL*

**La url de acceso a la base de datos y el controlador JDBC**

*database.url=jdbc:mysql://ovh1.firmaprofesional.com:3306/ejbca_3_5_2*
*database.driver=com.mysql.jdbc.Driver*

**Nombre y contraseña del usuario administrador de la base de datos**

*database.username=ejbca*
*database.password=ejbca*

En el archivo *ejbca.properties* habrá que realizar el mismo procedimiento, se descomentarán las líneas que definen los valores de las variables *appserver* y *ca,* como los datos del servidor de aplicaciones que se utilizará y la configuración del certificado de la CA que se creará por defecto. A continuación se muestran las variables más relevantes de este archivo con los valores que se le dieron, solo se modificaron algunas de las variables correspondientes a la configuración básica de dicha CA.

**El servidor de aplicaciones a utilizar y su ubicación**

*appserver.type=jboss*
*appserver.home=${env.JBOSS_HOME}*

**El nombre y DN de la CA por defecto**

*ca.name=AdminCA1*
*ca.dn=CN=AdminCA1,O=EJBCA Sample,C=SE*

**Las propiedades del token software de la CA**

*ca.tokentype=soft*
*ca.tokenpassword=null*
*ca.tokenproperties=conf/catoken.properties*

**Algunas características de su certificado**

*ca.keyspec=2 04 8 ca.keytype=RSA*
*ca.signaturealgorithm=SHA1WithRSA*
*ca.validity=3650*
*ca.policy=null*

**Las contraseñas para proteger sus keystores en la base de datos**

*ca.keystorepass=firma69*
*ca.ocspkeystorepass=firma69*
*ca.xkmskeystorepass=firma69*
*ca.cmskeystorepass=firma69*

De los valores de estas variables se observa que la primera CA que la herramienta creará por defecto tendrá claves software, que su certificado será valido por diez años y que la ubicación de JBoss se obtiene del contenido de la variable de entorno creada anteriormente.

En el archivo *web.properties* se descomentarán las líneas que definen los valores de las variables *java.trustpassword, httpserver* y *superadmin,* que ente otras cosas definen la configuración del servidor web y las propiedades del usuario superadministrador creado por defecto. Este usuario es el que tiene los privilegios para acceder a la admin web. A continuación se muestran las variables más relevantes de este archivo con los valores que se le dieron.

**La contraseña para el keystore de certificados de confianza de EJBCA**

*java.trustpassword=java69*

**La contraseña para el keystore del superadministrador**

*superadmin.password=admin69*
*superadmin.batch=true*

**Configuración del certificado SSL**

*httpsserver.password=server69*
*httpsserver.hostname=localhost*
*httpsserver.dn=CN=ovh1.firmaprofesional.com,O=EJBCA Sample,C=SE*

**Los puertos que JBoss escuchará**

*httpserver.pubhttp=8 08 0 httpserver.pubhttps=84 42 httpserver.privhttps=8 443*

**Las interfaces que JBoss escuchará**

*httpsserver.bindaddress.pubhttp=0.0.0.0*
*httpsserver.bindaddress.pubhttps=0.0.0.0*
*httpsserver.bindaddress.privhttps=0.0.0.0*

De los valores de estas variables se observa que el certificado de servidor SSL que se creará por defecto tiene como *Common Name* el nombre DNS del servidor en el que se instaló la herramienta, esto es importante para evitar advertencias del explorador de Internet, y que JBoss escuchará en todas las interfaces las peticiones dirigidas a la

herramienta.

#### 5.3.5.9. Utilización de los comandos ANT

Los pasos que se muestran a continuación son los que hay que seguir para terminar la instalación de EJBCA, en estos pasos se usará la herramienta *ant.* Ant es una especie de *make* pero hecho en JAVA, lo que asegura que sea independiente del sistema operativo, es útil porque nos evita realizar las tareas mecánicas y repetitivas propias de las fases de instalación y de construcción de aplicaciones, por eso utilizando el comando *apt-get* es recomendable primero asegurarse si está instalada.

*apt-get install ant ant-doc ant-gcj ant-optional ant-optional-gcj*

Luego de haber instalado *ant,* dentro de *ejbca_home* se ejecutará el siguiente comando desde la consola:

*ant bootstrap*

Este comando lo que hace es compilar todos los archivos y agruparlos en archivos JAR, luego todos los JAR en archivos WAR, y por último todos los WAR en un archivo EAR, finalmente desplegará este archivo EAR a JBoss. También creará dentro de *ejbca_home,* los directorios *dist/, tmp/, hwtoken/* y *ocsp-dist/;* dentro de *jboss_home/Server/default/deploy/* los servicios *ejbca.ear, ejbca-ds.xml* y *ejbca-mail- service.xml;* y por último la estructura de la base de datos. En este punto podría ocurrir que el siguiente error:

*OutOfMemoryError: Java heap space java.lang*

La solución a este error se tratará más adelante, en el punto *"Errores detectados"*.

