DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

TRABAJO DE TITULACIÓN PREVIO LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA

TEMA: IMPLEMENTACIÓN DE UNA ARQUITECTURA PKI PARA EL EJÉRCITO ECUATORIANO, UTILIZANDO SOFTWARE LIBRE

AUTOR: CRUZ ALMEIDA FIDEL OSWALDO

DIRECTOR: ING. ESCOBAR QUIÑA LUIS ALBERTO

SANGOLQUÍ, 2017

UNIVERSIDAD DE LAS FUERZAS ARMAS – ESPE

Departamento De Ciencias De La Computación

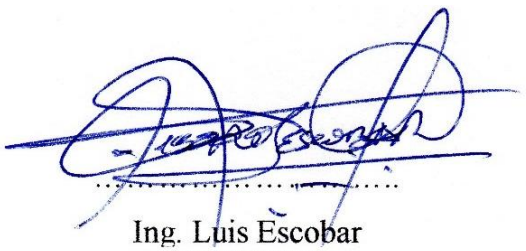
Carrera de Ingeniería en Sistemas e Informática

CERTIFICADO

En calidad de Director del Proyecto, **CERTIFICO**: que el trabajo titulado "IMPLEMENTACIÓN DE UNA ARQUITECTURA PKI PARA EL EJÉRCITO ECUATORIANO, UTILIZANDO SOFTWARE LIBRE", realizado por el Sr. Cbop. De I.M. Fidel Oswaldo Cruz Almeida, fue revisado en su totalidad y analizado con el software anti-plagio; y el mismo cumple con los requisitos teóricos, científicos y legales establecidos en la normativa de la Universidad de Fuerzas Armadas – ESPE y de la Dirección de Comunicaciones y Sistemas del Ejército.

Debido a que constituye un trabajo de excelente contenido científico y apoyará al desarrollo tecnológico e institucional el Ejército del Ecuador, se recomienda su publicación y autorizó su presentación ante los organismos pertinentes.

Sangolquí, 03 de mayo de 2017



Ing. Luis Escobar
DIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMAS – ESPE

Departamento De Ciencias De La Computación

Carrera de Ingeniería en Sistemas e Informática

AUTORÍA

Yo, Cbop. De I.M. Fidel Oswaldo Cruz Almeida, portador de la cédula de ciudadanía Nro. 060414896-5, **DECLARO:** que el trabajo de titulación "IMPLEMENTACIÓN DE UNA ARQUITECTURA PKI PARA EL EJÉRCITO ECUATORIANO, UTILIZANDO SOFTWARE LIBRE", ha sido desarrollado con base a una investigación exhaustiva, respetando los derechos intelectuales de terceros, conforme a las fuentes que se incorpora en la bibliografía.

Consecuentemente este trabajo es de mi autoría, y en virtud de esta declaración, me responsabilizo del contenido veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 03 de mayo de 2017

Fidel Oswaldo Cruz Almeida
Cbop. De I.M.
AUTOR DEL PROYECTO

UNIVERSIDAD DE LAS FUERZAS ARMAS – ESPE

Departamento De Ciencias De La Computación

Carrera de Ingeniería en Sistemas e Informática

AUTORIZACIÓN

Yo, Cbop. De I.M. Fidel Oswaldo Cruz Almeida, portador de la cédula de ciudadanía Nro. 060414896-5, **AUTORIZO:** que el trabajo de titulación "IMPLEMENTACIÓN DE UNA ARQUITECTURA PKI PARA EL EJÉRCITO ECUATORIANO, UTILIZANDO SOFTWARE LIBRE", sea publicado en la Biblioteca Virtual de la Universidad de Fuerzas Armadas – ESPE; cuyo contenido, ideas y criterios con de mi autoría y responsabilidad.

Sangolquí, 03 de mayo de 2017

Fidel Oswaldo Cruz Almeida
Cbop. De I.M.
AUTOR DEL PROYECTO

DEDICATORIA

El presente trabajo se lo dedico a todos aquellos que me acompañaron en este largo andar, y sobre todo a mis abuelitos Delia Castelo y Pedro Almeida (†), quienes han sido la base del aprendizaje de valores y principios, como el amor, el trabajo y la constancia… "*No importa que tan duro se ponga el camino, lo importante es que tan fuertes seámonos para avanzar".*

Oswaldo Cruz

AGRADECIMIENTO

Agradezco a toda mi familia y amigos, pero de sobremanera mi madre Anita, a mi esposa Rebeca, y a mi hija Valeria por ser la fuente de inspiración para seguir adelante… *Un paso a la vez se llega lejos, pero es mejor caminar en compañía de quienes amamos.*

Gracias por ser mis acompañantes.

Oswaldo Cruz

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En esencia el presente documento estudia lo que es una infraestructura de clave pública o PKI, además de las leyes que regulan el uso de dicha infraestructura y las herramientas a ser utilizadas para la implementación de la misma. Una infraestructura de clave pública o PKI, es un conjunto conformado por software, hardware y políticas de seguridad, con el fin de garantizar la integridad y confidencialidad de la información. Esta infraestructura se basa en la criptografía asimétrica, es decir se basa en el uso de un par de claves, una que cifra la información y otra que la descifra, según el caso una puede ser privada y la otra pública. Por otro lado, tenemos las leyes que rigen todos los procesos orientados a la implementación y uso de la PKI. Para ello también se debe tener un profundo conocimiento de la "Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos", y por las políticas emitidas por la ARCOTEL (Agencia de Regulación y Control de las Telecomunicaciones); enfocándonos específicamente al Capítulo II "De los Certificados de Firma Electrónica", y al Capítulo III "De las Entidades de Certificación de Información". Y finalmente, se analizan las herramientas a ser utilizadas para la implementación de la infraestructura, y el proceso de instalación y configuración de dichas herramientas y plataformas a ser utilizas. Además de esto se va a programar una nueva herramienta llamada pkiEjercitoEC, cuyo fin es acoplar toda esta infraestructura a la realidad tecnológica del Ejército Ecuatoriano. Como podemos apreciar el presente documento más que teórico es práctico, y la implementación de esta nueva tecnología en las Fuerzas Armadas del Ecuador, permitirán una mejora significativa de las Seguridades Informáticas e incrementará el nivel de confianza en los Sistemas que desarrollamos.

PALABRAS CLAVE:

- ✓ **INFRAESTRUCTURA DE CLAVE PÚBLICA**
- ✓ **FIRMA ELECTRÓNICA**
- ✓ **SEGURIDAD INFORMÁTICA**
- ✓ **PROGRAMACIÓN**
- ✓ **CONFIGURACIÓN LINUX**

ABSTRACT

In essence this document explores what is a PKI, or public key infrastructure as well as the laws governing the use of such infrastructure and tools to be used for the implementation of the same. A PKI, or public key infrastructure is a set consisting of software, hardware and security policies, in order to ensure the integrity and confidentiality of the information. This infrastructure is based on asymmetric cryptography, i.e. based on the use of a pair of keys, one that encrypts information and other decoding, depending on the case one can be private and the other public. On the other hand, we have the laws that govern all of the processes aimed at the implementation and use of the PKI. This also should be a deep knowledge of the "law of electronic commerce, data messages and electronic signatures", and by the policies issued by the ARCOTEL (Agency for the regulation and Control of telecommunications); focusing specifically on chapter II "Of the certificates of signature electronic", and chapter III "certification of information entities". And finally, is analyzed the tools to be used for the implementation of the infrastructure, and the process of installation and configuration of these tools and platforms to be use. In addition to this, you will schedule a new tool called pkiEjercitoEC, which aims to connect all this infrastructure to the technological reality of the Ecuadorian army. As can appreciate the present document rather than theoretical is practical, and the implementation of this new technology in them forces armed of the Ecuador, will allow an improves significant of them securities computer e will increase the level of confidence in the systems that develop.

KEYWORDS:

- ✓ **PUBLIC KEY INFRASTRUCTURE**
- ✓ **ELECTRONIC SIGNATURE**
- ✓ **COMPUTER SECURITY**
- ✓ **PROGRAMMING**
- ✓ **LINUX CONFIGURATION**

CAPITULO I
INTRODUCCIÓN

1.1 Antecedentes

El Ejército Ecuatoriano, es la entidad encarga de garantizar la seguridad y soberanía territorial, a la vez que apoya al desarrollo del país.

En la Fuerza Terrestre (F.T.), la Dirección de Comunicaciones e Informática (DCI), es el órgano rector de las comunicaciones e informática.

Actualmente la DCI, está buscando el mejoramiento de la seguridad de información, por cuanto la información clasificada se halla a disposición de personal que no es calificado para usarla, violentando las normas para el manejo de información calificada especificada en el Reglamento RT-3-IV. El solo hecho que dicha información caiga en manos de agentes antagónicos, es sumamente peligroso para la institución.

Por otra parte, la seguridad informática ha ido evolucionando rápidamente; ya que, con cada nuevo método de ataque cibernético, se generan nuevas formas de protección, una de estas formas de protección es la implementación de una arquitectura PKI.

Está arquitectura está siendo muy utilizada a nivel mundial, ya que incrementa la seguridad digital frente a una infraestructura tradicional.

La gran mayoría de los ataques informáticos se los podría realizar utilizando métodos de suplantación de identidad e ingeniería social; buscando obtener, modificar o destruir datos de interés institucional.

La infraestructura de llave pública o PKI, es la integración hardware, software, políticas y procedimientos de emisión de certificados, que pueden ser usados para diferentes propósitos, como son: Firmas Electrónicas, Correo Seguro, Identificación de Usuarios, Web Segura, etc. Además, que garantiza la autenticidad, confidencialidad, integridad y el no repudio de la información.

1.2 Problemática

La F.T., al ser una Institución de carácter militar, puede ser víctima de un ataque de tipo cibernético, ya que la información es un objetivo estratégico para el enemigo durante una guerra o para agentes antagónicos.

Incluso en tiempos de paz un ataque informático es muy probable, ya que existen muchos hackers, crackers, etc., que realicen intrusiones a los aplicativos, bases de dato, correo electrónico, etc., para obtener información con fines delictivos.

Otro problema que actualmente tiene el Ejército, es que el sistema de gestión documental SIGOB, no tiene la funcionalidad para poder firmar los documentos electrónicamente, por lo tanto, estos documentos pueden ser facialmente alterados.

Además de que muchas de las unidades de la Fuerza son muy lejanas, por lo tanto, no tienen acceso a la red intranet; teniendo que conectarse con el Comando de la Fuerza a través del Internet, siendo esta comunicación insegura. Incluso en algunos casos se remiten documentos referentes a operaciones militares en texto claro y a través de correos no seguros.

Se requiere que la información de la Fuerza Terrestre que viaja a través de la Internet, sea encriptada para evitar la fácil interceptación de la misma, y a su vez se requiere incrementar el nivel de seguridad en la autenticación de usuarios y los sistemas alojados en los servidores mediante certificados digitales.

También se requiere usar dichos certificados para la implementación de un Protocolo Ligero/Simplificado de Acceso a Directorios o LDAP en la DCI.

A la vez que se requiere utilizar dichos certificados digitales para la comunicación entre servidores de la Fuerza, tanto entre los servidores de aplicaciones, bases de datos, servidores de correo y entidades de certificación delegadas.

1.3 Justificación

1.3.1 Justificación operativa

La Fuerza Terrestre por su naturaleza requiere incrementar la seguridad informática, para el manejo de documentos de carácter clasificado, que van desde reservado hasta secretísimo; por lo que se requiere firmar y cifrar mensajes de correo electrónico con los certificados que se emitan a través de la Entidad Certificadora que se implementará con la tesis.

Cabe recalcar que por ser una institución militar y por conservar el secreto en las operaciones militares, no pueden depender de terceros para administrar y gestionar esta arquitectura, y por ende es necesario que sea implementada internamente y se lo haga con personal perteneciente a la Institución.

1.3.2 Justificación técnica

Es necesario implementar una Arquitectura PKI para que los sistemas informáticos pertenecientes al SIFTE, puedan emplear certificados digitales para los diferentes propósitos: autenticación, firma electrónica y cifrado; y de ese modo garantizar la integridad y autenticidad de la información, a la vez que no se permita el repudio de la misma.

Por otro lado, el sistema de gestión documental SIGOB que actualmente dispone la Fuerza y posteriormente el QUIPUX, requieren que sus documentos tengan valides legal, y para cuyo fin se requiere implementar la firma electrónica, usando para dicho propósito los certificados digitales generados por Entidad Certificadora.

También se prevé implementar un LDAP o Protocolo Ligero de Acceso a Directorios, y un protocolo SSL para web segura, además del uso de los certificados para cifrar, firmar y enviar correos seguros.

1.4 Objetivos

1.4.1 Objetivo General

Implementar una arquitectura PKI para el Ejército Ecuatoriano, utilizando Software Libre.

1.4.2 Objetivos Específicos

a. Analizar el marco jurídico enfocado a la Infraestructura de Clave Pública.
b. Realizar el diagnóstico de la seguridad de información.
c. Diseñar la Arquitectura PKI ajustándose a la realidad técnica de la fuerza.
d. Implementación de las Autoridades Certificadoras Raíz y Subordinada, Autoridades de Registro, Autoridad de Validación, Autoridad de Sellado de Tiempo y Repositorio de Certificados.
e. Definir las Políticas y Prácticas de Certificación.

1.5 Alcance

Diseño, instalación y configuración de una Autoridad Certificadora Raíz en un servidor independiente y fuera de línea; y una Autoridad Certificadora Subordinada, que emitirá los certificados para a los usuarios finales; además de cuatro autoridades de registro, una para cada División del Ejército.

CAPITULO II
MARCO JURÍDICO

2.1 Ley de comercio electrónico, firmas electrónicas y mensajes de datos

La Ley de Comercio Electrónico, que está vigente en el Ecuador es la No. 2002-67, la cual consta de 64 artículos clasificados en cinco títulos.

En nuestro caso para la implementación de la arquitectura PKI, se enfoca brevemente en el Titulo II referente a firmas electrónicas, certificados de firma electrónica, entidades de certificación de información, organismos de promoción de los servicios electrónicos, y de regulación y control de las entidades de certificación acreditadas.

Citando a esta ley en el Art.- 13 dice que "*Firma electrónica. - 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*" (Ley de comercio electrónico, 2002).

Además, dice que "*la firma electrónica tendrá igual validez y se le reconocerán los mismos efectos jurídicos que a una firma manuscrita en relación con los datos consignados en documentos escritos, y será admitida como prueba en juicio*" (Ley de comercio electrónico, 2002).

También detalla los requisitos para la firma electrónica y las obligaciones del titular, las cuales vemos a continuación:

Tabla 1:

Requisitos para Entidades de Certificación

Art.	Requisito
a.	Ser individual y estar vinculada exclusivamente a su titular;
b.	Que permita verificar inequívocamente la autoría e identidad del signatario, mediante dispositivos técnicos de comprobación establecidos por esta Ley y sus reglamentos;
c.	Que su método de creación y verificación sea confiable, seguro e inalterable para el propósito para el cual el mensaje fue generado o comunicado.
d.	Que, al momento de creación de la firma electrónica, los datos con los que se creare se hallen bajo control exclusivo del signatario; y,
e.	Que la firma sea controlada por la persona a quien pertenece.

Fuente: (Ley de comercio electrónico, 2002)

Como podemos apreciar el requisito más importante hace referencia al método de creación, ya que este debe ser confiable. Este método de creación debe cumplir normas internacionales, y a su vez ofrecer altos niveles de seguridad a fin de garantizar que los certificados ahí generados son válidos y no pueden ser alterados por cualquier persona.

En el caso que no se pueda cumplir con dicho requisito, no se puede implementar una autoridad certificadora. Ya que todo se basa en la confianza que se tenga en esta.

En el siguiente capítulo hace referencia a los certificados de firma electrónica, y a los requisitos que debe cumplir dicho certificado:

Tabla 2:

Certificados de Firma Electrónica

Art.	Requisito
a.	Identificación de la entidad de certificación de información;
b.	Domicilio legal de la entidad de certificación de información;
c.	Los datos del titular del certificado que permitan su ubicación e identificación;
d.	El método de verificación de la firma del titular del certificado;
e.	Las fechas de emisión y expiración del certificado;
f.	El número único de serie que identifica el certificado;
g.	La firma electrónica de la entidad de certificación de información;
h.	Las limitaciones o restricciones para los usos del certificado; e,
i.	Los demás señalados en esta ley y los reglamentos.

Fuente: (Ley de comercio electrónico, 2002)

En el capítulo III habla de las entidades de certificación sus obligaciones y responsabilidades que deben cumplir para emitir certificados electrónicos. También hace referencia a la protección de los datos y la garantía que estas deben ofrecer.

A continuación, citaremos las obligaciones que deben cumplir estas entidades de certificación:

Tabla 3:

Obligaciones de las Entidades de Certificación

Art.	Requisito
a.	Encontrarse legalmente constituidas, y estar registradas en el Consejo Nacional de Telecomunicaciones;
b.	Demostrar solvencia técnica, logística y financiera para prestar servicios a sus usuarios;
c.	Garantizar la prestación permanente, inmediata, confidencial, oportuna y segura del servicio de certificación de información;
d.	Mantener sistemas de respaldo de la información relativa a los certificados;
e.	Proceder de forma inmediata a la suspensión o revocatoria de certificados electrónicos previo mandato de la Superintendencia de Telecomunicaciones, en los casos que se especifiquen en esta ley;
f.	Mantener una publicación del estado de los certificados electrónicos emitidos;
g.	Proporcionar a los titulares de certificados de firmas electrónicas un medio efectivo y rápido para dar aviso que una firma electrónica tiene riesgo de uso indebido;
h.	Contar con una garantía de responsabilidad para cubrir daños y prejuicios que se ocasionaren por el incumplimiento de las obligaciones previstas en la presente ley, y hasta por culpa leve en el desempeño de sus obligaciones. Cuando certifiquen límites sobre responsabilidades o valores económicos, esta garantía será al menos del 5% del monto total de las operaciones que garanticen sus certificados; e,
i.	Las demás establecidas en esta ley y los reglamentos.

Fuente: (Ley de comercio electrónico, 2002)

2.2 Reglamento a la Ley de comercio electrónico.

El reglamento en vigencia es emitido mediante Decreto No. 3496, el cual consta de 23 artículos, en nuestro caso de estudio se enfoca los artículos relacionados a los elementos de la infraestructura de firma electrónica, duración de certificados, listas de

revocación, reconocimiento internacional, responsabilidades de las entidades de certificación y tiempos de sellado.

En el Art. 10, se habla de los elementos que debe contener una infraestructura de firma electrónica y los principios en la cual se basa:

Tabla 4:

Elementos de una Infraestructura de Firma Electrónica

Art.	Requisito
a.	No-discriminación a cualquier tipo de firma electrónica, así como a sus medios de verificación o tecnología empleada;
b.	Prácticas de certificación basadas en estándares internacionales o compatibles a los empleados internacionalmente;
c.	El soporte lógico o conjunto de instrucciones para los equipos de cómputo y comunicaciones, los elementos físicos y demás componentes adecuados al uso de las firmas electrónicas, a las prácticas de certificación y a las condiciones de seguridad adicionales, comprendidas en los estándares señalados en el literal b);
d.	Sistema de gestión que permita el mantenimiento de las condiciones señaladas en los literales anteriores, así como la seguridad, confidencialidad, transparencia y no-discriminación en la prestación de sus servicios; y,
e.	Organismos de promoción y difusión de los servicios electrónicos, y de regulación y control de las entidades de certificación.
f.	No-discriminación a cualquier tipo de firma electrónica, así como a sus medios de verificación o tecnología empleada;
g.	Prácticas de certificación basadas en estándares internacionales o compatibles a los empleados internacionalmente;
h.	El soporte lógico o conjunto de instrucciones para los equipos de cómputo y comunicaciones, los elementos físicos y demás componentes adecuados al uso de las firmas electrónicas, a las prácticas de certificación y a las condiciones de seguridad adicionales, comprendidas en los estándares señalados en el literal b);
i.	Sistema de gestión que permita el mantenimiento de las condiciones señaladas en los literales anteriores, así como la seguridad, confidencialidad, transparencia y no-discriminación en la prestación de sus servicios; y,
j.	Organismos de promoción y difusión de los servicios electrónicos, y de regulación y control de las entidades de certificación.

Fuente: (Ley de comercio electrónico, 2002)

Por otro lado, la duración de los certificados se definirá en un contrato entre el titular de la firma y la entidad que entrega el certificado.

En el artículo 16, se hace referencia al "*reconocimiento internacional de certificados de firma electrónica, la cual indica que los certificados de firma electrónica emitidos en el extranjero tendrán validez legal en Ecuador una vez obtenida la revalidación respectiva emitida por el CONATEL (actualmente ARCOTEL), él deberá comprobar el grado de fiabilidad de los certificados y la solvencia técnica de quien los emite"* (Ley de comercio electrónico, 2002).

Finalmente revisaremos los artículos 18 y 23, que hablan de las responsabilidades de la autoridad certificadora y la autoridad sellado de tiempo.

La Autoridad certificadora o en su defecto de la Autoridad de registro debe verificar la autenticidad de todos los datos que entrega el usuario y que van a permitir la generación de la Firma Electrónica.

El CONATEL actualmente conocido como ARCOTEL podrá realizar auditorías y verificar los datos que contienen los certificados digitales.

Para la Autoridad de sellado de tiempo, su única responsabilidad será grabar en el la hora y fecha exacta en que el mensaje de datos fue recibido por la entidad certificadora; y la fecha y hora exacta en dicho mensaje que fue entregado al destinatario.

2.3 Requisitos para la acreditación de una entidad de certificación.

Para acreditar una entidad de certificación según Decreto Nro. 1356, se debe presentar los siguientes documentos:

Tabla 5:

Requisitos de Certificación de una Entidad Certificadora

Art.	Requisito
a.	Solicitud dirigida a la Secretaria Nacional de Telecomunicaciones, detallando nombres y apellidos completos del representante legal, dirección domiciliaria de la empresa unipersonal o compañía.
b.	Copia de la cédula de ciudadanía del representante legal o pasaporte según corresponda.
c.	Copia del certificado de votación del último proceso eleccionario (correspondiente al representante legal, excepto cuando se trate de ciudadanos extranjeros).
d.	Copia certificada e inscrita en el Registro Mercantil (excepto las instituciones públicas) del nombramiento del representante legal.
e.	Copia certificada debidamente registrada en el Registro Mercantil, de la escritura de constitución de la empresa unipersonal o compañía y reformas en caso de haberlas (exceptos las instituciones públicas).
f.	Original del certificado de cumplimiento de obligaciones emitido por la Superintendencia de Compañías o Bancos y Seguros según corresponda, a excepción de las instituciones del Estado.
g.	Diagrama esquemático y descripción técnica detallada de la infraestructura a ser utilizada, indicando las características técnicas de la misma.
h.	Descripción detallada de cada servicio propuesto y de los recursos e infraestructura disponibles para su presentación. La SENATEL podrá ordenar inspecciones o verificaciones a las instalaciones del peticionario cuando lo considere necesario;
i.	Documentos de soporte que confirmen que se disponen de mecanismos de seguridad para evitar la falsificación de certificados, precautelar la integridad, resguardo de documentos, protección contra siniestros, control de acceso y confidencialidad durante la generación de claves, descripción de sistemas de seguridad, estándares de seguridad, sistemas de respaldo.
j.	Ubicación geográfica inicial, especificando la dirección de cada nodo o sitio seguro.

Art.	Requisito
a.	a. Diagrama técnico detallado de cada "Nodo" o "Sitio Seguro" detallando especificaciones técnicas de los equipos.
b.	b. Información que demuestre la capacidad económica y financiera para la presentación de servicios de certificación de información y servicios relacionados.
c.	c. En caso de solicitud de renovación de la acreditación y de acuerdo con los procedimientos que señala el CONATEL, deberán incluirse los requisitos de carácter técnico, la certificación de cumplimiento de obligaciones por parte de la Superintendencia de Telecomunicaciones, en la que constará el detalle de la imposición de sanciones, en caso de haberlas y el informe de cumplimiento de obligaciones por parte de la Secretaría Nacional de Telecomunicaciones.

Fuente: (ARCOTEL, 2016)

2.4 Acreditación para entidades del Estado.

En el caso del Ejército Ecuatoriano, se debe poner énfasis en el siguiente artículo que textualmente dice: "*Las Instituciones del Estado señaladas en el artículo 118 de la Constitución Política de la República, de acuerdo a lo señalado en la disposición Octava de la Ley 67, podrán prestar servicios como Entidades de Certificación de Información y Servicios Relacionados, previa Resolución emitida por el CONATEL*" (ARCOTEL, 2016).

Las Instituciones públicas obtendrán certificados de firma electrónica, únicamente de las Entidades de Certificación de Información y Servicios Relacionados Acreditadas de derecho público (ARCOTEL, 2016).

2.5 Garantía de Responsabilidad.

En el apartado “h” del artículo 30 de la Ley de Comercio Electrónico, las Entidades de Certificación deberán entregar una garantía monetaria para asegurar que los usuarios puedan ser cubiertos contra daños ocasionados por un mal manejo de sus datos o certificados digitales. Esta garantía será incondicional y de cobro inmediato.

a. Para el primero año de operaciones, la entidad deberá entregar una garantía igual o mayor a cuatrocientos mil dólares (USD $400.000,00). En el contrato deberá incluir el monto asignado a cada usuario, mecanismos de reclamación y restitución de dinero en caso de prejuicios al usuario.

b. Para el segundo año deberá contar con una garantía que estará en función de un valor base por cada certificado y que será establecido por el ARCOTEL.

CAPITULO III
INFRAESTRUCTURA PKI

3.1 Definiciones y conceptos generales

3.1.1 Arquitectura

Una arquitectura en el área de la informática se refiere al diseño de un sistema, tanto su estructura como mecanismos de funcionamiento, incluyendo a las políticas para el desarrollo y uso de los mismos.

3.1.2 Infraestructura de Clave Pública (PKI)

Una infraestructura de clave pública o PKI es un conjunto de medios técnicos, en este caso hardware y software, que combinados con políticas y procedimientos de aseguramiento de la información permiten garantizar la autenticidad del autor y la integridad del contenido de un mensaje de datos.

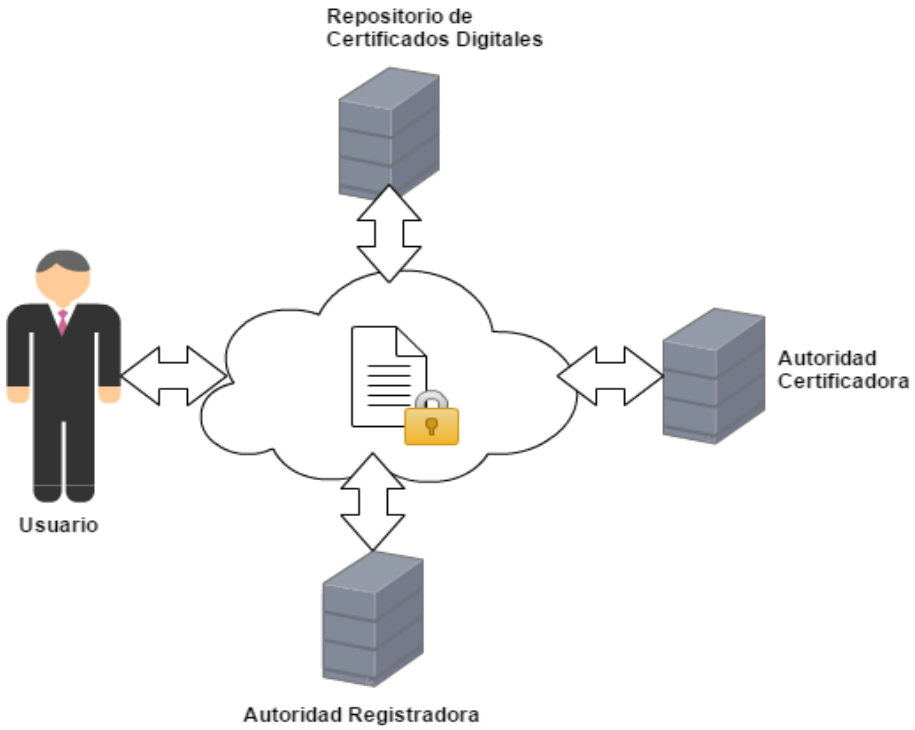


Figura 1: Infraestructura de Clave Pública

En el caso de una arquitectura de PKI tema de este proyecto se puede decir que es el diseño funcional de una infraestructura de clave pública aplicada a la Fuerza Terrestre, a su vez que se busca implementarla para con el fin de aportar al mejoramiento de la seguridad informática en la institución. Para lo cual se requiere de conocer algunos aspectos técnicos como los siguientes:

- Criptografía
- Función Hash
- Clave Publica y Clave Privada
- Certificado Digital
- Firma Electrónica

Cada uno de estos elementos conforma la base de una infraestructura de clave pública.

3.1.3 Criptografía

La criptografía es una técnica de escribir con procedimientos, claves secretas o de un modo enigmático; de tal forma que lo escrito solamente sea inteligible para quien sepa descifrarlo.

El rol que cumple en la criptografía dentro de una infraestructura de clave pública, es muy importante ya que es la base de la confianza de la misma, mientras más complejo sea el nivel de cifrado, más difícil es que se pueda alterar su contenido.

3.1.4 Función Hash

Una función hash es un algoritmo que permite obtener un código resumen de todo el contenido de un documento o fragmento de información, a fin de conocer posteriormente si este fue modificado.

Si un dato es cambiado en el documento, ya sea un punto y coma este código resumen cambia totalmente. Esta función es utilizada conjuntamente con la criptografía para firmado electrónico de documentos.

3.1.5 Clave Publica y Clave Privada

El par de llaves, es una técnica base para una infraestructura de clave pública, en la cual se genera una llave para encriptar un documento o fragmento de información, y otra que permite descifrar dicho contenido.

No se puede descifrar el contenido usando la misma llave que se usó para encriptarla. A esto se le llama criptografía asimétrica.

Si se puede descifrar el contenido con la misma llave con la que se encriptó se le llama criptografía simétrica.

Se utiliza este tipo de técnica criptográfica dentro PKI ya que en base a esté método se general los certificados digitales.

3.1.6 Certificado Digital

Un certificado digital es un archivo que contiene los del propietario de la entidad o autoridad certificadora, más la firma electrónica de dicha entidad.

Un certificado contiene la toda esta información encriptada, usando el método de par de claves, de tal forma que el resto del mundo pueda descifrar dicha información usando la clave pública. Sin embargo, solo el propietario puede encriptar su información con su clave privada. A su vez dicho archivo es enviado a una autoridad certificadora para que valide la información y firme dicho certificado con su llave privada.

3.1.7 Firma Electrónica

La firma electrónica es un método para evitar alteraciones a los documentos digitales, a la vez que permite conocer la procedencia del mismo.

Para firmar un documento el programa obtiene el código resumen o hash del documento, y lo encripta usando la clave privada del propietario, este hash encriptado se adjunta al documento digital.

Para validar el documento se procede a descifrar el hash con la clave pública. Esta clave pública forma parte de un certificado digital, que a su vez es firmado por una entidad o autoridad certificadora con validez legal.

3.1.8 Protocolo Ligero de Acceso a Directorios (LDAP)

Son las siglas de Lightweight Directory Access Protocol (en español Protocolo Ligero/Simplificado de Acceso a Directorios) que hacen referencia a un protocolo a que permite el acceso a un directorio que se encuentra en un servidor de archivos.

En la fuerza terrestre a más de tener un sistema de firmas electrónicas, también ser va a configurar un LDAP, para controlar el acceso que los usuarios tengan a la información de la institución.

3.2 Elementos de una Arquitectura PKI

3.2.1 Autoridad Certificadora (CA) [Raíz]

Una autoridad Certificadora Raíz es aquella que genera un certificado y se auto firma el mismo, para posteriormente ser usado por las autoridades subordinadas para generar certificados que serán entregados a los usuarios finales.

Esta autoridad deberá estar fuera de línea y solo tendrá comunicación con las autoridades subordinadas, además debe contar con un sistema de seguridad perimetral físico y lógico que cumpla altos estándares internacionales.

3.2.2 Autoridad Certificadora (CA) [Subordinada]

Una autoridad certificadora subordinada es aquella que entrega certificados a los usuarios finales.

Se comunica directamente con la Autoridad de Registro, y se encarga de generar las claves y firmarlas con el certificado entregado por la Autoridad Certificadora Raíz.

3.2.3 Autoridad de Registro (RA)

Una Autoridad de Registro es aquella que se encarga de realizar funciones administrativas, no requiere de un software especial simplemente se encarga de recolectar los datos, verificarlos y enviarlos a la entidad certificadora. La cual firma el certificado y lo entrega de nuevo a la Autoridad de Registro, para que este sea finalmente entregado al usuario final.

3.2.4 Autoridad de Validación (VA)

Una VA es una entidad de certificación neutral, que trabaja con protocolos multivalidación, que asegura la autenticidad de las operaciones más críticas, como por ejemplo compras y ventas por internet. Su trabajo se enfoca en proporcionar datos sobre la vigencia de los certificados.

3.2.5 Autoridad de Sellado de Tiempo o TIMESTAMPING (TS)

Una TS es aquella que confirma la validez de un certificado digital en un determinado periodo de tiempo.

Garantizando de este modo que la información contenida en el mensaje de datos no se ha modificado desde el momento en que fue firmado.

3.2.6 Repositorios de Certificados (CRL)

Los CRL son las encargadas de almacenar los certificados y las listas de revocación de los mismos. Además de los datos de la Autoridades emisoras de dichos certificados.

En una lista de revocación de certificados se guardan los certificados han dejado de ser válidos antes de la fecha fin normal establecida en cada uno.

CAPITULO IV
IMPLEMENTACIÓN DE LA PKI

4.1 Mecanismo De Funcionamiento

Para realizar la solitud, generación y entrega de los certificados digitales dentro del Ejército Ecuatoriano se deberá proceder con los siguientes pasos:

El usuario deberá acercarse a la Autoridad de Registro correspondiente a la División de Ejército que pertenezca.

Deberá Presentar un documento de identificación y llenar un formulario de registro.

a. El agente registrador, procederá a ingresar al sistema SIFTE, y en el menú seleccionará la opción Certificados Digitales.

b. Una vez que el agenta haya llenado los datos del solicitante, hará clic en botón Enviar Solicitud.

c. El Sistema deberá comprobar dicha información con la base de datos el Ejército, y enviarla a la autoridad de registro delegada la cual procederá a generar una Clave Privada y el Certificado Digital el cual contendrá una Clave Publica la información concerniente al solicitante y la Autoridad Certificadora.

d. Posteriormente la Autoridad Certificadora procederá a firmar dicho certificado y a enviarlo a la Autoridad de Registro, para que esta entregue dichos archivos al usuario final.

4.2 Diseño

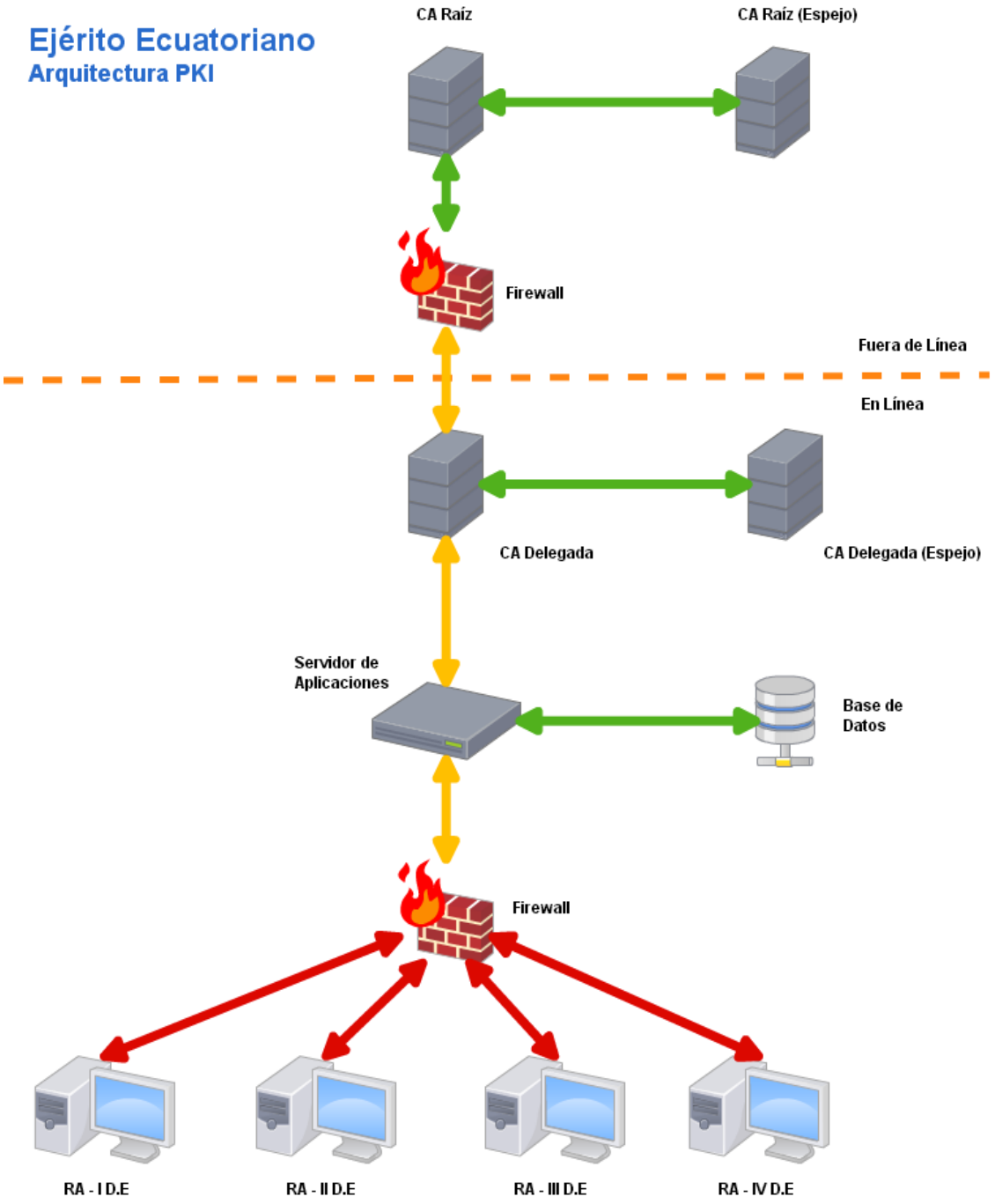


Figura 2: Diseño de la PKI aplicado al Ejército Ecuatoriano

4.3 Instalación y Configuración

4.3.1 Sistema operativo

Seleccionamos CentOS v7 como nuestro sistema operativo, porque es la versión libre y con soporte de Red Hat, por lo cual aseguramos que es un sistema operativo con base Linux, muy eficiente y confiable.

