UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CERTIFICADO

Que el trabajo titulado INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID realizado por: ESTEBAN EDUARDO CANDO PEÑAHERRERA Y DARIO JAVIER VELA ZAMBRANO, ha sido guiado y revisado periódicamente y cumple las normas estatuarias establecidas por la ESPE, en el reglamento estudiantil de la Universidad de las fuerzas Armadas ESPE.

El mencionado trabajo consta de un documento empastado y un disco compacto en el cual contiene los archivos en formato portátil de Acrobat (PDF)

Autorizan a ESTEBAN EDUARDO CANDO PEÑAHERRERA y DARÍO JAVIER VELA ZAMBRANO, entregar el mismo al Biblioteca.

Sangolquí, agosto de 2015

ING. FERNANDO GALARRAGA
Director

ING. ARTURO DE LA TORRE
Codirector

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

ESTEBAN EDUARDO CANDO PEÑAHERRERA

DARÍO JAVIER VELA ZAMBRANO

DECLARAMOS QUE:

El proyecto de pregrado titulado: "INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID", ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las citas que constan el pie de páginas correspondiente, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de pregrado en mención.

Sangolquí, agosto 2015

Esteban Eduardo Cando Peñaherrera

Darío Javier Vela Zambrano

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, ESTEBAN EDUARDO CANDO PEÑAHERRERA Y DARIO JAVIER VELA ZAMBRANO, autorizamos a la Universidad de las Fuerzas Armadas ESPE la publicación, en la biblioteca virtual de la institución, del trabajo de titulación "INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, agosto del 2015

Esteban Eduardo Cando Peñaherrera

Darío Javier Vela Zambrano

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

CANDO PENAHERRERA ESTEBAN EDUARDO

DARIO JAVIER VELA ZAMBRANO

DIRECTOR DE CARRERA DE INGENIERIA EN SISTEMAS E INFORMATICA

ING. MAURICIO CAMPANA

SANGOLQUÍ, AGOSTO 2015

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA

TEMA: INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID

AUTORES: ESTEBAN EDUARDO CANDO PEÑAHERRERA

DARIO JAVIER VELA ZAMBRANO

DIRECTOR: ING. FERNANDO GALARRAGA

CODIRECTOR: ING. ARTURO DE LA TORRE

SANGOLQUÍ

2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CERTIFICADO

Que el trabajo titulado INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID realizado por: ESTEBAN EDUARDO CANDO PEÑAHERRERA Y DARIO JAVIER VELA ZAMBRANO, ha sido guiado y revisado periódicamente y cumple las normas estatuarias establecidas por la ESPE, en el reglamento estudiantil de la Universidad de las fuerzas Armadas ESPE.

El mencionado trabajo consta de un documento empastado y un disco compacto en el cual contiene los archivos en formato portátil de Acrobat (PDF)

Autorizan a ESTEBAN EDUARDO CANDO PEÑAHERRERA y DARÍO JAVIER VELA ZAMBRANO, entregar el mismo al Biblioteca.

Sangolquí, agosto de 2015

ING. FERNANDO GALARRAGA	ING. ARTURO DE LA TORRE
Director	Codirector

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

ESTEBAN EDUARDO CANDO PEÑAHERRERA

DARÍO JAVIER VELA ZAMBRANO

DECLARAMOS QUE:

El proyecto de pregrado titulado: “INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las citas que constan el pie de páginas correspondiente, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de pregrado en mención.

Sangolquí, agosto 2015

Esteban Eduardo Cando Peñaherrera Darío Javier Vela Zambrano

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, ESTEBAN EDUARDO CANDO PEÑAHERRERA Y DARIO JAVIER VELA ZAMBRANO, autorizamos a la Universidad de las Fuerzas Armadas ESPE la publicación, en la biblioteca virtual de la institución, del trabajo de titulación "INFRAESTRUCTURA DE SERVICIOS DE CERTIFICACIÓN DIGITAL EN UN AMBIENTE SMARTGRID", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, agosto del 2015

Esteban Eduardo Cando Peñaherrera Darío Javier Vela Zambrano

DEDICATORIA

Cuando escriba estas líneas

Y el tiempo pase, sea pronto o sea tarde

No debo olvidarme de ti.

Todo el tiempo que pasamos

Han de convertirse en recuerdos

Ilusiones pasajeras que espero

Aun conservarlas con agrado.

Este trabajo investigativo está dedicado para:

Mi madre, gran amiga

Mi padre, mejor ejemplo

Mis hermanos, grandes alegrías

Mis amigos, buenos confidentes

Y para tod@s aquellos que ya no están, porque su apoyo fue importante.

Esteban Eduardo Cando P.

DEDICATORIA

Dedico este proyecto de tesis a mis padres, porque han estado conmigo a cada paso que he dado, cuidándome y dándome fortaleza para continuar. A mis padres quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ello que soy lo que soy ahora. Los amo con mi vida.

Darío Javier Vela Zambrano

AGRADECIMIENTOS

No podría escribir estas letras si no fuera por todo el apoyo de mi familia, en especial de mi padre Luis; La persona que siempre me demostró su afecto a través de sus consejos y aunque en muchas ocasiones no lo hubiera entendido hoy son tan claros. Gracias por siempre estar allí con tu ejemplo y guía.

A mi madre Isabel; La persona que ha demostrado con su paciencia y cariño, el abnegado e infinito amor de Dios. Gracias por ser mi confidente y mi amiga.

Debo agradecer a mis hermanos Luis, David y en especial a Jonathan, por ser pacientes conmigo y aceptarme tal y como soy. Espero llegar a ser en ellos un ejemplo de virtud.

A mi familia, que llevo siempre en mi mente. Gracias por apoyarme y socorrerme en mis dificultades. No podría haber logrado terminar este gran peldaño sin su apoyo incondicional.

A mi director de tesis Ing. Fernando Galarraga, y codirector Ing. Arturo de la Torre por su paciencia y consejos durante la realización de esta tesis, así como a las autoridades de la facultad que supieron canalizar positivamente mis solicitudes

Esteban Eduardo Cando P.

AGRADECIMIENTOS

Los resultados de este proyecto, están dedicados a todas aquellas personas que, de alguna forma, son parte de su culminación.

Mis sinceros agradecimientos están dirigidos hacia mi Madre, quien con su ayuda desinteresada, me brindó todo el apoyo necesario en momentos difíciles. A todas las personas que ayudaron a plasmar los resultados necesarios para el éxito de este proyecto. A mi familia por siempre brindarme su apoyo, tanto sentimental, como económico. Pero, principalmente mis agradecimientos están dirigidos hacia la excelentísima autoridad que es nuestro director de carrera Mauricio Campaña y director de tesis Fernando Galarraga, sin los cuales no hubiésemos podido salir adelante.

Gracias Dios, gracias a todos, gracias padre y hermanos, y en especial, gracias a mi madre.

Darío Javier Vela Zambrano

TABLA DE CONTENIDO

TABLA DE FIGURAS

RESUMEN

Una infraestructura de servicios de certificación digital en ambientes Grid, coordinados mediante un middleware de integración, es la mejor forma de garantizar el buen funcionamiento de los procesos de autorización, autenticación y acceso en los sistemas informáticos. Una autenticación eficiente, abre una sesión de servicios en un nodo de la Grid, para que este sea aprovechado en los procesos tecnológicos. Este trabajo de investigación tiene como propósito contribuir a las empresas públicas y privadas que han decidió emplear una Smart Grid, y que han optado por una infraestructura segura. Bajo un marco investigativo, se utilizó las herramientas provistas por parte de Globus Alliance, “Globus Toolkit”, el cual permite la construcción de grids computaciones, y proveer herramientas para su administración y configuración, tanto en ambientes reales, como de investigación. Esta investigación fue ejecutada, conjuntamente con una recopilación de los conceptos básicos de certificación digital, elementos y mecanismos básicos de seguridad de una Infraestructura de clave pública, elementos propios que intervienen en un ambiente Grid.Este proyecto fue ejecutado bajo un sistema operativo Ubuntu, pero puede ser replicado en sistemas similares de tipo Unix, porque así lo permiten las herramientas utilizadas.Finalmente, se desarrolló un prototipo para la generación y administración de Certificados Digitales usando la Certificación Digital con una arquitectura Single Sign On, y un manual de procedimientos para la configuración de cada uno de los nodos de la Grid.

PALABRAS CLAVES:

- **SMART GRID**
- **INFRAESTRUCTURA**
- **GLOBUS TOOLKIT**
- **PKI**
- **SINGLE SIGN ON**

ABSTRACT

An infrastructure for digital certification services in Grid environments, coordinated by middleware of integration, is the best way to ensure the smooth operational processes of authorization, authentication and access to computer systems. An efficient authentication system, open a node on the Grid, for it to be exploited in technological processes. This research aims to contribute public and private companies that have decided to use a Smart Grid, and have opted for a secure infrastructure. All the research framework was used with the tools provided by the Globus Alliance, "Globus Toolkit," which allows us to build grids computations, and provides tools for administration and configuration, both in real environments, such research was used. This research was executed, with a compilation of the basics of digital certification, basic elements and security mechanisms of Public Infrastructure, key elements involved in a Grid environment. This project was implemented under a Ubuntu operating system, but we can be replicated on a similar Unix systems, because the tools allow to the users. Finally, we developed a prototype for the generation and management of digital certificates using the Digital Certificate with Single Sing architecture, and a manual of procedures for setting up each of the nodes of the Grid

KEYWORDS:

- **SMART GRID**
- **INFRASTRUCTURE**
- **GLOBUS TOOLKIT**
- **PKI**
- **SINGLE SIGN ON**

CAPÍTULO 1 INTRODUCCIÓN

1.1. Antecedentes

Se ha evaluado e identificado la necesidad que tienen las instituciones públicas y privadas de implementar técnicas de seguridad informática que permitan garantizar la integridad, confidencialidad y disponibilidad de la información.

Bajo este escenario, las organizaciones deben identificar arquitecturas que faciliten la interacción basadas en ambientes de “SmartGrid”, así como plantear investigaciones y proyectos que promuevan las nuevas tecnologías que cada día aparecen en el mercado mundial.

1.2. Planteamiento del Problema

No se han realizado estudios sobre GSI basado en SMARTGRID en las Universidades del Ecuador. Los aspectos más destacados de la certificación digital no forman parte de las prestaciones que ofrecen las instituciones de educación superior en el Ecuador.