Después de esto, habrá que iniciar JBoss desde la consola y observar detenidamente su inicio, se le debería ver desplegando todos los servicios que utiliza sin que haya errores. Si no ha habido errores, y sin parar JBoss, desde *ejbca_home* se deberá ejecutar el siguiente comando:

*ant install*

Este comando generará los certificados y pares de claves que se necesitan para empezar a utilizar la CA por defecto, y los almacenará, junto con otros datos más, en la base de datos. El comando *ant install* solo se puede ejecutar una vez, la primera vez que EJBCA se instala, si se intenta ejecutar una segunda vez se obtendrán errores.

Si todo ha ido correctamente se habrá creado el directorio *ejbca_home/p12,* este directorio debe contener el repositorio de claves del usuario superadministrador en formato P12 *(superadmin.p12),* el repositorio de claves del servidor SSL *(tomcat.jks)* y el repositorio de claves de las entidades de confianza de EJBCA *(truststore.jks).* El tema de los repositorios de claves (también llamados *keystores)* en EJBCA se tratará con más detalle más adelante en el siguiente capítulo.

Si la ejecución de *ant install* ha sido exitosa el último paso será volver a desplegar toda la herramienta a JBoss. Antes de esto es importante parar el servidor de aplicaciones (también se puede hacer con CTRL+C). Para volver a desplegar la herramienta, desde *ejbca_home,* se deberá ejecutar:

*ant deploy*

Este comando copiará los archivos de configuración modificados y los repositorios de claves creados a JBoss, y volverá a crear los servicios que se crearon con *ant bootstrap.* Si el despliegue se ha realizado correctamente probablemente EJBCA ya esté listo para ser utilizado, para comprobarlo habrá que volver a iniciar JBoss y verificar que no ocurran errores.

## 5.4. OBTENCION DE DOCUMENTOS ELECTRONICOS

Se realizará la recolección de 100 documentos electrónicos (pudiendo se de tipo DOC, XLS, JPG, PDF, TXT) de forma aleatoria de acuerdo a la ficha anexo 1.

### 5.5. PROCESAMIENTO DE DOCUMENTOS ELECTRONICOS

Una vez que se hayan obtenido por parte del estudiante los documentos electrónicos, serán cifrados y firmados digitalmente con el certificado digital de la PKI, y nuevamente serán subidos a una página web para que puedan ser descargado libremente y verificar si el contenido de los documentos electrónicos son recuperables por terceras personas, esta última tarea será controlado, para ello los estudiantes seleccionados de la muestra se les solicitará que seleccione 06 archivos e intente obtener los contenidos de los archivos, de los cuales 2 documentos serán nivel de cifrado bajo, 2 de nivel de cifrado medio y 2 con nivel de cifrado alto, los resultados serán registrados en la ficha del anexo 3.

### 5.6. TECNICAS ESTADISTICAS

Para las pruebas estadísticas se utilizará el SPSS(software estadístico) como herramienta para el análisis de los datos, adicionalmente se considerará las pruebas estadísticas, tales como cuadros de frecuencias de datos, análisis porcentual así mismo se hará uso de graficas de barrar para realizar comparaciones.

### 5.7. DEMOSTRACION DE HIPOTESIS

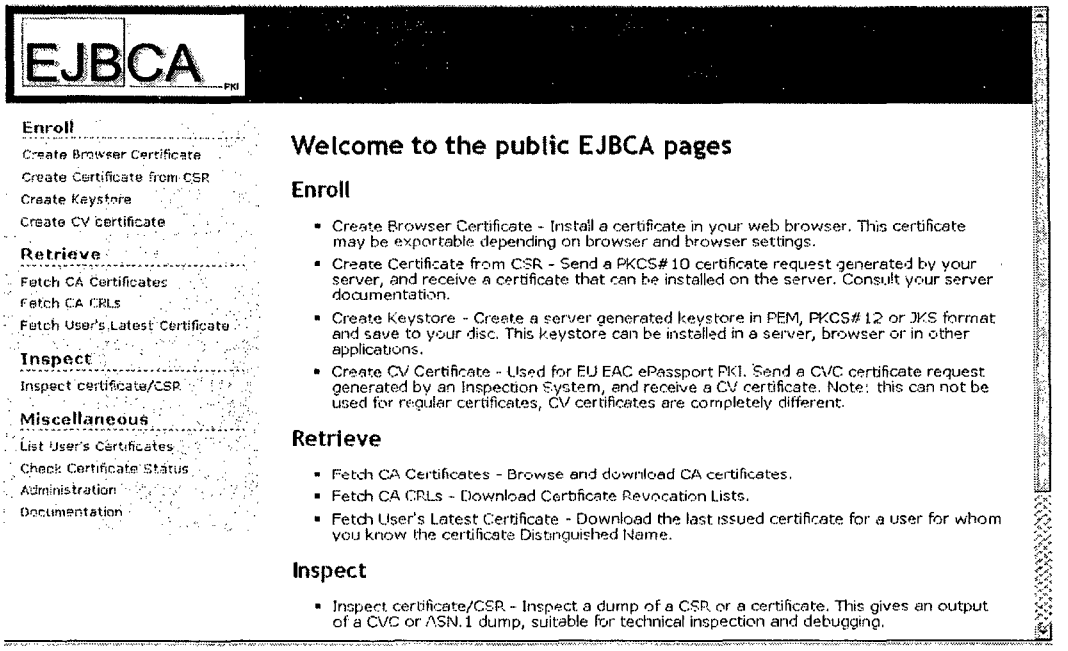
Para reducir el carácter formal del ambiente en el que se desarrollen las Para la demostración de la hipótesis será en base a los datos recolectados en la ficha anexo 3.