Para iniciar la instalación primero instalamos un gestor de descarga que agilite dicho proceso.

Descargamos el paquete

```
$ wget http://ufpr.dl.sourceforge.net/project/xdman/xdman.deb
```

Instalamos es paquete

```
$ sudo dpkg  i xdman.deb
```

Si sale error en dependencia ejecutamos el siguiente comando

```
$ sudo dpkg -f
```

Ahora vamos a las aplicaciones instaladas e iniciamos el gestor de descarga

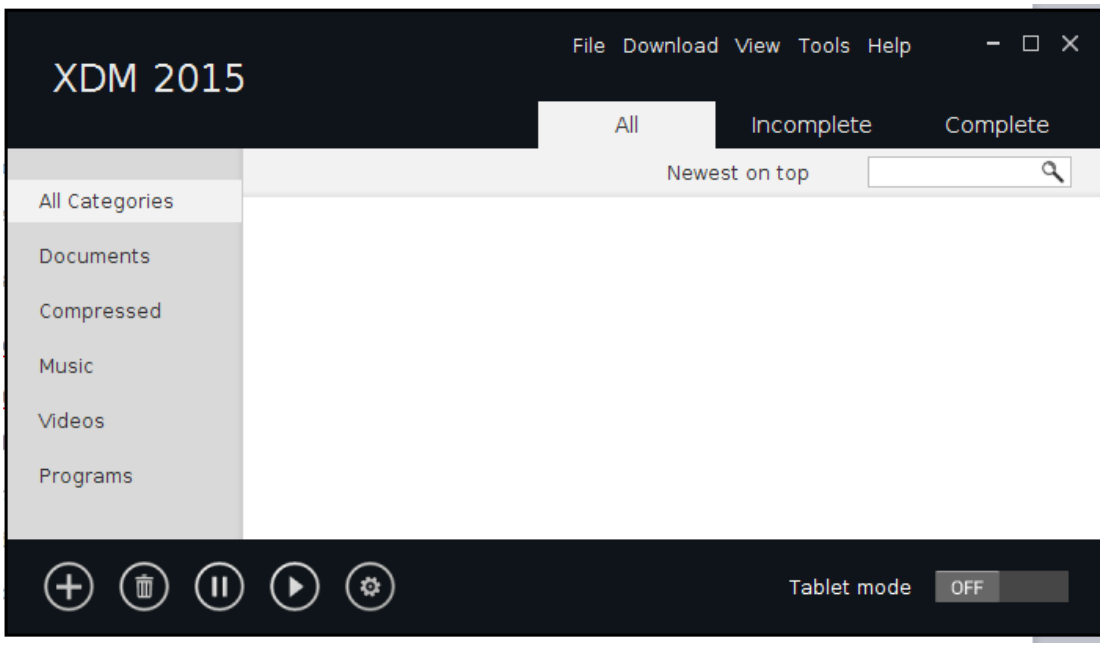


Figura 3: Instalación de la herramienta XDM

Ahora para descargar la imagen vamos a la página web de CentOS https://www.centos.org/download/ y seleccionamos la versión mínima.

Figura 4: Descarga del sistema operativo CentOS 7

Seleccionamos el origen para descargar nuestra imagen…

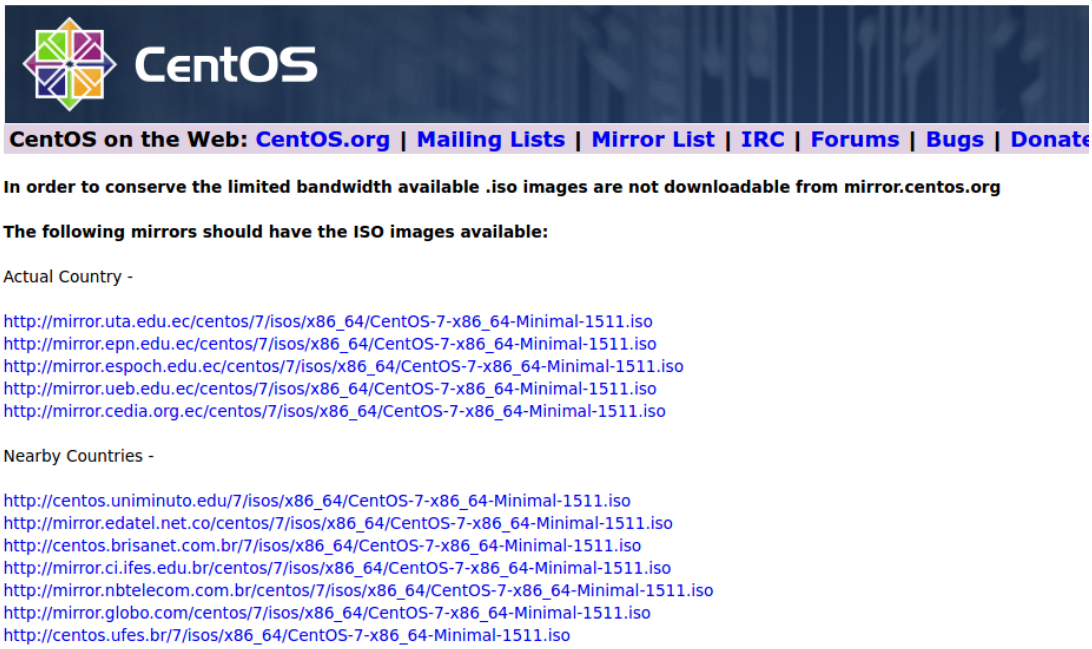


Figura 5: Selección de la versión del sistema operativo

Agregamos el URL seleccionado al xtreme download manager y damos clic en iniciar la descarga.

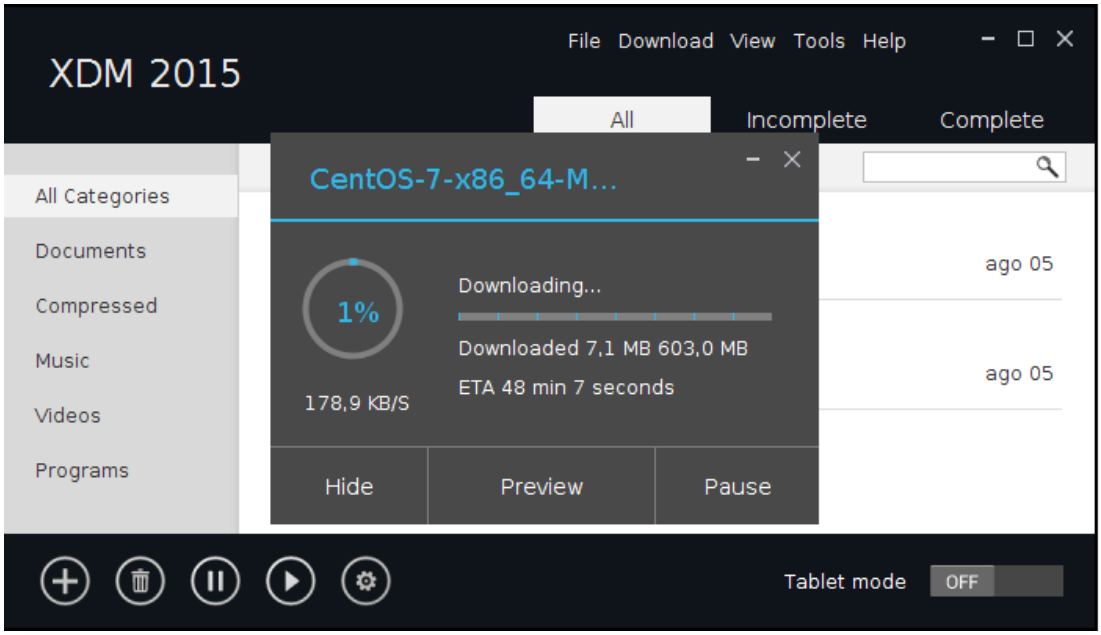


Figura 6: Proceso de descarga del sistema operativo

Una vez terminada la descarga procedemos a quemar un CD con la imagen descargada y proceder a instalar el sistema operativo.

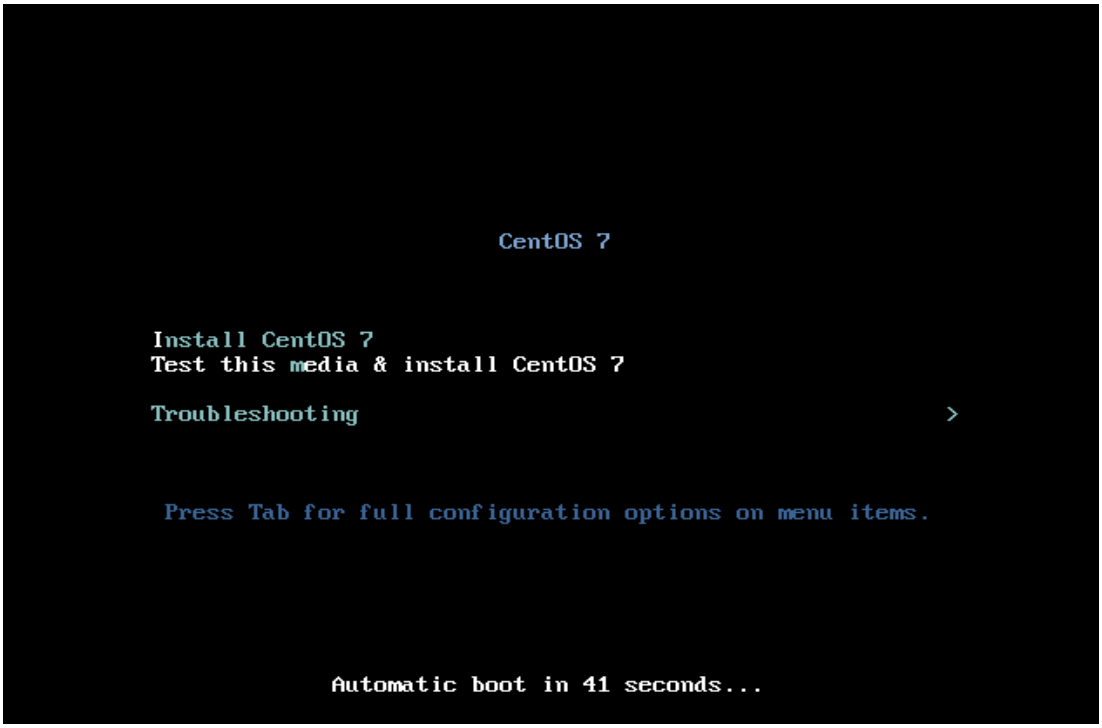


Figura 7: Instalación del Sistema Operativo

Seleccionamos español para el idioma de instalación…

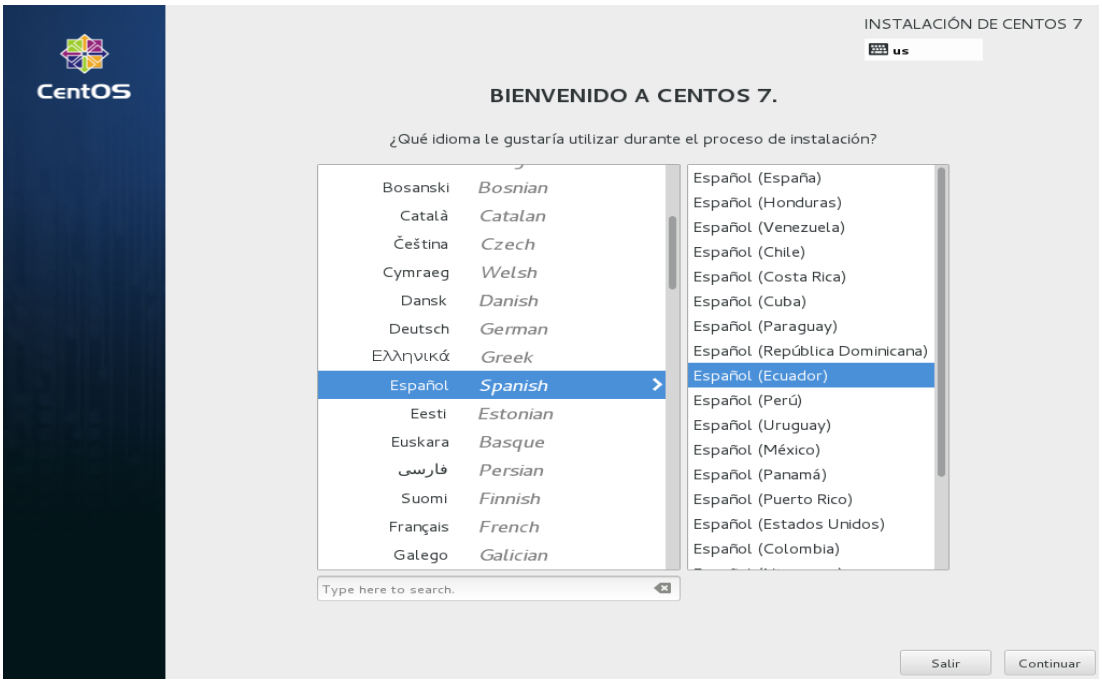


Figura 8: Selección del idioma

Seleccionamos el destino de la instalación…

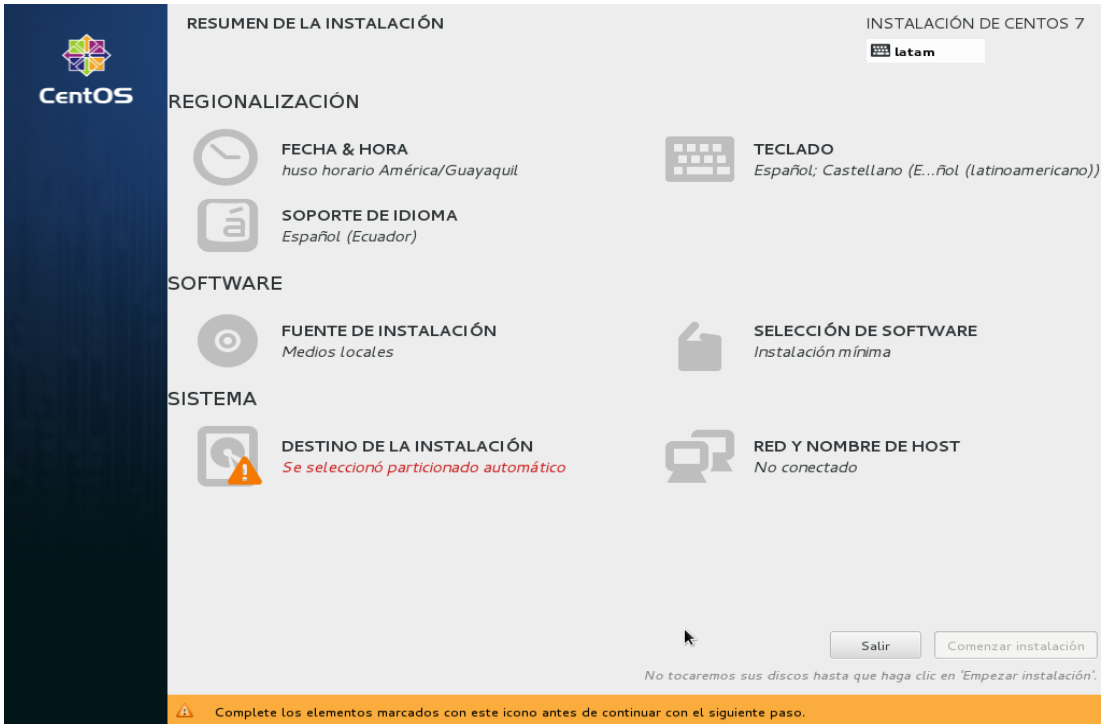


Figura 9: selección del destino de instalación

Vamos a tomar todo el disco para la instalación y hacemos clic en el botón “Listo”

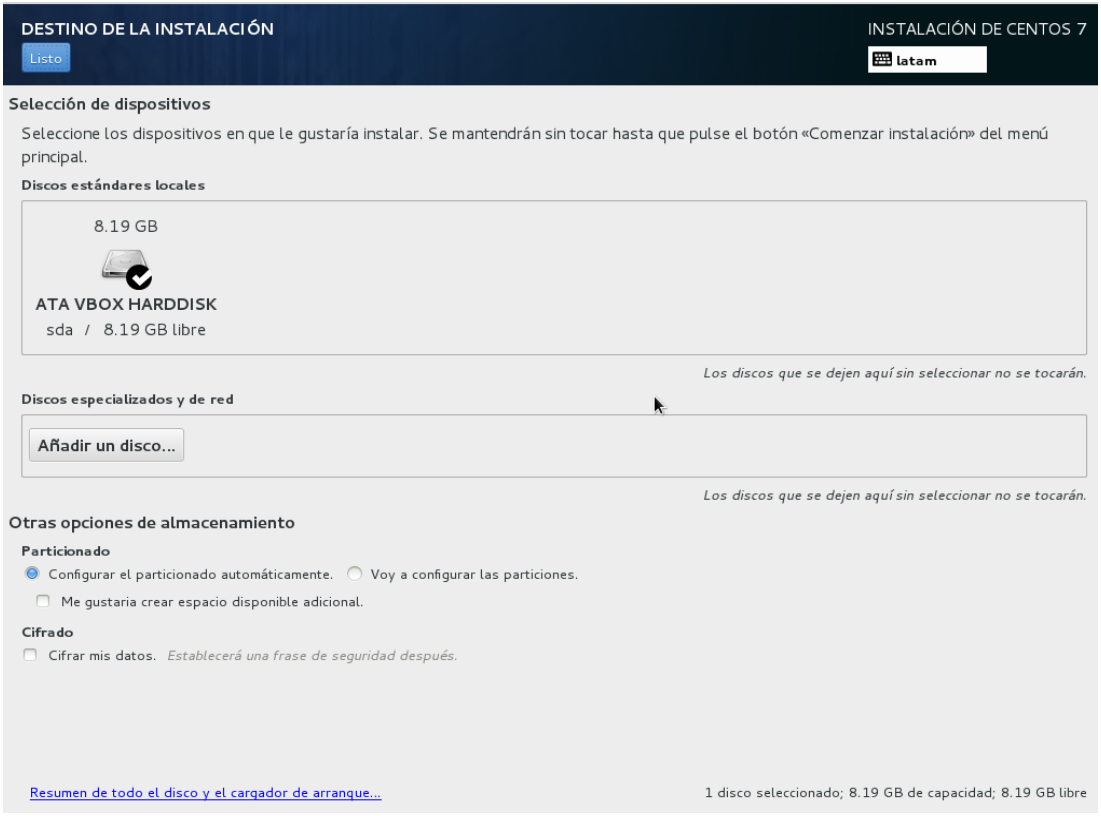


Figura 10: Asignación del disco de instalación

Para comenzar la instalación pide que se establezca la contraseña de root y la creación de un nuevo usuario…

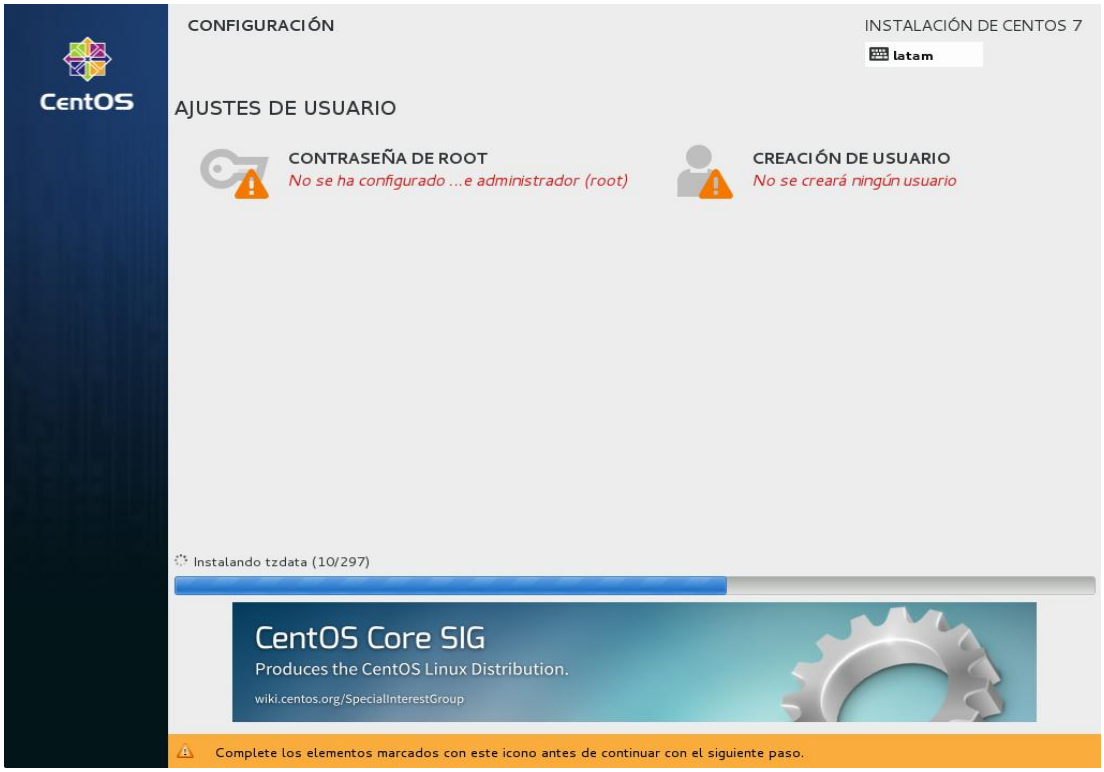


Figura 11: Ingresando a ajustes de usuario

En la creación de la contraseña de root el sistema indica si es una contraseña robusta o débil.

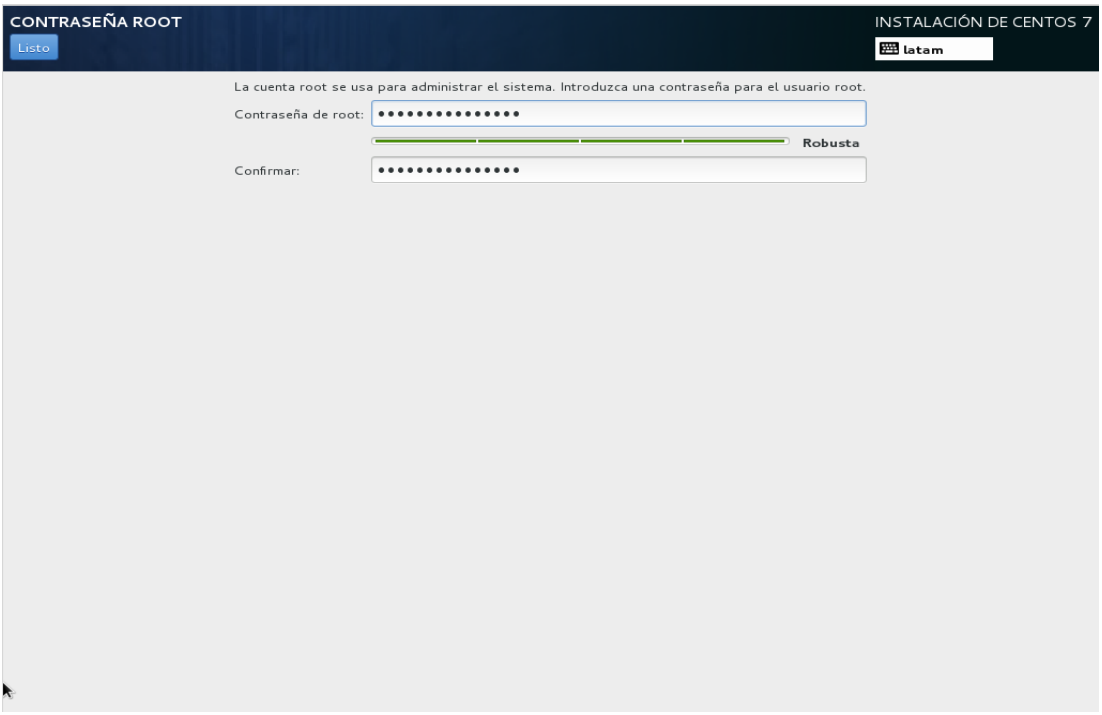


Figura 12: Estableciendo la contraseña de root

Para la creación del nuevo usuario pede un nombre de usuario y una clave de acceso.

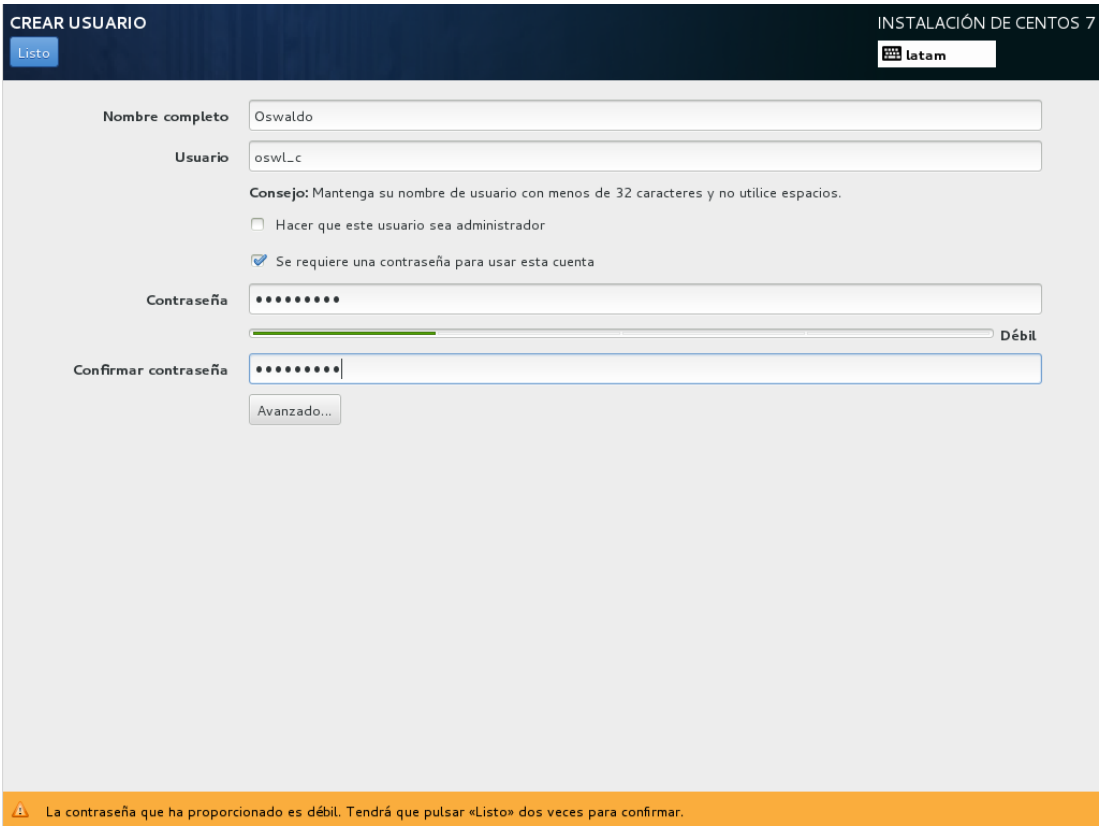


Figura 13: Creando un nuevo usuario y contraseña

Ahora procedemos a dar clic en el botón “finalizar la configuración” …

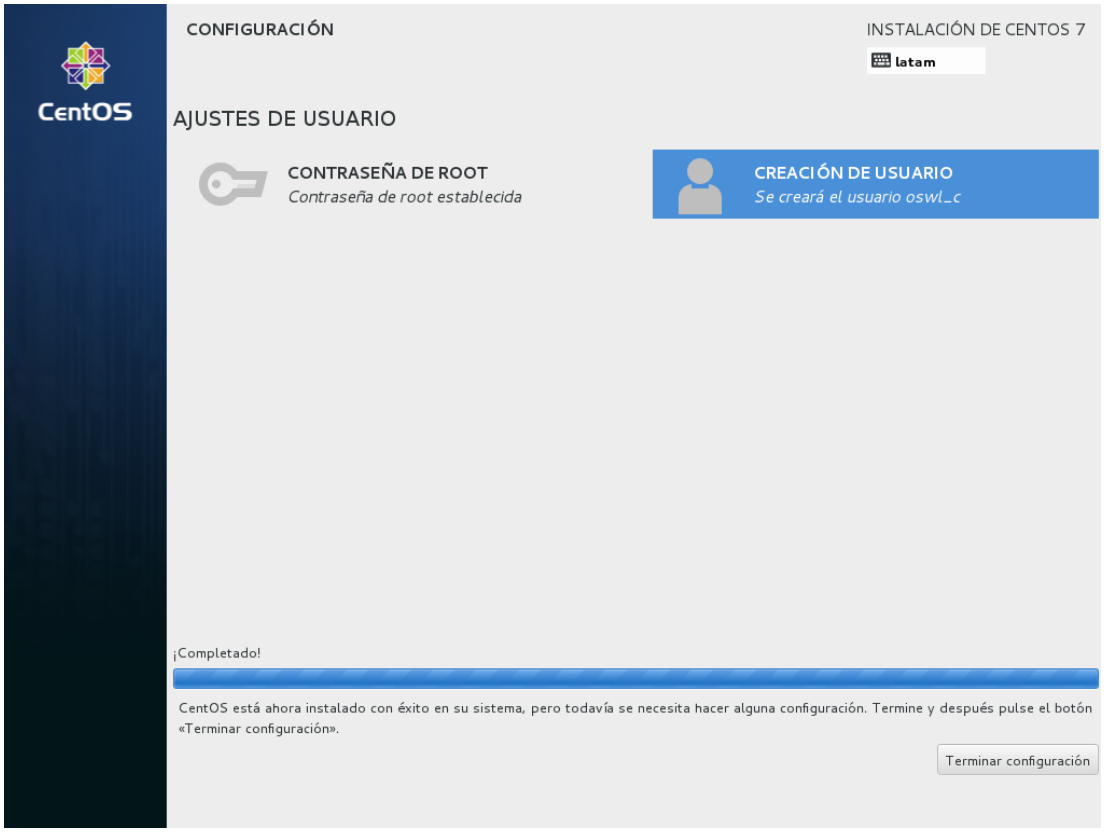


Figura 14: Inicio de Instalación

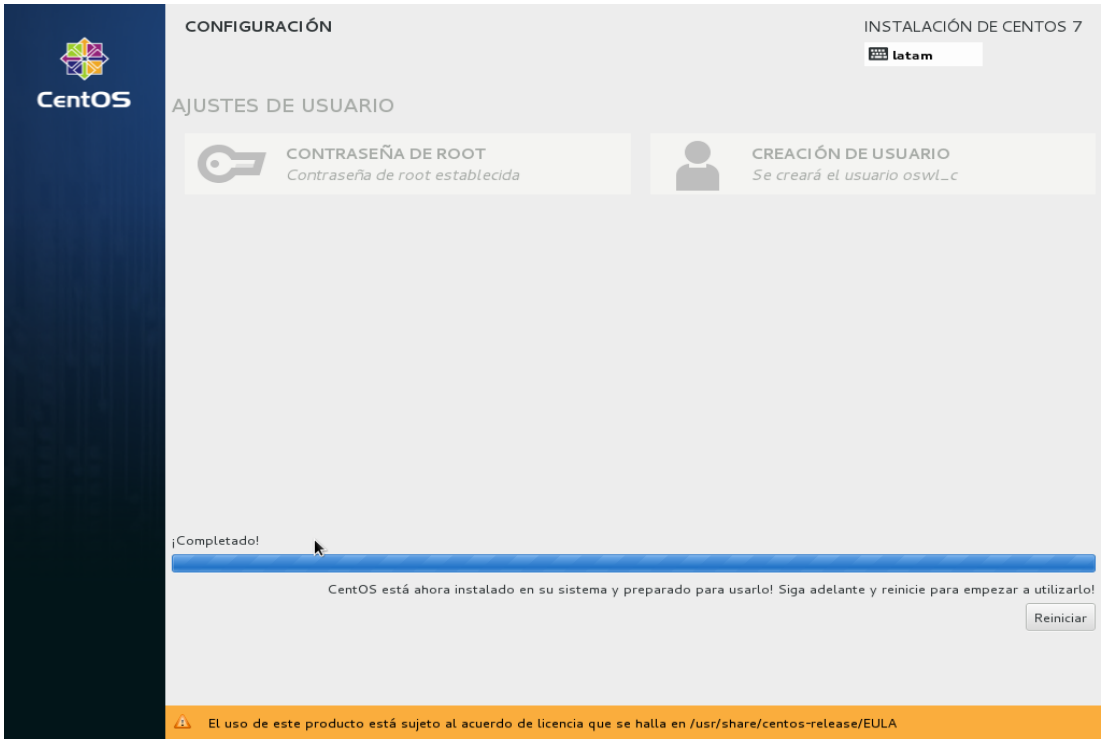


Figura 15: Proceso y finalización de instalación del sistema

Reiniciamos y tenemos listo nuestro sistema operativo

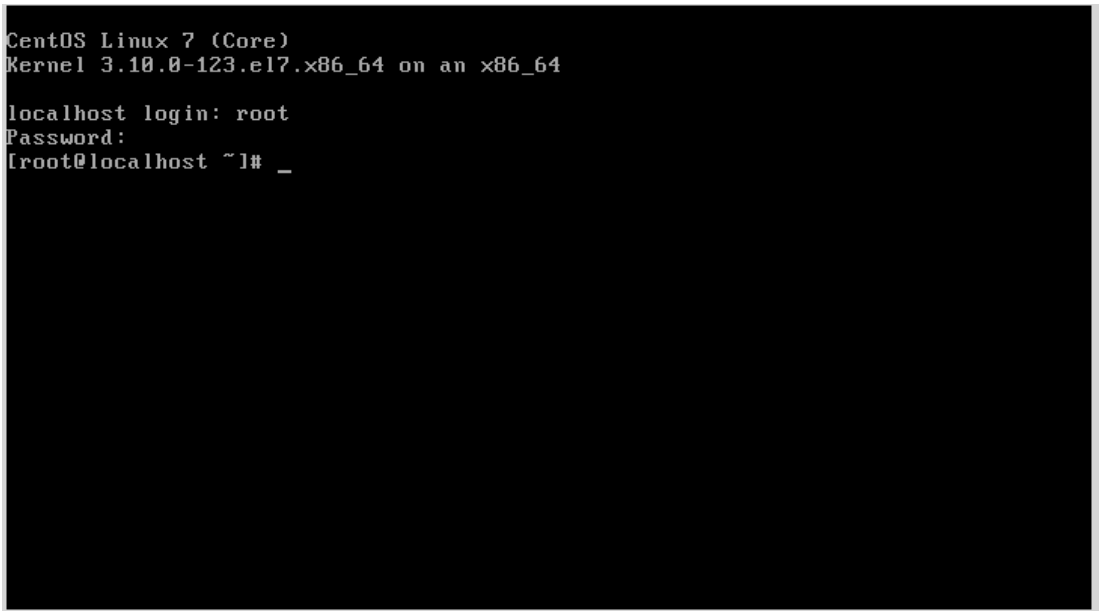


Figura 16: Acceso al sistema operativo

4.3.2 Herramientas PKI

4.3.2.1 OPENSSL

OpenSSL es un proyecto de software libre, que tiene herramientas de administración y bibliotecas relacionadas con la criptografía. Estas herramientas ayudan a implementar una capa de conexión segura o SSL (Secure Socket Layer), así como otros protocolos relacionados con la seguridad como la capa segura de transporte u TLS (Transport Layer Security). OpenSSL también permite crear certificados digitales que pueden aplicarse a un servidor, por ejemplo, Apache.