Actualmente, la información no tiene por qué ser aislada para el beneficio de una sola institución de educación superior sino que debe ser difundida de manera segura a través de un SMARTGRID para que otras instituciones se puedan beneficiar y a la vez aportar con conocimiento para que unificando esfuerzos se logre un crecimiento mutuo como formadoras de nuevos profesionales del Ecuador.

1.3. Justificación

De los estudios hechos anteriormente, se ha determinado y comprobado la importancia y ventajas que ofrece la Infraestructura de Clave Pública (PKI), y como ésta puede ser empleada como un potente mecanismo de seguridad en el intercambio de datos, transacciones y comunicaciones.

Proyectando esta visión a un futuro no distante, se desea unificar teorías y modelos innovadores como la arquitectura “SmartGrid” para implementar servicios de certificación digital, generando confianza entre las Instituciones, en lo referente a la información que éstos manejen, así como de los organismos que mantengan convenios entre sí.

1.4. Objetivos

1.5. Objetivo General

- Representar mediante un ambiente de simulación una infraestructura SmartGrid, que a mediante GSI, establece un modelo de terceras partes de confianza.

1.6. Objetivo Específico

- Implementar una infraestructura grid, mediante el simulador Globus ToolKit bajo un sistema operativo Unix.
- Obtener una SmartGrid segura, basada en un modelo de terceras partes confianza.
- Implementar una arquitectura Single Sign-On utilizando una entidad certificadora y certificados digitales verificados.

1.7. Alcance

La infraestructura de seguridad de mayor uso en SmartGrid es GSI[1], la misma utiliza certificados X.509 para identificar entidades[2] en el Grid, en ella cada entidad es autenticada por un único certificado digital, pero sin embargo, una entidad espera tener accesos a aplicaciones utilizando un Login y Password. Para cumplir simultáneamente con estos requisitos, los portales Grid operan utilizando mecanismos que automáticamente suplen las credenciales reconocidas en la organización virtual luego de que el usuario logra autenticarse utilizando una combinación de cuenta de usuario y palabra clave que coincide con una combinación almacenada en la base de datos de cuentas del dominio de seguridad al que el usuario pertenece.

GSI es, tal vez, la infraestructura que más dificultad ofrece por el carácter distribuido y heterogéneo de sus componentes que forman parte de la tecnología Grid Computing.

Con el desarrollo de este trabajo de grado se pretende realizar una investigación que indique los requisitos de la infraestructura y arquitectura tecnológica para simular un ambiente PKI viable en SmartGrid y así implementar una aplicación Single Sing-On en una SmartGrid, bajo la herramienta Globus ToolKit. Todo esto debe ser implementado en un sistema operativo Unix, el cual por comodidades didácticas se ha establecido como Ubuntu 12.02.

---

[1] Grid Security Infraestructure

[2] Se definen entidades todo aquello cuya existencia es perceptible por algún sistema

CAPÍTULO 2 MARCO TEÓRICO

2.1. Grid Computing

En la actualidad, muchas redes de computadoras han sido creadas para el mejor desempeño de las organizaciones, pero tienen la particularidad que sólo tiene acceso directo y exclusivo a su propio procesador o memoria para cada computador de esa red; Sin embargo, el Grid Computing da una nueva alternativa muy interesante e innovadora y, más que nada, una alternativa real para maximizar el aprovechamiento de las redes ya establecidas. Si se analiza por un momento, cualquier red de computadoras puede adaptarse fácilmente a una topología de Grid Computing, hablando más que de forma física, de manera lógica, de tal forma que se puede considerar que cualquier red, básicamente ya tiene la infraestructura física para implementar un Grid Computing. (Zavoral, Jung, & Costin , 2013)

Con el pasar del tiempo, las necesidades de unir redes de computadoras y distribuir el trabajo en distintas máquinas fueron cubiertas con el desarrollo de los sistemas distribuidos, sin embargo, en áreas tales como la ciencia y la ingeniería surgen problemas más complejos en los que se tienen que resolver procesos más demandantes, con cada vez mayor cantidad de información que a menudo, se encuentra distribuida geográficamente.

Además, surgen varios detonantes como el crecimiento del Internet, el desarrollo de hardware más rápido, software más sofisticado y el aumento de velocidad en las redes y su ancho de banda.

Es así como surge la idea del Grid Computing, que con bajas inversiones se puede obtener el máximo procesamiento con equipos existentes. El Grid Computing, básicamente permite a un computador que requiera de más procesamiento, tomarlo de máquinas que pertenecen a su red y no estén usando su capacidad máxima.

Por lo tanto, el Grid Computing puede considerarse como un sistema distribuido híbrido ya que es una tecnología que ha recogido muchas de las

bondades que pueden brindar las redes tal como utilizar y compartir simultáneamente recursos remotos para múltiples usuarios que pertenecen a las redes ya establecidas de las empresas.

2.2. Aplicaciones Grid

De una forma general y tratando de agrupar todas estas aplicaciones, se puede considerar que Grid Computing puede ser usado en cualquier área cuyas necesidades comprendan: cálculos intensos, compartir contenidos, acceso a software y servicios remotos, lograr una alta resolución en las imágenes, capturar y guardar grandes cantidades de información, simular para entender mejor conjuntos de datos, procesar datos en tiempo real o sobre demanda, hacer que estos datos esté disponible en cualquier parte del mundo y servicios orientados a cómputo.

Por las características propias de la Grid Computing, esta ha sido aplicada en áreas científicas como:

- La ingeniería
- Diseño mecánico
- Diseño electrónico
- Análisis de elementos y fallas
- La bioinformática
- La biotecnología
- Genómica
- Interpretación digital
- Visión artificial
- Climatología
- Sismología
- Defensa
- Investigación en física
- Astronomía
- La ingeniería civil
- El desarrollo aeroespacial
- El diagnóstico médico
- Video sobre demanda
- El desarrollo de productos
- Proyectos de comunidades
- Medios digitales
- Educación
- Entretenimiento on-line

2.3. Tipos de Grid

Para tener una idea más clara sobre el Grid Computing, se ha realizado una clasificación desde el punto de vista del usuario final, de acuerdo a los servicios que puede proveer estos se dividen en:

- Grid De Servicio entre Computadores, que ejecuta trabajos bajo recursos de otros equipos.
- Grid De Servicios de Datos, que permiten el acceso de información bajo el concepto de utilizar los recursos de toda la red.
- Grid De Servicio de Aplicaciones, que buscan el mejor desempeño del software en base a recursos externos de la empresa.

No se tiene la intención de generalizar ningún tipo, ya que sería muy difícil tratar de dar una descripción general de cómo funciona cualquier software de Grid Computing, ya que algunas características pueden variar de acuerdo al Hardware o Software; Sin embargo, se puede puntualizar que un software de este tipo debe proveer de algunas características básicas tales como la autentificación, las políticas de autorización, el descubrimiento y ubicación de recursos, el acceso remoto a datos y la alta velocidad de transferencia, el manejo de recursos y fallas, el monitoreo, garantizar el rendimiento, el manejo de cuentas y pagos, adaptación, etc.

2.4. Caracterización de Grid Computing

Grid Computing consta de tres operaciones básicas: almacenamiento, recuperación y actualización de la información, y dichas operaciones necesitan una manera adecuada y normalizada para el acceder a la información bajo autentificación. (Zavoral, Jung, & Costin , 2013)

Este es el caso de Grid Computing que es una infraestructura de hardware y software para resolver tareas de cómputo complejo y gran cantidad de datos que provee a usuarios de puntos de acceso a recursos distribuidos. De esta forma, se puede decir que ya se cuenta con una solución eficiente para la

administración de recursos, programando y distribuyendo la ejecución de procesos en las computadoras que forman una red.

2.5. Proyectos en Grid Computing

El Grid Computing en la actualidad se encuentra estudiada y aplicada a nivel mundial, y este se refleja en un sin número de proyectos internacionales como:

a) Proyecto SETI.

SETI es el acrónimo del inglés *Search for ExtraTerrestrial Intelligence*, o Búsqueda de Inteligencia Extraterrestre. Este proyecto fue diseñado con el propósito de encontrar vida extraterrestre mediante el análisis de varias señales electromagnéticas que han sido capturadas por radios telescopios, así como enviando señales que esperan sean contestadas.

b) Find a Drug

La filosofía de la tecnología Grid es la interconexión de recursos de computación para poder afrontar problemas de gran tamaño que, con la capacidad de cálculo de las máquinas actuales, no se podría abarcar.

c) TeraGrid

La TeraGrid es una infraestructura abierta la investigación científica que combina los recursos de clase de liderazgo a los once sitios asociados los recursos computacionales persistentes.

2.6. Infraestructura de Seguridad para Grid

El GSI, formalmente llamado GLOBUS SECURITY INFRSTRUCTURE, es una especificación para una seguridad y delegación de comunicación entre usuarios o servicios que estén usando encriptación asimétrica. La infraestructura en seguridad GRID (GSI) es seguridad enfocada a redes basadas básicamente en claves públicas encriptados.

Unos de los principales objetivos de GSI es de establecer una autentificación, pero que sea confidencial, de comunicación segura entre los elementos computacionales de la GRID. Entonces el GSI habilita la seguridad

entre la organización establecida en la empresa sin tener que involucrar cualquier sistema de seguridad centralizado. (Huaxiong, Pieprzyk, & Varadharajan, 2008)

El GSI normalmente soporta SINGLE SIGN-ON para usuario de la GRID, y con esto delega permisos y credenciales de acceso a servicios computacionales involucrando múltiples recursos computacionales. Normalmente un usuario o un servicio de la GRID son identificados por certificación de firmas para su autentificación. Un certificado contiene información pertinente para la identificación y autenticación para el acceso de los servicios de cada usuario.

2.6.1. Autenticación

La Autenticación se la realiza con el uso de tecnología de firmas digitales; esta autenticación segura permite acceder a recursos o aplicaciones definidas en el sistema para bloquear cualquier información o dejar abierta solo a las personas que se desee dar el acceso solicitado.

La autorización se presenta en base a la validación de datos obtenidos con una autentificación y se puede proceder a dar el acceso solicitado, pero también un problema, casualmente los servicios tienen que obtener información o recursos de usuarios independientes a la GRID, pero para realizar este proceso debe tener los privilegios pertinentes para obtener lo solicitado. El proceso de autorización permite: la creación de privilegios a delegar, crear una nueva clave, delegar, firmar y de esta manera obtiene el acceso solicitado y es autenticado en la GRID.