## 6. RESULTADOS

### 6.1. PRUEBA PILOTO

Con la finalidad de poner a prueba el software EJBCA y se encuentre en funcionamiento, se ejecutó la aplicación y se realizó las capturas de las pantallas correspondientes, tal como se muestra a continuación:

Figura 01: Captura de la pantalla principal después de haber instalado el EJBCA




Se ha podido verificar que el software de EJBCA encargado de generar los certificados digitales se encuentra en correcto funcionamiento, no presentando ningún tipo de error.

**Figura 02: Pantalla para registrar los certificados digitales**

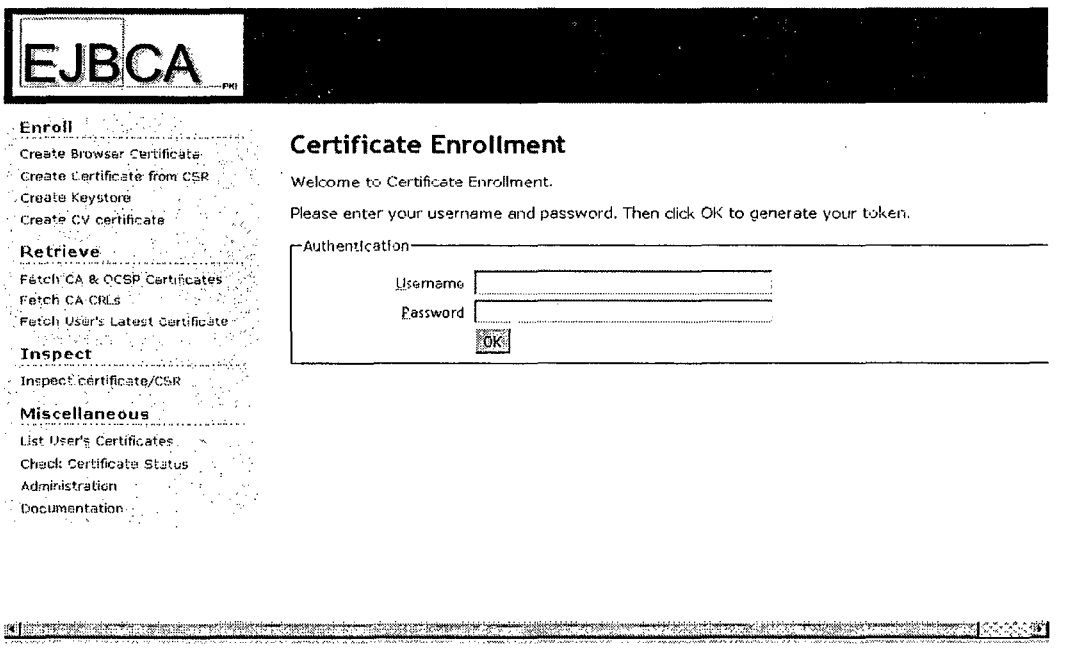


**Figura 03: Pantalla para registrar la solicitud de certificado digital**

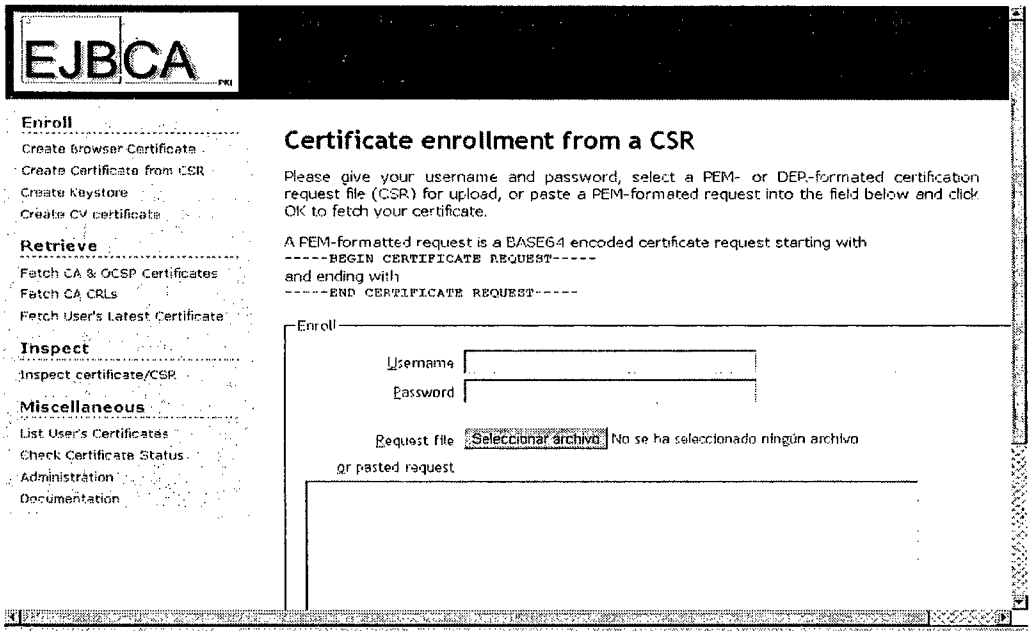
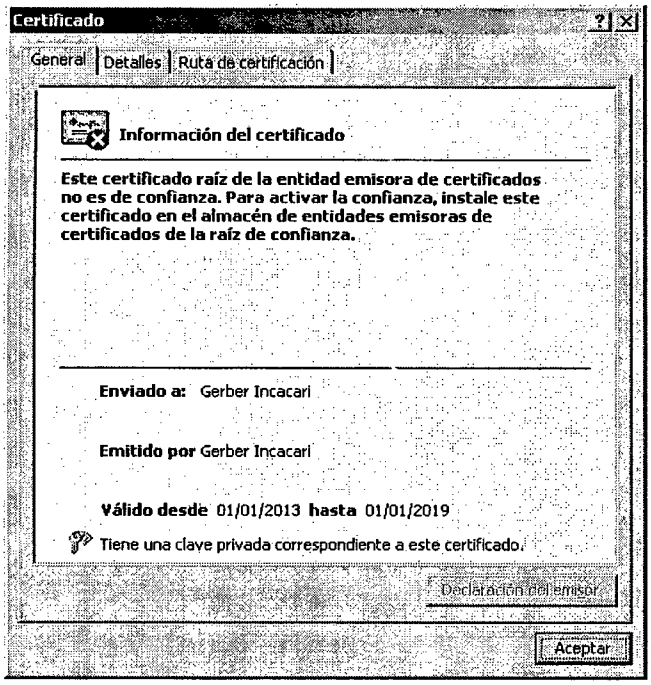



**Figura 04: Pantalla para verificar el estado de un certificado digital**

**Figura 05: Imagen con la generación de un certificado digital de prueba**




## Resultados obtenidos

**Tabla N° 1: Resumen de caso**

| | Casos | | | | | |
|---|---|---|---|---|---|---|
| | Válido | | Perdidos | | Total | |