Para instalar OpenSSL ejecutamos el siguiente comando…

```
# yum install openssl
```

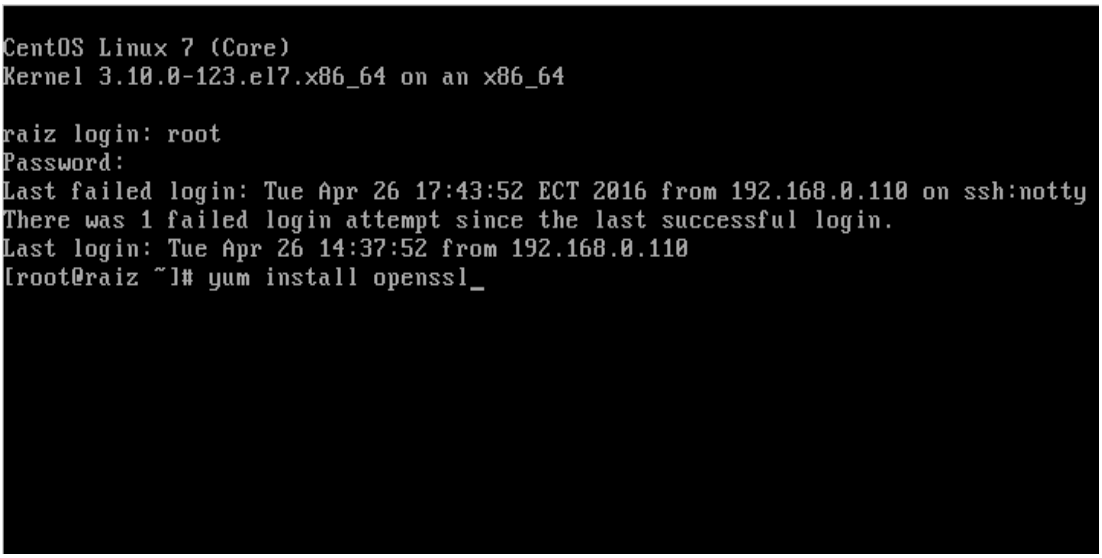


Figura 17: Instalación de la herramienta openssl

Si el sistema pregunta si deseamos descárganos los paquetes ingresamos la letra "y" …

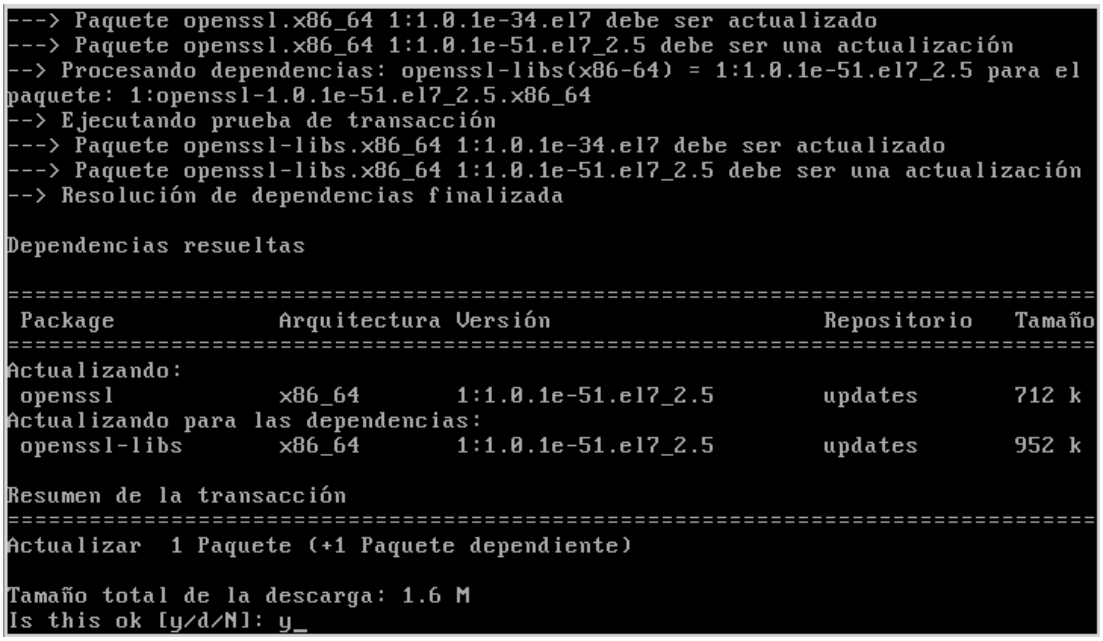


Figura 18: Caga de dependencias

Como podemos apreciar la instalación de esta herramienta es muy sencilla al igual que su uso.

4.3.2.2 OPENCA

OpenCA es una herramienta de software libre que utiliza OpenSSL para generar claves y tiene un entorno de web para la administración, pero su instalación y uso son un poco más complejos.

Vamos a proceder a instalar las siguientes herramientas que son necesarias para un correcto funcionamiento de OpenCA.

1.- Mysql Server

2.- Httpd

3.- OpenSSL

4.- openca-tools

5.- openca-base

Instalar mysql-server con el siguiente comando

```
# yum install mysql mysql-server
```

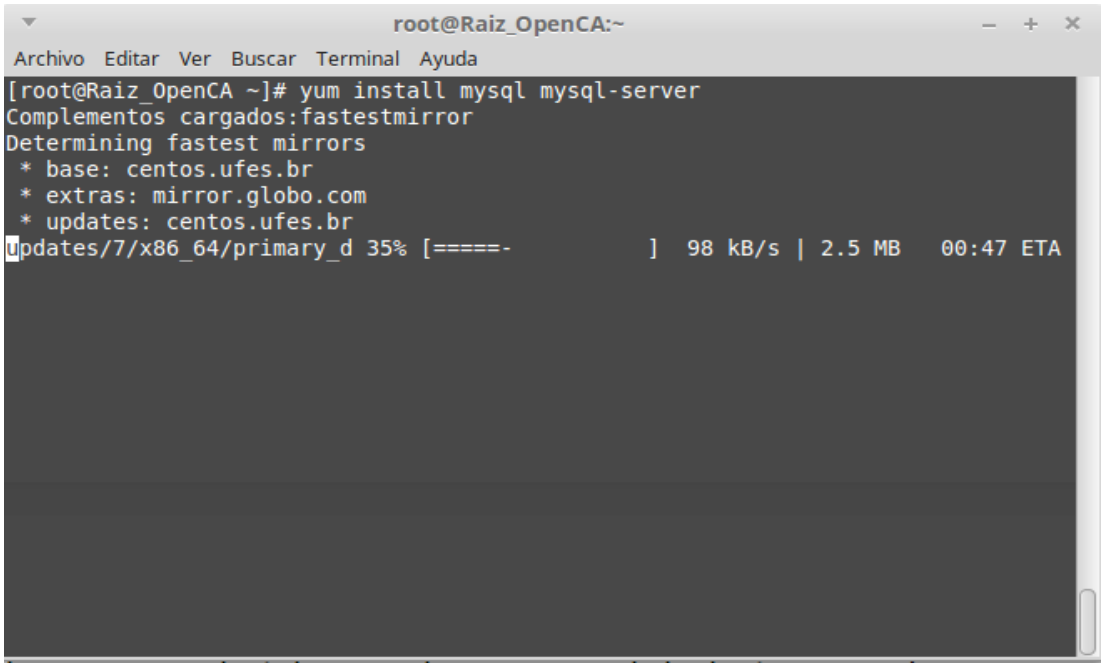


Figura 19: Instalación del servidor de base de datos

Ahora procedemos a iniciar el servidor mysql…

```
# systemctl start mysqld
```

Lo activamos para que inicie con el sistema operativo…

```
# systemctl enable mysqld
```

Ejecutamos el siguiente comando para agregar el password al usuario root de mysql.

```
# mysql_secure_installation
```

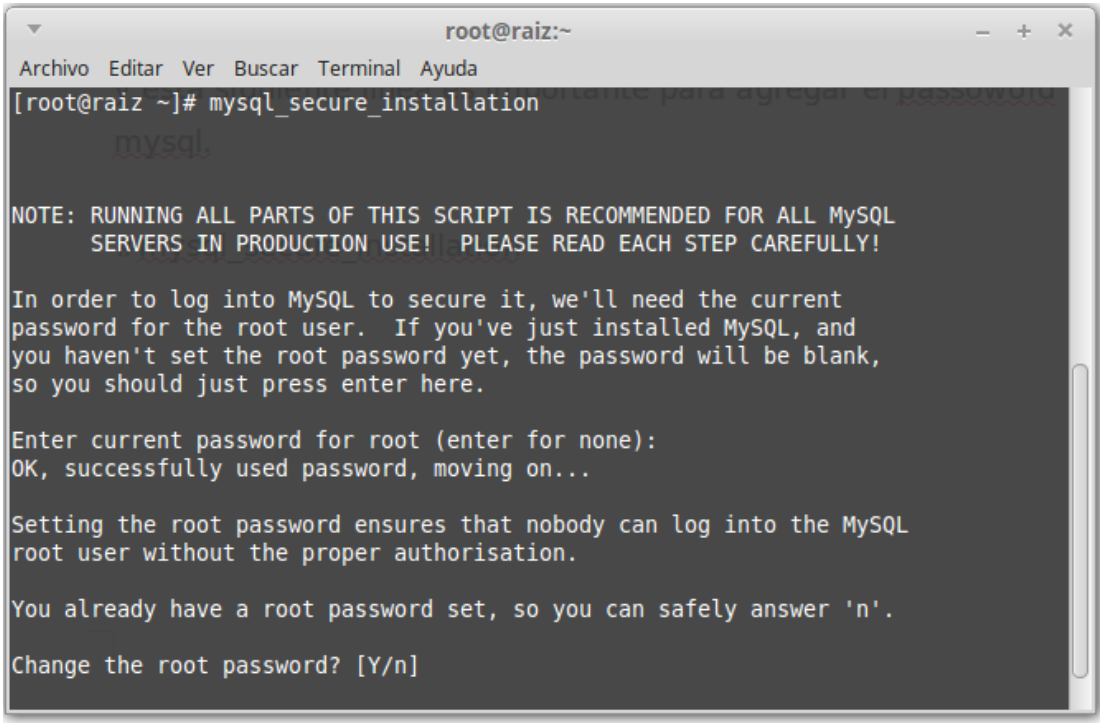


Figura 20: Creación de conexión segura a mysql

Ahora vamos a crear la base de datos con el nombre de openca, para lo cual ingresamos a mysql usando el siguiente comando…

```
# mysql -u root -p
```

Y creamos la base de datos…

```
mysql> create database openca;
```

También necesitamos instalar el paquete openssl-devel y mod_ssl…

```
# yum install openssl-devel

# yum install mod_ssl
```

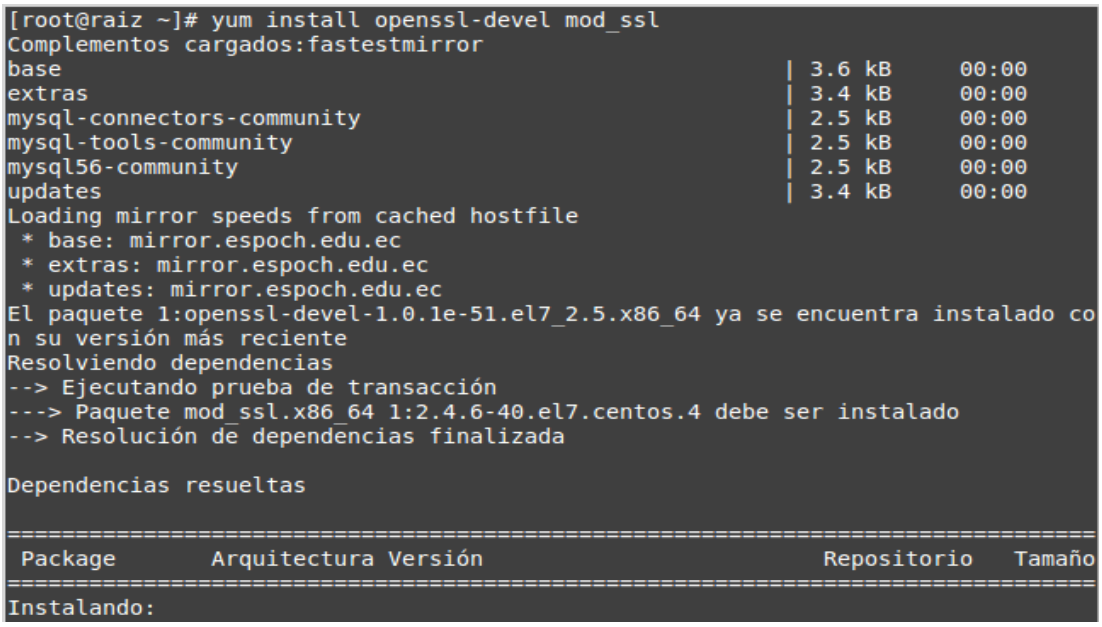


Figura 21: Instalación de openssl-devel y mod_ssl

Ahora procedemos a instalar el servidor web httpd de apache.

```
#yum install httpd
```

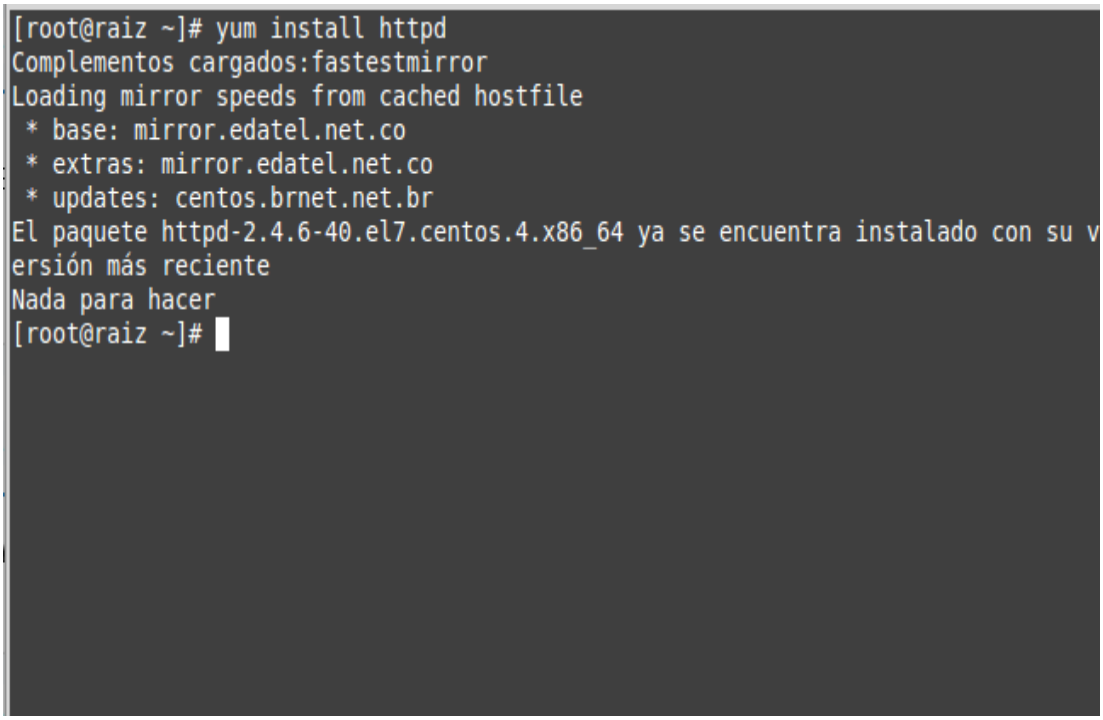


Figura 22: Instalación del servidor web httpd

Arrancamos el servidor con el comando…

```
# systemctl start httpd.service
```

Y lo agregamos al arranque con el sistema…

```
# systemctl enable httpd.service
```

Probamos si ya tenemos el servicio usando un navegador y colocando la IP del servidor, por ejemplo:

http://192.168.0.108

Para que salga una página personalizada colocamos el comando…

```
# nano /var/www/html/index.html
```

Y colocamos el siguiente código…

```
<html>
 <body>
  <center>
   <h1>hola mundo</h1>
  </center>
 </body>
</html>
```

Usando el navegador veremos lo siguiente…

Figura 23: Funcionamiento del servidor web

Ahora procedemos a instalar openldap…

```
# yum -y install openldap openldap-clients openldap-servers
```

```
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete openldap.x86_64 0:2.4.39-3.el7 debe ser actualizado
---> Paquete openldap.x86_64 0:2.4.40-9.el7_2 debe ser una actualización
--> Resolución de dependencias finalizada

Dependencias resueltas

================================================================================
 Package          Arquitectura   Versión                 Repositorio     Tamaño
================================================================================
Actualizando:
 openldap         x86_64         2.4.40-9.el7_2          updates         348 k

Resumen de la transacción
================================================================================
Actualizar  1 Paquete

Tamaño total de la descarga: 348 k
Is this ok [y/d/N]: y
```

Figura 24: Instalando openldap

Iniciamos el servicio…

```
# systemctl start slapd
```

Y lo agregamos al inicio…

```
# systemctl enable slapd
```

Y cambiamos el password del admin de openldap

```
# slappasswd
```

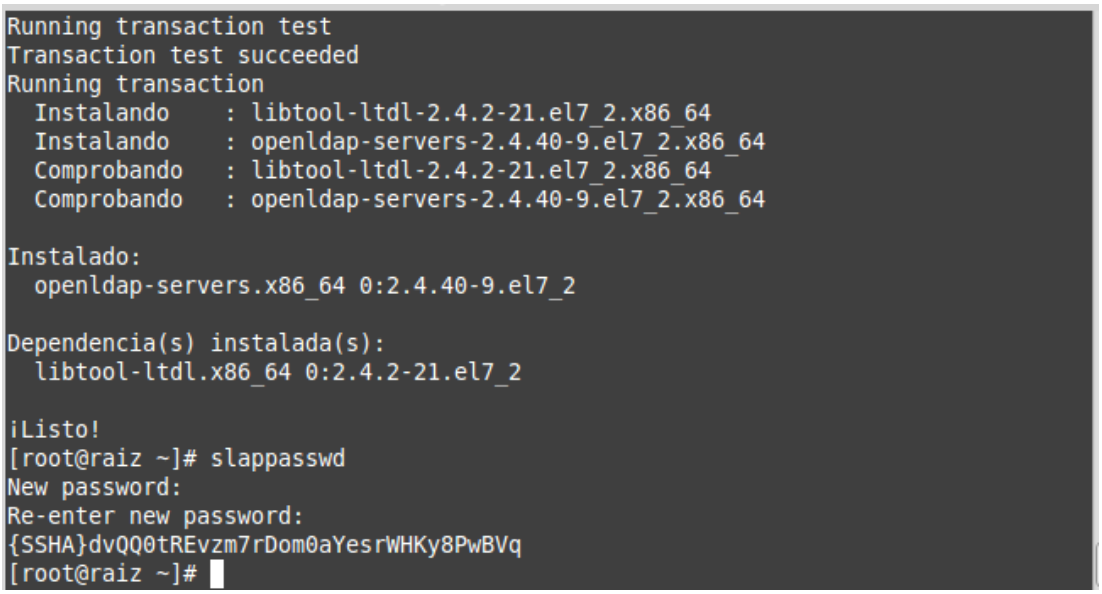


Figura 25: Estableciendo password a openldap

Instalar los repositorios epel…

```
# yum install -y epel-release
```

Ahora instalamos el resto de módulos…

```
# yum install db4-devel

# yum install expat-devel

# yum install perl-Digest-SHA1

# yum install perl-Digest-MD5

# yum install mod_perl

# yum install mysql-devel
```

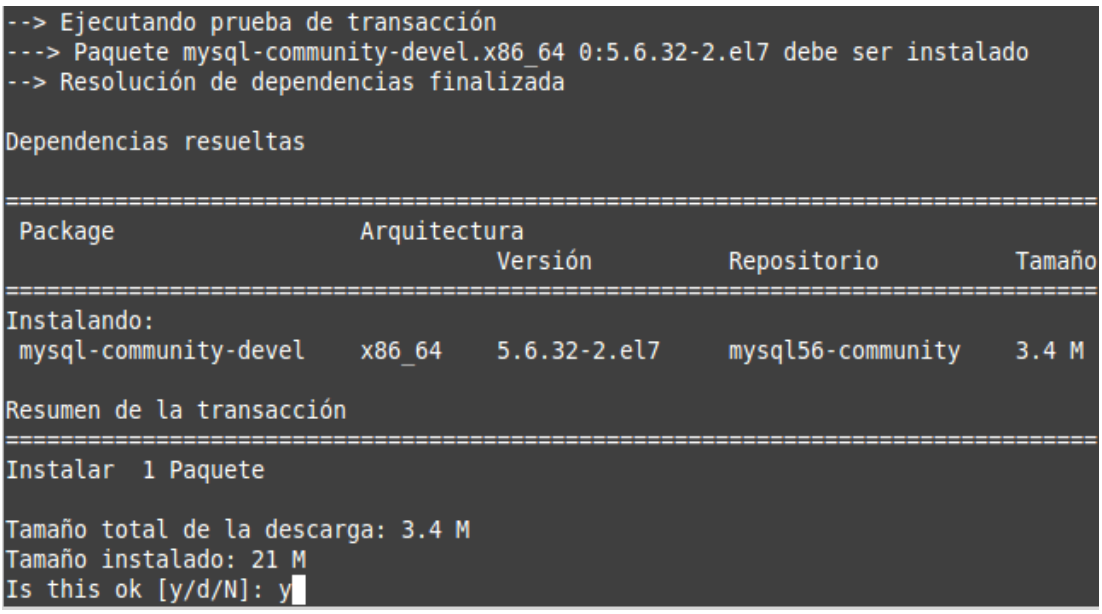


Figura 26: Instalación de los repositorios epel

Ahora ingresamos a la página de openca para descargar los paquetes necesarios

https://www.openca.org/projects/openca/downloads.shtml

Ahí descargamos el Source de OPENCA TOOLS y el OPENCA pki (Base) en formato tar.gz

Figura 27: Descarga de OpenCA Tools

Figura 28: Descarga de OpenCA PKI

Ahora procedemos a instalar dichos paquetes en nuestro sistema. Para lo cual, los copiamos mediante el protocolo sftp.

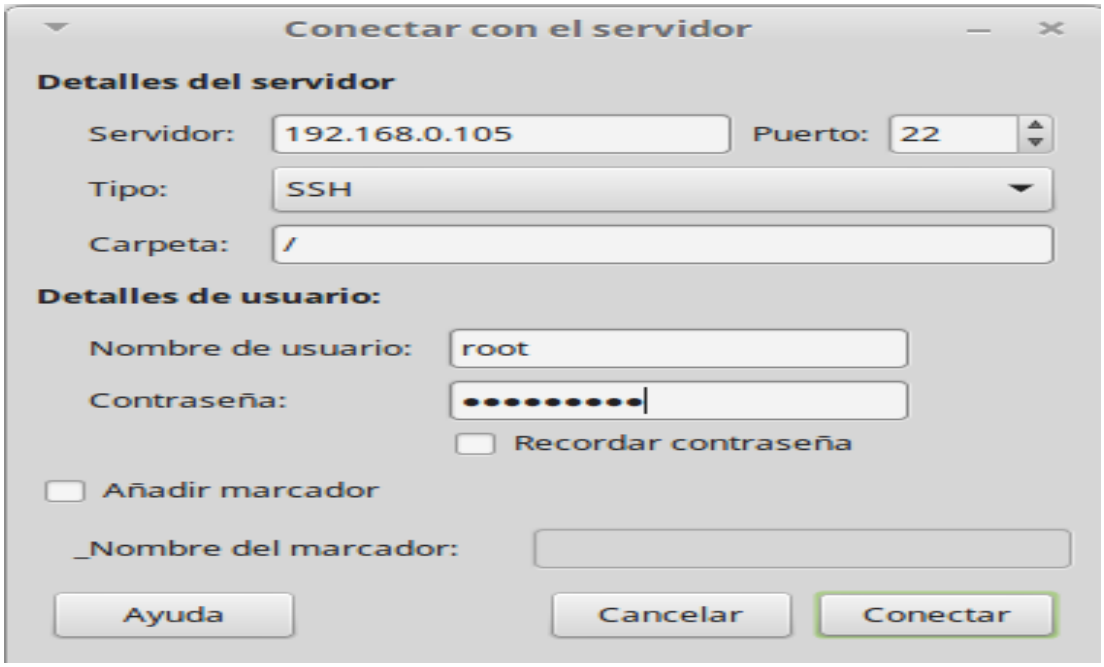


Figura 29: Acceso al servidor mediante el protocolo sftp

Y quedará los archivos de la siguiente forma…

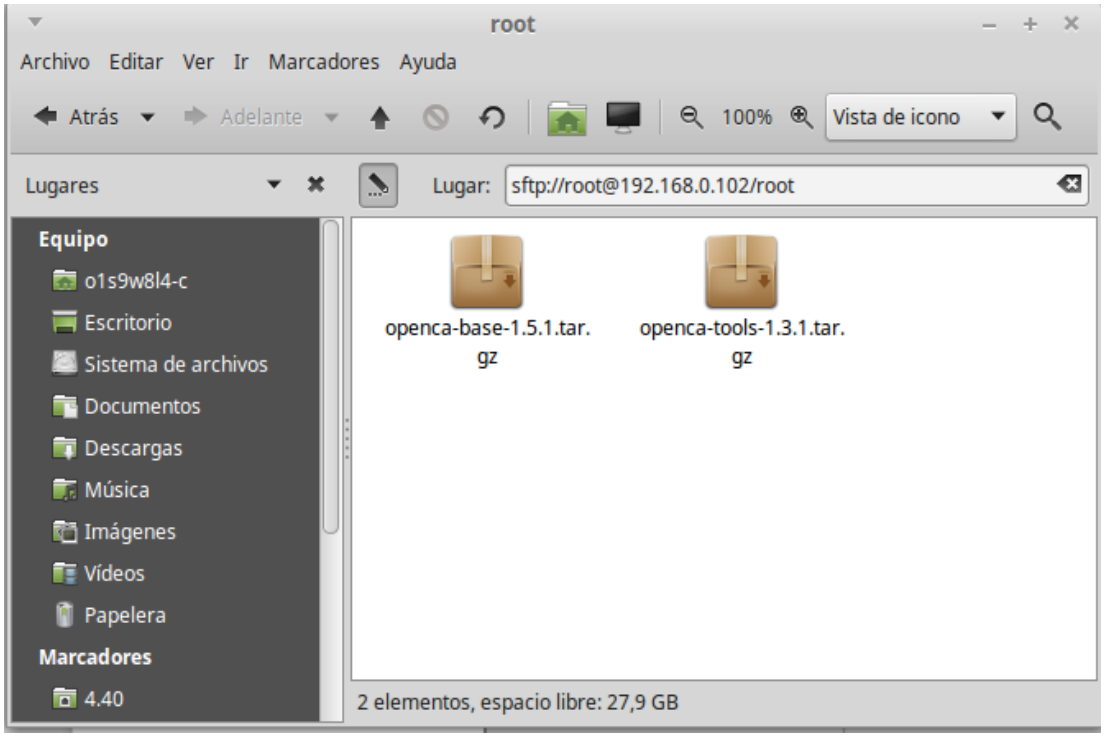


Figura 30: Navegando por el servidor

Y en la consola procedemos a desempaquetar y a instalar.

```
# tar -xvf openca-tools-1.3.1.tar.gz
```

```
openca-tools-1.3.1/src/crmf/Makefile.am
openca-tools-1.3.1/src/crmf/crmf_asn1.c
openca-tools-1.3.1/src/crmf/crmf_bio.c
openca-tools-1.3.1/src/crmf/Makefile.in
openca-tools-1.3.1/src/crmf/crmf.c
openca-tools-1.3.1/src/sv/
openca-tools-1.3.1/src/sv/apps.c
openca-tools-1.3.1/src/sv/tools.c
openca-tools-1.3.1/src/sv/callback.c
openca-tools-1.3.1/src/sv/sv.c
openca-tools-1.3.1/src/sv/Makefile.am
openca-tools-1.3.1/src/sv/Makefile.in
openca-tools-1.3.1/src/sv/sign-crypto.c
openca-tools-1.3.1/src/sv/verify-crypto.c
openca-tools-1.3.1/acinclude.m4
openca-tools-1.3.1/AUTHORS
openca-tools-1.3.1/install-sh
openca-tools-1.3.1/aclocal.m4
[root@raiz ~]# ls
openca-base-1.5.1.tar.gz  openca-tools-1.3.1  openca-tools-1.3.1.tar.gz
[root@raiz ~]#
```

Figura 31: Descomprimiendo OpenCA

Ingresamos a la carpeta openca…

```
# cd openca-tools-1.3.1
```

Ahora ejecutamos la configuración…

```
# ./configure
```

Si sale este error *"no acceptable C compiler found in $PATH"*, procedemos a instalar los siguientes paquetes…

```
# yum groupinstall "Development tools"
```

```
checking pkg-config is at least version 0.9.0... yes
checking for OPENSSL... yes
openssl 0.9.7 or greater found via pkgconfig

enable engine support   : true
enable debug messages   : false
install prefix          : /usr
checking for builder... no
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/sv/Makefile
config.status: creating src/scep/Makefile
config.status: creating src/crmf/Makefile
config.status: creating docs/Makefile
config.status: creating contrib/install-builder/openca-tools.xml
config.status: creating contrib/install-builder/openca/tools.xml
config.status: creating include/openca/config.h
config.status: include/openca/config.h is unchanged
config.status: executing libtool commands
config.status: executing depfiles commands
[root@raiz openca-tools-1.3.1]#
```

Figura 32: Configurando OpenCA Tools

```
 zip                    x86_64    3.0-10.el7              base       260 k
Actualizando para las dependencias:
 bzip2-libs             x86_64    1.0.6-13.el7            base        40 k
 elfutils-libelf        x86_64    0.163-3.el7             base       200 k
 elfutils-libs          x86_64    0.163-3.el7             base       260 k
 libgcc                 x86_64    4.8.5-4.el7             base        95 k
 libgomp                x86_64    4.8.5-4.el7             base       130 k
 libstdc++              x86_64    4.8.5-4.el7             base       298 k
 rpm                    x86_64    4.11.3-17.el7           base       1.2 M
 rpm-build-libs         x86_64    4.11.3-17.el7           base       103 k
 rpm-libs               x86_64    4.11.3-17.el7           base       272 k
 rpm-python             x86_64    4.11.3-17.el7           base        79 k

Resumen de la transacción
================================================================================
Instalar    26 Paquetes (+36 Paquetes dependientes)
Actualizar             ( 10 Paquetes dependientes)

Tamaño total de la descarga: 80 M
Is this ok [y/d/N]: y
Downloading packages:
```

Figura 33: Instalando Development tools

Ahora procedemos de nuevo a ejecutar el comando de configuración

```
# ./configure
```

Continuamos con las configuraciones

```
# make clean

# make

# make install

# cd ..
```

Ahora procedemos a instalar el openca-base

```
# tar -xvf openca-base-1.5.1.tar.gz

# cd openca-base-1.5.1

# ls
```

```
openca-base-1.5.1/build/
openca-base-1.5.1/build/config.sub
openca-base-1.5.1/build/install-sh
openca-base-1.5.1/build/missing
openca-base-1.5.1/build/config.guess
openca-base-1.5.1/build/compile
openca-base-1.5.1/build/ltmain.sh
openca-base-1.5.1/aclocal.m4
[root@raiz ~]# ls
openca-base-1.5.1         openca-tools-1.3.1
openca-base-1.5.1.tar.gz  openca-tools-1.3.1.tar.gz
[root@raiz ~]# cd openca-base-1.5.1
[root@raiz openca-base-1.5.1]# ls
aclocal.m4    docs             m4                    relative_ln_s.sh
build         I18N             Makefile              relative_ln_s.sh.in
configs       INSTALL          Makefile.devel        RELEASE-NOTES
configure     INSTALL.SOLARIS  Makefile.global-vars  src
configure.in  INSTALL.UBUNTU   Makefile.global-vars.in  STATUS
contrib       libtool          NOTES.Chain           THANKS
ChangeLog     LICENSE          README                VERSION
[root@raiz openca-base-1.5.1]#
```

Figura 34: Instalando OpenCA base

Ahora procedemos a ejecutar la configuración con el siguiente código que pegaremos en la consola.

```
./configure \
-prefix=/opt/openca \
-with-ca-organization=”Ejercito” \
-with-ca-country=EC \
-with-ca-locality=Quito \
-with-httpd-fs-prefix=/var/www \
-with-httpd-main-dir=pki \
-with-openca-user=apache \
-with-openca-group=apache \
-with-db-name=openca \
-with-db-host=localhost \
-with-db-user=root \
-with-db-passwd=****** \
-with-db-type=mysql \
-with-service-mail-account=”oswl_c@ecuasof.com”
```

Ahora ejecutamos el comando…

```
# make
```

```
Building (Digest::MD5::251) ... Ok.
Building (libintl::perl::120) ... Ok.
Building (Net::SSLeay::140) ... Ok.
Building (IO::Socket::SSL::131) ... Ok.
Building (IO::stringy::211) ... Ok.
Building (File::Temp::022) ... Ok.
Building (MIME::tools::550) ... Ok.
Building (MailTools::209) ... Ok.
Building (MIME::Lite::302) ... Ok.
Building (Net::Server::200) ... Ok.
Building (URI::152) ... Ok.
Building (XML::Parser::241) ... Ok.
Building (Parse::RecDescent::194) ... Ok.
Building (X500::DN::029) ... Ok.
Building (XML::SAX::Base::108) ... Ok.
Building (perl::ldap::043) ... Ok.
Building (FCGI::074) ... Ok.
Building (CGI.pm::359) ... Ok.
Building (XML::Twig::339) ... Ok.

[root@raiz openca-base-1.5.1]#
```

Figura 35: Configurando OpenCA base

Procedemos a instalar la autoridad de certificación y de registro…

```
# make install-offline install-online
```

Ahora procedemos a desactivar el selinux

```
# cd /etc/selinux

# nano config
```

Buscamos la línea

```
SELINUX=enforcing
```

Y la cambiamos por

```
SELINUX=disabled
```

Nos quedará el código de la siguiente forma:

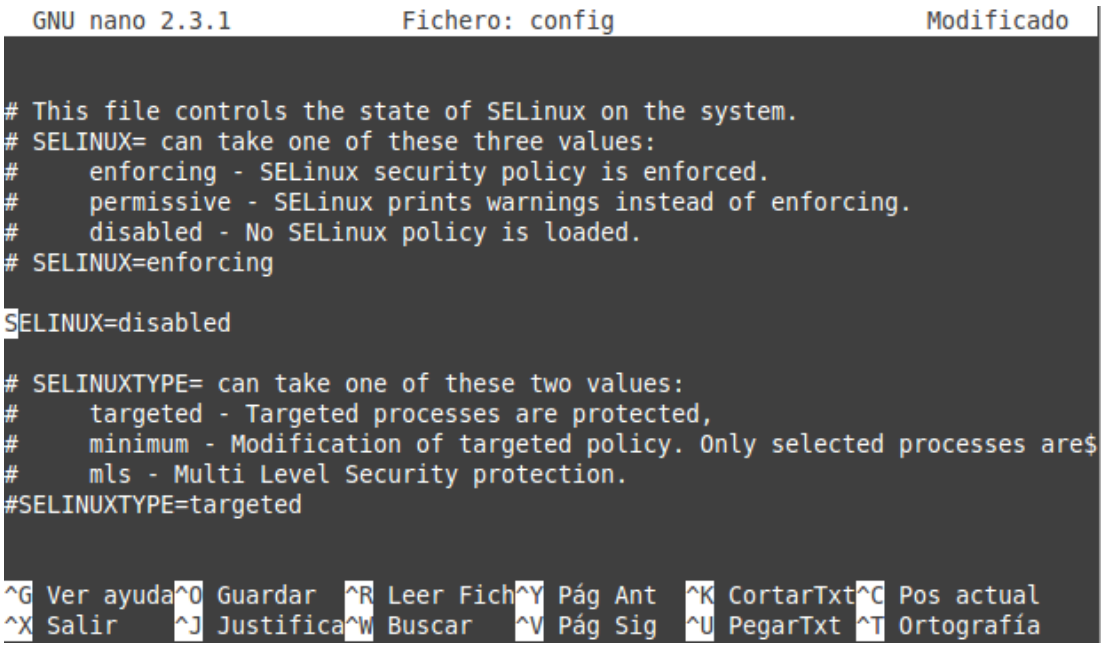


Figura 36: Configurando SELINUX

Reiniciamos la maquina…

```
# shutdown -r now
```

Ingresamos nuevamente y vamos al siguiente directorio…

```
# cd /opt/openca/etc/openca/
```

Y ejecutamos el comando…

```
# ./configure_etc.sh
```

```
UPDATE => 1
status: To be updated
--------------end file-----------------
-------------begin file-----------------
template: ./mails/it_IT/certsMail.msg.template
target: ./mails/it_IT/certsMail.msg
UPDATE => 1
status: To be updated
--------------end file-----------------
-------------begin file-----------------
template: ./mails/it_IT/verifyMail.msg.template
target: ./mails/it_IT/verifyMail.msg
UPDATE => 1
status: To be updated
--------------end file-----------------
-------------begin file-----------------
template: ./mails/it_IT/confirm_cert_sign.msg.template
target: ./mails/it_IT/confirm_cert_sign.msg
UPDATE => 1
status: To be updated
```

Figura 37: Ejecutando el comando configure_etc.sh

Finalizado este proceso vamos a reiniciar el servidor web…

```
# systemctl restart httpd.service
```

Creamos un enlace para que automatizar el arranque del servicio de openca…

```
# ln -s /opt/openca/etc/init.d/openca /etc/init.d/openca
```

Arrancamos el servicio nuevamente…

```
# /etc/init.d/openca start
```

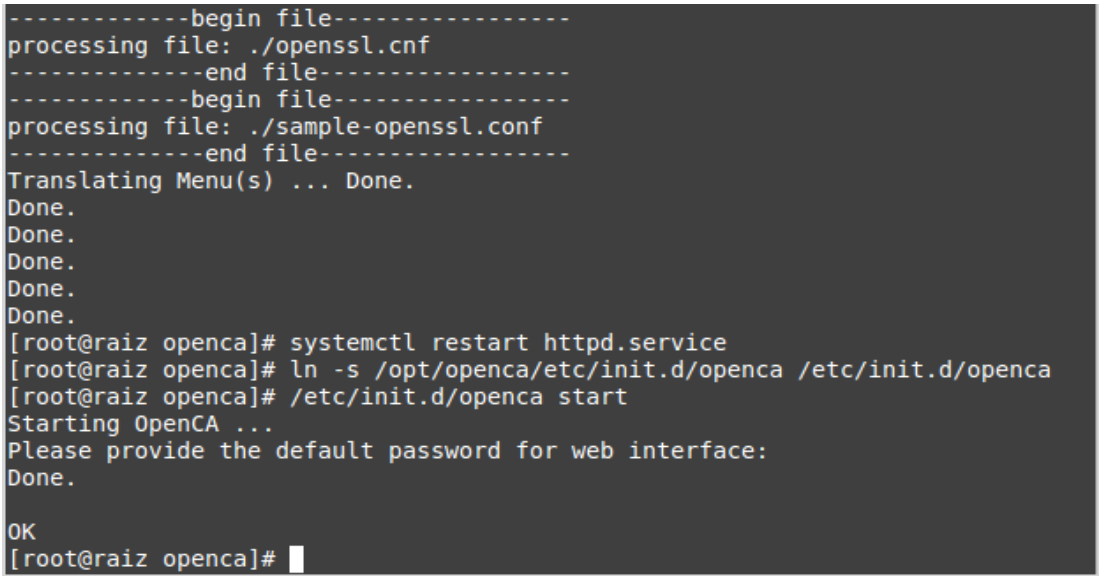


Figura 38: Finalizando la instalación de OpenCA

Ahora probamos el servicio en nuestro navegador usando la siguiente dirección

http://192.168.0.102/pki/

Y preguntará si deseamos agregar este sitio como excepción.