Las seguridades en GRID ofrecen también el acceso total y autentificado a:

a) Supercomputación distribuida

- Simulaciones
- Cálculos numéricos intensos
- Análisis de datos de alto rendimiento
- Extracción de almacenes

b) Sistemas distribuidos en tiempo real

- Sistemas médicos, como el tratamiento de imágenes virtuales.

c) Proceso intensivo de datos

- Gestionar bases de datos distribuidos.

d) Entornos virtuales de colaboración

- Compartición de recursos entre servidores virtuales para mejor procesamiento de sus sistemas operativos.

e) Servicios puntuales de aplicaciones

- Aplicaciones que permitan el acceso fiable y optimizado.

2.7. Arquitecturas e Infraestructuras GRID

La seguridad es un aspecto vital en el Grid Computing, destacándose las tres principales características de seguridad que se deben proveen en un ambiente grid:

a) **Login**: significa que una entidad es capaz de registrarse haciendo uso de sus credenciales de seguridad y tener acceso a los servicios del Grid computing por un cierto periodo de tiempo.
b) **Autenticación**: significa ser capaz de crear una identidad, como por ejemplo cuando accedes con login en tu cuenta de mail te auténticas con el servidor dando tu usuario y tu contraseña, así otorgando acceso a los servicios requeridos.
c) **Autorización:** es el proceso de gestionar y controlar los privilegios de las entidades. (Wilkinson, 2013)

2.7.1. Sistemas de autenticación reducida

Los sistemas de autenticación reducida están en función del ítem que utilizan para la verificación y estos se dividen en varias categorías entre las cuales se detallan las más importantes.

2.7.2. Sistemas basados en conocimiento.

A estos sistemas se los conoce por el uso de Password o Passphrase. Estos son los más conocidos; Un ejemplo claro de esto sería una contraseña o clave, la cual es una forma de autentificación que utiliza información secreta para controlar el acceso hacia algún recurso. La contraseña normalmente debe mantenerse en secreto ante aquellos a quien no se le permite el acceso. Aquellos que desean acceder a la información se les solicitan una clave; si conocen o no conocen la contraseña, se concede o se niega el acceso a la información según sea el caso.

2.7.3. Sistemas basados en objetos

Un claro ejemplo de esta categoría es una tarjeta de identidad, una tarjeta inteligente o *smartcard*, dispositivo usb tipo epass token, o dongle criptográfico. Uno muy conocido es el sistema Single Sign-On que tiene una doble utilidad, para las entidades que soliciten acceso supone la comodidad de identificarse una sola vez y mantener la sesión válida para el resto de aplicaciones que hacen uso del SSO. Además le permite identificarse usando los siguientes métodos de autenticación:

- Usando su usuario y contraseña
- Usando su TOKEN
- Usando su sistemas de autentificación reducida

Para los desarrolladores y administradores de aplicaciones es una manera de simplificar enormemente la lógica de sus aplicaciones, al poder delegar completamente la tarea de autenticar a los usuarios a un sistema independiente de las mismas.

2.7.4. Sistemas basados en característica física del usuario

En este tipo de sistemas se tiene por ejemplo la verificación de voz, así como la escritura, las huellas digitales, de patrones oculares del iris, siendo estas únicas por usuario.

a) Sistemas de autenticación biométrica

Esta categoría es más amigable para el usuario ya que no va a necesitar recordar contraseñas o números de identificación complejos, y, como se suele decir, el usuario puede olvidar una tarjeta de identificación en casa, pero nunca se olvidará de su mano o su ojo; Además son mucho más difíciles de falsificar que una simple contraseña o una tarjeta magnética; las principales razones por la que no se han impuesto es por su elevado precio, fuera del alcance de muchas organizaciones, y su dificultad de mantenimiento.

b) Verificación de voz

En los sistemas de reconocimiento de voz no se intenta, como mucha gente piensa, reconocer lo que el usuario dice, sino identificar una serie de sonidos y sus características para decidir si el usuario es quien dice ser. Para autenticar a un usuario utilizando un reconocedor de voz se debe disponer de ciertas condiciones para el correcto registro de los datos, como ausencia de ruidos, reverberaciones o ecos; Idealmente, estas condiciones han de ser las mismas siempre que se necesite la autenticación.

Cuando un usuario desea acceder al sistema pronunciará unas frases en las cuales reside gran parte de la seguridad del protocolo; en algunos modelos, los denominados de texto dependiente, el sistema tiene almacenadas un conjunto muy limitado de frases que es capaz de reconocer: por ejemplo el usuario se limita a pronunciar su nombre, de forma que el reconocedor lo entienda y lo autentique.

c) Verificación de escritura

Aunque la escritura, no es una característica estrictamente biométrica, se suele agrupar dentro de esta categoría; de la misma forma que sucedía en la verificación de la voz, el objetivo aquí no es interpretar o entender lo que el usuario escribe en el lector, sino autenticarlo basándose en ciertos rasgos tanto de la firma como de su rúbrica.

La verificación en base a firmas es utilizado día a día en documentos personales, no obstante existe una diferencia fundamental entre el uso de las firmas que se hace en nuestra vida cotidiana y los sistemas biométricos; mientras que habitualmente la verificación de la firma consiste en un simple análisis visual sobre una impresión en papel, estática, en los sistemas automáticos no es posible autenticar usuarios en base a la representación de los trazos de su firma. En los modelos biométricos se utiliza además la forma de firmar, las características dinámicas, por eso se les suele denominar DSV[2], y el cual se encarga de analizar el tiempo utilizado para rubricar, las veces que se separa el bolígrafo del papel, el ángulo con que se realiza cada trazo

d) Verificación de huellas

Típicamente la huella dactilar de un individuo ha sido un patrón bastante bueno para determinar su identidad de forma inequívoca, ya que está aceptado que dos dedos nunca poseen huellas similares, ni siquiera entre gemelos o entre dedos de la misma persona. Por tanto, parece obvio que las huellas se convertirían antes o después en un modelo de autenticación biométrico: desde el siglo pasado hasta nuestros días se vienen realizando con éxito clasificaciones sistemáticas de huellas dactilares en entornos policiales, y el uso de estos patrones fue uno de los primeros en establecerse como modelo de autenticación biométrica.

---

[2] Dynamic Signature Verification = Comprobación Dinámica de la firma

2.8. Single Sign On

El concepto de Single Sign On se refiere al acceso a múltiples recursos por medio de un único proceso de ingreso. Gran cantidad de las arquitecturas implementadas en diferentes organizaciones han sido diseñadas con el objeto de dar acceso a los usuarios a múltiples servicios Web y/o aplicaciones. En la mayoría de los casos se encuentra que cada uno de los servicios o aplicaciones cuenta con su propio componente de seguridad, lo cual generalmente compromete la seguridad de todo el sistema, dado que el nivel de seguridad de todo un sistema es igual al nivel de seguridad del componente más inseguro que lo compone. Una de las posibles soluciones a este problema es implementar la estrategia Single Sign-On. Se presentarán cuatro arquitecturas que permiten implementar SSO: Password vault, Administración centralizada con almacenamiento local de credenciales, Administración y almacenamiento de credenciales centralizados y Arquitectura SSO totalmente distribuida, explicando sus características, ventajas y desventajas.

El principal objetivo de una arquitectura que implemente Single Sign-On es transferir la funcionalidad y complejidad de todos los componentes de seguridad a un solo servicio de Single Sign-On (SSO). En una arquitectura SSO, todos los mecanismos de seguridad se encuentran concentrados en el SSO, siendo éste el único punto de autenticación y registro en el sistema. Otro beneficio de una arquitectura Single Sign-On es que los usuarios deben hacer el proceso de ingreso una sola vez, a pesar de que continúan interactuando con múltiples componentes de seguridad en el sistema. (Rediris)

El concepto de Single Sign-On no necesariamente se refiere a una sincronización de *password*s, ya que en ese caso todas las aplicaciones y servicios funcionan con un mismo *password*. Aunque una sincronización de *password*s le permite al usuario experimentar las ventajas del SSO, ésta no puede considerarse una implementación real, ya que en lugar de fortalecer las características de seguridad del sistema, éstas se estarían debilitando, dado que cuando todas las aplicaciones o servicios utilizan un mismo *password*, se

corre el riesgo de que si un intruso logra conseguir el *password* de una de las aplicaciones o servicios, inmediatamente tendrá acceso a todas ellas.

Una implementación real de SSO, deberá contar con un agente SSO que se encarga de almacenar en una base de datos o directorio protegido los *password*s que le permiten al usuario acceder a cada una de las aplicaciones o servicios, en el momento que lo desee, ya que el proceso de *login* se realiza de manera transparente para el usuario, una vez que éste ha sido autenticado por medio de la arquitectura SSO.

A pesar de que en una verdadera implementación existe un *password* que permite el acceso a todas las aplicaciones o servicios, esta aparente debilidad se soluciona sometiendo al usuario a un proceso de autenticación fuerte en el momento de hacer el ingreso, haciendo que la arquitectura SSO aumente el nivel de seguridad del sistema completo, en lugar de disminuirlo.

Autenticación fuerte se refiere al proceso de autenticación en sistemas que requieren múltiples factores para realizar la identificación del usuario, los cuales utilizan tecnología avanzada como contraseñas dinámicas o certificados digitales.

El ejemplo más simple de autenticación con múltiples factores es la tarjeta débito, ya que ésta requiere algo que el usuario tiene como por ejemplo la tarjeta con banda magnética y algo que el usuario sabe por ejemplo su clave; con solamente una de las dos, el usuario no logrará autenticarse para realizar consultas o transacciones.

2.8.1. Arquitecturas

Existen diferentes tipos de arquitecturas que permiten implementar SSO. Cada una de ellas posee características que la hace más apropiada para algún tipo de organización. La decisión de adoptar una u otra arquitectura básicamente depende de los recursos computacionales y/o económicos disponibles, y las decisiones de diseño establecidas por el equipo del proyecto.

Las diferentes arquitecturas SSO están compuestas por tres componentes básicos:

- Interface: El modo en que el SSO interactúa con una determinada aplicación. Usualmente reside en el cliente, y es conocido como Agente SSO.
- Administración: El mecanismo que permite configurar, mantener y monitorear el proceso de SSO.
- Credenciales: Cada aplicación a la que se accede requiere información confidencial (nombre de usuario, contraseña, etc.), que agrupada recibe el nombre de credenciales. Las credenciales deben almacenarse de manera protegida para que sea únicamente el agente SSO quien pueda acceder a ellas.