| | N | Porcentaje | N | Porcentaje | N | Porcentaje |
| $Documentos[a] | 100 | 100,0% | 0 | 0,0% | 100 | 100,0% |

Se aprecia la evaluación de 100 documentos electrónicos utilizados para la presente investigación.

**Tabla N° 2: Tipo de documento**

| | | Frecuencia | Porcentaje | Porcentaje válido | Porcentaje acumulado |
|---|---|---|---|---|---|
| Válido | PDF | 20 | 20,0 | 20,0 | 20,0 |
| | DOC | 23 | 23,0 | 23,0 | 43,0 |
| | JPG | 17 | 17,0 | 17,0 | 60,0 |
| | TXT | 16 | 16,0 | 16,0 | 76,0 |
| | XLS | 24 | 24,0 | 24,0 | 100,0 |
| | Total | 100 | 100,0 | 100,0 | |

Se parecía en la tabla 2 que el 24% de archivos corresponde al tipo Excel (XLS) siendo uno de los más altos, por otra parte el 23% de los documentos corresponde a archivos de Word y seguido por los documentos en formato PDF con el 20%.

**Tabla N° 3: Nivel de cifrado**

| | | Frecuencia | Porcentaje | Porcentaje válido | Porcentaje acumulado |
|---|---|---|---|---|---|
| Válido | Bajo | 31 | 31,0 | 31,0 | 31,0 |
| | Medio | 33 | 33,0 | 33,0 | 64,0 |
| | Alto | 36 | 36,0 | 36,0 | 100,0 |
| | Total | 100 | 100,0 | 100,0 | |