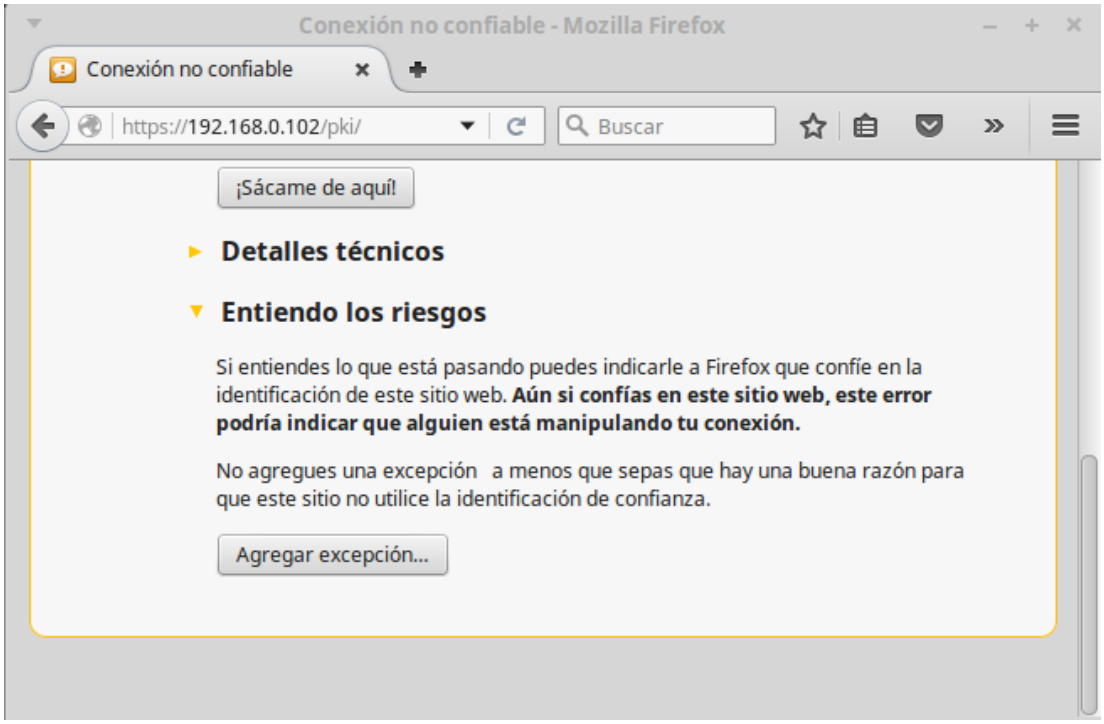


Figura 39: Accesando a OpenCA

Se desplegarán varias carpetas…

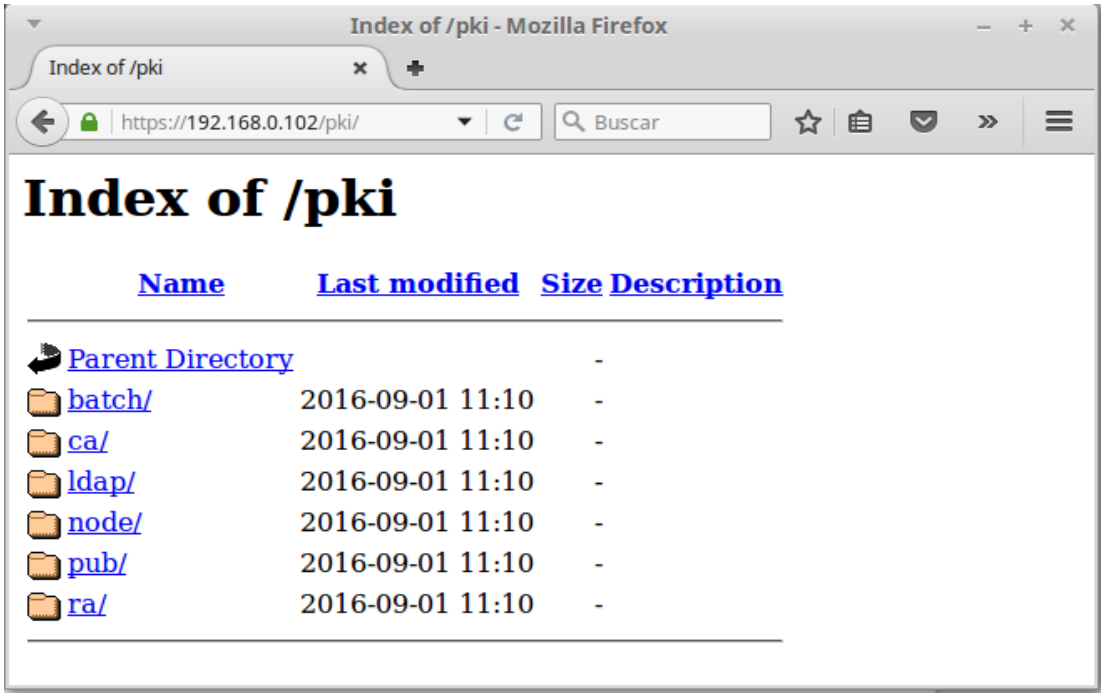


Figura 40: Navegando por las carpetas de OpenCA

Si muestra el siguiente error…

Figura 41: El error típico de OpenCA

Se debe cambiar de dueño de la carpeta /opt/openca

```
# chown apache:apache /opt/openca -R
```

También en el archivo openca_start debemos realizar los siguientes cambios:

```
# nano /opt/openca/etc/openca/openca_start
```

Buscamos las siguientes líneas httpd_user y httpd_group y las dejamos como se muestran a continuación…

```
$AUTOCONF {"httpd_user"}     = "apache";

$AUTOCONF {"httpd_group"}    = "apache";
```

A continuación, paramos el servicio y lo iniciamos nuevamente.

```
# cd /opt/openca/etc/openca/

# ./openca_stop

# ./openca_start
```

Y probamos si los cambios surtieron efecto

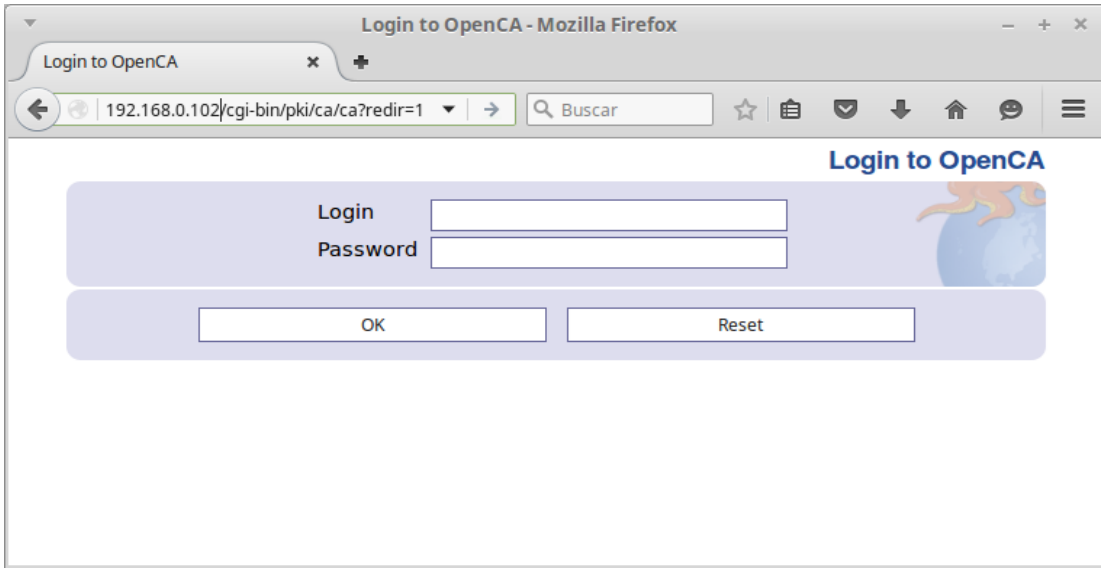


Figura 42: Pantalla de Login de OpenCA

Ingresamos el usuario admin y la clave creada anteriormente en el paso de instalación y configuración de openldap.

Una vez dentro del sistema podemos ya crear las autoridades certificadoras y demás elementos de la PKI.

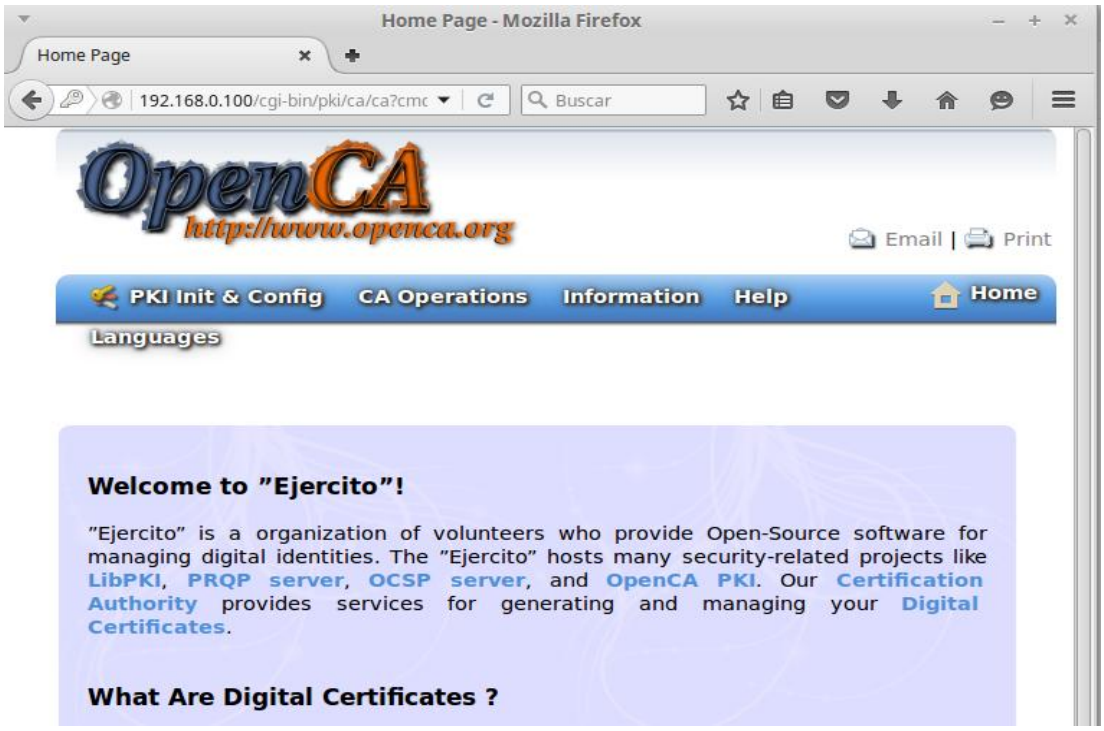


Figura 43: Uso aplicativo de OpenCA

4.3.2.3 EJBCA

EjbCa es una herramienta de software libre para construir una PKI basada en JAVA muy robusta, pero a la vez es compleja, desde la instalación hasta el uso de la misma, y para su instalación se requieren los siguientes paquetes:

1. EjbCA v6
2. jBoss v7.1.1
3. Apache Ant v1.9.7
4. JAVA v1.8

Para iniciar la instalación de EjbCa vamos a instalar wget y unzip:

```
# yum install wget

# yum install unzip
```

Figura 44: Descargando y descomprimiendo EjbCa

Ahora procedemos a descargar los siguientes paquetes: jdk, ejbc, jboss y apache ant y los copiamos al directorio /opt…

```
# cd /opt

# wget https://sourceforge.net/projects/ejbca/files/ejbca6/

# wget http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.Final/jboss-as-7.1.1.Final.zip

# wget http://www-us.apache.org/dist//ant/binaries/apache-ant-1.9.7-bin.zip

# wget http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
```

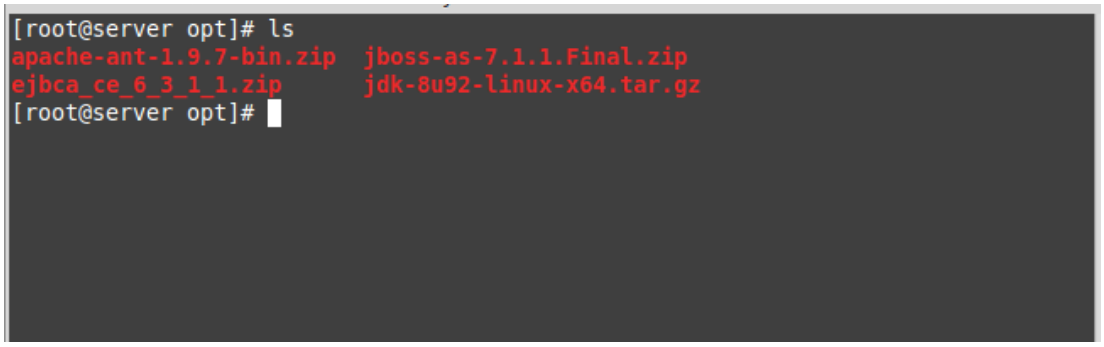
En su defecto se puede instalar el open jdk.

```
# yum install java-1.7.0-openjdk
```

Se puede cambiar de version de jdk por defecto usando el comando:

```
#  update-alternatives --config java
```

Una vez descargados los archivos ya los vamos a poder descomprimir




```
#unzip apache-ant-1.9.7-bin.zip

# unzip ejbca_ce_6_3_1_1.zip

# unzip jboss-as-7.1.1.Final.zip

# tar -xvf jdk-8u92-linux-x64.tar.gz
```

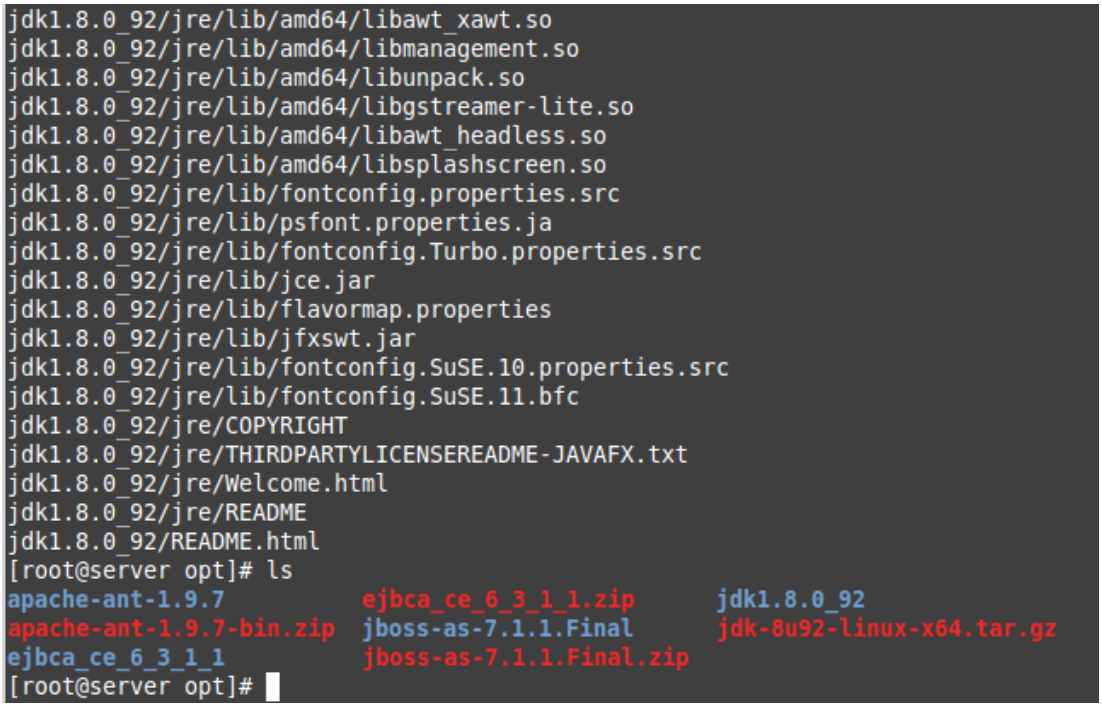


Figura 45: Navegando a través de EjbCA

Ahora procedemos a instalar el jdk de Oracle para lo cual ejecutamos los siguientes comandos:

```
# mkdir -p /usr/lib/jvm

# mv jdk1.8.0_92 java-8-oracle

# mv java-8-oracle /usr/lib/jvm/

# update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/java-8-oracle/bin/java" 1

# update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/java-8-oracle/bin/javac" 1

# update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/lib/jvm/java-8-oracle/bin/javaws" 1
```

Damos permisos de ejecución…

```
# chmod a+x /usr/bin/java

# chmod a+x /usr/bin/javac

# chmod a+x /usr/bin/javaws

# chown -R root:root /usr/lib/jvm/java-8-oracle
```

Finalmente seleccionamos la plataforma java usada por defecto

```
# update-alternatives --config java
```

```
[root@server opt]# update-alternatives --config java

Hay 3 programas que proporcionan 'java'.

  Selección    Comando
-----------------------------------------------
   1           /usr/lib/jvm/java-8-oracle/bin/java
   2           /usr/lib/jvm/jre-1.5.0-gcj/bin/java
*+ 3           /usr/lib/jvm/jre-1.7.0-openjdk/bin/java

Presione Intro para mantener la selección actual[+], o escriba el número de la selección:  3
[root@server opt]# java -version
java version "1.7.0_111"
OpenJDK Runtime Environment (rhel-2.6.7.2.el6_8-i386 u111-b01)
OpenJDK Client VM (build 24.111-b01, mixed mode, sharing)
```

Figura 46: Configurando JAVA

Y verificamos con el siguiente comando:

```
# java -version
```

Repetimos la operación con javaws y javac…

```
# update-alternatives --config javac

# update-alternatives --config javaws
```

Ahora vamos a proceder con la configuracion del mysql server, para ello ejecutamos los siguientes comandos:

```
#yum install mysql mysql-server
```

Ahora procedemos a iniciar el servicio y a activarlo para que inicie el con sistema.

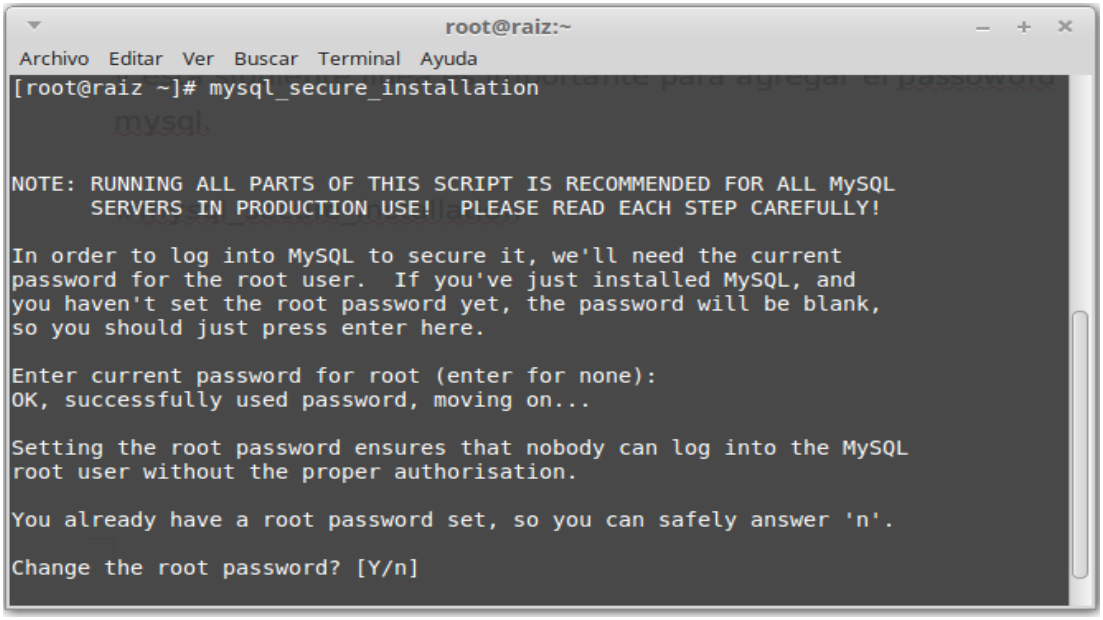
```
# chkconfig mysqld on

# service mysqld start
```

```
[root@server conf]# service mysqld start
Iniciando mysqld:                                          [  OK  ]
[root@server conf]#
```

Figura 47: Arrancando mysql

Creamos un password de acceso a mysql:

```
# mysql_secure_installation
```



Y vamos llenando los campos que pide y de esa forma creamos el password para mysql-server.

Ahora vamos a crear la base de datos con el nombre de openca, para lo cual ingresamos a mysql usando el siguiente comando.

```
# mysql -u root -p
```

Y creamos la base de datos con el comando:

```
mysql> create database ejbca;
```

El siguiente paso es proceder a instalar e iniciar el servidor jboss, para lo cual cambiamos el nombre de la carpeta:

```
# mv jboss-as-7.1.1.Final jboss
```

```
[root@server opt]# ls
apache-ant-1.9.7          ejbca_ce_6_3_1_1.zip      jdk1.8.0_92
apache-ant-1.9.7-bin.zip  jboss                     jdk-8u92-linux-x64.tar.gz
ejbca                     jboss-as-7.1.1.Final.zip
[root@server opt]#
```

Ingresamos a la carpeta jboss y arrancamos el servidor

```
# cd jboss/bin

# ./standalone.sh -Djboss.bind.address=0.0.0.0 -Djboss.bind.address.management=0.0.0.0&
```

Para probar el servidor en el navegador ponemos la IP y el puerto 8080:

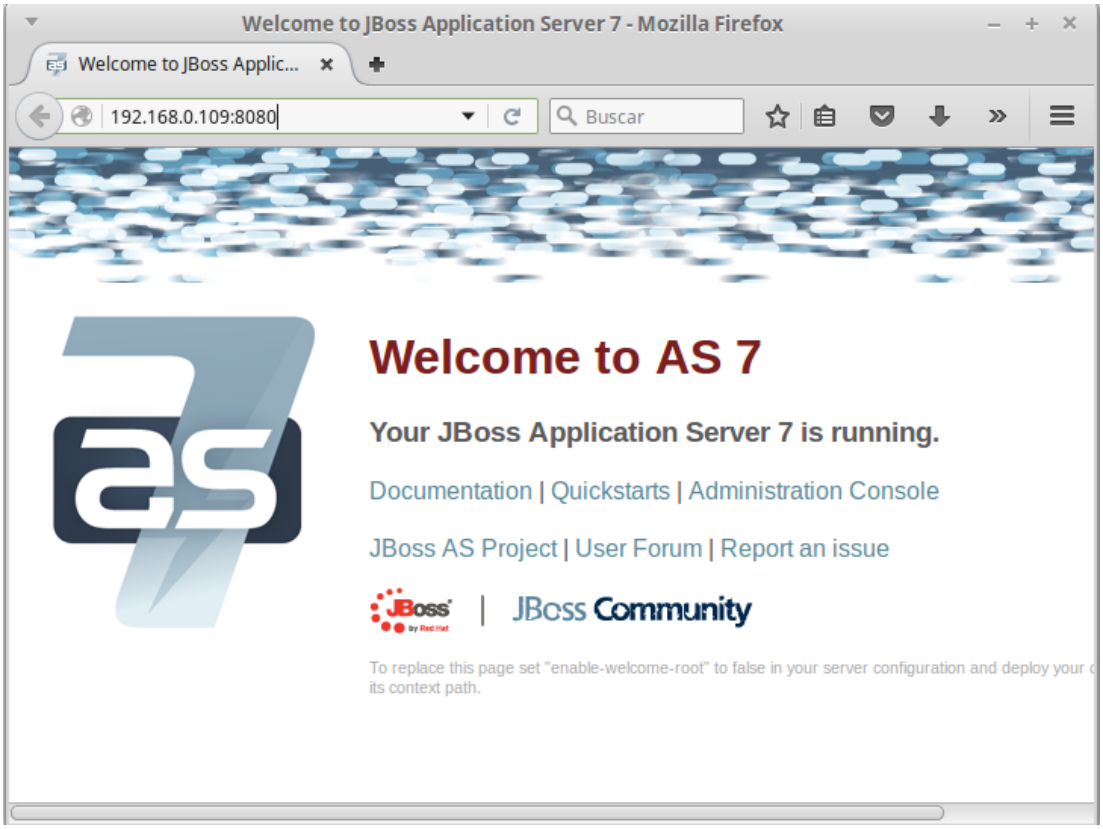


Figura 48: Prueba de Instalación de JBoss 7

Ahora procedemos a instalar el paquete Ant de Apache…

```
# yum install ant
```

```
[root@server opt]# yum install ant
Complementos cargados:fastestmirror
Configurando el proceso de instalación
Loading mirror speeds from cached hostfile
 * base: centos.brisanet.com.br
 * extras: centos.brisanet.com.br
 * updates: centos.brisanet.com.br
El paquete ant-1.7.1-15.el6.i686 ya se encuentra instalado con su versión más re
ciente
Nada para hacer
[root@server opt]#
```

Figura 49: Instalando el compilador ANT

Si se diera el caso y se requiriera instalar de forma manual, se lo puede hacer usando los paquetes descargados, para ello simplemente creamos links simbólicos:

```
# ln -s /opt/apache-ant-1.9.2 /opt/ant

# sh -c 'echo ANT_HOME=/opt/ant >> /etc/environment'

# ln -s /opt/ant/bin/ant /usr/bin/ant

y verifcamos con el siguiente comando

# ant -version
```

```
[root@server apache-ant-1.9.7]# ant -version
Apache Ant version 1.7.1 compiled on May 10 2016
```

Ahora procedemos a compilar e instalar el paquete de ejbca, para ello usamos el comando ant para que se cree el archivo ebjca.ear el cual se desplegara en el servidor de aplicaciones jBoss v7. Cambiamos el nombre de la carpeta ejbca

```
# mv ejbca_ce_6_3_1_1 ejbca
```

Ejecutamos un comando ls para verificar…

```
[root@server opt]# ls
apache-ant-1.9.7          ejbca_ce_6_3_1_1.zip      jdk1.8.0_92
apache-ant-1.9.7-bin.zip  jboss-as-7.1.1.Final      jdk-8u92-linux-x64.tar.gz
ejbca                     jboss-as-7.1.1.Final.zip
```

Ingresamos a la carpeta de ejbca…

```
# cd ejbca
```

Y ejecutamos el comando ls para ver que archivos contiene…

```
[root@server ejbca]# ls
bin         Changelog.txt   docs.xmli  modules              removed.xml
build.xml   deprecated.xml  echo.xml   propertyDefaults.xml  src
conf        doc             lib        README               test.xmli
```

Ingresamos a la carpeta conf y procedemos y ejecutamos un ls para verificar que archivos contiene.

```
bin         Changelog.txt   docs.xmli  modules              removed.xml
build.xml   deprecated.xml  echo.xml   propertyDefaults.xml  src
conf        doc             lib        README               test.xmli
[root@server ejbca]# cd conf
[root@server conf]# ls
batchtool.properties.sample       jndi.properties.jbosseap6
cache.properties.sample           jndi.properties.weblogic
catoken.properties.sample         jndi.properties.websphere
certstore.properties.sample       log4j-glassfish.xml.sample
cesecore.properties.sample        log4j-jboss6.xml.sample
cmptcp.properties.sample          log4j-jbosseap6.xml.sample
crlstore.properties.sample        log4j-weblogic.xml.sample
custom.properties.sample          log4j-websphere.xml.sample
database.properties.sample        logdevices
ejbca.properties.sample           mail.properties.sample
extendedkeyusage.properties       ocsp.properties.sample
externalra-gui.properties.sample  plugins
externalra.properties.sample      scepra.properties.sample
install.properties.sample         systemtests.properties.sample
jaxws.properties.sample           va.properties.sample
jndi.properties.glassfish         va-publisher.properties.sample
jndi.properties.jboss             web.properties.sample
jndi.properties.jboss7            xkms.properties.sample
[root@server conf]#
```

Figura 50: Carpetas de configuración de EjbCA

Como vemos la mayoría de archivos son de ejemplo, ahora vamos a editar los archivos más importantes que son el de configuración y el de base de datos.

Antes de ello copiamos los ejemplos para trabajar sobre ellos.

```
# cp ejbca.properties.sample ejbca.properties

# cp database.properties.sample database.properties
```

Ahora abrimos dichos archivos y los configuramos:

```
# nano ejbca.properties
```

Agregamos las siguientes lineas

```
appserver.home=/opt/jboss

appserver.type=jboss

ejbca.productionmode=false

allow.external-dynamic.configuration=true
```

Guardamos el archivo usando las teclas control + o y control + x para salir.

```
# nano database.properties
```

Agregamos las siguientes líneas…

```
database.name=mysql

database.url=jdbc:mysql://127.0.0.1:3306/ejbca

database.driver=com.mysql.jdbc.Driver

database.username=root

database.password=V*****
```

Guardamos y salimos del archivo de configuracion y procedemos a compilar usando el comando Ant de Apache.

```
# cd ..

# ant deploy

# ant install
```

Si sale el error ***"/opt/ejbca/propertyDefaults.xml:102: No supported regular expression matcher found: java.lang.ClassNotFoundException: org.apache.tools.ant.util.regexp.Jdk14RegexpRegexp".*** *Instalamos* el siguiente paquete:

```
# yum install ant-apache-regexp
```

Nuevamente ejecutamos los comandos anteriores y agregamos las variables de entorno.

```
# export EJBCA_HOME="/opt/ejbca"

 # export JAVA_HOME="/usr/lib/jvm/java-1.7.0-openjdk"

 # export APPSRV_HOME="/opt/jboss"

 # export JBOSS_HOME="$APPSRV_HOME"

 # export ANT_HOME="/usr/share/ant"

 # export PATH="$JAVA_HOME/bin:$ANT_HOME/bin:$PATH"
```

Como siguiente paso creamos la conexion con la base de datos desde jboss a mysql.

```
mkdir -p /opt/jboss/modules/com/mysql/main/
```

Descargar el driver de mysql y colocarlo en la siguiente carpeta:

```
cp /opt/mysql-connector-java-5.1.23-bin.jar /opt/jboss/modules/com/mysql/main/
```

Crear un archivo xml indicando el driver-class

```
# nano /opt/jboss/modules/com/mysql/main/module.xml
```

Y colocar el siguiente código…

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.23-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Ahora cargamos el driver, para lo cual ingresamos a la siguiente dirección…

```
# cd /opt/jboss/bin
```

Y ejecutamos…

```
# ./jboss-cli.sh
```

Tecleamos la palabra "***connect",*** y cargamos el driver usando el siguiente código:

```
/subsystem=datasources/jdbc-driver=com.mysql.jdbc.Driver:add(driver-
name=com.mysql.jdbc.Driver,driver-class-name=com.mysql.jdbc.Driver,driver-module-
name=com.mysql,driver-xa-datasource-class-
name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)

:reload
```

Finalmente cargamos el recurso de conexión a la base de datos:

```
data-source add --name=ejbcads --driver-name="com.mysql.jdbc.Driver" --connection-
url="jdbc:mysql://127.0.0.1:3306/ejbca" --jndi-name="java:/EjbcaDS" --use-ccm=true --
driver-class="com.mysql.jdbc.Driver" --user-name="root" –password="Vale.2009" --
validate-on-match=true --background-validation=false --prepared-statements-cache-
size=50 --share-prepared-statements=true --min-pool-size=5 --max-pool-size=150 --pool-
prefill=true --transaction-isolation=TRANSACTION_READ_COMMITTED --check-valid-
connection-sql="select 1;"
```

Y ejecutamos unas configuraciones adicionales…

```
/system-property=org.jboss.as.logging.per-deployment:add(value=false)

/subsystem=logging/logger=org.ejbca:add

/subsystem=logging/logger=org.ejbca:write-attribute(name=level, value=DEBUG)

/subsystem=logging/logger=org.cesecore:add

/subsystem=logging/logger=org.cesecore:write-attribute(name=level, value=DEBUG)
```

Ejecutamos de nuevo el comando para desplegar el ejbca…

```
# cd /opt/ejbca

# ant deploy

# ant install

para glassfish

# ant clean

# ant bootstrap

También podemos utilizar las instrucciones:

# ant clean

# ant

# asadmin deploy --precompilejsp $EJBCA_HOME/dist/ejbca.ear
```

Hay que resaltar, que ejbca solo trabaja en la versión 2.1 de glassfish, mientras que en las más actuales aún no se encuentra soportado.