2.8.2. Password Vault

Se trata de la configuración más básica para implementar SSO utilizando credenciales. En este caso los tres elementos de la arquitectura se encuentran ubicados en el cliente y, por lo tanto, es justamente allí desde donde se accede a las aplicaciones, para lo cual se deben previamente almacenar las credenciales correspondientes, para que puedan ser suministradas a las aplicaciones cuando sea necesario, como se ilustra en la figura siguiente.

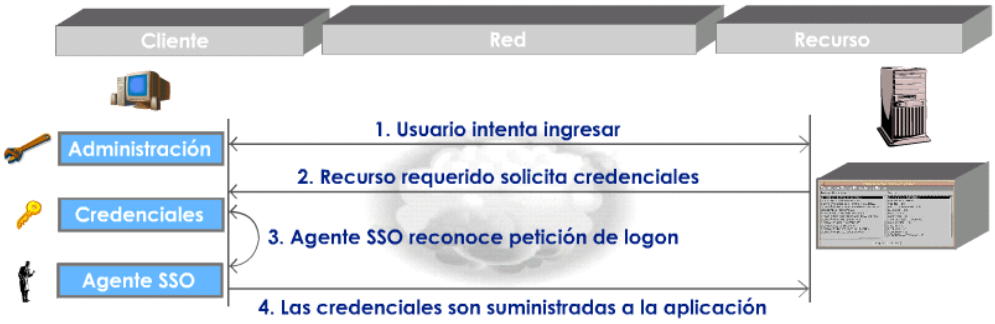

Figura 1 Manejo de Credenciales en Password Vault

Fuente: docs.fedoraproject.org

Características:

- Funciones administrativas limitadas. (La administración de cada uno de los clientes debe realizarse desde la estación correspondiente y por lo tanto generalmente termina quedando a cargo del usuario.)
- No es posible actualizar los clientes de manera masiva en toda la organización, requiere que se realice máquina por máquina.
- Ventajas:
- Su implementación no es mucho más complicada que instalar un nuevo software en el equipo cliente.
- Pocos recursos computacionales necesarios (Un servidor central donde residen las diferentes aplicaciones y los clientes necesarios).

Desventajas:

- La administración local obliga a tomar medidas adicionales de seguridad informática y control de acceso por parte de la empresa.
- El nivel de transparencia para el usuario es bajo ya que éste generalmente se encuentra comprometido con la administración del proceso de ingreso.
- El almacenamiento local de credenciales no permite que el usuario acceda a las aplicaciones desde múltiples estaciones.
- La información entre el cliente SSO y el servidor no viaja cifrada.

2.8.3. Administración con almacenamiento local de credenciales

Con el propósito de solucionar los principales inconvenientes que presenta la arquitectura *Password Vault*, surge la Administración centralizada con almacenamiento local de credenciales, ofreciendo un mecanismo para controlar y supervisar el proceso de ingreso, y eliminando la necesidad de configurar el SSO en cada uno de los clientes. La arquitectura en mención se ilustra a continuación

Figura 2 Almacenamiento local de credenciales

Fuente: docs.fedoraproject.org

Características:

- Incluye un servidor central que permite realizar labores de administración.
- El software cliente es autónomo durante el proceso de autenticación, debido a que durante este proceso, la labor de administración se restringe a realizar monitoreo de los clientes.
- Las credenciales permanecen en el cliente.

Ventajas:

- Control centralizado de la configuración y monitoreo del software del cliente.
- Las labores de administración tienen un bajo grado de complejidad.

Desventajas:

- El hecho de almacenar las credenciales en el cliente hace que se deban tomar medidas de control de acceso y confidencialidad de la información.
- Una vez el cliente se ha conectado, el administrador del SSO sólo puede monitorear la conexión y no podría efectuar acciones de desconexión o cambio de configuración de la misma.
- La información entre el cliente SSO y el servidor no viaja cifrada.

2.8.4. Administración de credenciales centralizadas

La arquitectura SSO con administración y almacenamiento centralizado de credenciales (Figura 3) pretende solucionar los principales inconvenientes encontrados en la arquitectura que almacena las credenciales localmente.

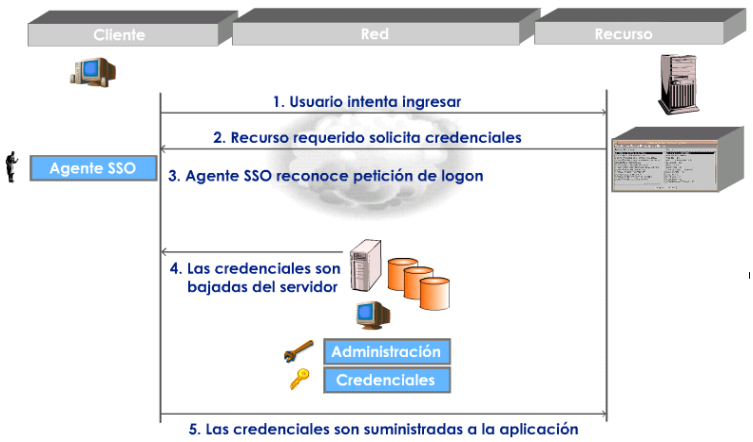

Figura 3 Almacenamiento de credenciales centralizado

Fuente: docs.fedoraproject.org

Características:

- Las credenciales son migradas a un servidor central, quien entrega las credenciales al cliente correspondiente en el momento de hacer el ingreso.

- El administrador determina la frecuencia con que se descargan las credenciales del servidor (Por sesión, por login, etc).

Ventajas:

- Permite a los usuarios el acceso a las aplicaciones desde cualquier estación, previa autenticación del mismo.
- Ofrece administración centralizada de credenciales disminuyendo posible manipulación de la misma en el cliente.

Desventajas:

- Se crea un único punto de falla, convirtiendo al SSO en un gateway para todos los recursos de la organización, ya que el servidor debe ser contactado cada vez que se realice un ingreso. El acceso a todas las aplicaciones de la organización depende del servidor central.
- La configuración carece de redundancia, recuperación entre fallas y respaldo.
- La información entre el cliente SSO y el servidor no viaja cifrada.

2.8.5. D. Arquitectura SSO totalmente distribuida

La arquitectura SSO totalmente distribuida (mostrada en la figura 4) se caracteriza principalmente por separar el servidor de la base de datos, lo cual la hace completamente modular. Esta arquitectura soluciona los problemas encontrados en las arquitecturas anteriormente presentadas y adicionalmente ofrece múltiples ventajas.

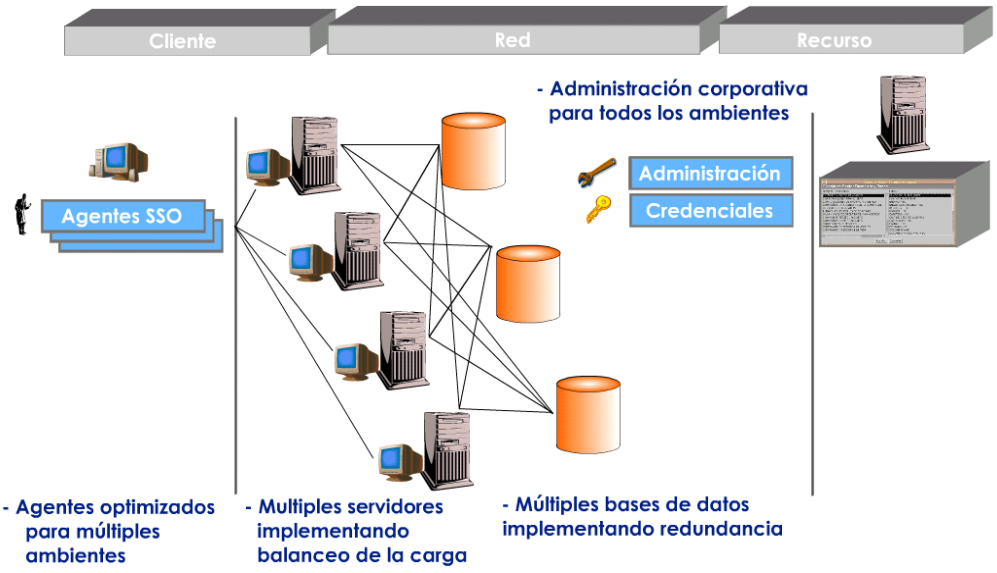

Figura 4 Arquitectura SSO totalmente distribuida

Fuente: docs.fedoraproject.org

Características:

- La información se accede en el momento de ingreso.
- Cuenta con SSOs avanzados que utilizan bases de datos escalables que soportan redundancia (i.e. SQL Server u Oracle).
- Las bases de datos se encuentran sincronizadas con el fin de lograr redundancia y respaldo.
- El proceso de ingreso ha sido migrado a un recurso de red. Siempre y cuando el agente SSO pueda establecer conexión IP a un servidor SSO, las credenciales podrán ser solicitadas y el ingreso podrá ser realizado. Además pueden ser almacenadas en memoria caché para realizar offline logon
- El servidor resulta ser una aplicación independiente que cuenta con un administrador diferente.

- La información es almacenada en bases de datos comerciales o en directorios de manera encriptada. Sin embargo, la información entre el cliente SSO y el servidor no viaja cifrada.

Ventajas:

- Los agentes SSO se encuentran optimizados para múltiples ambientes (Terminal Server, Web, Win32).
- Contiene múltiples servidores implementando balanceo de la carga para aumentar la disponibilidad y la atención de los requerimientos de autenticación.
- El hecho de contar con múltiples servidores adicionalmente hace que se disminuya la latencia (ver glosario) de la red.
- Contiene múltiples bases de datos sincronizadas, implementando redundancia.
- Permite realizar funciones de administración corporativa para todos los ambientes.

Desventajas:

- Solución altamente costosa por el ambiente distribuido requerido.
- Demorada implementación técnica por interacción entre múltiples sistemas operacionales.
- Soporte y administración complejos por la consideración anterior.

2.8.6. Administración con disponibilidad y redundancia

La arquitectura SSO con administración y almacenamiento centralizado de credenciales garantizando alta disponibilidad y redundancia (Figura 5) es una

adaptación de la arquitectura presentada en *C*, incorporando algunas de las ventajas de la arquitectura totalmente distribuida, presentada en *D*.