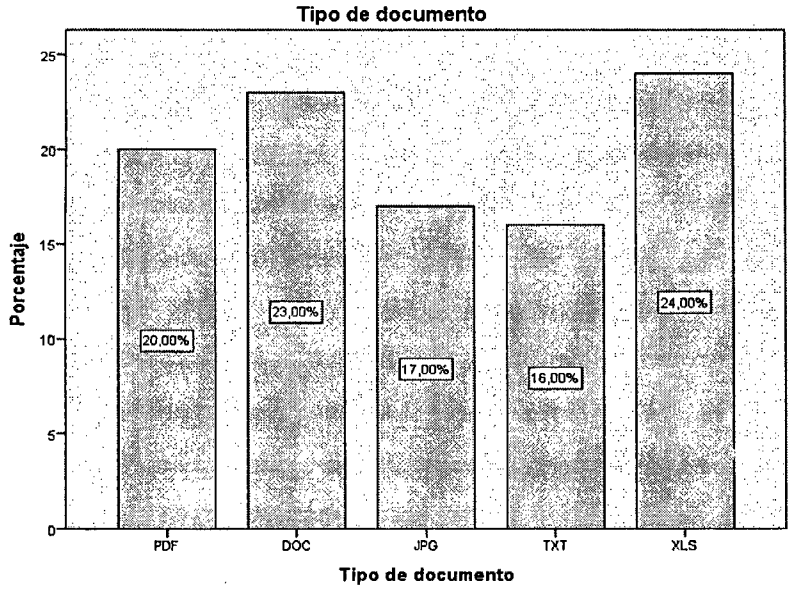
Se aprecia en la tabla N° 3 se aprecia que el el 36 por ciento de los casos tiene un cifrado alto, seguido de por los documentos cifrados medio con el 33 por ciento.

**Tabla N° 4: Resumen de procesamiento de casos**

| | Casos | | | | | |
|---|---|---|---|---|---|---|
| | Válido | | Perdidos | | Total | |
| | N | Porcentaje | N | Porcentaje | N | Porcentaje |
| Tipo de documento * Nivel de cifrado | 100 | 100,0% | 0 | 0,0% | 100 | 100,0% |

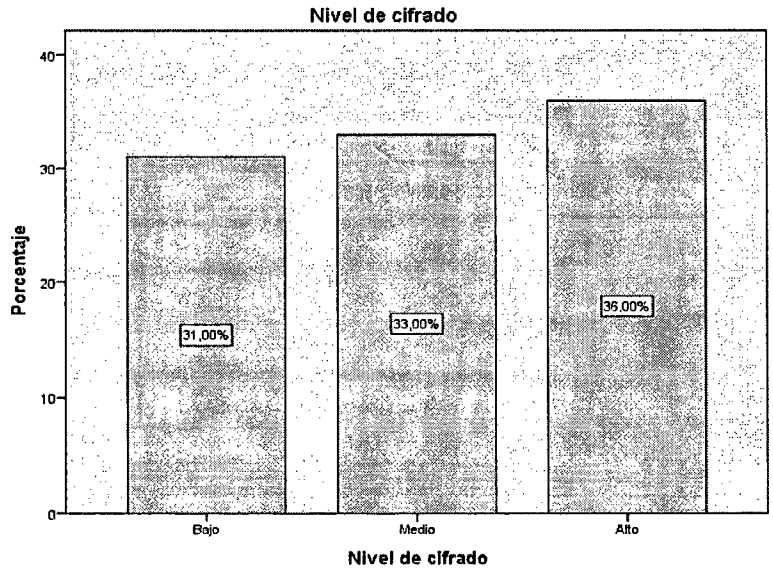
De la tabla N°4 se concluye que se pudo lograr cifrar el 100% de los documentos, es decir los 100 documentos.

**Imagen N° 1: Nivel de cifrado**




De la imagen N° 1, podemos afirma que el 24% y 23% provienen de documentos de Excel (XLS) y documentos de Word(DOC).

**Imagen N° 2: Nivel de cifrado**




De la imagen N° 2, el mayor porcentaje corresponde a documentos cifrado alto en un 36%, seguido por el cifrado de medio con el 33%.

**Hipótesis Planteadas**

**Hipótesis general:**

La seguridad de los documentos electrónicos mediante el cifrado y la firma digital por certificados digitales mejora a un grado alto.

Esto se puede afirmar debido a que ningún documento a podido ha podido ser recuperado se concluye que son seguros debido que dichos archivos presentan cifrado de grado alto.

**Hipótesis específico:**

1. El desarrollo del marco teórico de la autoridad certificadora y de la entidad de registro permite entender los conceptos PKI. Los conceptos desarrollados en la presente investigación permitieron entender mejor el concepto de infraestructura de clave pública PKI.
2. El Desarrollo de un modelo de Autoridad Certificadora (AC) permite validar la emisión de certificados digitales.

   El software EJBCA implementado permitió validar la emisión de certificados digitales.
3. El desarrollo de un modelo de Autoridad de Registro (AR) permite a los usuarios expedir sus certificados

   Toda Autoridad de registro al validar la identidad del usuario permitió expedir los certificados digitales directamente al estudiante.
4. La configuración de un sistema de administración de certificados y distribución permite una mejor administración de los certificados digitales a ser distribuidos.

La implementación del EJBCA asi como la administración de los certificados digitales permitió administrar mejor la distribución del as mismas.

5. La implementación del software PKI permite dar soporte a la generación a los certificados digitales.

   Se pudo verificar que la herramienta EJBCA empleada en el presente investigación permitió dar soporte a la generación de certificados digitales.