Una vez instalado procedemos a verificar la instalación. Colocamos en el navegador la IP del servidor, por ejemplo:

http://192.168.0.109:8080/ejbca/

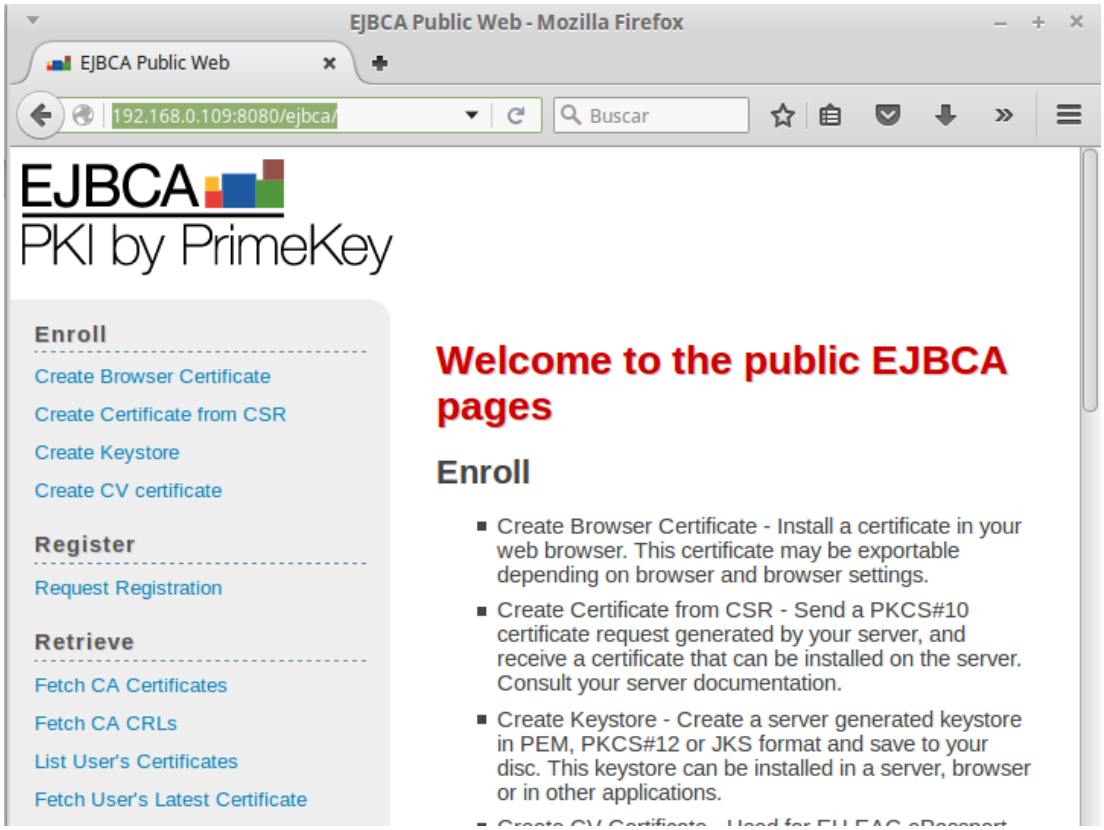


Figura 51: Probando EjbCA

4.3.3 Análisis y selección de la herramienta PKI a ser utilizada

Una vez instaladas las herramientas a simple vista se puede apreciar algunas diferencias significativas que se explican en la siguiente matriz comparativa:

Tabla 6:

Cuadro comparativo entre herramientas PKI

	OpenSSL	OpenCA	EjbCA
Administración	Línea de comandos	Consola grafica	Consola grafica
Plataforma	Multiplataforma	Multiplataforma	Multiplataforma
Lenguaje	C++	Perl	Java

Dificultad en Administración	Alta	Media	Media
Módulos	C++ Modules	Perl Modules	EJB
Basado en Componentes	No	Si	Si
Navegadores Soportados	Múltiples	múltiples	múltiples
Soporte LDAP	Si	Si	Si
Dificultad en Configuración	Fácil	Compleja	Muy Compleja
Autenticación	Si	Si	Si
No repudio	Si	Si	Si
Integridad	Si (mediante encriptación)	Si (mediante encriptación)	Si (mediante encriptación)
Confidencialidad	Si (mediante encriptación)	Si (mediante encriptación)	Si (mediante encriptación)
Escalabilidad	Baja	Media	Alta

Resultado de análisis y selección de la herramienta:

Como podemos apreciar entre las herramientas PKI basadas en software libre la más completa es EjbCA seguida por OpenCA, sin embargo, utilizaremos OpenSSL para implementar nuestra arquitectura PKI porque es una herramienta que trabaja a nivel consola, y se la puede acoplar fácilmente en el Sistema Integrado de la Fuerza Terrestre SIFTE, tanto para la crear la Autoridades Certificadoras, como para las entidades de registro, validación de certificados y firmas electrónicas.

4.4 Implementación De La Autoridad Certificadora (Ca) [Raíz]

Implementación de la Autoridad de Certificación Raíz

Creamos la carpeta root_ca

```
# mkdir root_ca
```

Ingresamos a la carpeta creada

```
# cd root_ca
```

Ahora procedemos a copiar en nuestra carpeta el arhivo de configuración de openssl

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

Lo abrimos y lo configuramos

```
# nano openssl.cnf
```

Nos vamos a la línea

```
RANDFILE              = $ENV::HOME/.rnd
```

Y la comentamos…

```
#RANDFILE             = $ENV::HOME/.rnd
```

Posteriormente vamos a la linea

```
dir             = /etc/pki/CA               # Where everything is kept
```

Y la modificamos

```
dir             = . #/etc/pki/CA            # Where everything is kept
```

El archivo va a quedar de la siguiente forma:

Figura 52: Fichero de configuración de OpenSSL

Guardamos los cambios usando Ctrl + o y salimos usando Ctrl + x

Ahora procedemos a crear los siguientes directorios

```
# mkdir certs
# mkdir crl
# mkdir newcerts
# mkdir private
```

Creamos el archivo serial y lo cargamos con el dato 0100

```
# touch serial
# echo 0100 > serial
```

Ahora vamos a crear el archivo index y crlnumber

```
# touch index.txt
# touch crlnumber
# echo 0100 > crlnumber
```

Generamos un número aleatorio

```
# openssl rand -out ./private/.rand 1024
```

Ahora generamos la clave privada de la Autoridad certificadora

```
# openssl genrsa -out ./private/cakey.pem -des3 -rand ./private/.rand 2048
```

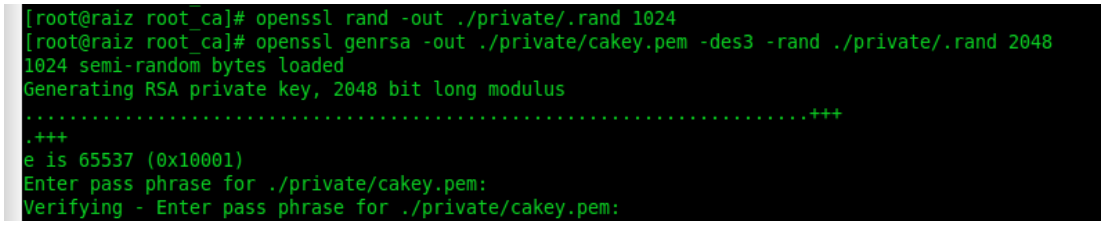


Figura 53: Generando la Clave Privada

Creamos un certificado autofirmado

```
# openssl req -x509 -new -key ./private/cakey.pem -out cacert.pem -config openssl.cnf
```

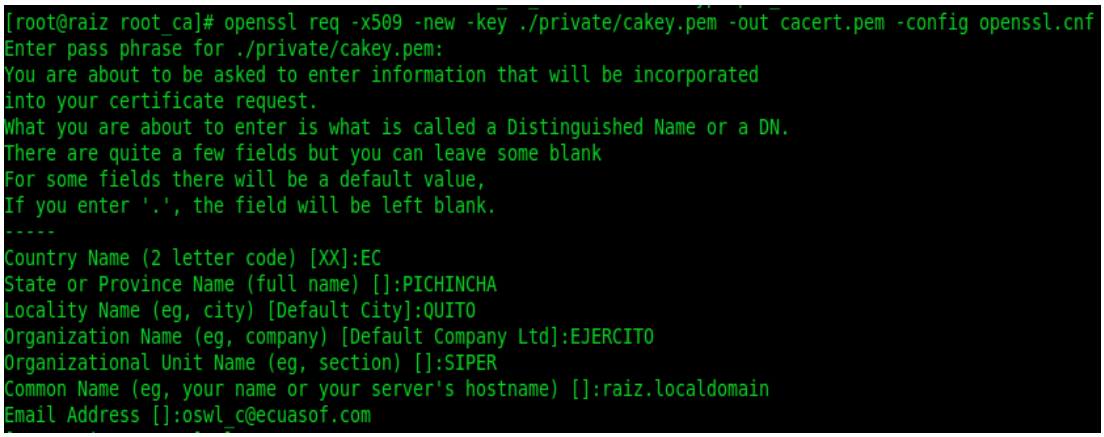


Figura 54: Generando el certificado raíz autofirmado

Lista la configuración de la Autoridad Certificadora Raíz Ahora procedemos ha configurar la Autoridad Certificadora Subordinada.

4.5 Implementación De La Autoridad Certificadora (Ca) [Subordinada]

Creamos la carpeta sub_ca

```
# mkdir sub_ca
```

Ingresamos a la carpeta creada

```
# cd sub_ca
```

Y copiamos el archivo de configuración de openssl

```
# cp /etc/pki/tls/openssl.cnf .
# nano openssl.cnf
```

Nos vamos a la línea

```
RANDFILE              = $ENV::HOME/.rnd
```

Y la comentamos…

```
#RANDFILE             = $ENV::HOME/.rnd
```

Posteriormente vamos a la línea

```
dir             = /etc/pki/CA               # Where everything is kept
```

Y la modificamos

```
dir             = . #/etc/pki/CA              # Where everything is kept
```

El archivo va a quedar de la siguiente forma:

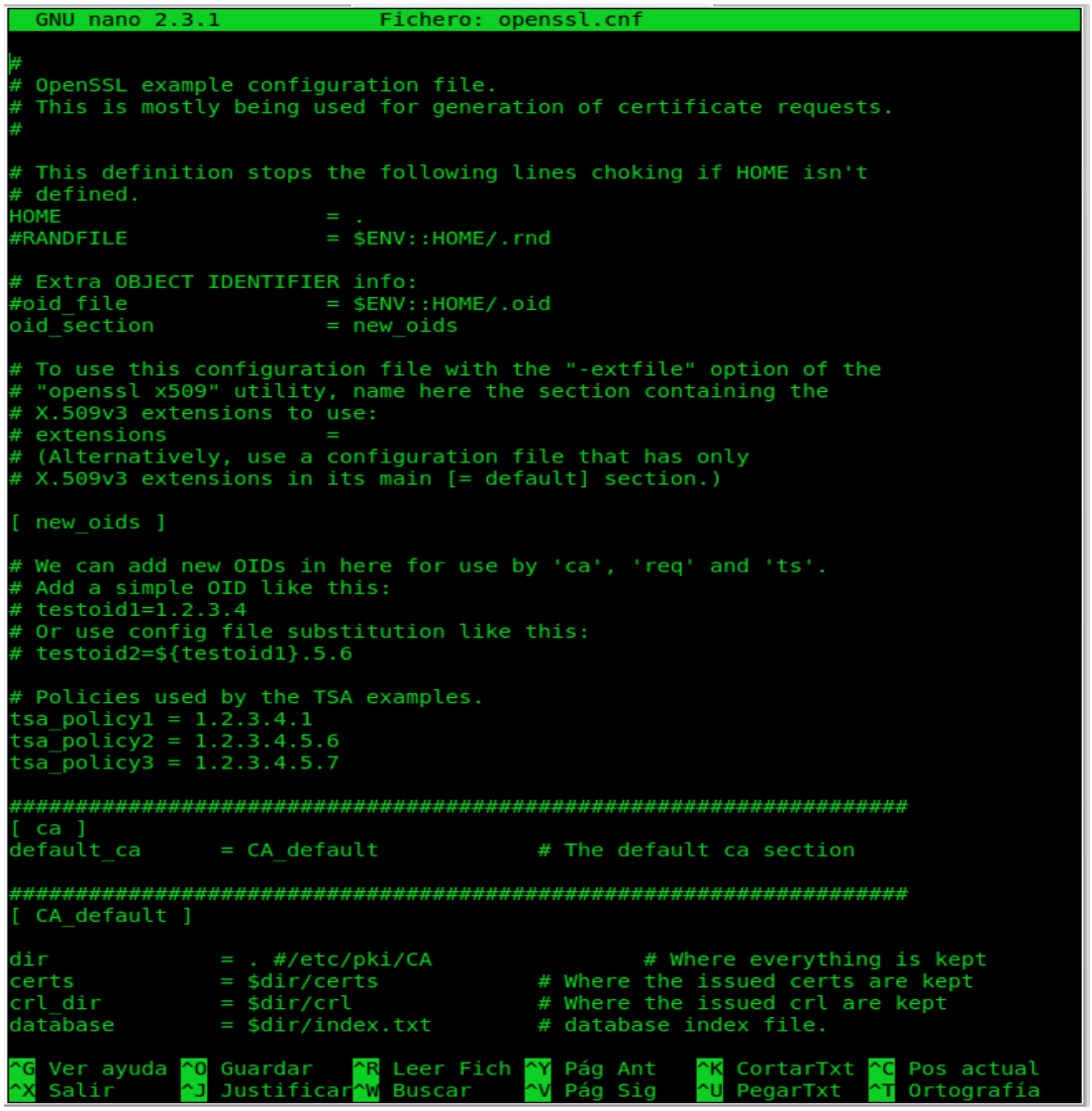


Figura 55: Cambio de políticas en OpenSSL

Guardamos los cambios usando Ctrl + o y salimos usando Ctrl + x

Ahora creamos los siguientes directorios:

```
# mkdir certs
# mkdir crl
# mkdir newcerts
# mkdir private
```

Y un archivo serial al cual lo cargamos con el valor 0100

```
# touch serial
# echo 0100 > serial
```

Y hacemos lo mismo para el archivo index y crlnumber

```
# touch index.txt
# touch crlnumber
# echo 0100 > crlnumber
```

Ahora regeneramos número aleatorios

```
# openssl rand -out ./private/.rand 1024
```

Generamos la clave privada de la Autoridad Subordinada

```
# openssl genrsa -out ./private/cakey.pem -des3 -rand ./private/.rand 2048
```

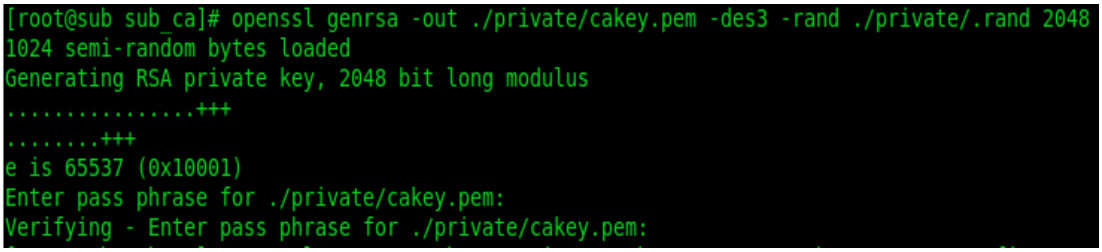


Figura 56: Clave privada de la autoridad certificadora subordinada

Crear el requerimiento de certificado

```
# openssl req -new -key ./private/cakey.pem -out subcareq.pem -config openssl.cnf
```

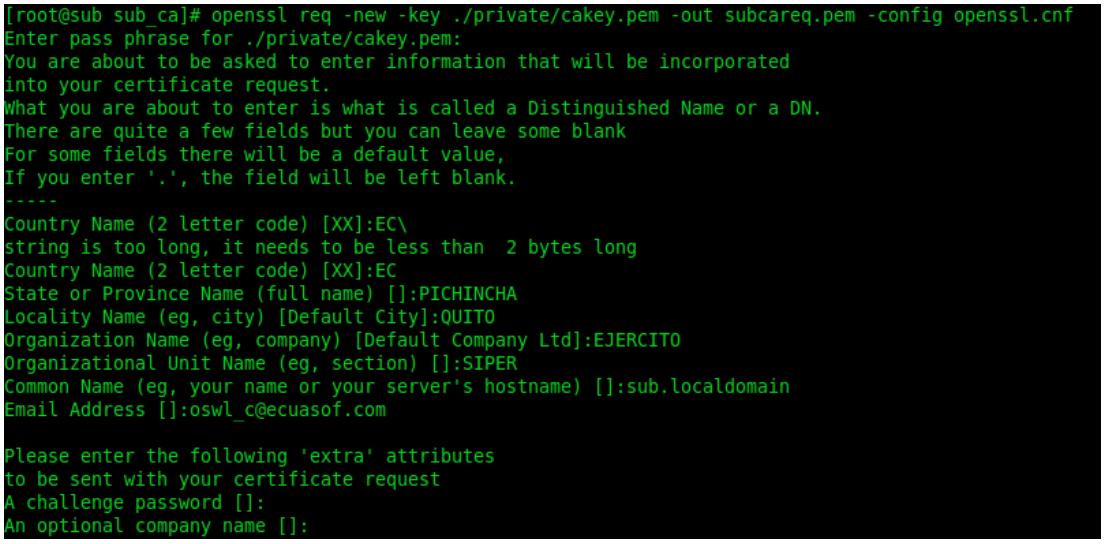


Figura 57: Requerimiento del certificado de entidad subordinada

Finalmente procedemos a copiar este requerimiento de certificado a nuestra autoridad de certificación raíz para generar un nuevo certificado firmado.

```
# scp subcareq.pem root@192.168.0.102:/root/root_ca
```

En la entidad certificadora raiz procedemos a firmar el certificado de la autoridad certificadora subordinada.

```
# openssl ca -in ../sub_ca/subcareq.pem -extensions v3_ca -config openssl.cnf
```

```
[root@raiz root_ca]# openssl ca -in subcareq.pem -extensions v3_ca -config
openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ./private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 256 (0x100)
        Validity
            Not Before: Sep 27 14:38:25 2016 GMT
            Not After : Sep 27 14:38:25 2017 GMT
        Subject:
            countryName               = EC
            stateOrProvinceName       = PICHINCHA
            organizationName          = EJERCITO
            organizationalUnitName    = SIPER
            commonName                = sub.localdomain
            emailAddress              = oswl_c@ecuasof.com
        X509v3 extensions:
            X509v3 Subject Key Identifier:
```

```
            7D:E0:11:E7:89:AC:48:02:17:32:38:E7:3D:81:56:BA:07:D0:21:77
        X509v3 Authority Key Identifier:
keyid:00:9B:F9:B1:8E:3B:F5:28:AE:4E:08:61:EA:18:62:C3:9B:99:A8:22


        X509v3 Basic Constraints:
          CA:TRUE
Certificate is to be certified until Sep 27 14:38:25 2017 GMT (365 days)
Sign the certificate? [y/n]:y



1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 256 (0x100)
   Signature Algorithm: sha256WithRSAEncryption
      Issuer: C=EC, ST=PICHINCHA, L=QUITO, O=EJERCITO, OU=SIPER,
CN=raiz.localdomain/emailAddress=oswl_c@ecuasof.com
      Validity
         Not Before: Sep 27 14:38:25 2016 GMT
         Not After : Sep 27 14:38:25 2017 GMT
      Subject: C=EC, ST=PICHINCHA, O=EJERCITO, OU=SIPER,
CN=sub.localdomain/emailAddress=oswl_c@ecuasof.com
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
```

```
        Public-Key: (2048 bit)
        Modulus:
            00:dd:02:4a:65:29:9f:72:bb:6f:e9:56:bd:9d:e2:
            99:bc:82:a7:44:80:7d:19:7c:84:5a:5f:0e:e8:1d:
            3d:3f:44:ab:51:7f:77:21:1e:83:1f:bd:44:4d:5c:
            34:0b:b4:72:26:96:ea:c1:2a:64:87:ed:0a:67:46:
            41:47:05:a3:9e:d9:7d:da:00:43:40:0d:70:28:de:
            2d:73:77:24:23:7d:03:93:c3:d7:63:7e:f2:e4:04:
            a1:7a:29:62:cb:09:d8:73:15:9b:22:88:3b:e2:d0:
            d3:25:78:62:ac:68:4a:22:e3:d0:2b:d9:6f:2b:05:
            f5:ab:0e:10:da:4c:f6:75:be:aa:e4:42:85:5b:61:
            d4:cb:07:60:27:28:c5:5d:3b:d8:b4:a9:d9:20:ae:
            f8:70:3f:07:bb:ed:be:c6:dc:84:28:2f:37:69:9b:
            7a:38:06:bd:71:48:40:cd:23:ff:45:3c:f2:d6:58:
            ac:20:b7:aa:e4:65:92:32:34:d7:2c:46:f0:ca:17:
            68:32:62:10:a9:08:86:f4:29:f1:36:04:c1:85:a0:
            0e:85:3b:d6:02:76:27:de:cd:4c:a6:ff:9c:e9:1e:
            ec:df:3c:89:c3:39:15:eb:14:ac:e6:85:8f:8a:a6:
            34:3b:1b:95:29:d0:55:e8:92:bc:4d:90:7b:ca:df:
            f4:af
        Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            7D:E0:11:E7:89:AC:48:02:17:32:38:E7:3D:81:56:BA:07:D0:21:77
        X509v3 Authority Key Identifier:
```

```
keyid:00:9B:F9:B1:8E:3B:F5:28:AE:4E:08:61:EA:18:62:C3:9B:99:A8:22

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
         93:8c:9e:b1:c2:87:ab:c6:6b:96:f7:14:20:b4:46:25:0b:97:
         04:c8:d8:2e:7c:bd:8a:c0:ad:c3:6a:1a:9f:90:0c:4e:80:9a:
         31:44:52:5f:1e:e6:cd:bf:24:13:aa:a1:27:e2:02:f2:3b:b3:
         13:72:ef:3d:de:3e:28:e2:1b:f0:38:3c:3e:96:67:f6:2e:d8:
         0c:cc:b8:36:db:dc:9a:7d:86:e2:08:ca:59:ed:f7:24:89:cc:
         9b:b8:f4:5e:95:ae:ef:22:6d:2f:a5:9d:78:45:03:a2:e0:b4:
         77:75:fa:81:4b:8d:72:1b:8c:86:76:d3:47:72:90:97:5c:2e:
         24:24:40:44:30:50:f1:18:e6:bf:36:f8:e9:da:0d:fb:dd:a7:
         7c:f5:cd:35:dc:5c:4b:4f:a8:35:13:81:34:7e:b3:12:2f:6c:
         5b:87:fd:2c:35:cb:f0:79:63:41:43:c5:58:fc:08:71:59:48:
         51:31:fe:c0:85:6b:5b:d6:e9:7e:33:fc:31:52:70:a8:b4:26:
         1e:53:80:77:de:7e:bb:13:2f:03:e1:17:de:d2:f0:13:f8:03:
         3f:52:37:08:ad:63:78:bc:35:a9:af:ed:ab:d8:69:ad:ee:94:
         ab:39:ea:2d:a9:8a:93:ba:10:81:96:69:61:f0:b3:36:01:cf:
         bc:2a:22:df
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgICAQAwDQYJKoZIhvcNAQELBQAwgZIxCzAJBgNVBAYTAkVD
MRIwEAYDVQQIDAlQSUNISU5DSEExDjAMBgNVBAcMBVFVSVRPMREwDwYDVQQKDAhF
SkVSQ0lUTzEOMAwGA1UECwwFU0lQRVIxGTAXBgNVBAMMEHJhaXou
```

bG9jYWxkb21h

aW4xITAfBgkqhkiG9w0BCQEWEm9zd2xfY0BlY3Vhc29mLmNvbTAeFw0xNjA5Mjcx

NDM4MjVaFw0xNzA5MjcxNDM4MjVaMIGBMQswCQYDVQQGEwJFQzESMBAGA1UECAwJ

UElDSElOQ0hBMREwDwYDVQQKDAhFSkVSQ0lUTzEOMAwGA1UECwwFU0lQRVIxGDAW

BgNVBAMMD3N1Yi5sb2NhbGRvbWFpbjEhMB8GCSqGSIb3DQEJARYSb3N3bF9jQGVj

dWFzb2YuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3QJKZSmf

crtv6Va9neKZvIKnRIB9GXyEWl8O6B09P0SrUX93IR6DH71ETVw0C7RyJpbqwSpk

h+0KZ0ZBRwWjntl92gBDQA1wKN4tc3ckI30Dk8PXY37y5ASheiliywnYcxWbIog7

4tDTJXhirGhKIuPQK9lvKwX1qw4Q2kz2db6q5EKFW2HUywdgJyjFXTvYtKnZIK74

cD8Hu+2+xtyEKC83aZt6OAa9cUhAzSP/RTzy1lisILeq5GWSMjTXLEbwyhdoMmIQ

qQiG9CnxNgTBhaAOhTvWAnYn3s1Mpv+c6R7s3zyJwzkV6xSs5oWPiqY0OxuVKdBV

6JK8TZB7yt/0rwIDAQABo1AwTjAdBgNVHQ4EFgQUfeAR54msSAIXMjjnPYFWugfQ

IXcwHwYDVR0jBBgwFoAUAJv5sY479SiuTghh6hhiw5uZqCIwDAYDVR0TBAUwAwEB

/zANBgkqhkiG9w0BAQsFAAOCAQEAk4yescKHq8ZrlvcUILRGJQuXBMjYLny9isCt

w2oan5AMToCaMURSXx7mzb8kE6qhJ+IC8juzE3LvPd4+KOIb8Dg8PpZn9i7YDMy4

Ntvcmn2G4gjKWe33JInMm7j0XpWu7yJtL6WdeEUDouC0d3X6gUuNchuMhnbTR3KQ

l1wuJCRARDBQ8Rjmvzb46doN+92nfPXNNdxcS0+oNROBNH6zEi9sW4f9LDXL8Hlj

```
QUPFWPwIcVlIUTH+wIVrW9bpfjP8MVJwqLQmHlOAd95+uxMvA+EX3tLwE/gDP1I3

CK1jeLw1qa/tq9hpre6UqznqLamKk7oQgZZpYfCzNgHPvCoi3w==

-----END CERTIFICATE-----

Data Base Updated
```

Figura 58: Certificado Digital

Ahora copiamos el archivo de certificado digital al nuestro servidor o entidad de certificación subordinada.

```
# scp 0100.pem root@192.168.0.101:/root/sub_ca
```

Cambiamos el nombre de nuestros archivos

```
# mv 0100.pem cacert.pem
```

Una vez ejecutados estos pasos tendremos lista nuestra autoridad de certificación subordinada lista para operar y el certificado para importarlo a nuestros navegadores.

4.6 Importación Del Certificado Digital Al Navegador

Este certificado en formato '.pem' lo descargamos y lo instalamos en las pc de los usuarios.

Para lo cual se conecta usando el protocolo sftp, ya sea desde Windows o desde Linux, en este caso se conectará desde Linux. Antes de esto vamos a obtener la IP del servidor CA raíz, con el comando "ip add".

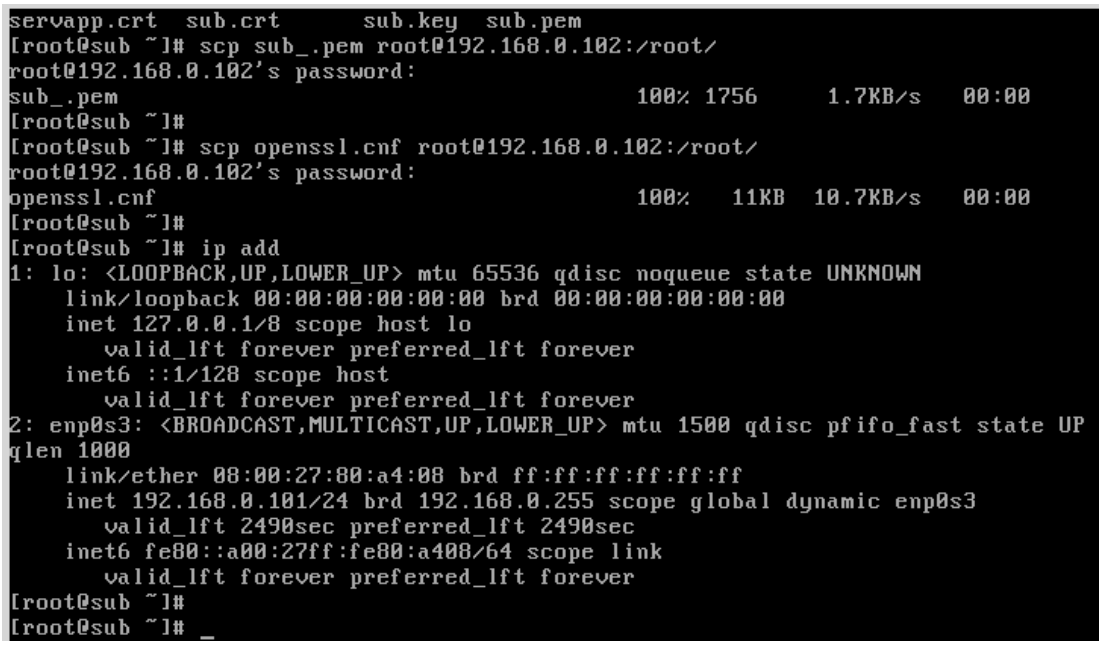


Figura 59: Verificando IP

Con esta ip vamos a conectarnos usando el protocolo sftp

sftp://root@192.168.0.101

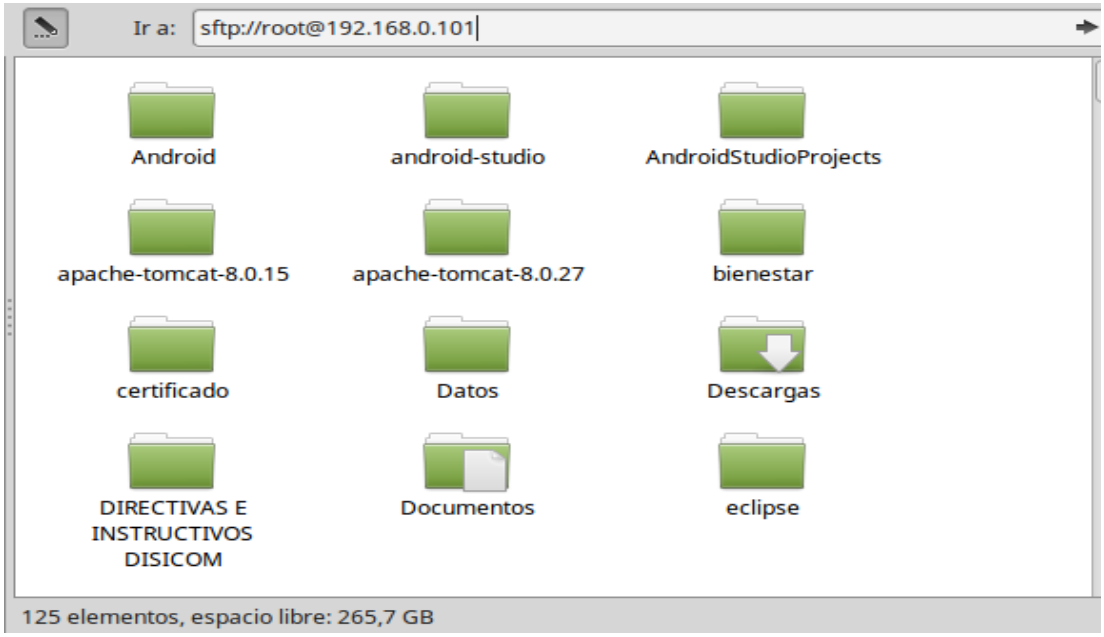


Figura 60: Navegando por la autoridad certificadora subordinada

Nos pedirá la contraseña del servidor

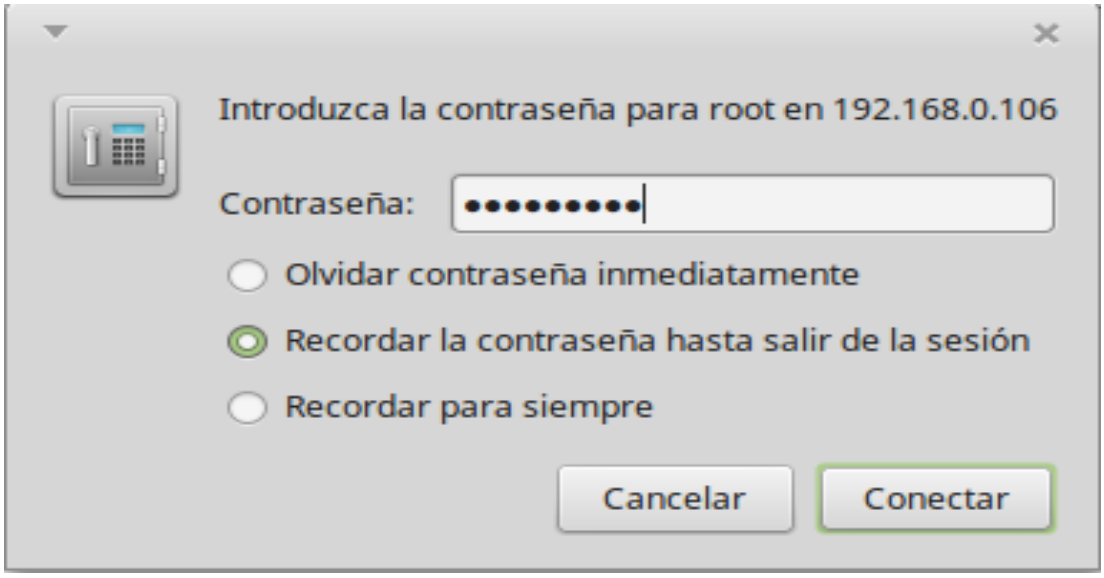


Figura 61: Pedido de clave por el servirdor

Nos dirigimos a la carpeta sftp://root@192.168.0.10/root, y buscamos el archivo cacert.pem

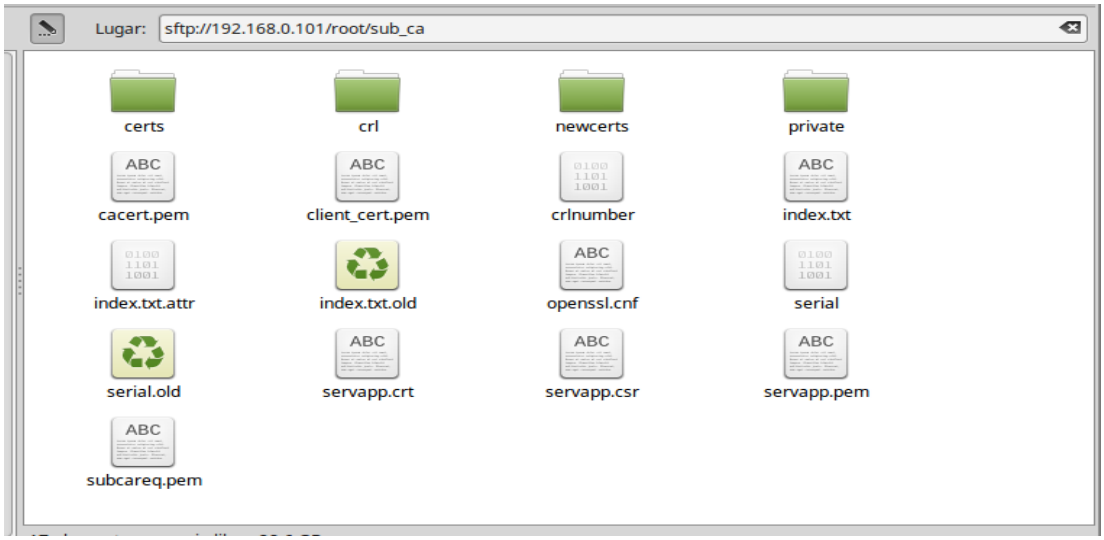


Figura 62: Árbol de directorios de la autoridad certificadora

Copiamos a nuestro pc el certificado cacert.pem, y hacemos doble clic y este mostrara la siguiente pantalla.

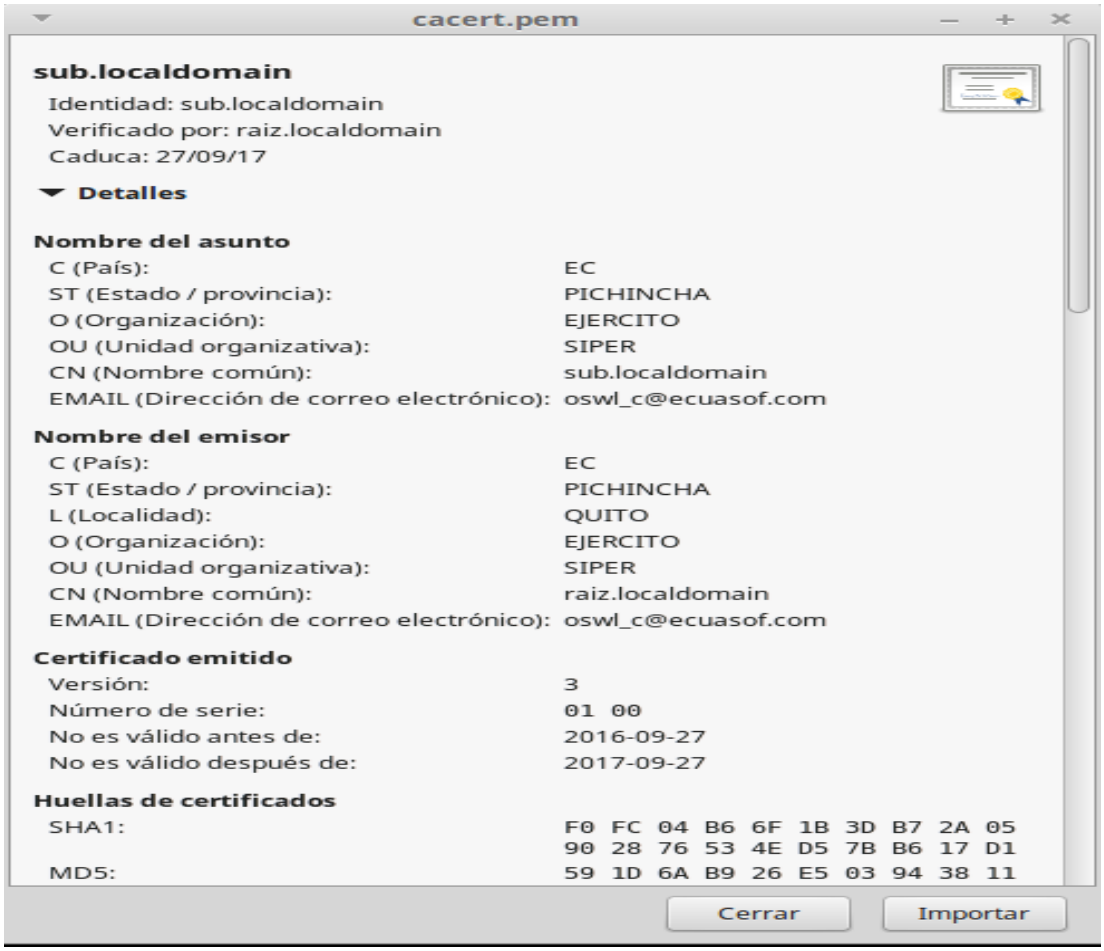


Figura 63: Certificado de la Autoridad Certificadora Subordinada

Damos clic en importar y luego en aceptar. Ahora vamos a importar este certificado en el navegador chrome, para ello abrimos chrome y colocamos lo siguiente en la barra de dirección: ***chrome://settings/*** y damos clic en donde dice configuración avanzada y navegamos hasta la sección https/ssl, y damos clic en el botón, Administrar Certificados.