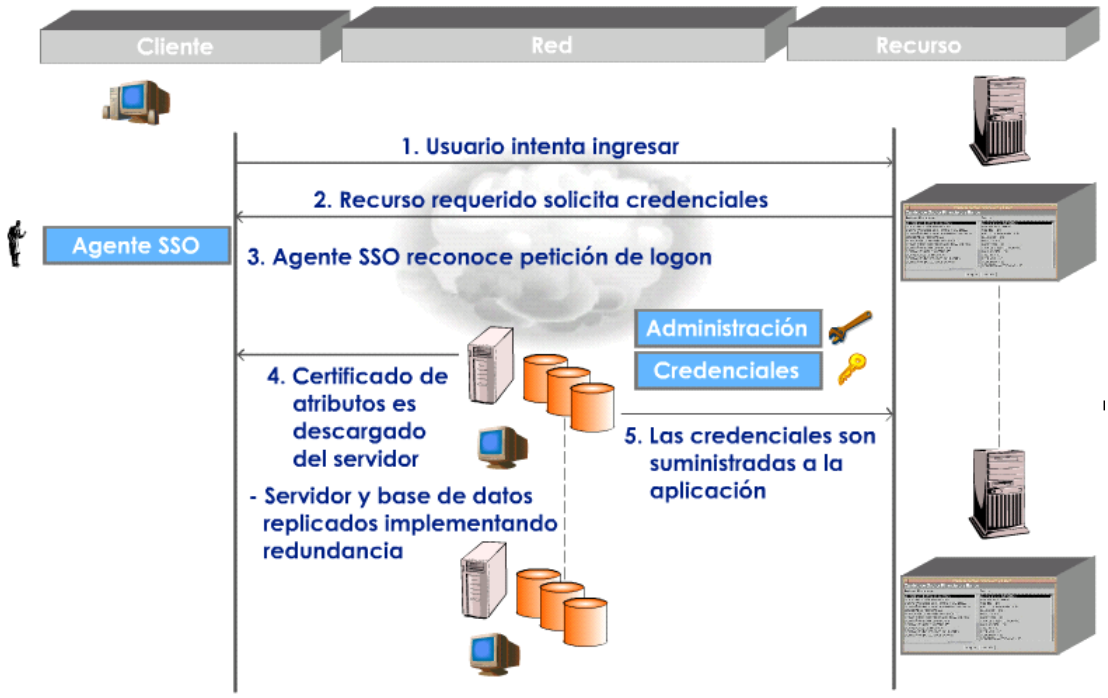

Figura 5 Administración y almacenamiento de credenciales

Fuente: docs.fedoraproject.org

Características:

- Las credenciales son almacenadas en un servidor central, quien entrega un certificado al cliente correspondiente, y las credenciales necesarias a la respectiva aplicación, en el momento de hacer el ingreso.
- Incorpora infraestructura replicada con el fin de manejar la contingencia y redundancia en tiempo real.
- Ofrece alta disponibilidad mediante software.

Ventajas:

- Permite a los usuarios el acceso a las aplicaciones desde cualquier estación.

- Ofrece administración centralizada.
- Su infraestructura duplicada permite implementar alta disponibilidad y redundancia.
- Tanto el hardware como el software se encuentran debidamente especificados para enfrentar una situación de contingencia.

Desventajas:

- La alta disponibilidad y redundancia que ofrece se basa en su infraestructura replicada, lo cual la hace costosa a nivel de hardware y software, como a nivel de administración y control de la misma.
- La información entre el cliente SSO y el servidor no viaja cifrada.

El sistema Single Sign-On por ejemplo soporta los siguientes protocolos:

- SAML 2.0
- CAS 1.0 y 2.0
- PAPI
- OpenSSO

Cada uno tiene sus particularidades, pero la descripción general de funcionamiento es la que consta a continuación:

Los usuarios reciben tokens, identificadores opacos que quedan asociados internamente a la identidad del usuario que hace uso del sistema.

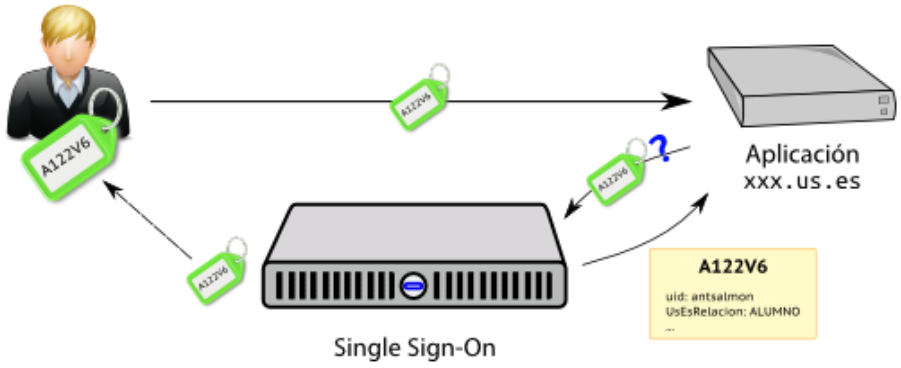

Figura 6 OpenSSO

Fuente: sic.us.es

Cuando un usuario se identifica usando el sistema, su navegador hace llegar a las aplicaciones su identificador opaco o, en algunos protocolos, su identidad completa cifrada y firmada.

Es por tanto tarea de las aplicaciones comprobar la presencia y validez de los tokens y atributos de identidad, consultando al sistema Single Sign-On si fuera necesario.

2.9. Modelo “Trusted Third Party”

2.9.1. Certificados digitales

El Certificado Digital es el medio por el cual se permitir garantizar técnicamente y bajo fundamentos legales la identidad de una persona en el Internet. Es indispensable para que las instituciones, públicas o privadas, para que puedan ofrecer servicios seguros a través de Internet;

El receptor de un documento firmado bajo un certificado digital puede tener la seguridad de que éste es el original y no ha sido manipulado y el autor de la firma electrónica no podrá negar la autoría de esta firma, tal y como hubiera sido ejecutado de manera física.

2.9.2. Claves Públicas

El titular del certificado debe mantener bajo su poder una clave privada y mantenerla bajo cuidado, ya que si ésta es sustraída, cualquiera podría suplantar la identidad del titular en la red. Al igual que cualquier identidad, está también pueden ser canceladas y ser nuevamente adquiridas bajo otros parámetros, como si de una nueva credencial física se tratase.

La clave pública forma parte de lo que se denomina Certificado Digital en sí, que es un documento digital que contiene la clave pública junto con los datos del titular, todo ello firmado electrónicamente por una Autoridad de Certificación, que es una tercera entidad de confianza que asegura que la clave pública se corresponde con los datos del titular.

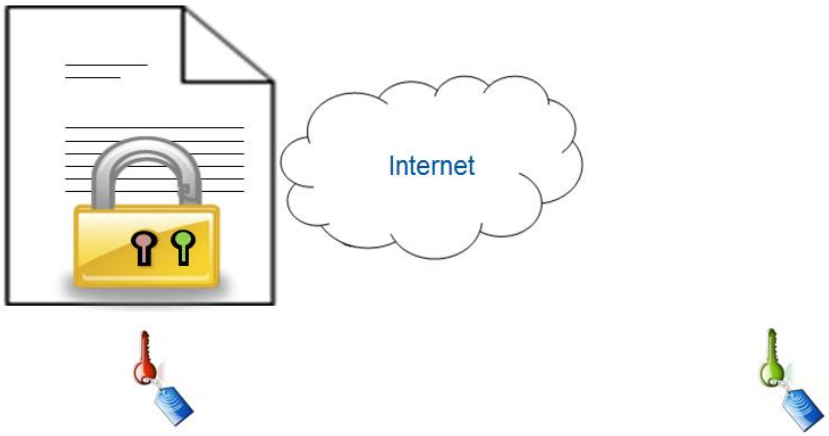

Figura 7 Modelo claves públicas y privadas

2.9.3. Terceras Partes de Confianza

El problema sobre un certificado viene al momento de definir su autenticidad, o veracidad del mismo; esta confianza es primordial en un entorno de clave público dado que todo el concepto se basa en la confianza que se coloca sobre ella.

Una metodología aplicada para poder llegar a confiar en un certificado proveniente de un usuario con el cual no se ha tenido ningún tipo de relación es mediante un agente externo, tercera parte de confianza.

Una TTP[3] es una autoridad de seguridad, por la cual las otras entidades encuentran valido las certificaciones con respecto a actividades relativas a la seguridad.

Están involucran diferente tipo de protocolos, permitiendo en medida que las entidades confíen unas con otras en especial en procesos de comunicación, dado que esta se comprometió a ejecutar tal y como el protocolo lo establece.

Las TTPs, permiten utilizar una colección de servicios de confianza, por ejemplo claves de cifrado público, y recuperación de estas; manejo de dinero electrónico, certificado de autenticidad de archivos y en casos judiciales también archivo y registro de evidencias.

2.9.4. Tipos de TTPs

Por la evolución global de la comunicación y la necesidade mecanismos de confianza han aparecido distintos tipos de TTPs, según la línea de negocio.

Por ejemplo se tiene:

- Autoridades de Certificación
- Autoridades de Registro
- Bancos Electrónicos
- Notarios Electrónicos
- Jueces Electrónicos

Esta variedad de servicios generan una gama de protocolos de seguridad, y que por lo general son manejadas por las reglas impuestas por la autoridad,

[3] Tercera arte de confianza

siendo esta aceptada por todas sin dudarlas; Sin embargo el problema aparece cuando se desea interactuar entre diferentes TTPs.

Una clasificación según el grado de participación de las TTPs en los protocolos de seguridad ayudan a mantener heterogeneidad en las comunicaciones; que gracias a esta se puede medir la capacidad de generación de confianza de los usuarios hacia las TTPs

2.9.5. Análisis de confianza

Una TTP es una organización con un grado de seguridad suficiente para que sea de confianza para otras organizaciones en relación con los servicios que presta. La relación de la confianza hacia esta puede ser tanto de los usuarios finales de una red telemática como con otras TTPs de diferente tipo. Sin embargo, esta relación puede ser distinta, es decir, el grado de confianza depositado en una TTP puede ser diferente y variar, dependiendo en cada caso del servicio de confianza proporcionado y la manera de dar soporte a este servicio.