6. El cifrado y firma de documentos electrónicos de los estudiantes de la FIIS-UNAC permite aumentar la seguridad.

   Tal como se pudo apreciar en las tablas N° 6, 7 , 8, 9, 10 , 11 que ninguno de los archivos se pudo recuperar su contenido debido a que este se encontraban cifrados

**Conclusión**

- La seguridad de los documentos electrónicos mediante el cifrado y la firma digital por certificados digitales mejora a un grado alto, esto se puede afirmar debido a que ningún documento a podido ha podido ser recuperado, por lo que se concluye que son seguros debido que dichos archivos presentan cifrado de grado alto.
- El desarrollo del marco teórico de la autoridad certificadora y de la entidad de registro permite entender los conceptos PKI. Los conceptos desarrollados en la presente investigación permitieron entender mejor el concepto de infraestructura de clave pública PKI.
- El Desarrollo de un modelo de Autoridad Certificadora (AC) permite validar la emisión de certificados digitales por medio de uso del software EJBCA permitió validar la emisión de certificados digitales.
- El desarrollo de un modelo de Autoridad de Registro (AR) permite a los usuarios expedir sus certificados el cual permitió validar la identidad del usuario



y al mismo tiempo expedir los certificados digitales directamente al estudiante.

- La implementación del EJBCA asi como la administración de los certificados digitales permitió administrar mejor la distribución del as mismas.
- Se verificó que la herramienta EJBCA empleada en el presente investigación permitió dar soporte a la generación de certificados digitales.
- El cifrado y firma de documentos electrónicos de los estudiantes de la FIIS-UNAC permite aumentar la seguridad, debido a que ninguno de los archivos utilizados se pudo recuperar el contenido ya que estos se encontraban cifrados.

## 7. DISCUSION

Primero. Según la tesis de R. Blanco de título análisis de algoritmos criptográficos y su aplicación al cifrado de archivos hasta el momento a demostrado niveles de seguridad razonables, dependiendo del algoritmo que se utilice lo cual beneficia su uso en el cifrado y la firma digital por certificados digitales.

Segundo. El desarrollo del marco teórico de la autoridad certificadora y de la entidad de registro permite entender los conceptos PKI. En la Tesis de A. de la Torre con título Análisis y diseño de una autoridad certificadora, afirma que el desarrollo de un modelo de Autoridad Certificadora (AC) permite implementar mecanismos de autenticación basados en certificados digitales de llave pública.

Tercero. La implementación del EJBCA así como la administración de los certificados digitales permitió administrar mejor la distribución de las mismas, en la Tesis de A. de la Torre con título Análisis y diseño de una autoridad certificadora, afirma que la mayoría de las veces la implementación de una autoridad certificadora se puede hacer mediante software diseñado para tal efecto, tal como se ha realizado en la presente investigación.

Cuarto. Según la tesis de W. García de título Implementación de firma digital en una plataforma de comercio electrónico, recomienda el uso de una infraestructura adecuada que permita firmar documentos y contratos en forma digital mediante el uso de certificados digitales.

## 8. REFERENCIALES


A. de la Torre (2009). *Analisis y diseño de una autoridad certificadora*, Instituto Politecnico NacionalMexico

A. Nash, (2002). *PKI infraestructura de claves públicas (la mejor tecnología para implernentar y administrar la seguridad electrónica de su negocio,* McGRAW-HILL, Ed.

A. Pierre-Muller, (1997).*Modelado de Objetos con UML.* Eyrolles Barcelona.

Committee on Government Reform, House of Representatives. (2001). *Advances and Remaining Challenges to Adoption of Public Key Infrastructure Technology, Report to the Chairman, Subcommittee on Government Efficiency, Financial Management and Intergovernmental Relations.*

*J. Leung., A. Jafri. (2003). PKI Deployment - Business Issues, FundServ Inc.*

L. Craig. (2003). *UML y PATRONES. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado. Segunda Edición. Prentice.* Hall Madrid.

M. Rivolta,. and Prandini, P, (2002). *Argentine Public Key Infrastructure Development - a comparative study of PKI experiences in Latin America, Internet Society.* http://inet2002.org/CD-ROM/lu65rw2n/papers/g11-a.pdf

OASIS PKI Action Plan, (2004). *Prepared and Published by the OASIS Public Key Infrastructure (PKI) Technical Committee (TC),* February 22, Version: 1.0

P. Gutmann, (2004). *How to build an X.509 PKI that works, University of Auckland.*

R. Blanco, (2010). *Analisis de algoritmos criptograficos y su aplicacion al cifrado de archivos*, Instituto Politecnico Nacional, Mexico

United States General Accounting Office. (2001). *Advances and Remaining Challenges to Adoption of Public Key Infrastructure Technology.*