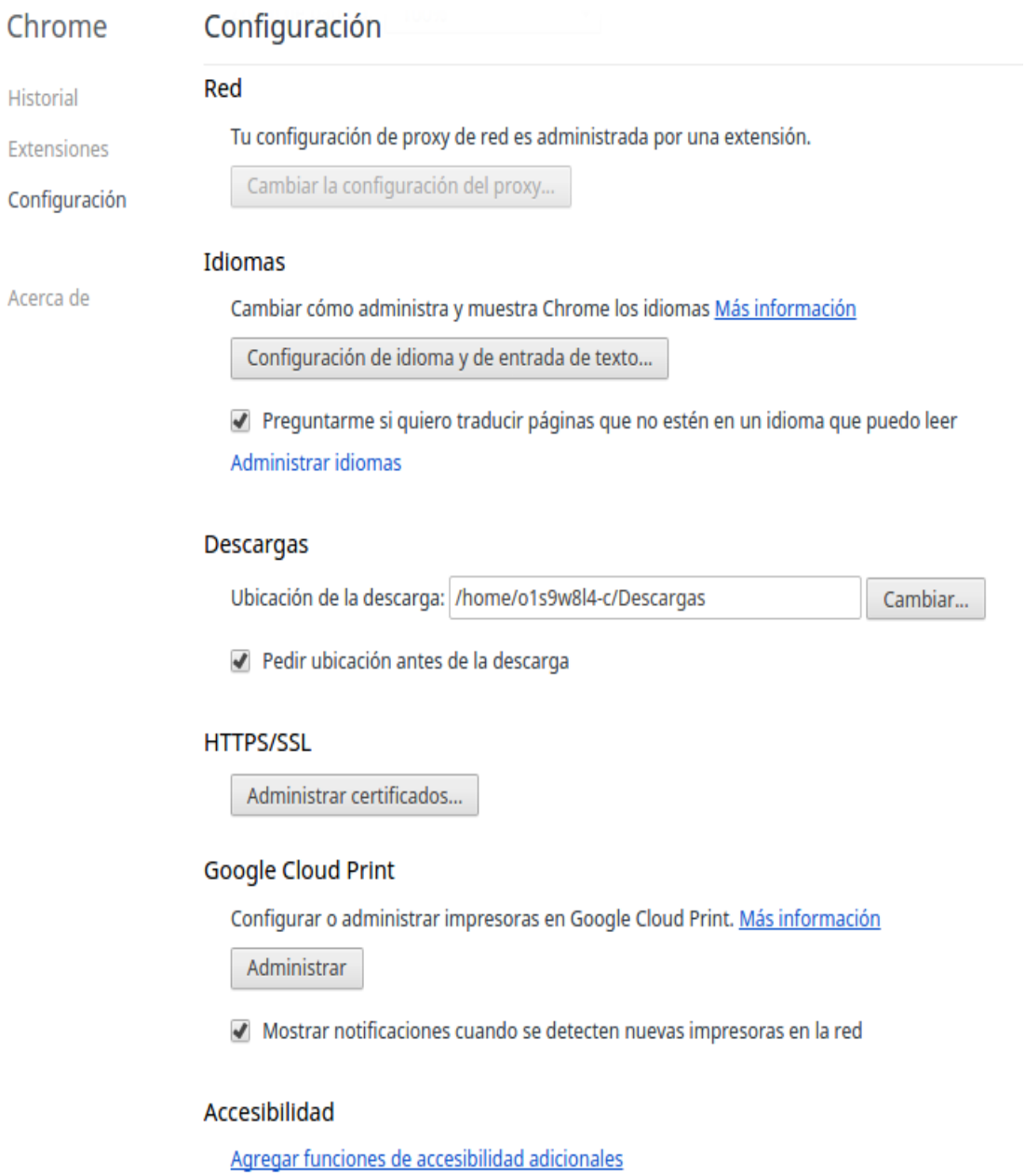
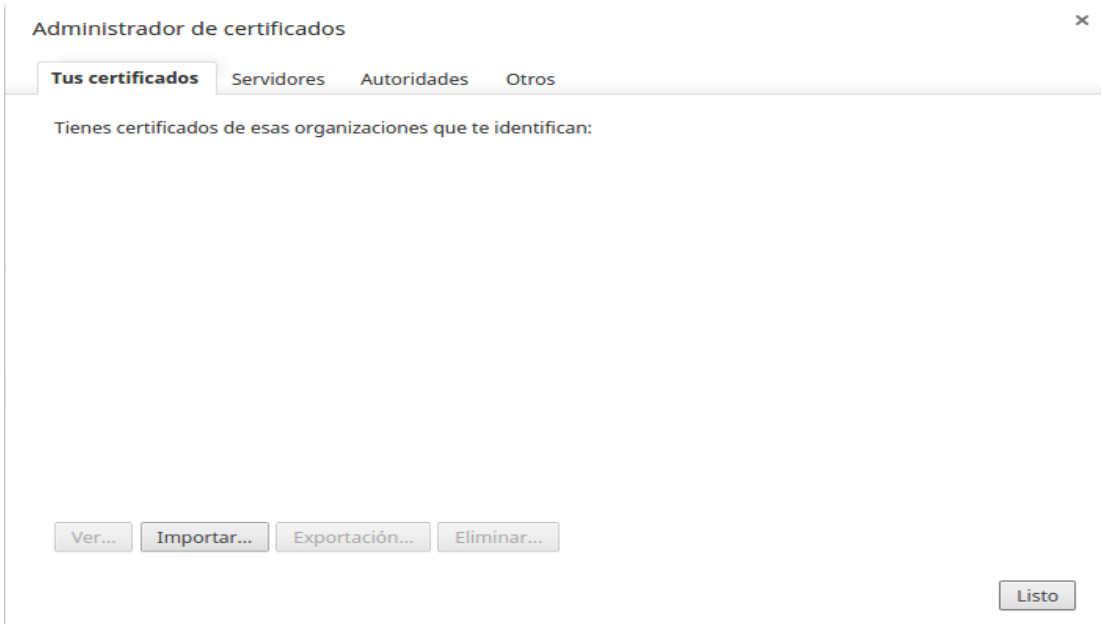


Figura 64: Importación del certificado al navegador

Y mostrara la siguiente ventana…



Y ahora vamos a la pestaña Autoridades:

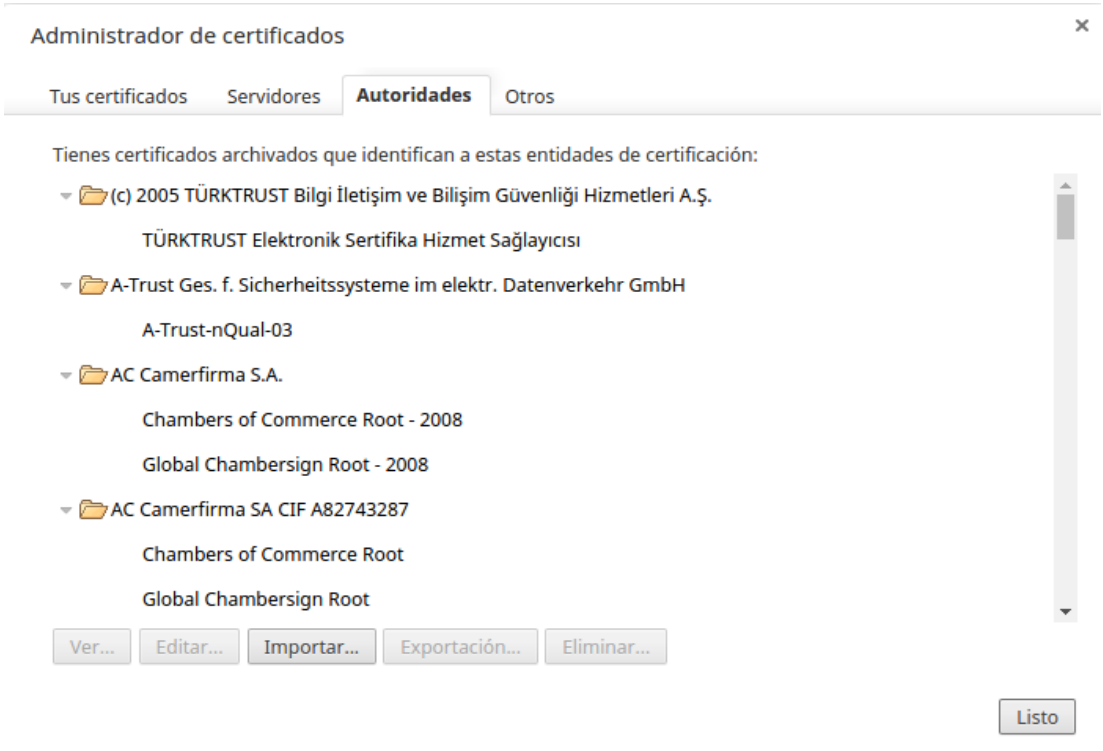


Figura 65: Autoridades de Confianza

Y hacemos clic en Importar y seleccionamos nuestro certificadora.

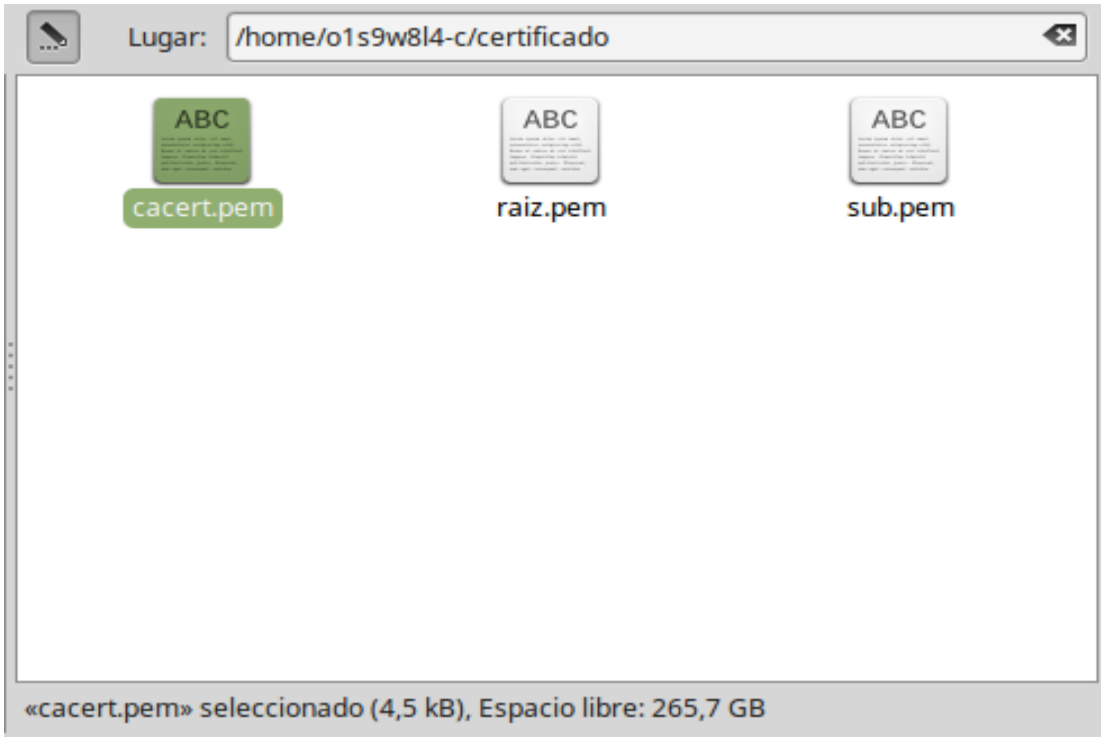


Figura 66: Parámetros de Importación del Certificado

Preguntará las opciones para confiar en este certificado

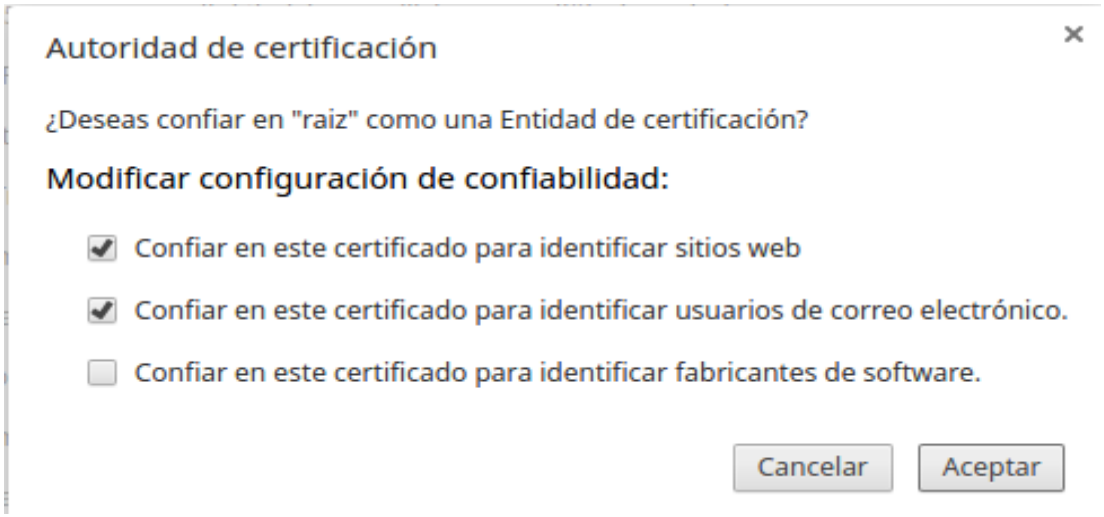


Figura 67: Archivo de certificado digital

Listo nuestro certificado está instalado en el navegador.

4.7 Implementación Del Servidor Web

Para verificar la validez de nuestras configuraciones procedemos a configurar un servidor web con la finalidad de generar un certificado y probar una página en el navegador.

Si no está instalado el servidor web procedemos a instalarlo con el siguiente comando:

```
# yum install httpd
```

Ahora procedemos a generar la clave privada

```
# openssl genrsa -out servapp.key 4096
```

```
[root@servapp ~]# openssl genrsa -out servapp.key 4096
Generating RSA private key, 4096 bit long modulus
.....................................................................................................+
+
.....++
e is 65537 (0x10001)
```

Y procedemos a generar una solicitud de certificado

```
# openssl req -new -key servapp.key -out servapp.csr
```

```
[root@servapp ~]# openssl req -new -key servapp.key -out servapp.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:EC
State or Province Name (full name) []:PICHINCHA
Locality Name (eg, city) [Default City]:QUITO
Organization Name (eg, company) [Default Company Ltd]:EJERCITO
Organizational Unit Name (eg, section) []:SIPER
Common Name (eg, your name or your server's hostname) []:servapp.localdomain
Email Address []:oswl_c@ecuasof.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Figura 68: Generando solicitud de certificado del servidor de aplicaciones

Procedemos a subir este requerimiento de certificado a la autoridad certificadora subordinada para que esta genere un nuevo certificado firmado.

```
# scp servapp.csr root@192.168.0.101:/root/sub_ca
```

En la autoridad certificadora procedemos a generar el certificado usando el siguiente código.

```
#  openssl ca -in servapp.csr -out servapp.pem -config openssl.cnf
```

```
[root@sub sub_ca]# openssl ca -in servapp.csr -out servapp.pem -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ./private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 256 (0x100)
        Validity
            Not Before: Sep 27 14:56:01 2016 GMT
            Not After : Sep 27 14:56:01 2017 GMT
        Subject:
            countryName               = EC
            stateOrProvinceName       = PICHINCHA
            organizationName          = EJERCITO
            organizationalUnitName    = SIPER
            commonName                = servapp.localdomain
            emailAddress              = oswl_c@ecuasof.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                61:1E:30:C1:54:08:DE:F9:90:D0:64:D5:10:DE:BB:69:AB:AA:BE:49
            X509v3 Authority Key Identifier:
                keyid:7D:E0:11:E7:89:AC:48:02:17:32:38:E7:3D:81:56:BA:07:D0:21:77

Certificate is to be certified until Sep 27 14:56:01 2017 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Figura 69: Cargando parámetros para la solitud de certificado digital

Y finalmente lo copiamos a nuestro servidor web.

```
# scp servapp.pem root@192.168.0.107:/root/
```

Listo ahora ya tenemos nuestro certificado creado y procedemos a instalarlo en nuestro servidor web.

4.8 Instalación De Los Certificados En El Servidor Web

Copiamos los archivos generados a la carpeta de configuración de apache

```
# cp servapp.key /etc/pki/tls/private/

# cp servapp.pem /etc/pki/tls/certs/
```

Ingresamos a la carpeta /etc/httpd/conf.d

```
# cd /etc/httpd/conf.d
```

Y abrimos el archivo ssl.conf

```
# nano  ssl.conf
```

Y buscamos las siguientes lineas para modificarlas

```
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost .key
```

Y las reemplazamos con

```
SSLCertificateFile /etc/pki/tls/certs/servapp.pem
SSLCertificateKeyFile /etc/pki/tls/private/servapp.key
```

Reiniciamos el servidor web

```
# service httpd restart
```

Ahora dentro de nuestra maquina vamos al archivo /etc/hosts y agregamos la ip y nombre del servidor web

```
$ sudo nano /etc/hosts
```

Y agregamos al principio esta linea

```
192.168.0.107   servapp servapp.localdomain
```

Quedándonos el archivo como se muestra a continuación

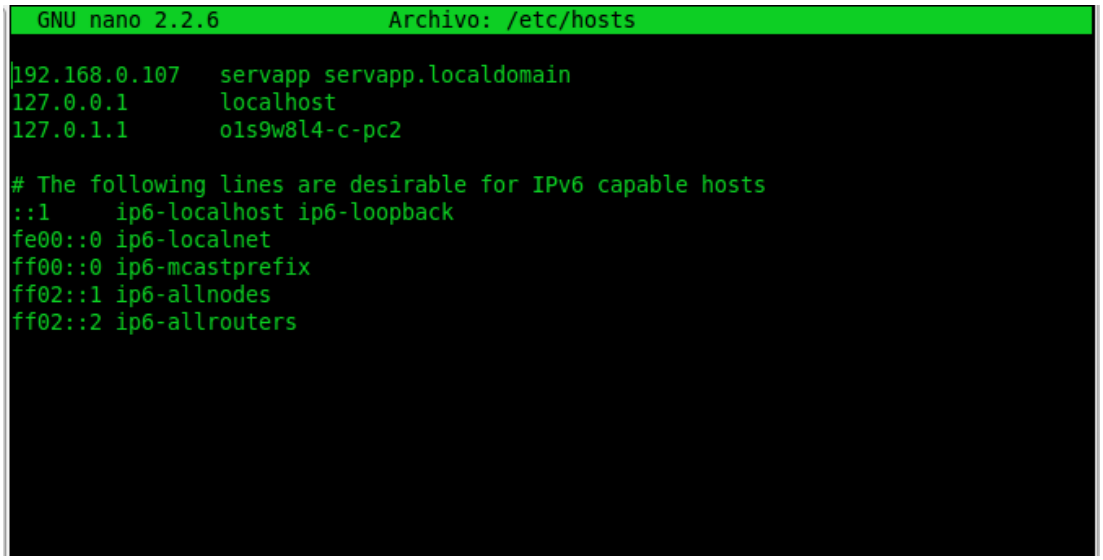


Figura 70: Configuración del archivo /etc/hosts

Ahora procedemos a abrir el navegador y colocamos en la barra de dirección

https://servapp.localdomain/

Como podemos ver el candadito se puso en verde. Ahora hacemos clic en el candado para ver qué información se despliega.

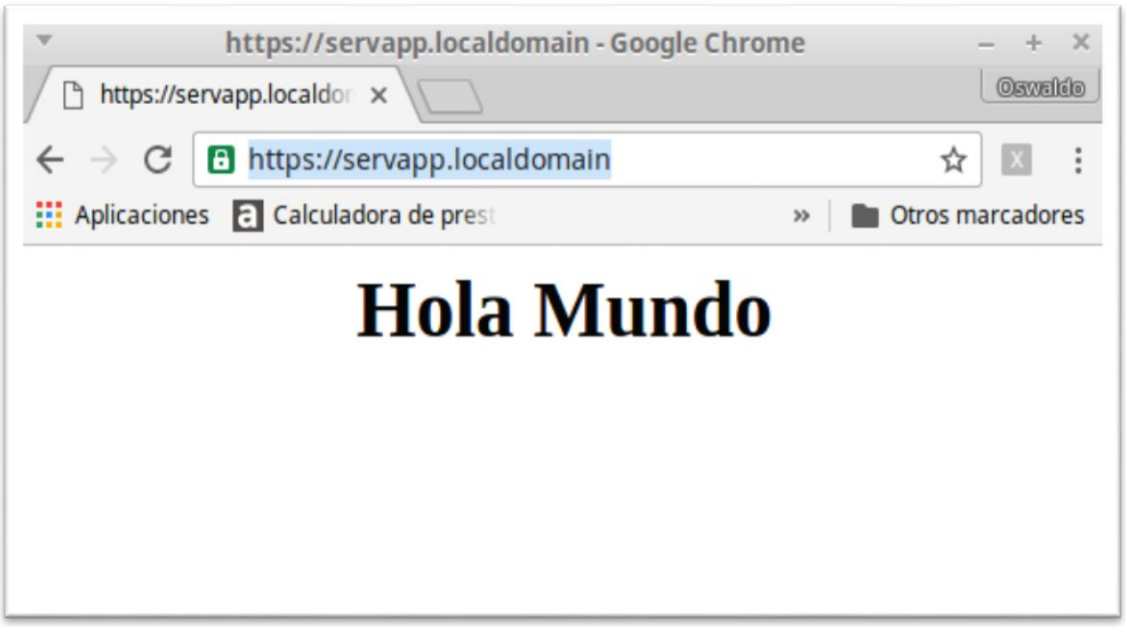


Figura 71: Prueba del servidor web con el certificado instalado

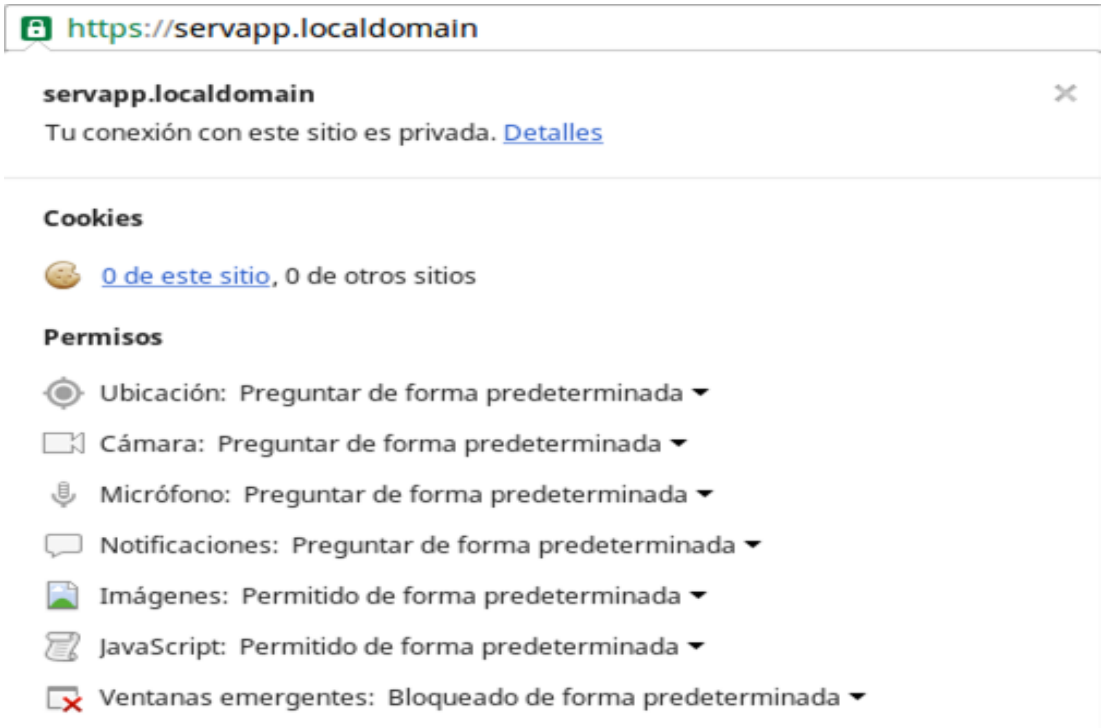


Figura 72: Detalles de conexión

Hacemos clic en Detalles

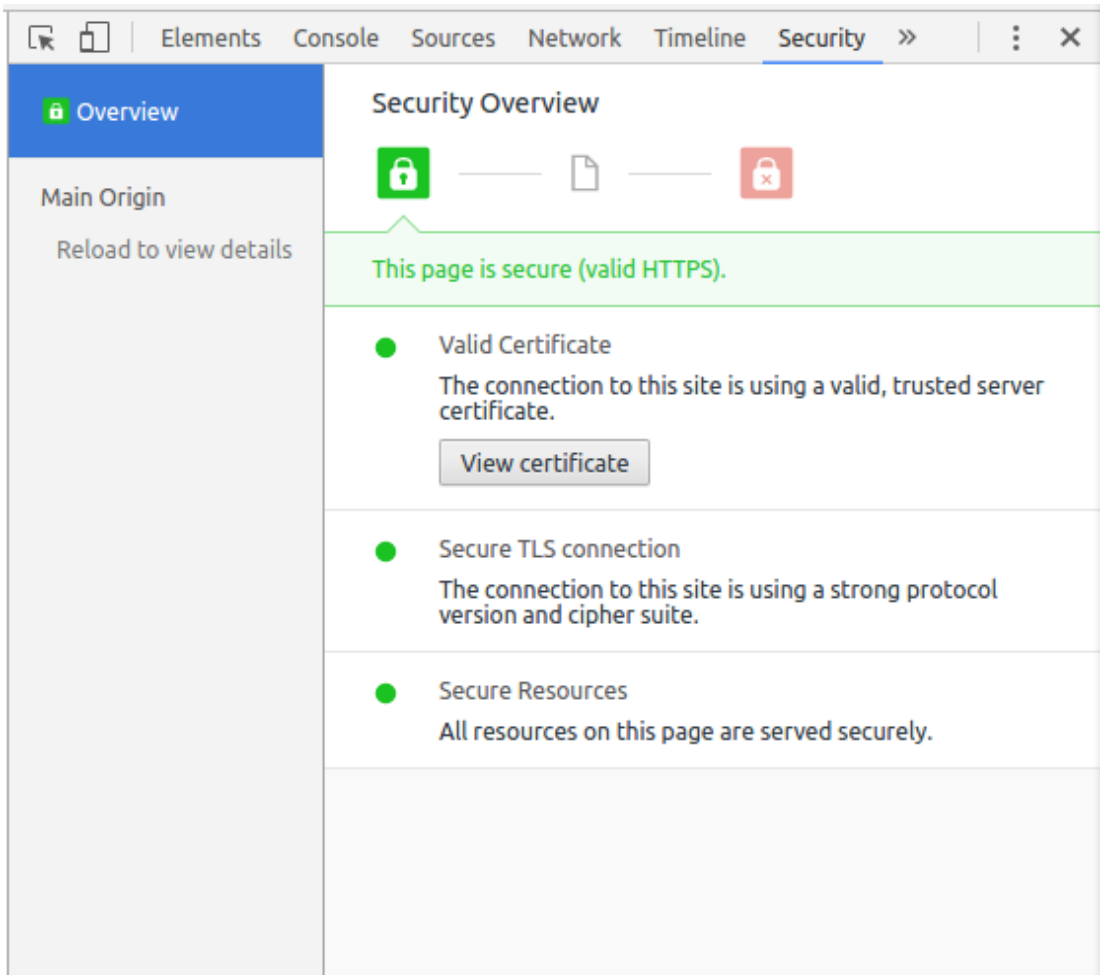


Figura 73: Verificando autenticidad del certificado

Damos clic en View Certificate

Figura 74: Detalles del certificado digital

Como podemos ver están los datos de verificación, emitido a, proporcionado por, etc.

Y listo nuestras autoridades certificadoras están funcionando y listas para continuar con el resto de configuraciones y desarrollo de aplicaciones.

4.9 Implementación pkiEjercitoEC

Una vez que seleccionamos a OpenSSL para implementar la infraestructura de clave pública, se puede determinar que es necesario la creación de una herramienta propia para la gestión de la PKI, para ello vamos a trabajar con los estándares establecidos en la Dirección de Comunicaciones e Informática del Ejército, estableciendo la siguiente arquitectura de proyectos.

Tabla 7:

Arquitectura de Proyectos PKI

Ord.	Proyecto	Descripción
01	pkiEjercitoEC	Contiene los servicios web necesarios para la gestión de la PKI, y será ubicado en el servidor destinado para la Autoridad Certificadora Subordinada, y tendrá comunicación solo con el servidor de aplicaciones.
02	PkiEjercitoEC_CA	Consume los servicios proporcionados por el proyecto pkiEjercitoEC, y está destinado para la generación de certificados digitales. Este proyecto será ubicado en el servidor de aplicaciones de la fuerza, y tendrá un puerto diferente al de las demás aplicaciones.
03	pkiEjercitoEC_Pub	Consume los servicios proporcionados por el proyecto pkiEjercitoEC, y está destinado para firmar electrónicamente los documentos tramitados en la Fuerza Terrestre. Este proyecto será ubicado en el servidor de aplicaciones de la fuerza, y tendrá un puerto diferente al de las demás aplicaciones. Además de que debe permitir la validación de los documentos firmados anteriormente.

04	pkiEjercitoEC_RA	Consume los servicios proporcionados por el proyecto pkiEjercitoEC, y está destinado para el registro de los nuevos usuarios. Este proyecto será ubicado en el servidor de aplicaciones de la fuerza, y tendrá un puerto diferente al de las demás aplicaciones.

4.9.1 Plataforma informática

Los proyectos antes mencionados se implementarán usando una plataforma tipo web, ya que dentro de los estándares de la Dirección de Comunicaciones y Sistemas del Ejército se debe realizar el desarrollo de aplicaciones en Lenguaje JAVA y usar el servidor de aplicaciones Glassfish.

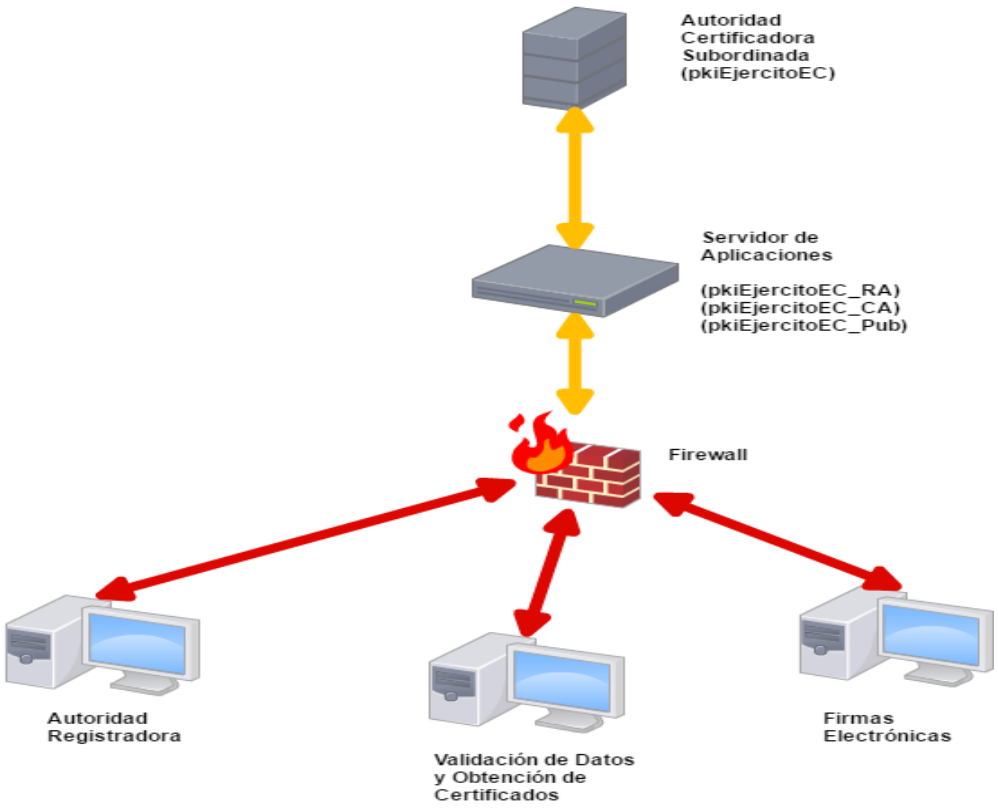


Figura 75: Modelo de plataforma web

Una vez que se crearon los proyectos con la plataforma establecida, nos va a quedar un árbol como el que muestra el Gráfico Nro. 76.

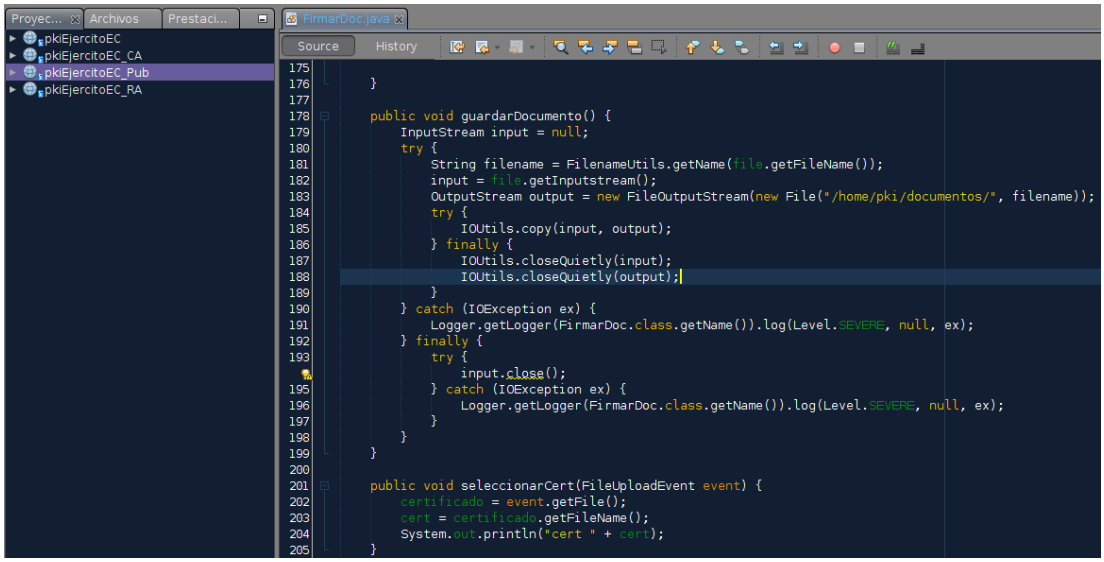


Figura 76: Estructura del proyecto pkiEjercitoEC

Una vez analizada la estructura del proyecto creados para gestionar la PKI de la Fuerza Terrestre, procedemos a analizar las clases internas de cada proyecto.

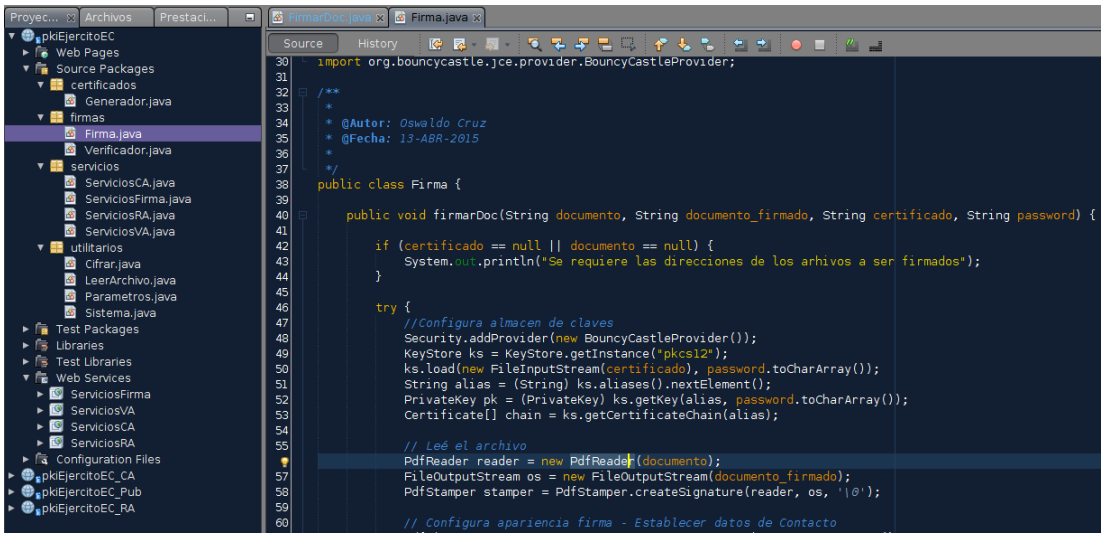


Figura 77: Clase destinada a la firma electrónica

4.9.2 pkiEjercitoEC

El proyecto pkiEjercitoEC, está basado en el protocolo SOAP, para proporcionar los siguientes servicios web: ServiciosRA, ServiciosCA, ServiciosVA, ServiciosFirma. Además, el proyecto contiene clases importantes para el proceso de generación de los certificados digitales.