El conjunto de criterios que pueden escoger para evaluar las TTPs tiene que ser reducido. Hay artículos que proponen sistemas de evaluación según un conjunto amplio de criterios en estos sistemas, sin embargo, la tabla de evaluación es difícil medir y para los usuarios es muy complejo analizarlo y compararlo con distintos servicios de confianza.

CAPÍTULO 3 INFRAESTRUCTURA SMART GRID

El problema real al concepto de lo que es una Grid es compartir recursos de forma coordinada y así poder resolver problemas en ambientes virtuales de naturaleza dinámica, y que formen parte más de una institución. La acción de compartir está referida al acceso directo a ordenadores, software, datos, y cualquier otro recurso, para solventar problemas en forma rápida y eficiente. Esta compartición tiene un alto control, por parte de los proveedores y consumidores o usuarios finales de los recursos, definiendo y aclarando lo qué se está compartiendo, también viendo quienes están autorizados a compartir y bajo qué condiciones se permite la acción de compartir. El conjunto de lo mencionado anteriormente determinados por las anteriores reglas, forman lo que se denomina una Organización Virtual.

3.1. Análisis del Hardware y Software para SMARTGRID

La paralelización una la técnica más utilizada en la computación de altas prestaciones y esta hace referencia a dos aspectos que son: la agregación de recursos hardware para que trabajen concurrentemente y la partición de las tareas software para que puedan ser asignadas a los diferentes elementos de proceso. Hoy en día, los procesadores de cualquier computador personal ya incorporan algún tipo de división funcional orientada al procesamiento concurrente. Tratan de vender la idea de que al poseer dos procesadores en el mismo chip se obtiene el doble de rendimiento. Por desgracia, en la actualidad todavía existen pocos programas comerciales que aprovechen esta característica. Entre los sistemas masivamente paralelos se encuentran diferentes arquitecturas: sistemas multiprocesador, computadores vectoriales y computadores matriciales.

Constituyen una única máquina o sistema fuertemente acoplado ya que todos los procesadores suelen compartir la memoria como recurso común. Para necesidades de computación menos extremas se cuenta con los clúster. Se clasifican dentro de los sistemas multicomputador, es decir aquellos donde se

agregan computadores completos que podrían trabajar independientemente. Son sistemas débilmente acoplados aunque físicamente las máquinas están próximas o incluso residen en el mismo bastidor. No comparten la memoria y, en consecuencia, la comunicación entre unidades de proceso se realiza mediante paso de mensajes. Existen clústeres llave en mano diseñados por fabricantes especializados pero también se han extendido los clústeres *off the shelf* que no son más que construidos por los propios usuarios finales siguiendo el esquema del Beowulf computing[4].

Su difusión es debida a dos factores: por un lado, la relación precio/rendimiento de los ordenadores personales es muy buena y, por otra parte, las redes de interconexión convencionales de hasta 1.000 Mbps permiten comunicar máquinas con una sobrecarga tolerable. La conectividad se puede montar sobre un protocolo común de comunicación como por ejemplo el estándar TCP/IP, y la ejecución concurrente de tareas se puede implantar utilizando un paradigma de programación basado en memoria distribuida y pasó de mensajes, como por ejemplo MPI[5]. Representan una alternativa viable a los supercomputadores aunque su precio también puede desbordar las posibilidades habituales.

El Grid computing representa un modelo emergente de computación. Proporciona una alta productividad gracias a que admite la agregación dinámica de cualquier recurso computacional que esté conectado a una red global de computación como puede ser Internet. Se comparten de manera distribuida recursos tales como capacidad de cómputo en grid: Su funcionamiento se basa en la instalación de una capa middleware que sirve de enlace entre los recursos

---

[4] Beowulf es un sistema de cómputo paralelo basado en clusters de ordenadores personales conectados a través de redes informáticas standard, sin el uso de equipos desarrollados específicamente para la computación paralela.

[5] Menssage Passing Interface

independientemente del hardware y del sistema operativo de cada máquina. En el plano teórico, un grid de computación es un sistema multicomputador pero aún más desacoplado que los clústeres ya que engloba máquinas pertenecientes a dominios administrativos diferentes. La idea original consiste en formar una infraestructura a nivel mundial que proporcione servicios a petición del usuario de manera completamente transparente a él.

En realidad la principal tarea o propósito de las tecnologías Grid es el de integrar juntamente con la posibilidad de virtualizar y gestionar los recursos o servicios dentro de Organizaciones Virtuales (VO) distribuidas, heterogéneas y dinámicas a través de diferentes organizaciones físicas, esto sin el problema que conllevaría integrar nuevos recursos.

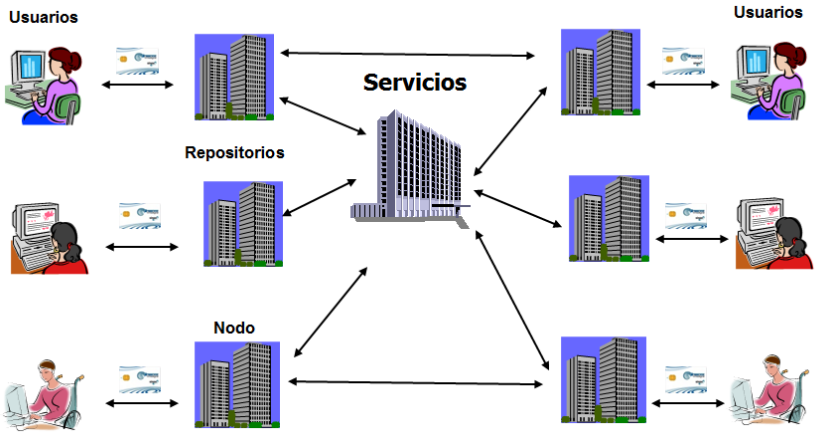

Figura 8 Organización VO

3.2. Networking

Es esencial para el uso efectivo de la *Grid computing,* la red deba tener la capacidad de ancho de banda necesaria para la comunicación interna. Los requisitos de ancho de banda serán dominados por la replicación de los datos

en la etapa de análisis del procesamiento de datos. Los experimentos del LHC[6] utilizaron en un principio herramientas de simulación, como OptorSim[7], para entender la "economía" de su modelo de computación. Las primeras estimaciones indicaron que la tasa de transferencia de datos podría aumentar en al menos un factor de 5 en un par de años. La experiencia con las transferencias de datos actuales de producción de datos indica que un problema habitual generalmente son los cuellos de botella en el sistema. Estos tienden a no estar relacionado con la red, pero son en general asociados con el enlace de las redes MAN[8]

La Grid debe optimizar todos los recursos bajo su disposición para así lograr obtener el mayor ancho de banda posible. La gestión de recursos incluye envió de trabajo remotamente, revisar frecuentemente el status cuando está en progreso de peticiones de recursos y obtener una salida satisfactoria al final de la ejecución. Cuando un proceso es solicitado y es enviado, los recursos disponibles se ponen a disposición a través del directorio de servicios

[6]Large Hadron Collider o LHC, traducido a El Gran Colisionador de Hadrones.
[7]OptorSim es un simulador de red diseñado para probar las estrategias dinámicas de replicación utilizados en la optimización de la ubicación de datos dentro de una cuadrícula
[8]Metropolitan Area Network o MAN, traducida a red de área metropolitana

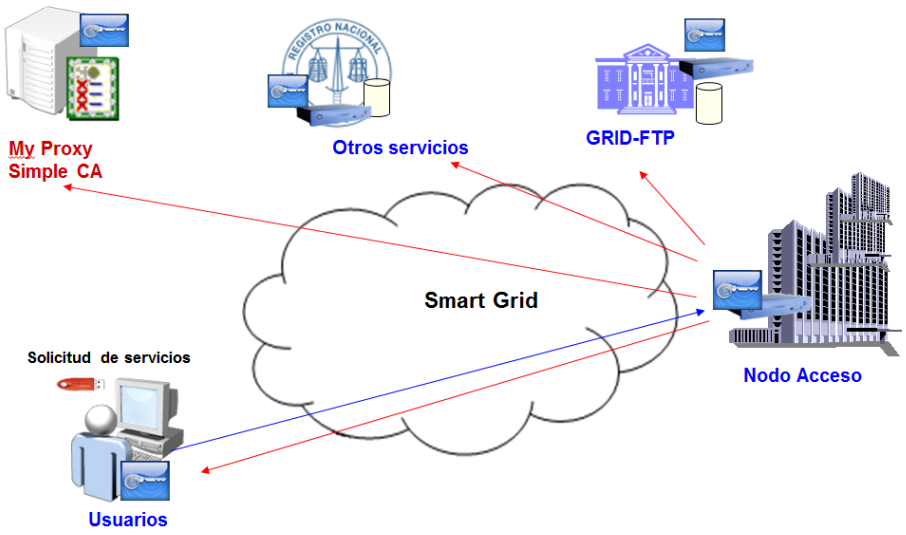

Figura 9 Esquema Red Grid

3.3. Arquitectura Grid

Se describe la arquitectura Grid en modo de capas, donde cada una cumple una función específica. Las capas altas de esta arquitectura se enfocan y acercan más a la relación directa con el usuario, mientras que las de nivel más bajo se aproximan al hardware, es decir a las computadoras y redes (Sun Grid Engine.).

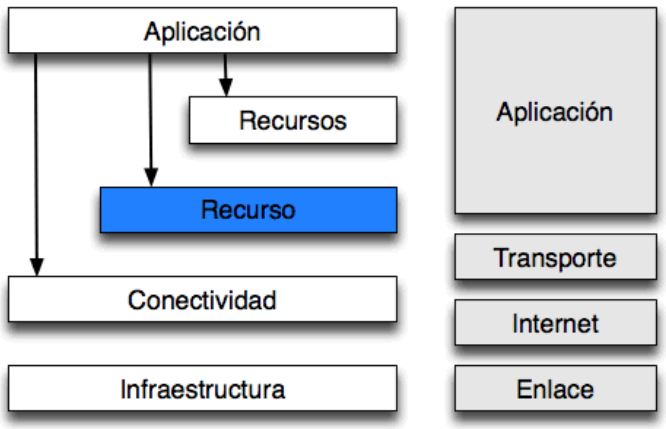

Figura 10 Esquema de Arquitectura Grid

3.4. Capa Fabric

Esta capa proporciona los recursos en los que el acceso compartido es a través de protocolos. Un recurso puede ser un sistema de archivo distribuido, un clúster de cualquier computador de la red o un pool de computadoras distribuido, en esos casos la utilización del recurso puede involucrar protocolos internos. La funcionalidad de esta capa permite la compartición de operaciones. Estas pueden ser detalladas bajo los siguientes recursos específicos detallados a continuación.

a) Recursos computacionales

Son mecanismos requeridos para iniciar los programas del computador, poder controlar y monitorear la ejecución de procesos solicitados por trabajos específicos.

b) Recursos alojados

Se requieren algunos mecanismos o procesos para el guardado y recuperación de archivos. También mecanismos para la lectura y escritura de archivos. El manejo de estos mecanismos permite el control sobre los recursos alojados, para transferencia de datos y otras funciones específicas.

c) Recursos de red

Mecanismos de manejo y control sobre los recursos y la transferencia en la red, funciones que provean las características de red.

d) Repositorios de recursos

Es una forma dedicada al almacenamiento de recursos que requiere mecanismos de manejo de versiones de fuentes y código de objetos.

e) Catálogos

Forma que está dedicada al almacenamiento donde están los mecanismos para implementar consultas y actualización de operaciones, por ejemplo el acceso a bases de datos relacionadas.