W. García, (2009). *Tesis Implementación de firma digital en una plataforma de comercio electrónico*, PUCP, Perú

## 9. APENDICE

**Tabla N° 5: Tipo de documento por Nivel de cifrado**

| | | | Nivel de cifrado | | | |
|---|---|---|---|---|---|---|
| | | | Bajo | Medio | Alto | Total |
| Tipo de documento | PDF | Recuento | 6 | 8 | 6 | 20 |
| | | % del total | 6,0% | 8,0% | 6,0% | 20,0% |
| | DOC | Recuento | 7 | 7 | 9 | 23 |
| | | % del total | 7,0% | 7,0% | 9,0% | 23,0% |
| | JPG | Recuento | 3 | 8 | 6 | 17 |
| | | % del total | 3,0% | 8,0% | 6,0% | 17,0% |
| | TXT | Recuento | 7 | 3 | 6 | 16 |
| | | % del total | 7,0% | 3,0% | 6,0% | 16,0% |
| | XLS | Recuento | 8 | 7 | 9 | 24 |
| | | % del total | 8,0% | 7,0% | 9,0% | 24,0% |
| Total | | Recuento | 31 | 33 | 36 | 100 |
| | | % del total | 31,0% | 33,0% | 36,0% | 100,0% |

**Tabla N° 6: Tipo de documento por Primer archivo**

| | | | Primer archivo | |
|---|---|---|---|---|
| | | | Sin obtener contenido | Total |
| Tipo de documento | PDF | Recuento | 20 | 20 |
| | | % | 100,0% | 100,0% |
| | DOC | Recuento | 23 | 23 |
| | | % | 100,0% | 100,0% |
| | JPG | Recuento | 17 | 17 |
| | | % | 100,0% | 100,0% |
| | TXT | Recuento | 16 | 16 |
| | | % | 100,0% | 100,0% |
| | XLS | Recuento | 24 | 24 |
| | | % | 100,0% | 100,0% |
| Total | | Recuento | 100 | 100 |
| | | % | 100,0% | 100,0% |

**Tabla N° 7: Tipo de documento por Segundo archivo**

| | | | Segundo archivo | |
|---|---|---|---|---|
| | | | Sin obtener contenido | Total |
| Tipo de documento | PDF | Recuento | 20 | 20 |
| | | % | 100,0% | 100,0% |
| | DOC | Recuento | 23 | 23 |
| | | % | 100,0% | 100,0% |
| | JPG | Recuento | 17 | 17 |
| | | % | 100,0% | 100,0% |
| | TXT | Recuento | 16 | 16 |
| | | % | 100,0% | 100,0% |
| | XLS | Recuento | 24 | 24 |
| | | % | 100,0% | 100,0% |
| Total | | Recuento | 100 | 100 |
| | | % | 100,0% | 100,0% |

**Tabla N° 8: Tipo de documento por Tercer archivo**

| | | | Tercer archivo | |
|---|---|---|---|---|
| | | | Sin obtener contenido | Total |
| Tipo de documento | PDF | Recuento | 20 | 20 |
| | | % | 100,0% | 100,0% |
| | DOC | Recuento | 23 | 23 |
| | | % | 100,0% | 100,0% |
| | JPG | Recuento | 17 | 17 |
| | | % | 100,0% | 100,0% |
| | TXT | Recuento | 16 | 16 |
| | | % | 100,0% | 100,0% |
| | XLS | Recuento | 24 | 24 |
| | | % | 100,0% | 100,0% |
| Total | | Recuento | 100 | 100 |
| | | % | 100,0% | 100,0% |

**Tabla N° 9: Tipo de documento por Cuarto archivo**

| | | | Cuarto archivo | |
|---|---|---|---|---|
| | | | Sin obtener contenido | Total |
| Tipo de documento | PDF | Recuento | 20 | 20 |
| | | % | 100,0% | 100,0% |
| | DOC | Recuento | 23 | 23 |
| | | % | 100,0% | 100,0% |
| | JPG | Recuento | 17 | 17 |
| | | % | 100,0% | 100,0% |
| | TXT | Recuento | 16 | 16 |
| | | % | 100,0% | 100,0% |
| | XLS | Recuento | 24 | 24 |
| | | % | 100,0% | 100,0% |
| Total | | Recuento | 100 | 100 |
| | | % | 100,0% | 100,0% |

**Tabla N° 10: Tipo de documento por Quinto archivo**

| | | | Quinto archivo | |
|---|---|---|---|---|
| | | | Sin obtener contenido | Total |
| Tipo de documento | PDF | Recuento | 20 | 20 |
| | | % | 100,0% | 100,0% |
| | DOC | Recuento | 23 | 23 |
| | | % | 100,0% | 100,0% |
| | JPG | Recuento | 17 | 17 |
| | | % | 100,0% | 100,0% |
| | TXT | Recuento | 16 | 16 |
| | | % | 100,0% | 100,0% |
| | XLS | Recuento | 24 | 24 |
| | | % | 100,0% | 100,0% |
| Total | | Recuento | 100 | 100 |
| | | % | 100,0% | 100,0% |