En el paquete utilitarios tenemos las siguientes clases que apoyan a los servicios web proporcionados por el proyecto, como son la clase Cifrar.java, LeerArchivo.java, Parametros.java y Sistema.java

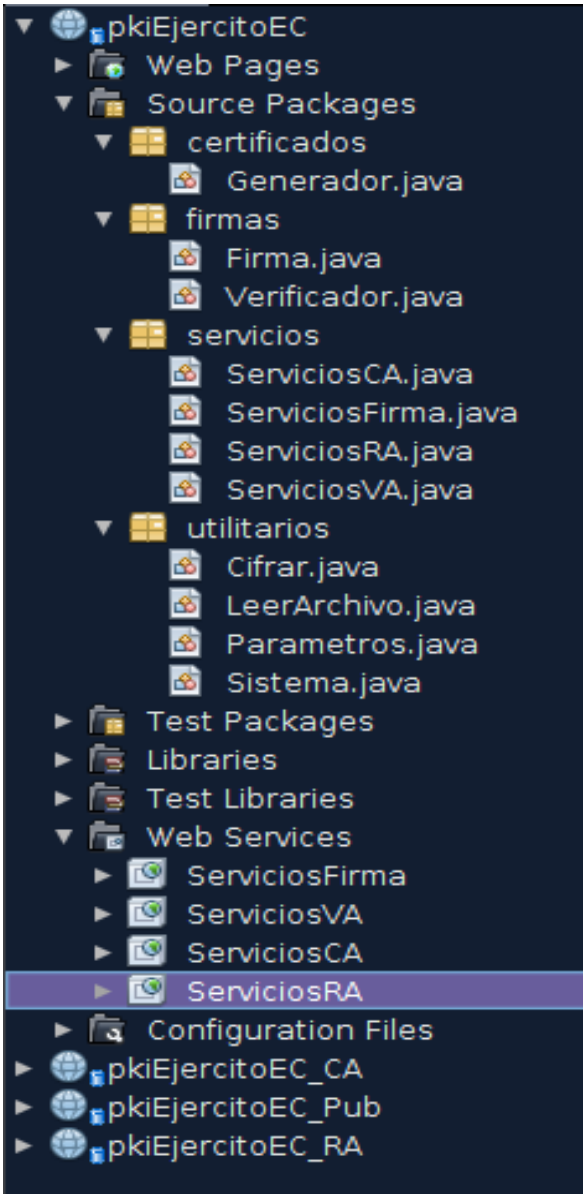


Figura 78: Servicios creados dentro del proyecto pkiEjercitoEC

Los servicios creados son utilizados para realizar los siguientes procesos, firmar documento, verificar documento, generar certificado digital, generar firma digital, realizar el registro de usuario.

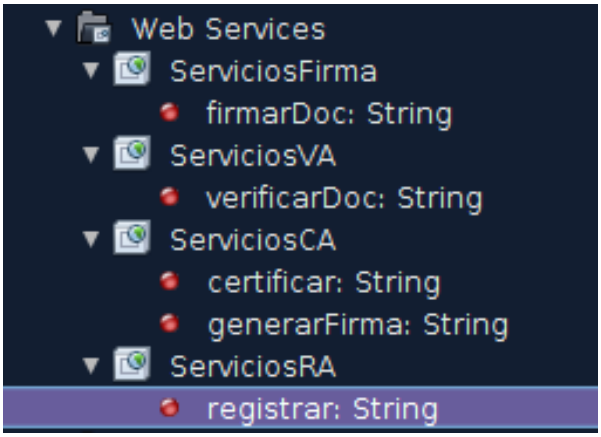


Figura 79: Método registrar del servicio RA

Cada servicio web es programado mediante una clase en JAVA, y para que funcionen como tal, se debe poner la anotación @WebService sobre el nombre de la clase a programar.

Por otro lado, se crean los métodos que serán los que hagan la ejecución de los procesos para ello se procede como normalmente se hace con un método normal, sin embargo, se debe agregar la anotación @WebMethod sobre la declaración del nombre del método, y para los parámetros se agrega la anotación @WebParam y el nombre del parámetro.

En el parámetro subj (Sujeto), se precarga el país, la provincia, el cantón y la institución. Y el resto de parámetros son enviados a través de las aplicaciones que consumen este servicio.

Para probar el método programado simplemente seleccionamos el servicio web y damos clic en el botón TEST.

```java
/**
 *
 * @Autor Oswaldo Cruz
 * @Fecha 20-DIC-2016
 *
 */
@WebService(serviceName = "ServiciosRA")
public class ServiciosRA {

    Generador generadorCert = new Generador();

    /**
     * @param cn
     * @return
     */
    @WebMethod(operationName = "registrar")
    public String registrar(@WebParam(name = "cn") String cn) {
        //Genero clave privada
        generadorCert.generarClavePrivada(cn);

        //Genero solicitud de certificado
        String subj = "/C=EC/ST=PICHINCHA/L=QUITO/O=EJERCITO/CN=" + cn;
        generadorCert.generarSolicitudCert(cn, subj);

        return "Ok";

    }
}
```

Figura 80: Detalle del servicio RA

Figura 81: Prueba del servicio RA

```java
/**
 *
 * @Autor: Oswaldo Cruz
 * @Fecha: 26-DIC-2016
 */
@WebService(serviceName = "ServiciosCA")
public class ServiciosCA {

    Generador generadorCert = new Generador();

    /**
     * This is a sample web service operation
     *
     * @param cn
     * @return
     */
    @WebMethod(operationName = "certificar")
    public String certificar(@WebParam(name = "cn") String cn) {
        //Genero certificado firmado
        generadorCert.generarCerttificado(cn);

        return "Ok";
    }

    @WebMethod(operationName = "generarFirma")
    public String generarFirma(@WebParam(name = "cn") String cn, @WebParam(name = "pwd") String pwd) {
        //Genero firma
        generadorCert.generarPFX(cn, pwd);

        return "Ok";
    }

}
```

Figura 82: Detalle del servicio CA

Figura 83: Prueba del servicio CA

4.9.3 Clase Cifrar

La clase cifrar está basada en la codificación de los caracteres en base 64, y justamente utiliza la clase de JAVA llamada java.util.Base64. Este cifrado es simétrico, y se utiliza para cifrar la comunicación de los servicios web con los proyectos consumidores.

A diferencia del cifrado usado para generar los certificados digitales que son asimétricos, es decir no se puede descifrar con la misma clave que se cifraron los datos, se utiliza una clave para cifrar y otra para descifrar.

Depende el uso una puede ser pública y la otra privada, en otros casos se puede usar a la inversa. En el caso de la firma electrónica la clave privada es con la que se cifra y la pública es con la que se descifra. Al contrario de los certificados SSL que, al dar clic en enviar el formulario de una página web, el contenido es cifrado con la clave pública y es descifrado con la privada.

En cualquiera que sea el caso es muy difícil que se puede descifrar por fuerza bruta o usando otros métodos no legales. Por ello es un método aceptado a nivel mundial.

```java
package utilitarios;

import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class Cifrar {

    public static String ejecutarCifrado(String a) {
        Base64.Encoder encoder = Base64.getEncoder();
        String b = encoder.encodeToString(a.getBytes(StandardCharsets.UTF_8));
        return b;
    }

    public static String ejecutarDesifrado(String a) {
        Base64.Decoder decoder = Base64.getDecoder();
        byte[] decodedByteArray = decoder.decode(a);

        String b = new String(decodedByteArray);
        return b;
    }

}
```

Figura 84: Clase Cifrar

4.9.4 Servicio web para firma electrónica

Para implementar el servicio web de firma electrónica Gráfico Nro.85, hacemos la llamada a la clase firma Grafico Nro. 87 la cual a su vez utiliza clases internas de JAVA como Security y KeyStore.

Para que funcione adecuadamente este servicio se le debe asignar parámetros obligatorios como son:

Tabla 8:

Parámetros usados para firma electrónica

Ord.	Parámetro	Detalle
01	Documento	Nombre del documento a ser firmado
02	Documento_firmado	Nombre del documento que el sistema entregara ya firmado.
03	Certificado	Nombre del certificado de firma electrónica que se va a utilizar.
04	Password	Contraseña que se utiliza para poder descifrar el certificado de firma electrónica.

```
/**
 *
 * @Autor: Oswaldo Cruz
 * @Fecha: 26-DIC-2016
 */
@WebService(serviceName = "ServiciosFirma")
public class ServiciosFirma {

    /**
     * This is a sample web service operation
     *
     * @param documento
     * @param documento_firmado
     * @param certificado
     * @param password
     * @return
     */
    @WebMethod(operationName = "firmarDoc")
    public String firmarDoc(@WebParam(name = "documento") String documento,
            @WebParam(name = "documento_firmado") String documento_firmado,
            @WebParam(name = "certificado") String certificado,
            @WebParam(name = "password") String password) {

        Firma firma = new Firma();
        firma.firmarDoc(Parametros.directorio_documentos + documento,
                Parametros.directorio_firmados + documento_firmado,
                Parametros.directorio_firma + certificado, password);
        return "Ok";
    }
}
```

Figura 85: Servicios Firma

Figura 86: Prueba del servicio de Firma

```java
public class Firma {

    public void firmarDoc(String documento, String documento_firmado, String certificado, String password) {

        if (certificado == null || documento == null) {
            System.out.println("Se requiere las direcciones de los arhivos a ser firmados");
        }

        try {
            //Configura almacen de claves
            Security.addProvider(new BouncyCastleProvider());
            KeyStore ks = KeyStore.getInstance("pkcs12");
            ks.load(new FileInputStream(certificado), password.toCharArray());
            String alias = (String) ks.aliases().nextElement();
            PrivateKey pk = (PrivateKey) ks.getKey(alias, password.toCharArray());
            Certificate[] chain = ks.getCertificateChain(alias);

            // Leé el archivo
            PdfReader reader = new PdfReader(documento);
            FileOutputStream os = new FileOutputStream(documento_firmado);
            PdfStamper stamper = PdfStamper.createSignature(reader, os, '\0');

            // Configura apariencia firma - Establecer datos de Contacto
            PdfSignatureAppearance appearance = stamper.getSignatureAppearance();
            appearance.setReason("Ejército Ecuatoriano");
            appearance.setLocation("Quito-Ecuador");

            //Configura apariencia firma - Ubicar firma Bottom right
            float width = 100, height = 50;
            Rectangle cropBox = reader.getCropBox(1);
            Rectangle rectangle = new Rectangle(cropBox.getRight(width), cropBox.getBottom(), cropBox.getRight(), cropBox.getBottom(height));
            appearance.setVisibleSignature(rectangle, 1, "Firma");

            //Firma el archivo
            ExternalSignature es = new PrivateKeySignature(pk, "SHA-256", "BC");
            ExternalDigest digest = new BouncyCastleDigest();
            MakeSignature.signDetached(appearance, digest, es, chain, null, null, null, 0, MakeSignature.CryptoStandard.CMS);

        } catch (IOException | DocumentException | GeneralSecurityException e) {
            System.out.println(e);
        }

    }

}
```

Figura 87: Clase para firmar el documento

4.9.5 Proyecto pkiEjercitoEC_RA

El proyecto pkiEjercitoEC_RA, tiene como finalidad facilitar el trabajo a la Autoridad Registradora o RA, ya que permite realizar el proceso de una forma más sencilla, utilizando ventanas amigables para el usuario y que a su vez le brinda mayor seguridad ya que este proyecto se desplegará solamente a través de un puerto determinado y solo será habilitada una maquina especifica.

El árbol del proyecto se muestro en el Gráfico Nro. 88, y se explica a mayor detalle en la Tabla Nro. 4.

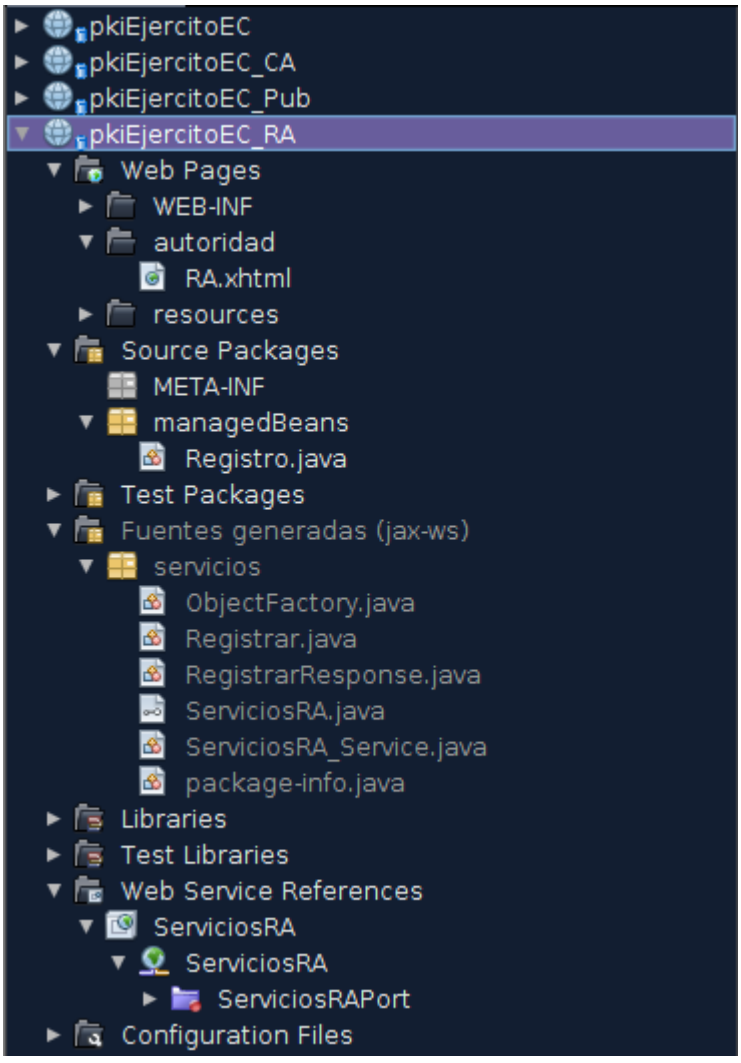


Figura 88: Árbol del proyecto para la autoridad registradora

Tabla 9:

Ficheros principales en el proyecto pkiEjercitoEC_RA

Ord.	Fichero	Descripción
01	RA.xhtml	Interface principal que muestra la pantalla donde el usuario ingresa los datos de registro del personal que solicita un certificado digital.
02	Registro.java	Clase principal que gestiona el registro de datos y la generación de la petición de certificado digital.
03	ServiciosRA	Cliente del servicio web generado en el proyecto pkiEjercitoEC con el propósito de ser consumido para que la autoridad de registro pueda generar las peticiones de certificado.
04	ServiciosRAPort	Puerto a través del cual se puede hacer las llamadas a los procedimientos que se encuentran inmersos en el servicio web.

Proyecto pkiEjercitoEC_CA

De igual forma que el proyecto pkiEjercitoEC_RA, el objetivo es facilitar al usuario la validación de datos y la generación de nuevos certificados de forma automática, evitando el tramite manual ya que los datos del personal militar se encuentran ya registrados en la base de datos de la Fuerza.

El proyecto pkiEjercitoEC_CA, tiene una estructura interna como se muestra en el Gráfico Nro. 89, y se explica a mayor detalle en la Tabla Nro. 5.

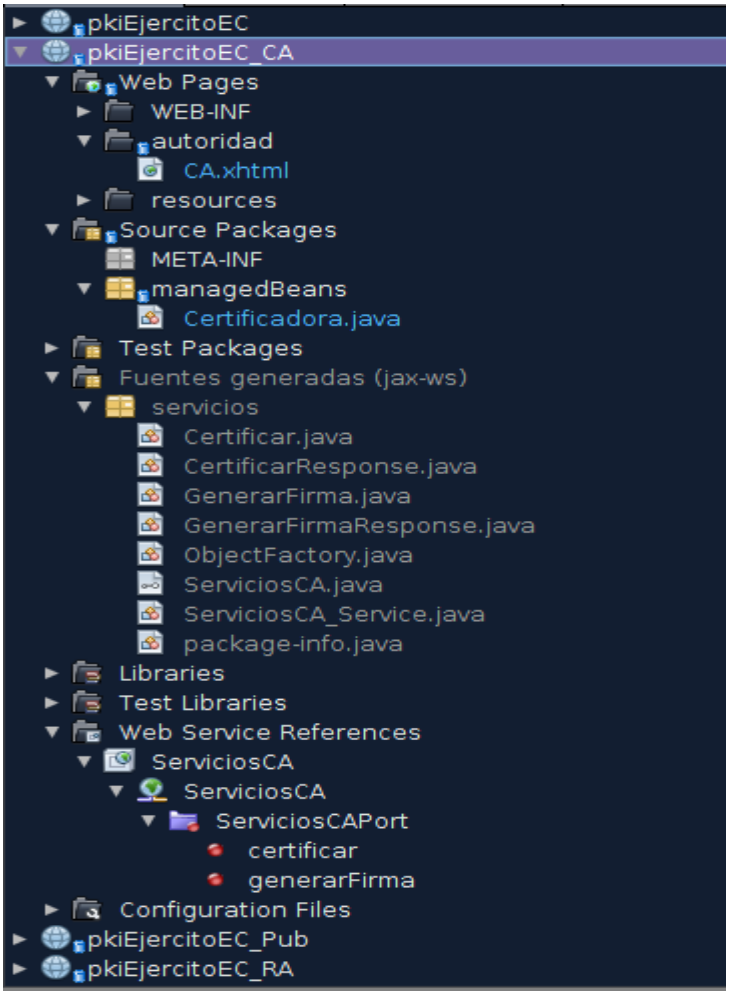


Figura 89: Árbol de proyecto para autoridad certificadora

Tabla 10:

Ficheros principales en el proyecto pkiEjercitoEC_CA

Ord.	Fichero	Descripción
01	CA.xhtml	Interface principal que muestra la pantalla donde el usuario valida los datos de registro y genera el certificado digital y la firma electrónica.
02	Certificadora.java	Clase principal que gestiona la validación de datos, la generación del certificado digital y la firma electrónica.
03	ServiciosCA	Cliente del servicio web generado en el proyecto pkiEjercitoEC con el propósito de ser consumido para que la autoridad certificadora pueda generar los certificados digitales.
04	ServiciosCAPort	Puerto a través del cual se puede hacer las llamadas a los procedimientos que se encuentran inmersos en el servicio web.

4.9.6 Proyecto pkiEjercitoEC_Pub

Este proyecto está orientado al manejo público, será publicado como parte del SIFTE, y tendrán acceso todos aquellos que requieran firmar y verificar documentos digitales.

Es importante aclarar que mientras no se realice el tramite con la ARCOTEL, y se pague la garantía correpondiente, las firmas que ejecute este sistema solo tendrán valides dentro de la Fuerza Terrestre.

La estructura del proyecto se muestra en el Gráfico Nro. 90, y se explica de mejor manera en la Tabla Nro. 6.

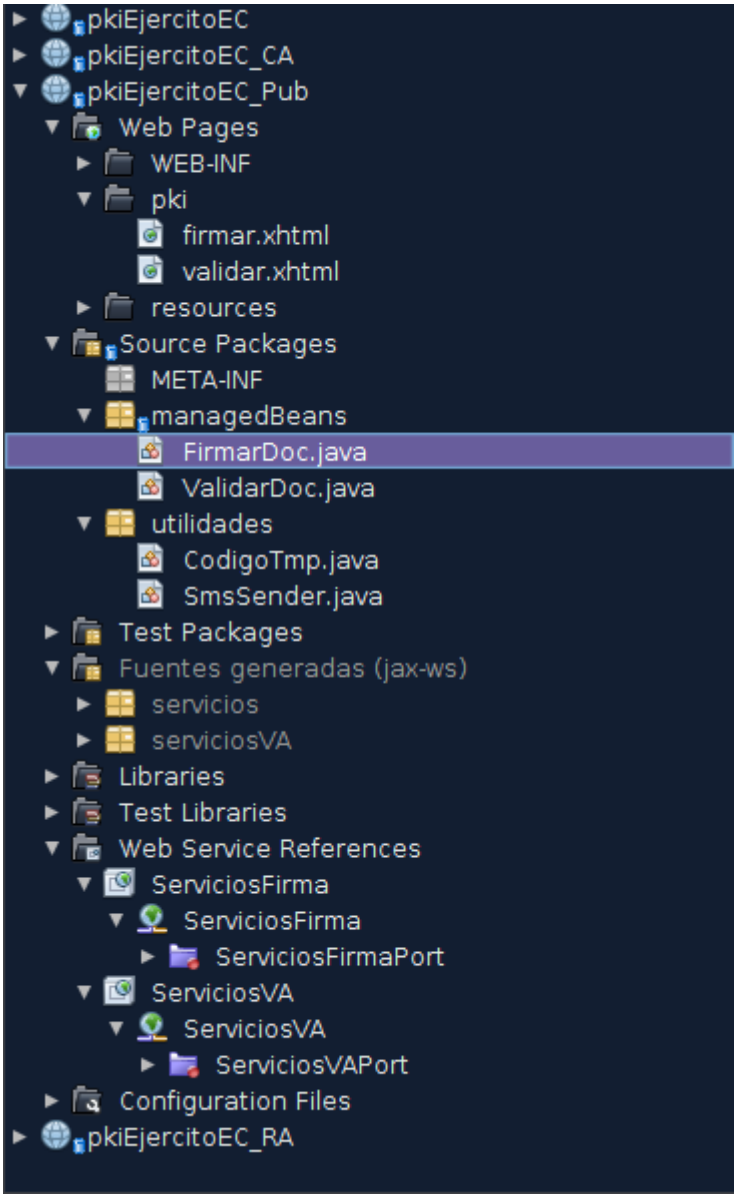


Figura 90: Árbol de proyecto para la firma electrónica

Tabla 11:

Ficheros principales en el proyecto pkiEjercitoEC_Pub

Ord.	Fichero	Descripción
01	firmarDoc.xhtml	Pantalla donde el usuario carga el archivo a ser firmado, y esta a su vez realiza una validación mediante un código temporal enviado al celular y una contraseña ya ingresada anteriormente.
02	FirmarDoc.java	Clase principal que gestiona las firmas electrónicas y los certificados digitales de cada firmante.
03	ServiciosFirma	Servicio web que proporciona las funcionalidades para firmar los documentos con los certificados digitales anteriormente generados.
04	ServiciosFirmaPort	Puerto a través del cual se puede hacer las llamadas a los procedimientos que se encuentran inmersos en el ser servicio web.

4.10 Uso del Sistema

4.10.1 Ventana de acceso

La ventana de acceso es la que se utiliza normalmente en el SIFTE, con la diferencia que el momento que realiza la selección en el menú el sistema se re-direccionará a un nuevo puerto.

Figura 91: Ventana de acceso al SIFTE

El usuario y el password serán los mismos que usamos habitualmente para los otros aplicativos

Figura 92: Login del SIFTE

4.10.2 Autoridad de Registro

El menú se mostrará como en el Gráfico Nro. 93, sin embargo, se puede asignar el perfil de acceso según sea el requerimiento.

La carga del menú se lo realiza mediante una consulta a la base de datos, pero es necesaria la modificación del menú para que pueda realizar el cambio de puerto en el caso que se trate de la Autoridad de Registro.

El proyecto que despliega la ventana de la Autoridad de Registro debe ser cargado en un servidor virtual a fin de poderle cambiar el puerto de escucha. Y mediante el firewall se puede dar acceso a una computadora especifica. En este caso, se dará acceso a una por división.

Si es necesario se debe dar acceso mediante una VPN, para dar mayor seguridad a la comunicación.

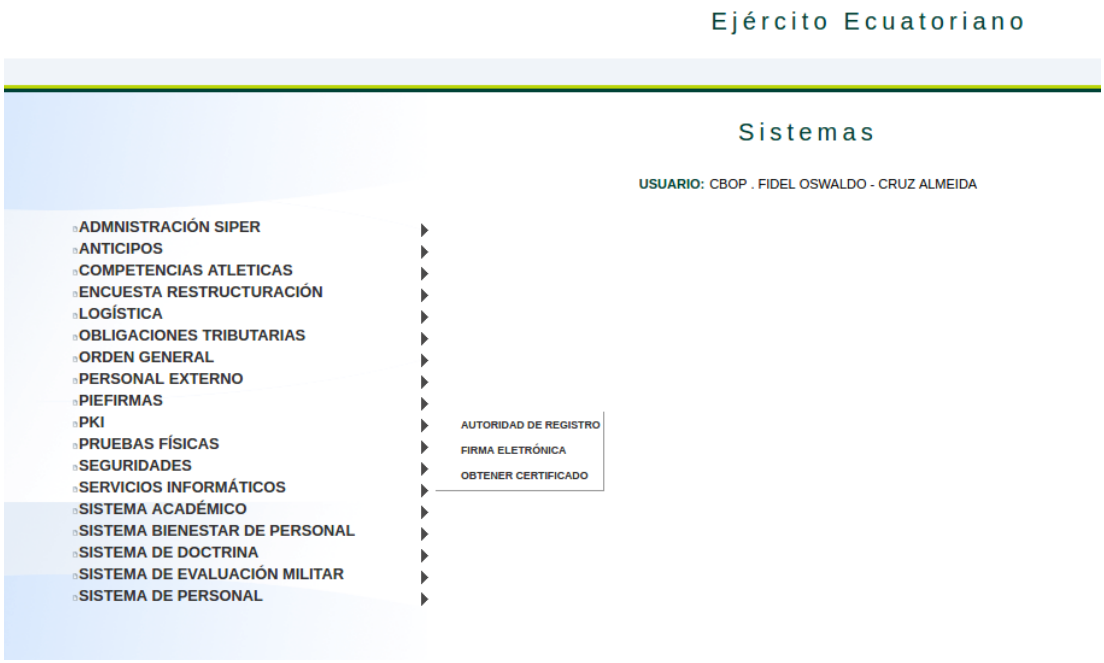


Figura 93: Menú del SIFTE para Autoridad de Registro

La ventana de registro se mostrará como en el Gráfico Nro. 94 y deberá ser llenado cuidadosamente en vista que es necesario una validación de documentos, como son cédula y tarjeta militar. Además, se requerirá llenar un formulario donde conste el requerimiento del certificado y las firmas que autorizan la emisión del mismo.

Figura 94: Ventana de registro

Cada registro genera una petición de certificado como lo muestra el Gráfico Nro. 95, y se guardará en la carpeta request dentro del servidor que proporciona los servicios web de la autoridad certificadora subordinada.

Y no se podrá tener acceso por ningún motivo a través de protocolos como FTP o SSH, ni cualquier otro ya que solo podrán acceder a estos archivos a través de los servicios web anteriormente mencionados.

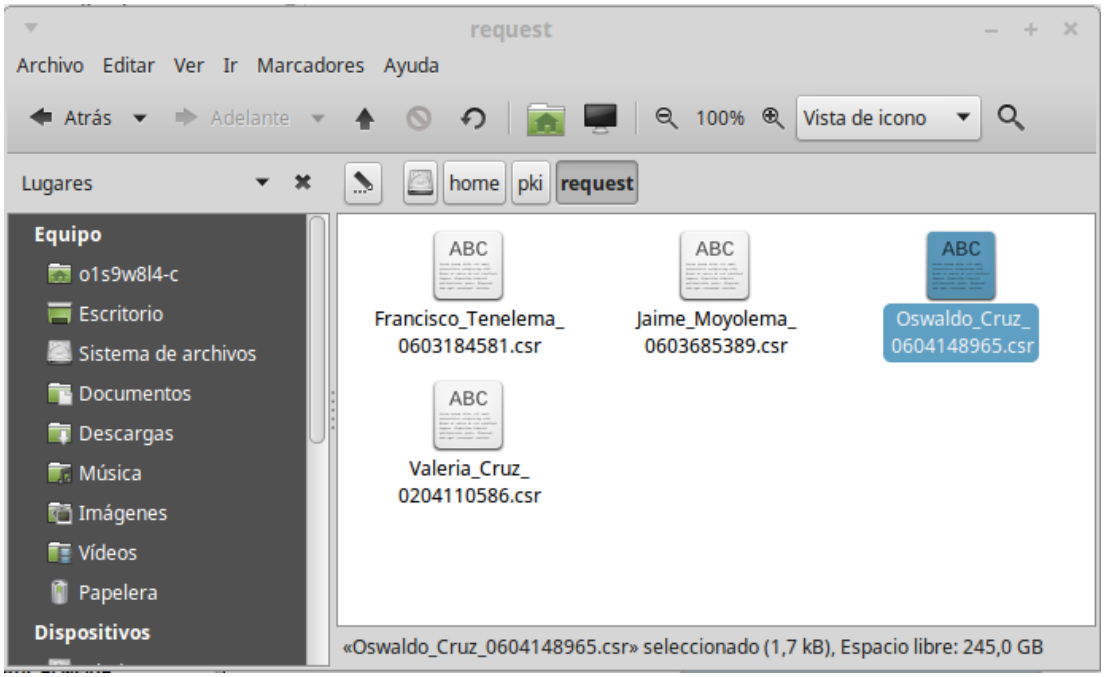


Figura 95: Carpeta request

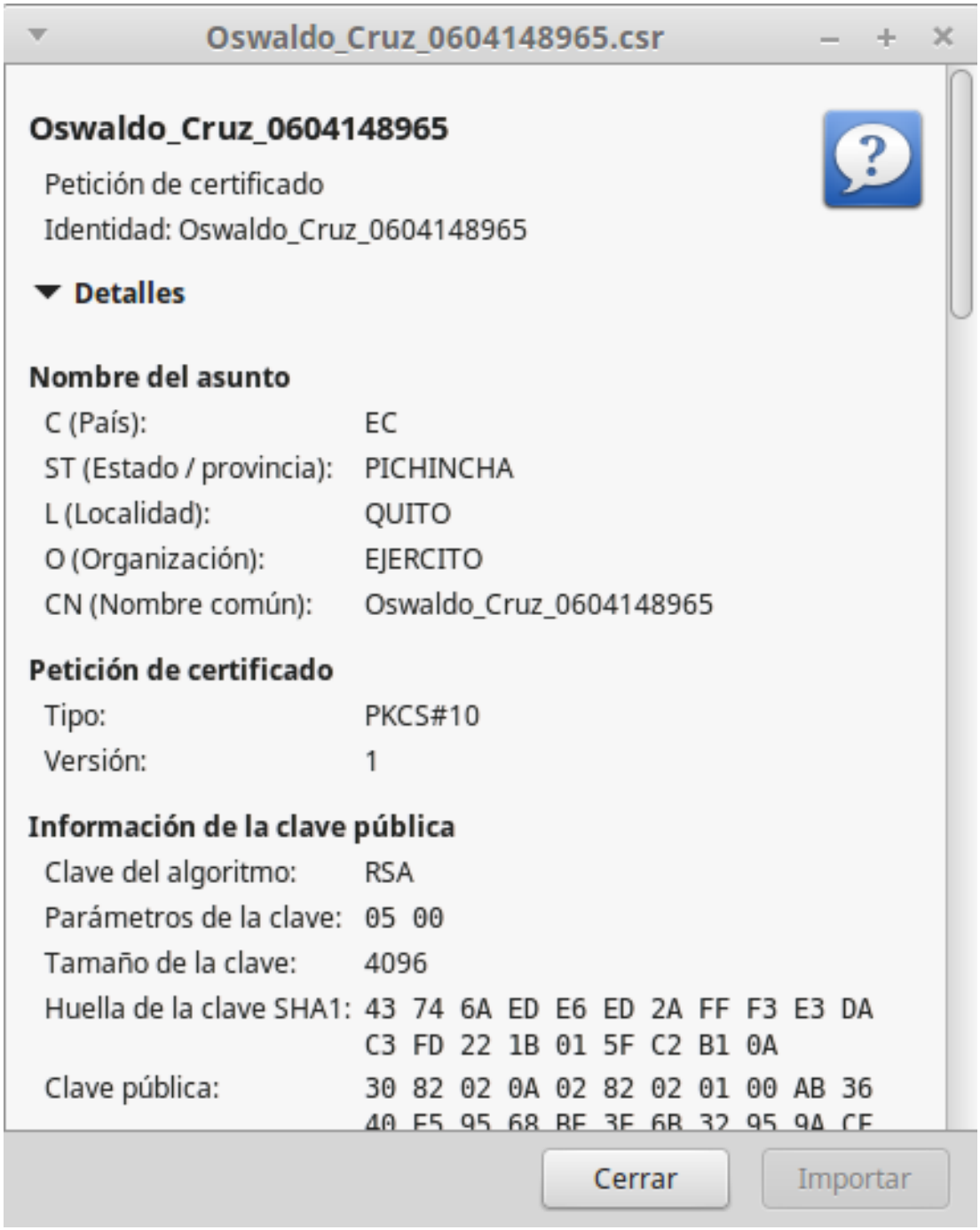


Figura 96: Archivo de petición de certificado

Por otro lado, tendremos el archivo de clave privada, la cual se guardará en la carpeta prívate. Como lo muestra el Gráfico Nro. 97, y el detalle de fichero en el Gráfico Nro. 98, el cual está cifrado a nivel 4096 bits.

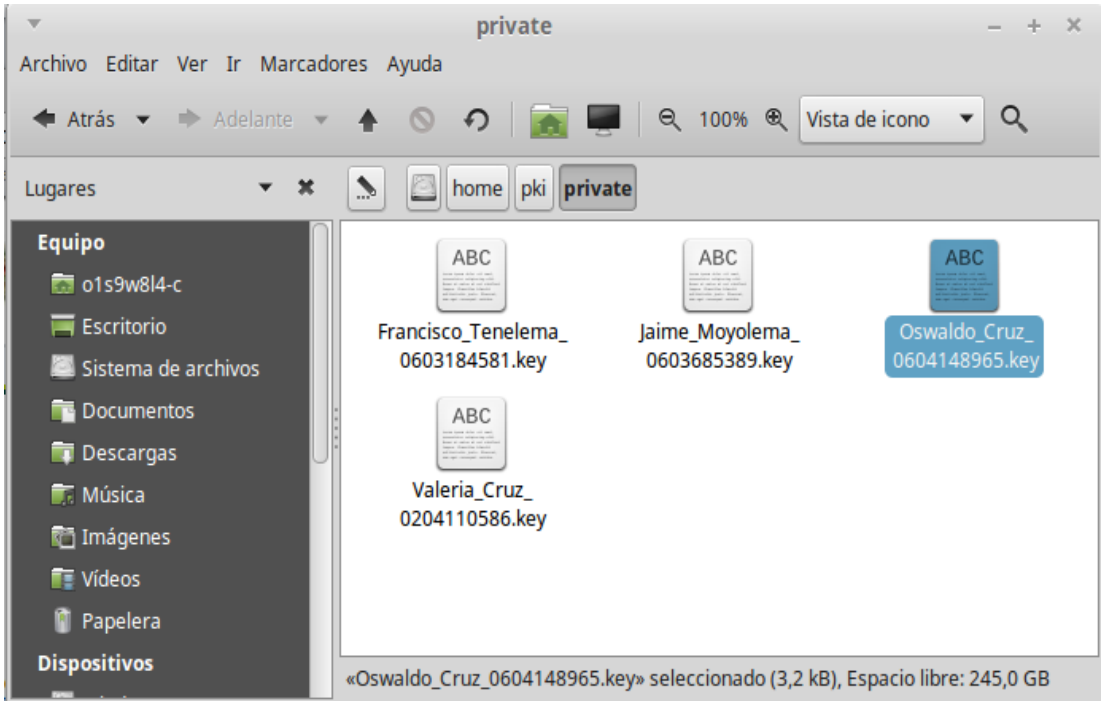


Figura 97:Carpeta de claves privadas

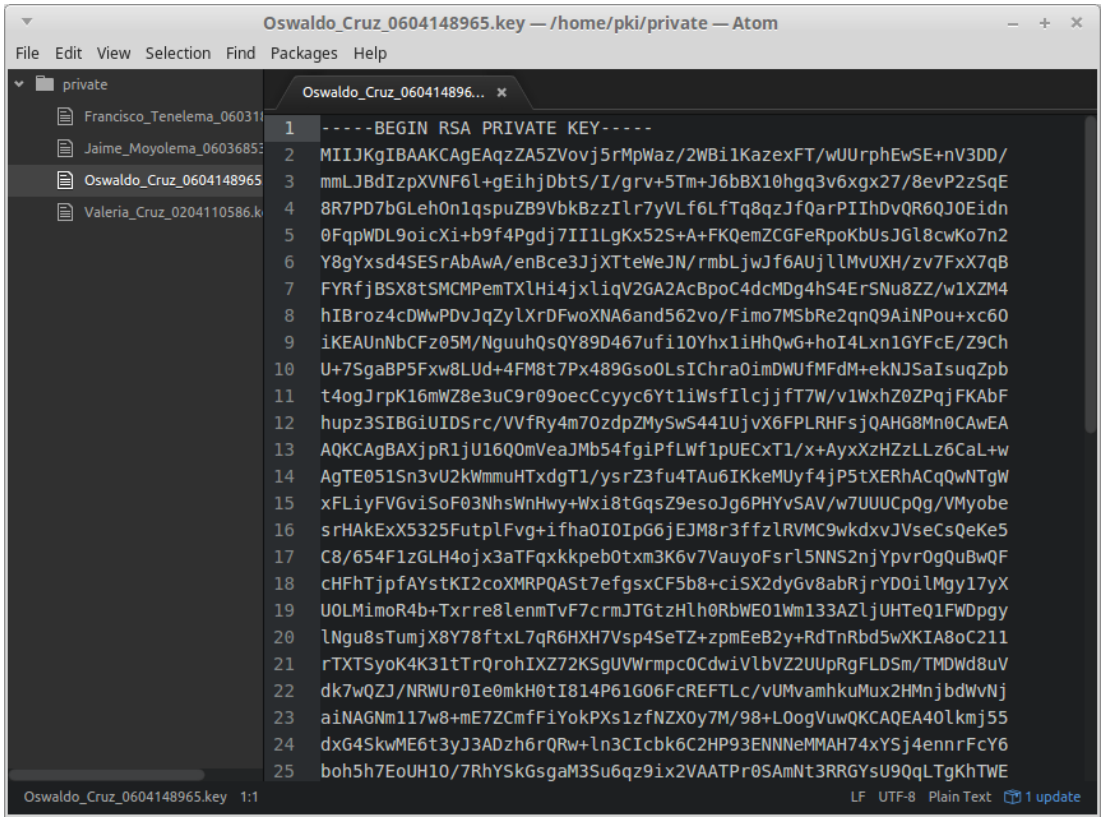


Figura 98: Archivo de clave privada

4.10.3 Obtener Certificado

Para obtener el certificado se debe ingresar nuevamente los datos como cédula, nombre y apellido, y dar clic en el botón Validar y Generar el Certificado como lo muestra el Gráfico Nro. 99, de este modo se genera el certificado digital como lo muestra el Gráfico Nro. 101 y el detalle se lo muestra en el Gráfico Nro. 102.