3.5. Capa Conectividad

La capa conectividad es el centro de todas las comunicaciones y protocolos de autenticación en cualquier red y en sus trabajos requeridos para las transacciones en la red Grid. Los Recursos y todos los protocolos de conectividad mantienen funcionando cada una de las transacciones específicas de Grid entre diferentes computadoras y otros recursos solicitados. La red empleada por GRID es Internet ya que esta es la misma red usada por la web y varios servicios.

Esta capa define el todas las comunicaciones y autenticación de protocolos requeridos para transacciones específicas de Grid dentro de la red. Los protocolos de comunicación generan el intercambio de todos los datos entre los recursos de la capa Fabric que se solicitaron. Los protocolos de autenticación construyen la comunicación de servicios para proveer mecanismos seguros para la verificación e identificación de usuarios y recursos.

Para la autenticación de algunos entornos se puede tener características de Log In, Delegación e Integración con algunas soluciones de seguridad. Las soluciones de la Grid en lo que se trata de seguridad se puede interoperar con varias soluciones locales como por ejemplo usuarios basados en las relaciones confiables en el que el sistema de seguridad no debe requerir que cada proveedor de recursos coopere sino los necesarios.

3.6. Capa Recurso

Esta se encuentra sobre la capa de Conectividad, con protocolos de Comunicación y Autenticación con los cuales se podrá definir negociaciones seguras, iniciación, monitoreo y control sobre recursos. Esta implementación llama a las funciones de capa Fabric para acceder y controlar los recursos. Los protocolos en esta capa se refieren a recursos individuales. Existen dos tipos de protocolos que se hablaran en esta capa:

3.7. Protocolo de información

Este se usa para obtener información estado de los recursos y su estructura.

3.8. Protocolos de manejo

Este se usa para negociar el acceso a un recurso que esta compartido, especificando sus requerimientos y operaciones para así poder optimizar estos recursos en base al requerimiento, como ser la creación de procesos o acceso a datos del trabajo a realizar. Ya que el manejo de protocolos es el responsable de la inicialización de todas las relaciones de compartir recursos disponibles.

3.9. Capa Colectiva

A esta capa se la denomina capa colectiva por que los componentes de capa colectiva construyen sobre la capa Recursos y la capa Conectividad, y con esto implementar una amplia variedad de comportamientos para el sharing sin la localización de nuevos requerimientos solicitados sobre los recursos almacenados. Por ejemplo:

a) Servicios de directorio

Este permite a varios participantes de las OV descubrir la existencia y/o propiedades de los recursos de OV. Permite a sus usuarios hacer consultas para recursos por el nombre y/o por atributos, como ser tipo, disponibilidad o carga.

b) Co-Allocation, Scheduling y servicios brokering

Permiten a recursos de participantes de las OV averiguar sobre más recursos para propósitos específicos y para la programación de tareas sobre los recursos apropiados.

c) Monitoreo y diagnóstico de servicios

Soportan el monitoreo de los recursos de las OV para fallas, ataques de adversarios, detección de intrusos, sobrecarga y más.

d) Servicios de replicación de datos

Soporta el manejo de almacenamiento de los recursos pertenecientes a las OV para maximizar el rendimiento en los accesos a datos con las respectivas métricas como el tiempo de respuesta, costo, etc.

e) Sistemas de programación GRID-ENABLED

Permite usar en los entornos GRID modelos de programación familiar usando varios servicios Grid para localizar recursos, seguridad, etc.

f) Sistemas de manejo de Workload y Collaboration Frameworks

Conocidos como entornos de solución de problemas“PSE´s”, provee para la descripción, uso y manejo de multi-steps, asíncronos, flujos de trabajo multicomponentes, etc.

g) Sistemas de ubicación de software

Descubre y selecciona las mejores implementaciones de software y ejecución de plataformas basadas en parámetros del problema a resolver. Ejemplo: NetSolve y Ninf.

h) Servidor de autorización de la comunidad

Refuerza el gobierno de las políticas de acceso a recursos generando capacidades que los miembros de la comunidad pueden usar para el acceso a recursos comunitarios.

i) Servicios de cuentas de la comunidad y pagos

Junta información acerca del uso de información para el propósito de manejo de cuentas, pagos, y/o limitación de usos de recursos a miembros de la comunidad.

j) Servicios de colaboración

Soporta el intercambio coordinado de información dentro de comunidades de usuarios potencialmente grandes. Mientras los protocolos de capa Recursos deben ser generales en naturaleza y desarrollados ampliamente, los protocolos de capa Collective expanden su espectro desde propósitos generales a aplicaciones altas o dominios específicos sólo entre OV específicas. Las funciones de esta capa pueden implementarse como servicios persistentes con protocolos asociados o SDK`s designados para enlazarse con ciertas

aplicaciones. En ambos casos sus implementaciones pueden construir en capa Recurso protocolos y API´s.

Los componentes de capa Collective pueden crearse para requerimientos de usuarios de comunidades específicas, OV, o dominios de aplicaciones como por ejemplo un SDK que implementa protocolos de coherencia de aplicaciones específicas, o un servicio co-reservado para un conjunto específico de recursos en la red. Otros componentes de esta capa pueden ser de propósitos más generales por ejemplo, servicios de replicación que manejen una colección internacional de sistemas de almacenamiento para múltiples comunidades o un servicio directorio designado a permitir el descubrimiento de OV´s.

Los servicios colectivos (collective services) se basan en protocolos: protocolos de información que obtienen datos sobre la estructura y estado de los recursos, y protocolos de manejo que negocian el acceso a recursos de una forma uniforme.

3.3. Seguridad en una SMARTGRID

La interconexión de sistemas informáticos de cualquier tipo de procesos empresariales con sistemas de control, se puede tomar como un indicio de que las fallas de seguridad de los sistemas informáticos tradicionales (Windows, Linux, Unix, protocolos TCP/IP, etc.) Impactarán los sistemas de control que hasta la fecha se encuentran centralizados o aislados.

La smartgrid se puede tomar como una recopilación de gran cantidad de datos sobre los cuales se toma en base al funcionamiento de la red y la administración remota de todos sus componentes, permitiendo una gestión más eficiente. Pero la Smart Grid también implica una serie de nuevos riesgos para el sector. Especialmente cuando existen riesgos de ciberataques malintencionados para adquirir cualquier tipo de información.

La Smart Grid es un objetivo muy admirado para los ataques, que podrían ser entre otros trabajadores descontentos, agencias de inteligencia extranjeras, crimen organizado, extorsionadores, terroristas, etc.

Estos grupos de inicio no tienen el conocimiento de cuales vulnerabilidades pueda tener una y probablemente van a atacar los puntos más débiles de la infraestructura para intentar alcanzar sus objetivos si es que logran encontrarlos. (Knapp & Samani, 2013)

Los accesos desde la red corporativa de cualquier empresa siempre pone o expone estas redes son muy grandes y difíciles de controlar en su totalidad. Existen usuarios conectando desde varias partes del mundo, oficinas repartidas físicamente, distintos tipos de accesos a Internet, conexiones con clientes y proveedores, etc. Estas redes suelen contar con varios métodos de seguridad pero también han sido el objetivo tradicional por parte de los atacantes y por lo tanto es donde estos cuentan con mayor facilidad de atacar. Además, en muchos casos no se realiza un correcto control de accesos hacia los sistemas empresariales que permiten operar las distintas infraestructuras.

Con el nacimiento de las redes inteligentes, aplicar, mantener y gestionar la seguridad de estas redes se convierte en el principal objetivo cualquier empresa de contenga información valiosa.

Por eso para las seguridades en SmartGrid se debe tener como objetivo principal el descubrir y analizar los mecanismos de seguridad para contrarrestar cualquier tipo de amenazas y vulnerabilidades que existen y futuras en los ambientes SMARTGRID. Por lo cual se debería tener en cuenta que para cualquier implementación tomar en cuenta lo siguiente:

- Analizar las existentes iniciativas y estándares de seguridad en la SMARTGRID.
- Estudiar las arquitecturas de seguridad en el ámbito de la SMARTGRID.
- Analizar los protocolos de red y elementos de red inteligentes.
- Analizar las metodologías de organización y de la seguridad en la SMARTGRID.
- Utilizar herramientas de seguridad (Wireshark, Sniffing, etc.) para el estudio de los posibles ataques.

Como beneficios de la seguridad en una SmartGrid después de completar cualquier implementación en la empresa se debería ser capaz de:

- Identificar y definir los conceptos de Ciber seguridad Empresarial y Protección de Infraestructuras Críticas aplicados en el ámbito de la REDES INTELIGENTES o SMARTGRID.
- Descubrir y analizar el estado del arte de las Infraestructuras Críticas a nivel de poder solventar o bloquear cualquier ataque.
- Discernir tipos de tecnología y madurez de la seguridad para diferentes topologías y entornos en la SMARTGRID

La adquisición de información no es el único objetivo que puede llamar la atención. El año pasado diversos medios publicaron detalles técnicos de Stuxnet, un gusano cuyo propósito era atacar un conjunto definido de sistemas de control industrial, conocidos en inglés como Supervisory Control and Data Acquisitionsystems, SCADA. Los investigadores de Symantec señalan que el gusano se valía de varias vulnerabilidades, incluyendo una vulnerabilidad desconocida hasta el momento (zero-dayexploit) para ingresar a un sistema y propagarse, además incorporaba técnicas sofisticadas para evadir programas antivirus. Una vez el gusano entraba en un sistema, era capaz de reprogramarlo ocasionando daño físico a algunos componentes. Stuxnet demuestra que un programa malicioso puede ocasionar daño físico a algunos elementos del mundo real. Aunque los expertos en programas maliciosos están de acuerdo en que Stuxnet es sumamente sofisticado y seguramente no se produzca un programa malicioso de calidad semejante con frecuencia, el riesgo es real, un gusano con las características de Stuxnet podría alterar o dañar componentes físicos de una red inteligente. (Knapp & Samani, 2013)

Aunque la información que se registra por cada usuario, esta información puede ser usada con fines diferentes. En el caso de las redes inteligentes, la información describe el comportamiento del usuario o de los usuarios a quienes pertenece. A partir de esto, es posible obtener información privada, como el

número aproximado de personas que viven en un hogar, la hora a la que salen, la hora a la que regresan y en algunos casos, y mucha información personal.