**Tabla N° 11: Tipo de documento por Sexto archivo**

| | | | Sexto archivo | |
|---|---|---|---|---|
| | | | Sin obtener contenido | Total |
| Tipo de documento | PDF | Recuento | 20 | 20 |
| | | % | 100,0% | 100,0% |
| | DOC | Recuento | 23 | 23 |
| | | % | 100,0% | 100,0% |
| | JPG | Recuento | 17 | 17 |
| | | % | 100,0% | 100,0% |
| | TXT | Recuento | 16 | 16 |
| | | % | 100,0% | 100,0% |
| | XLS | Recuento | 24 | 24 |
| | | % | 100,0% | 100,0% |
| Total | | Recuento | 100 | 100 |
| | | % | 100,0% | 100,0% |

## 10. ANEXOS

**ANEXO 1**
**FICHA DE RECOLECCION DE DOCUMENTOS ELECTRONICOS**
**NRO FICHA: _____**

1. **Datos Generales**
   - Nombre : _____
   - Escuela profesional : _____
   - Sexo : _____
   - Fecha : _____
2. **Identificación del documento electrónico**
   - Nombre del documento electrónico : _____
   - Tipo [ ] DOC [ ]XLS [ ]PDF [ ]TXT
   - Tamaño: _____
   - Fecha de creación: _____

**ANEXO 2**
**FICHA DE CIFRADO DEL DOCUMENTOS ELECTRONICOS**

| N° | Nro Ficha | Nombre del documento electrónico | Tipo | Tamaño | Nivel de Cifrado |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| ... | | | | | |
| 100 | | | | | |

**Nivel de Cifrado: Alto, medio, bajo**

**ANEXO 3**
**FICHA DE RESULTADOS**
**NRO FICHA: _____**

1. **Datos Generales**
   - Nombre : _____
   - Escuela profesional : _____
   - Sexo : _____
   - Fecha : _____

2. **Registro de resultados**

| N° | Nro Ficha | Nombre del documento electrónico | Nivel de Cifrado | Tiempo utilizado | Éxito | Parcial | Fracaso |
|---|---|---|---|---|---|---|---|
| 1 | | | Bajo | | | | |
| 2 | | | Bajo | | | | |
| 3 | | | Medio | | | | |
| 4 | | | Medio | | | | |

**Glosario de términos**

**Clave**: Contraseña de acceso a un sistema o software.

**Clúster**: Conjunto de computadores que simulan ser un único computador.

**CRL**: Lista de revocación de certificados digitales. Es usada para informar si algún certificado digital ya no es válido.

**EC**: Entidad de Certificación.

**ER**: Entidad de Registro. La EC y ER forman parte de la IOFE.

**EJBCA**: Software PKI de código abierto para la administración de una Autoridad Certificadora.

**HSM**: Hardware Security Module (Modulo de Seguridad en Hardware). Appliance criptográfico.

**IOFE**: Infraestructura Oficial de Firma Electrónica. Nombre específico para una plataforma PKI con respaldo legal. Nombre acuñado por INDECOPI.

**JBoss**: Servidor de aplicaciones basado en lenguaje JAVA, propiedad de la empresa REDHAT.

**LDAP**: Lightweight Directory Access Protocol o Protocolo Ligero de Acceso a Directorios que se encuentra en la versión 3 estandarizado por el RFC 4510. Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. El ejemplo más común es el directorio telefónico, que consiste en una serie de nombres (personas u organizaciones) que están ordenados alfabéticamente por nombre teniendo una dirección y un número de teléfono adjunto. En el caso específico de una PKI es utilizado como repositorio de certificados digitales, debido a su velocidad de búsqueda y su simplicidad.

**Llave pública y privada (par de claves):** La estructura PKI se basa en tres componentes principales los cuales son: par de claves, certificado digital y entidad certificadora. De estos tres puntos el par de claves corresponden a un algoritmo asimétrico (ejemplo RSA) por el cual se generan dos claves distintas pero con una relación matemática muy estrecha, por la cual una llave pública se corresponde con una única llave privada. En otras literaturas se puede encontrar como clave pública y clave privada, pero por motivos de entendimiento los llamaremos "claves".

**OCSP**: Online Certificate Status Protocol, es un método para determinar el estado de revocación de un certificado digital (X.509) usando otros medios que no sean el uso de **CRL** (Listas de Revocación de Certificados). Este protocolo se describe en el RFC 2560 y está en el registro de estándares de Internet.

**OID**: Object Identifier, o Identificador de Objeto, es una secuencia de números que se asignan jerárquicamente y que permite identificar objetos en la red, siendo usados con gran cantidad de protocolos.

**OpenSSL**: Herramienta de código abierto para la administración y bibliotecas que suministran funciones criptográficas (SSL y TLS).

**PKI**: Infraestructura de llave pública, es una combinación de hardware, software, políticas y procedimientos de seguridad que permiten la ejecución de procesos con garantías de operaciones criptográficas como el cifrado, la firma digital o el no repudio de transacciones electrónicas. Una PKI básicamente está formada por una entidad certificadora la cual cuenta con un certificado digital raíz, la cual a la vez emite certificados digitales para otros terceros formado una red de confianza a la cual se denomina webtrust o red de confianza.

**SSL**: Secure Sockets Layer o protocolo de capa de conexión segura, son protocolos criptográficos que proporcionan comunicaciones seguras por una red o Internet.