Posteriormente en el Gráfico Nro. 100, se muestra la creación del password y la generación del certificado de Firma Electrónica, como lo muestra el Gráfico 103.

Figura 99: Formulario de Validación y Generación del Certificado

Figura 100: Formulario con datos

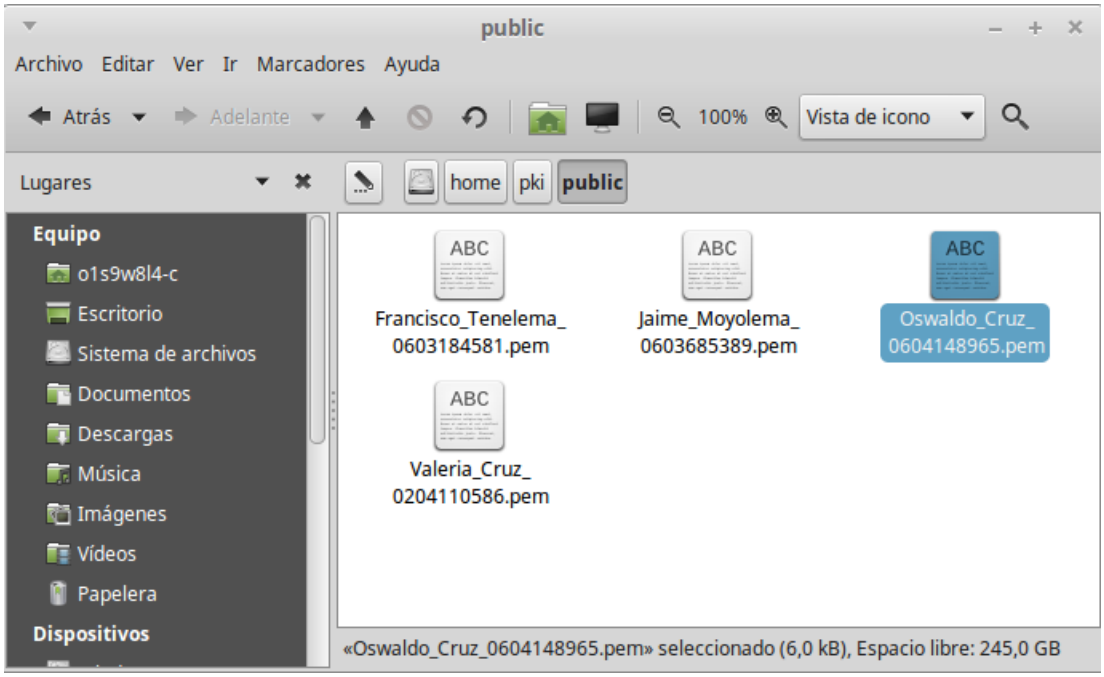


Figura 101: Carpeta de Certificados

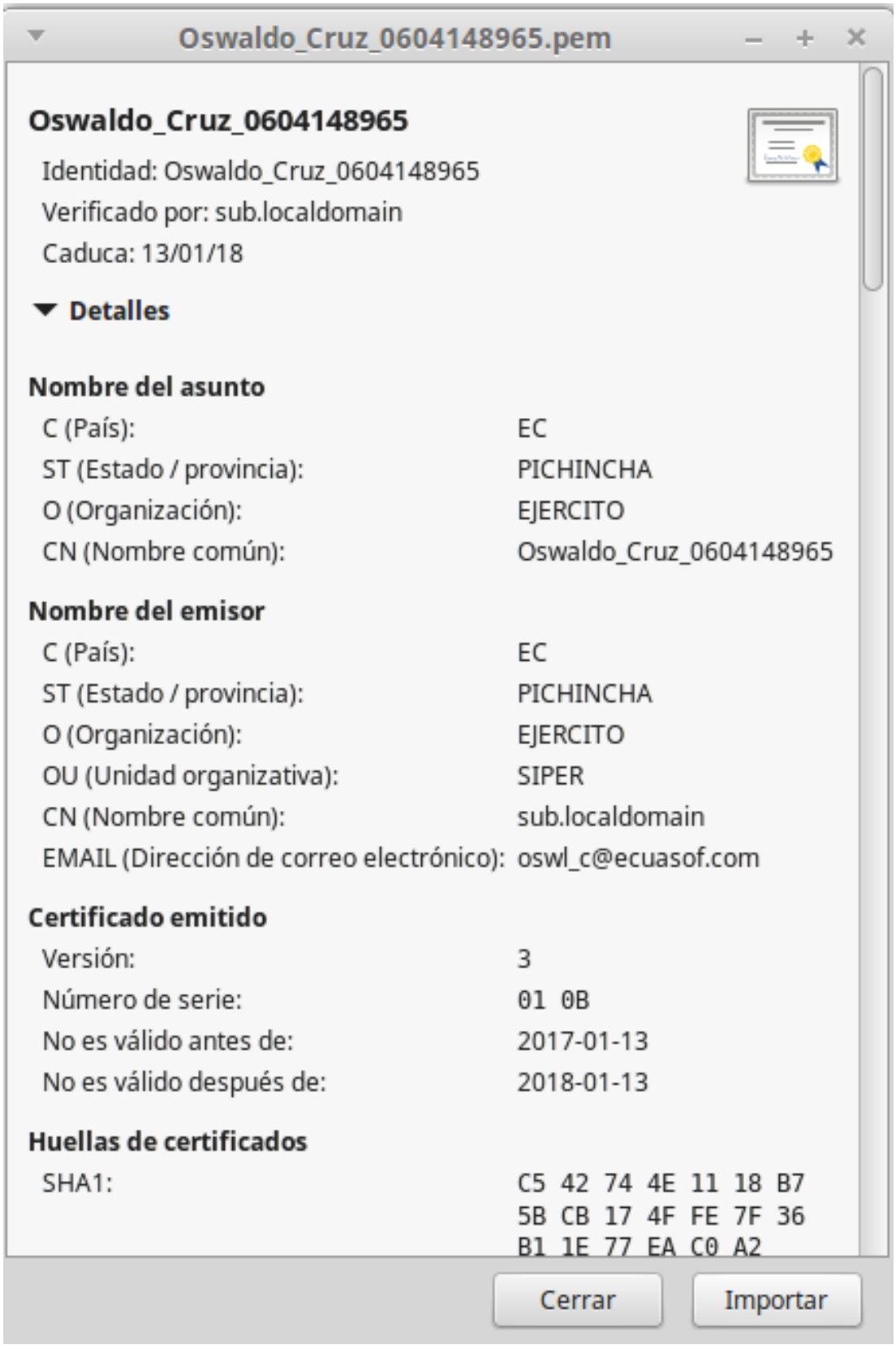


Figura 102: Certificado Digital

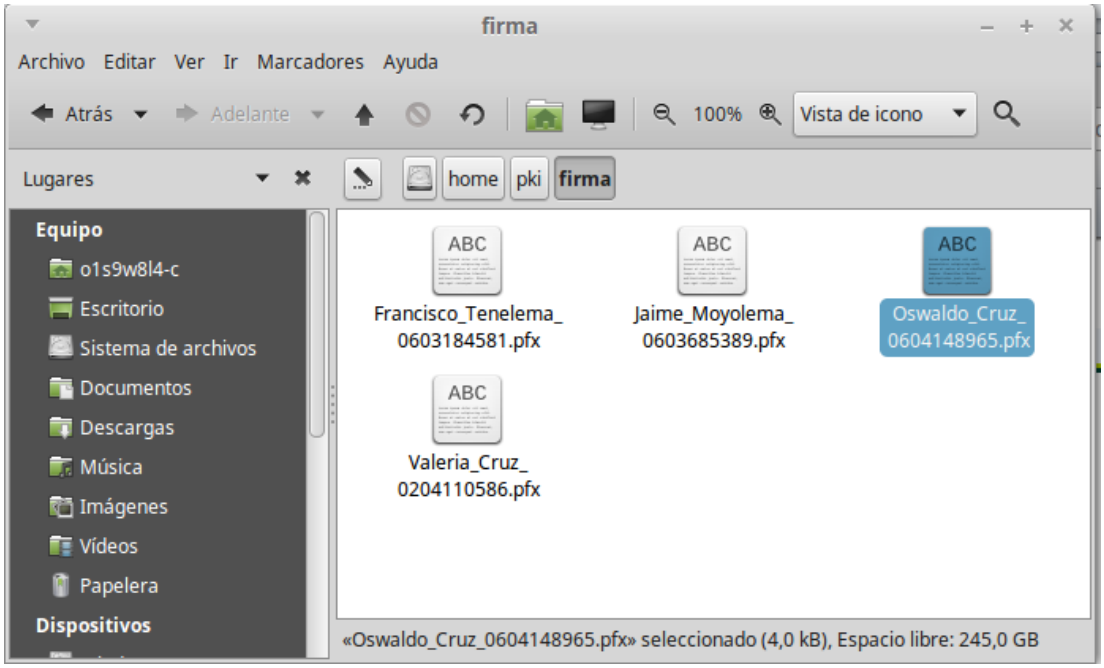


Figura 103: Carpeta de Firmas Electrónicas

Para tener acceso al certificado de firma electrónica, se debe ingresar el password ingresado en la venta de obtención de certificado anteriormente.

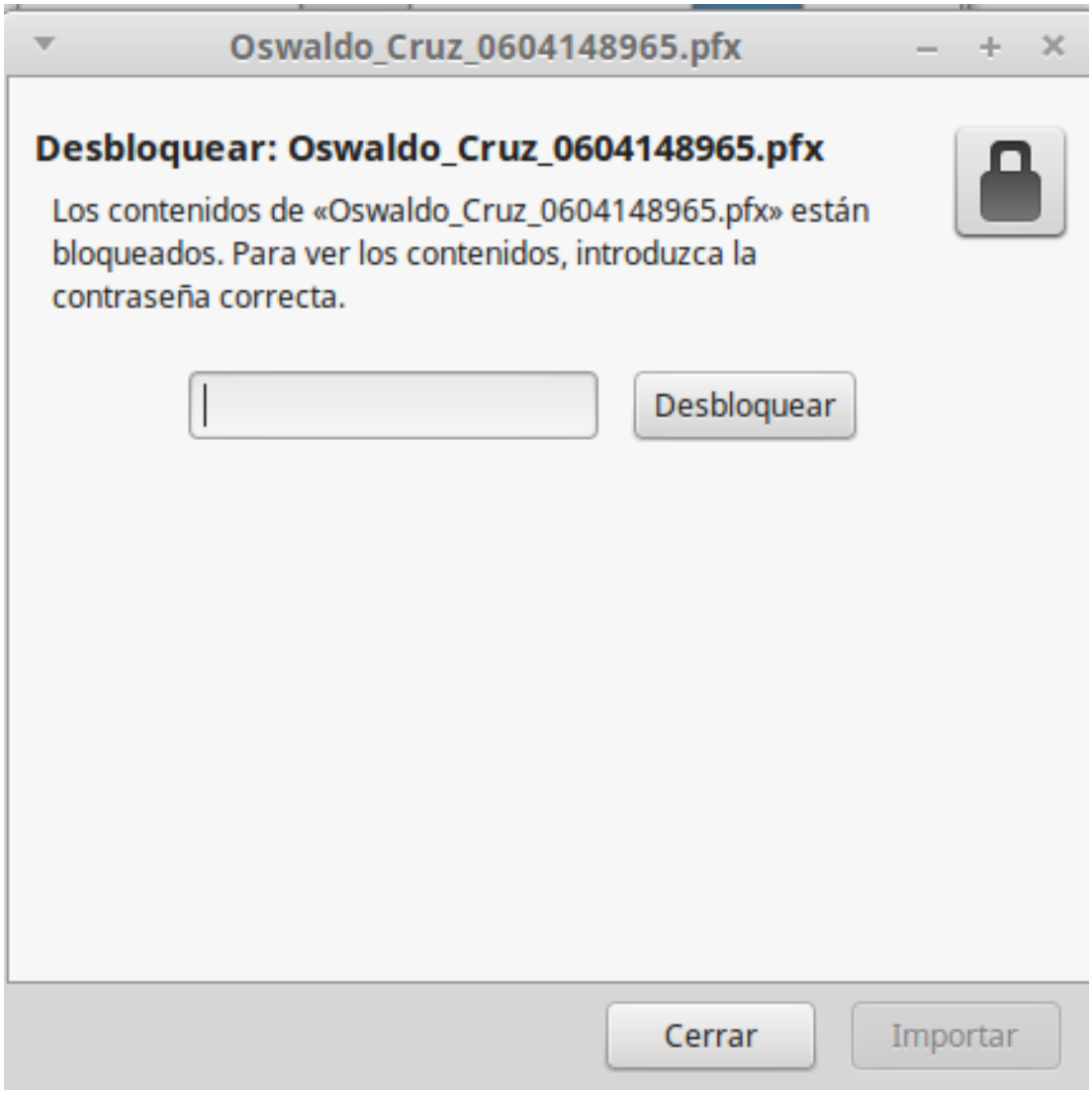


Figura 104: Archivo de firma electrónica bloqueada

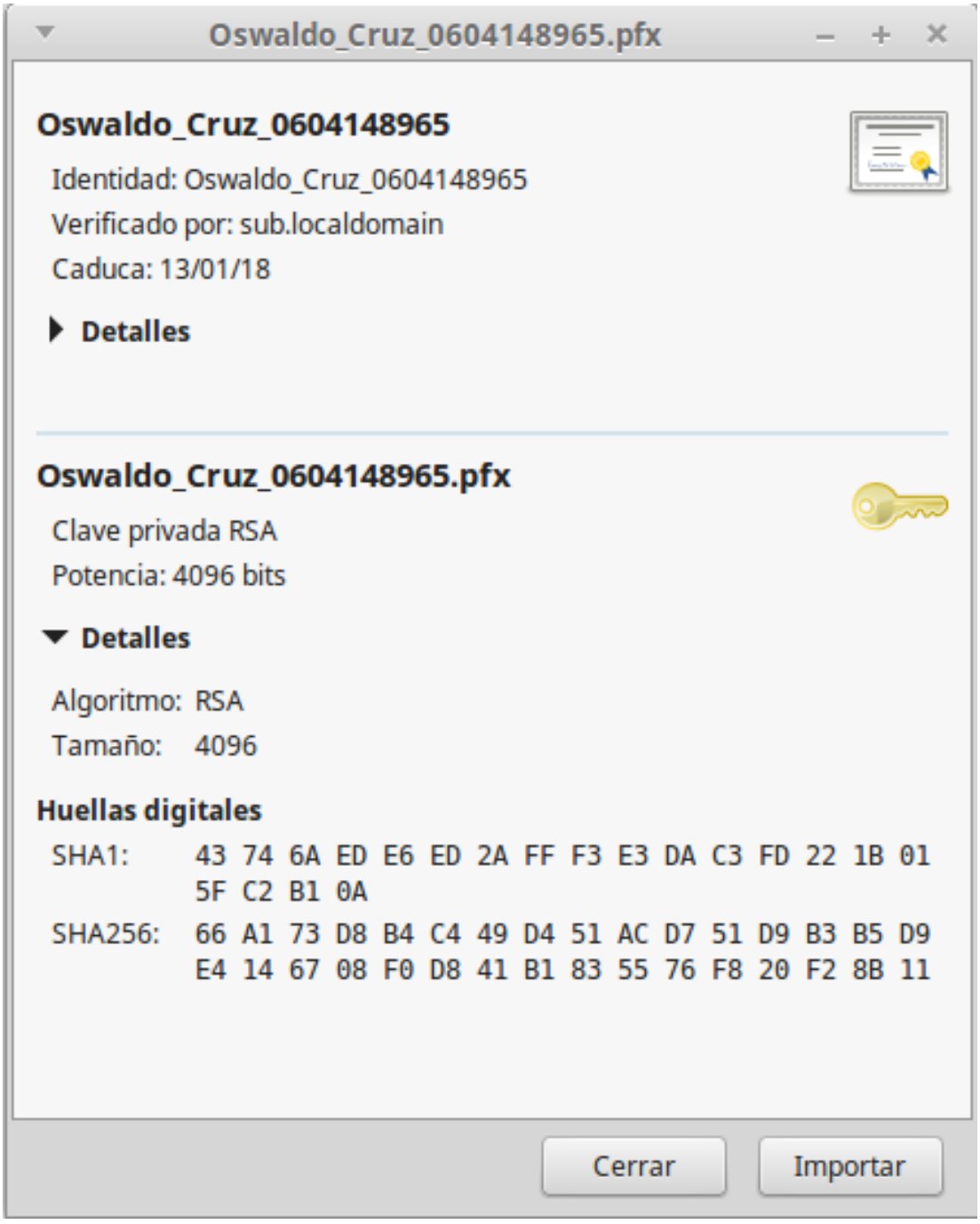


Figura 105: Archivo de firma electrónica desbloqueada

4.10.4 Firmar Documento

Para realizar la firma de documentos se procede a ingresar al menú Firma Electrónica, como lo muestra en el Gráfico Nro. 106, y posteriormente y posteriormente se procede a ingresar la información solicitada como lo muestra el Gráfico Nro. 107.

Dentro de dicho formulario se debe cargar el documento a ser firmado dando clic en el botón Seleccionar y Cargar, al igual debemos hacer para seleccionar el certificado de firma que vamos a utilizar.

Además, debemos hacer clic en botón Enviar Código Temporal, y nos va a llegar el código el celular que tengamos registrado en la base de datos. Ingresamos este código y un password el sistema procede a ejecutar la firma del documento.

Y posterior a esto el sistema a permitir la descarga de dicho documento ya firmado, como lo muestra el Gráfico Nro. 108.

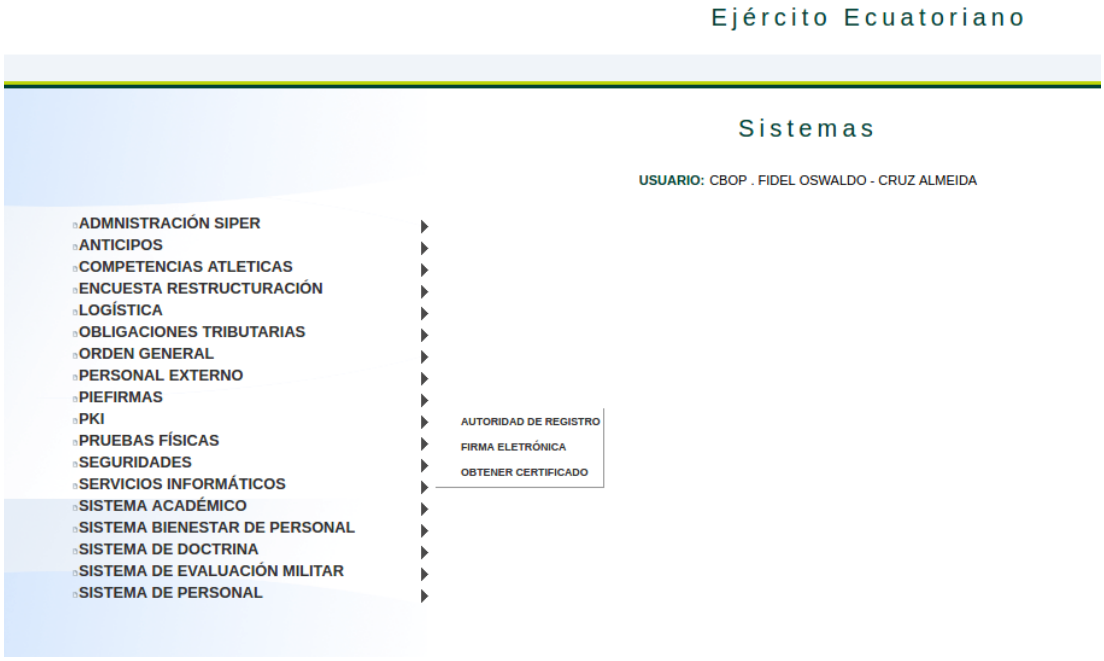


Figura 106: Menú del SIFTE para Firma Electrónica

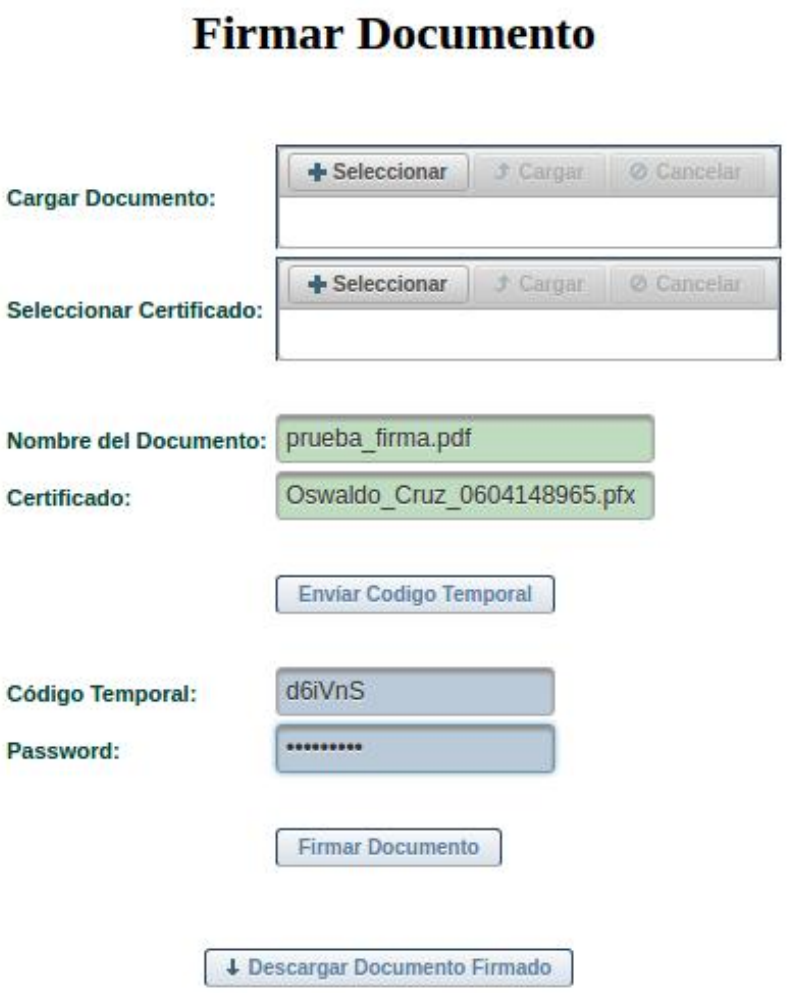


Figura 107: Formulario de Firma Electrónica

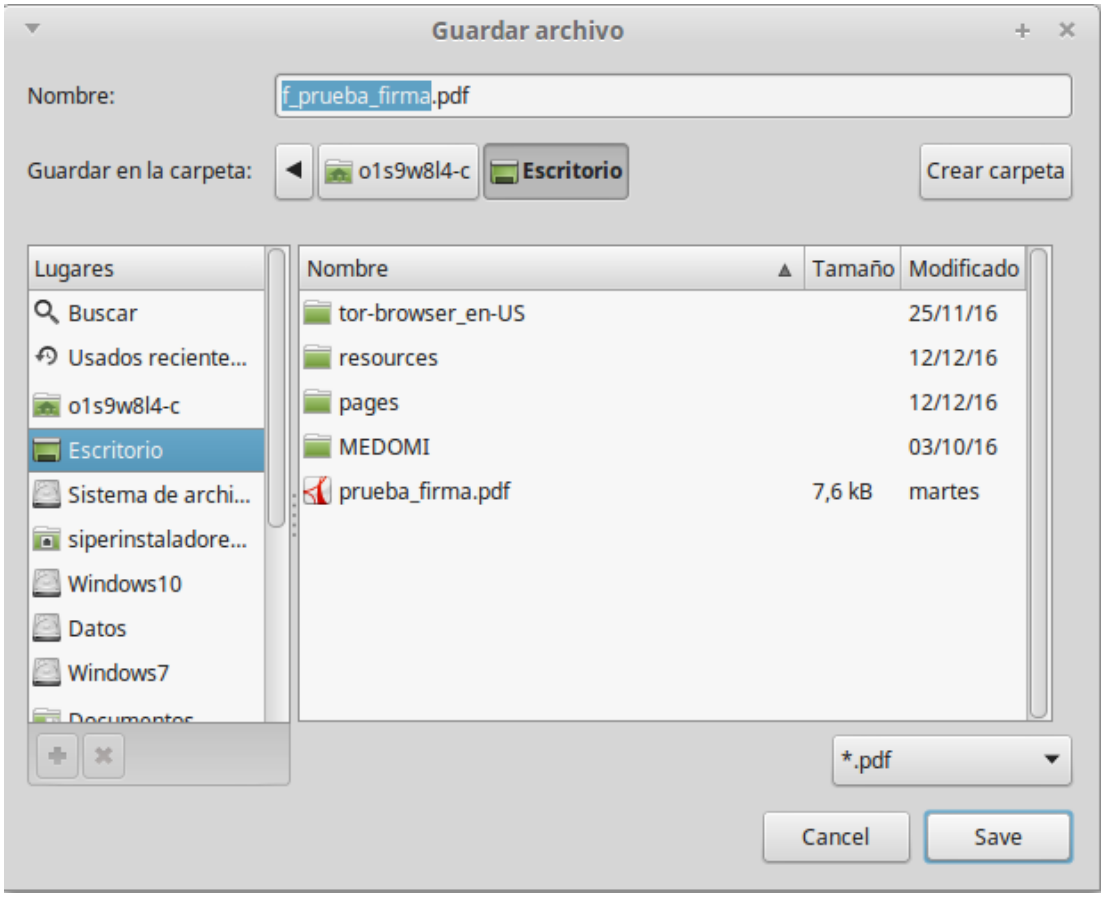


Figura 108: Descarga de documento firmado

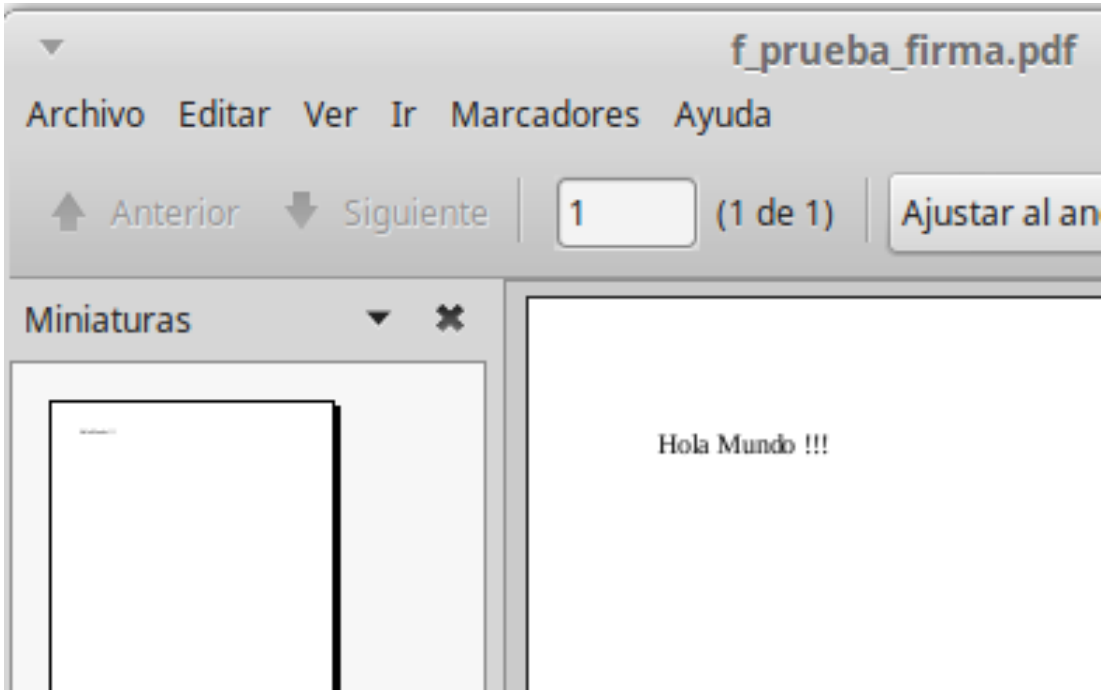


Figura 109: Contenido del documento firmado

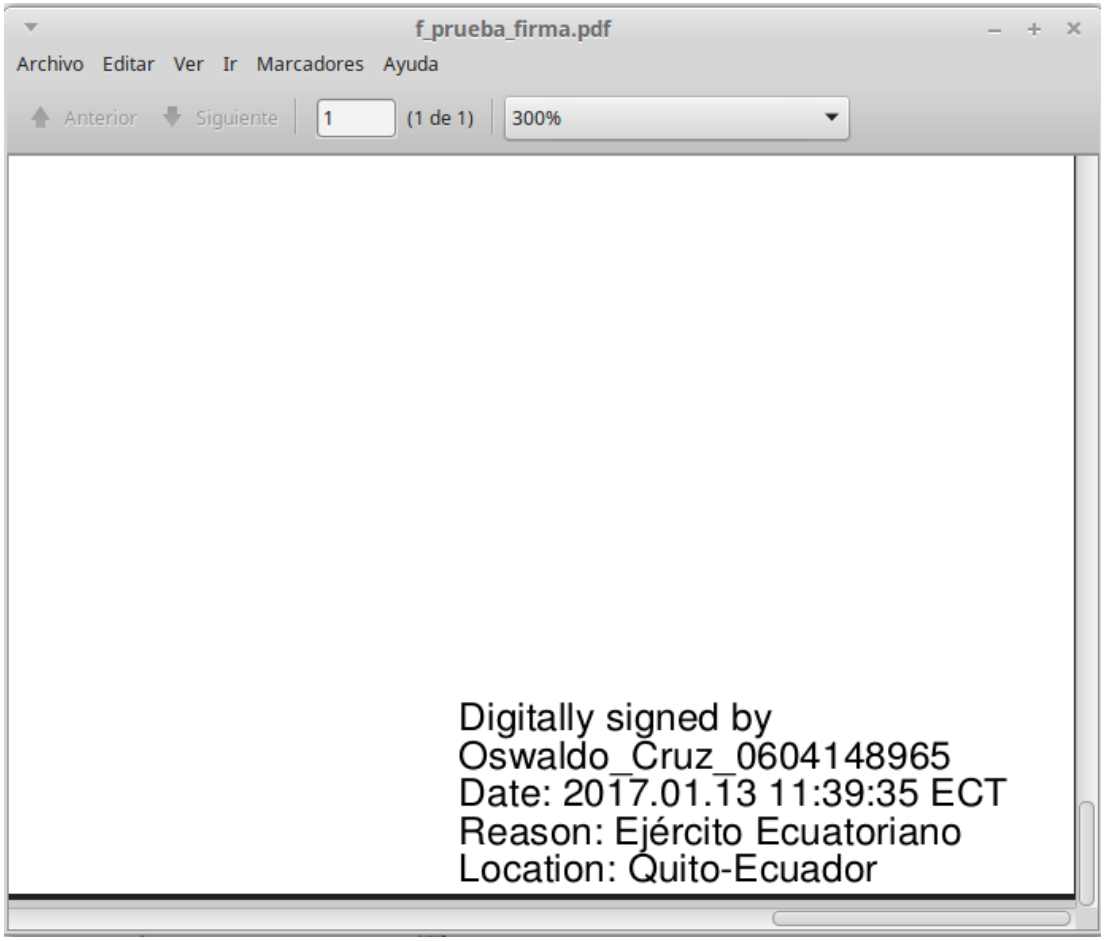


Figura 110: Sello de firma electrónica generado por el aplicativo

Como se puede apreciar el proceso firma internamente el documento y a la vez crea una etiqueta indicando que el documento está firmado, y marca la identidad del firmante y la fecha de la firma (Gráfico Nro. 111). Se lo puede verificar con cualquier software del mercado, pero se recomienda implementar un software propio de la institución.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El trabajo realizado está enmarcado en la actual "LEY DE COMERCIO ELECTRÓNICO, FIRMAS ELECTRÓNICAS Y MENSAJES DE DATOS", y por las políticas emitidas por la ARCOTEL (Agencia de Regulación y Control de las Telecomunicaciones); enfocándonos específicamente al Capítulo II "DE LOS CERTIFICADOS DE FIRMA ELECTRÓNICA", y al Capítulo III "DE LAS ENTIDADES DE CERTIFICACIÓN DE INFORMACIÓN".

- Los certificados digitales y firmas electrónicas están basados en la criptografía asimétrico (Es decir con una clave se cifra y con otra diferente se descifra la información), y existen funciones matemáticas que permiten un cifrado extremadamente seguro, por ello varias instituciones gubernamentales, bancos, y otras entidades grandes a nivel mundial confían en estos métodos.

- Las herramientas más utilizadas para implementar una PKI, son OpenSSL, OpenCA y EjbCA. Sin embargo, son herramientas muy genéricas y por la naturaleza de la Institución Armada, se tuvo que desarrollar una nueva herramienta llamada pkiEjercitoEC, la cual se adapta específicamente a la infraestructura informática y a las políticas de seguridad ya establecidas en el Ejército Ecuatoriano.

- Estamos viviendo en la era del conocimiento, y los Sistemas Informáticos son el pilar fundamental de muchas instituciones a nivel mundial e incluso de muchos estados. Y para su implementación es necesario aplicar Ingeniería, porque contempla procesos de desarrollo ya estandarizados y probados. Se puede afirmar también, que la programación es la base primordial de la carrera de Ingeniería en Sistemas, ya que el desarrollo de aplicativos es lo que permite

el crecimiento tecnológico y por ende es un aporte a la humanidad ya que mejora la calidad de vida de las personas.

5.2 Recomendaciones

- La principal recomendación, es que el trabajo realizado sea implantado en la Fuerza Terrestre y que para ello es necesario realizar el trámite respectivo en la ARCOTEL, y a su vez conseguir los recursos económicos que se establecen en los reglamentos de la misma.

- Se recomienda además la continua capacitación de personal técnico y de programadores pertenecientes a la Dirección de Comunicaciones y Sistemas del Ejército, ya que sin gente preparada se hace muy difícil desarrollar nuevas tecnologías.

- Finalmente se recomienda que invierta más en investigación científica y tecnológica, tanto en la ESPE como en la Fuerza Terrestre. Ya que esta es la forma en que la Institución y el país podrán progresar de verdad.

REFERENCIAS BIBLIOGRAFICAS

- ✓ Anzaldo, J. (2015). *wordpress*. Obtenido de https://janzaldo.wordpress.com/2005/11/15/%C2%BFque-es-una-arquitectura-de-sistemas-de-informacion/
- ✓ ARCOTEL. (2016). *www.acotel.gob.ec*. Obtenido de https://www.acotel.gob.ec
- ✓ Congreso Nacional. (2002). Ley de comercio electrónico, firmas electrónicas y mensajes de datos. Ecuador.
- ✓ Cordero, E. (2016). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Arquitectura
- ✓ dnielectronico. (2015). Obtenido de http://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_800
- ✓ Ley de comercio electrónico, f. e. (2002). Obtenido de http://www.arcotel.gob.ec/
- ✓ Repol, M. (2015). Obtenido de https://es.wikipedia.org/wiki/Protocolo_Ligero_de_Acceso_a_Directorios