Para todas estas inseguridades en un grid, siempre depende de los mecanismos de protección que se tomen para estar protegidos y por eso muchas industrias y universidades en diferentes países adelantan proyectos de investigación y desarrollo para mejorar la tecnología de las redes inteligentes. Tales proyectos incluyen los diseños de medidas de seguridad inteligentes con coprocesadores eficientes para cifrar y descifrar la información, tratando de mantener siempre esta seguridad segura, desarrollo de herramientas que las empresas usen en seguridades inteligentes y así puedan usarla para adelantar evaluaciones de penetración de forma semiautomática y algoritmos para ajustar el perfil de seguridad, de tal manera que resuma la información que el operador de distribución necesita, mientras elimina las características que revelan detalles del comportamiento del usuario.

3.4. Middlewares en una SMARTGRID

La smart grid presenta características muy similares a las aplicaciones distribuidas, siendo la heterogeneidad y la escalabilidad algunas de las cuestiones más relevantes. La aplicación de middlewares de comunicaciones distribuidos orientados a objetos ayuda en gran medida a resolver muchos de los problemas presentados en otros esquemas similares. (Hernández & Moltó, 2006)

3.4.1. Globus Toolkit

El código abierto Globus Toolkit es una tecnología que permite fundamental dejar que la gente comparta recursos y así aumentar la potencia de cálculo, procesamiento de bases de datos y otras herramientas online de forma segura a través de límites corporativos, institucionales y geográficas sin sacrificar la autonomía local. Este incluye los servicios de software y bibliotecas para el control de los recursos, además de la seguridad y la gestión de archivos. Además de ser una parte central de los proyectos de ciencia e ingeniería, el

Globus Toolkit es un soporte sobre el cual las principales empresas de TI están construyendo productos comerciales significativos.

El Globus Toolkit incluye software para la seguridad, la infraestructura de la información, gestión de recursos, gestión de datos, la comunicación, la detección de fallos, y la portabilidad. Se empaqueta como un conjunto de componentes que se pueden utilizar de forma independiente o en conjunto para desarrollar aplicaciones. Cada organización tiene modos singulares de operación, y la colaboración entre varias organizaciones se ve obstaculizada por la incompatibilidad de los recursos tales como archivos de datos, computadoras y redes. El Globus Toolkit fue creado para eliminar los obstáculos que impiden toda colaboración entre recursos sin problemas. Sus servicios, interfaces y protocolos permiten a los usuarios acceder a recursos remotos como si estuvieran situados dentro de su propio cuarto de máquinas, preservando al mismo tiempo el control local sobre quién puede usar los recursos y cuándo. (Sotomayor & Childers, 2006)

El Globus Toolkit ha crecido a través de una gran estrategia y sea creado para ser de código abierto similar al sistema operativo Linux, y distinto de los intentos de propiedad de software de intercambio de recursos. Esto anima más a una adopción más rápida y conduce a una mayor innovación técnica, ya que la comunidad de código abierto proporciona mejoras continuas al producto.

Algunos de los paquetes con los que cuenta el Globus TOOLKIT vienen ya configurados para el uso del mismo, esto permitiendo tener los que es compartición de archivos, seguridades, administración.

Estos paquetes son:

- **globus-gridftp:** GridFTP cliente y herramientas de servidor, para compartir archivos.
- **globus-gram5:** GRAM5 cliente y herramientas de servidor, para compartir archivos y documentos.

- **globus-gsi:** Herramientas Infraestructura Globus de seguridad para la gestión de certificados y proxies.
- **globus-gestión-servidor de datos:** Herramientas de servidor para el despliegue de un servidor GridFTP.
- **globus-gestión-cliente de datos:** Herramientas de cliente para la gestión de datos, incluidos los programas cliente GridFTP y globus-url-copy
- **globus-data-gestión-sdk:** Cabeceras de desarrollo y documentación para escribir aplicaciones que utilizan las API GridFTP.
- **globus-recursos-gestión-servidor:** Herramientas de servidor para el despliegue de un administrador de recursos GRAM5
- **globus-recursos-gestión-cliente:** Herramientas de cliente para la gestión de recursos, incluyendo la herramienta globusrun, y los * Herramientas-globus trabajo-.
- **globus-recursos-gestión-sdk:** Cabeceras de desarrollo y documentación para escribir aplicaciones que utilizan las API GRAM5.

La Instalación y funcionalidad de este se lo vera en el Capítulo 4 donde se explica la Implementación de FTP y seguridades.

CAPÍTULO 4 IMPLEMENTACIÓN GSI

Antes de empezar a implementar la infraestructura de la tecnología Grid, se deseaa definir el concepto fundamental que aparecerá a lo largo del capítulo y es necesario dejarlo claro para poder entender la filosofía de la Smart Grid. El problema real que subyace al concepto Grid, es como prestar seguridad al momento de compartir recursos de forma coordinada y como se resuelve esta problemática en organizaciones virtuales de naturaleza dinámica.

Dicho de otro modo, el corazón de la Grid será la adecuada y controlada forma de autentificar a los usuarios de nuestra Smart Grid, y para eso, en este capítulo se analizara las configuraciones necesarias propias de las herramientas provistas por parte de Globus Toolkit como son el simpleCA y Myproxy.

4.1. Globus Toolkit

Para el desarrollo de este proyecto, se ha utilizado dos máquinas virtuales bajo un sistema operativo Ubuntu 12.02.

Se ha denominado al servidor principal `protel-VirtualBox`, y al cliente `protel2-VirtualBox`.Cada uno de los comandos ejecutados para las configuraciones se han dividido en grupos con las siguientes nomenclaturas:

Comandos de usuario root:

root@protel-VirtualBox:/#

root@protel2-VirtualBox:/#

Los comandos que inicien con la nomenclatura descrita anteriormente, deben ser ejecutados como súper usuarios o usuarios root, en el servidor o cliente

Comandos de usuario myProxy:

$

Los comandos con el signo numeral, deben ser ejecutados con el usuario `myproxy`, en el servidor protel-VirtualBox. Este usuario es creado automáticamente al momento de instalar el paquete myproxy-server

Comandos de usuario común:

protel@protel-VirtualBox%
protel@protel2-VirtualBox%

Los comandos como los ejemplos anteriores deben ser ejecutados con un usuario normal, ya que pretenden interactuar con los servicios propios del Globus Toolkit, sean en el cliente o en el servidor principal. Para este ejemplo se ha usado los usuarios `protel y protel2`, del servidores y clientes respectivamente, pero estos pueden ser utilizados por cualquier usuario de los equipos conectados a la Grid.

4.2. Requisitos previos

A continuación se detallan las herramientas necesarias como requisito para la implementación de Globus Toolkit.

4.2.1. Instalación Virtual Box

Virtual box es un programa de virtualización capaz de instalar en el ordenador cualquier sistema operativo de forma “virtual”. Esta herramienta es ideal para empezar a conocer nuevos sistemas operativos y probar aplicaciones de software sin alterar el sistema. El software se encuentra en forma totalmente gratuita desde su página Oficial (Virtual Box, 2006).

En primer lugar se instala Virtual Box en el computador, para esto se ejecuta el instalador del programa y se abrirá la primera pantalla tal como aparece en la figura 11.

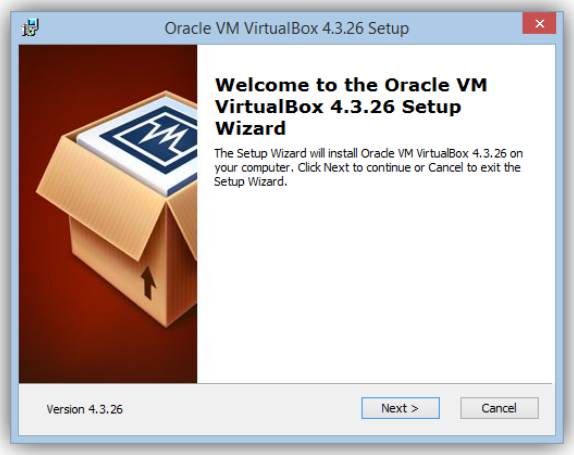

Figura 11 Pantalla de instalación Virtual Box

Fuente: www.VirtualBox.com

La pantalla de bienvenida informa y sobre la instalación de Virtual Box en el sistema, se pulsa en "*Next*" y en la siguiente pantalla se debe elegir los componentes y la ruta donde se lo instalara, además se solicitara seleccionar si se desea crear accesos directos.

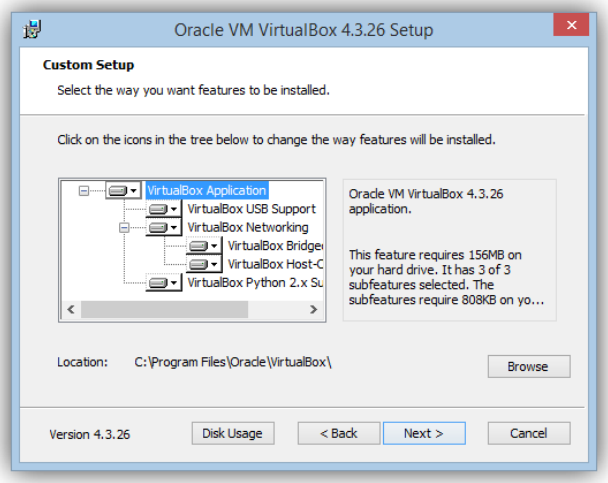

Figura 12 Pantalla de selección de componentes.

Fuente: www.VirtualBox.com

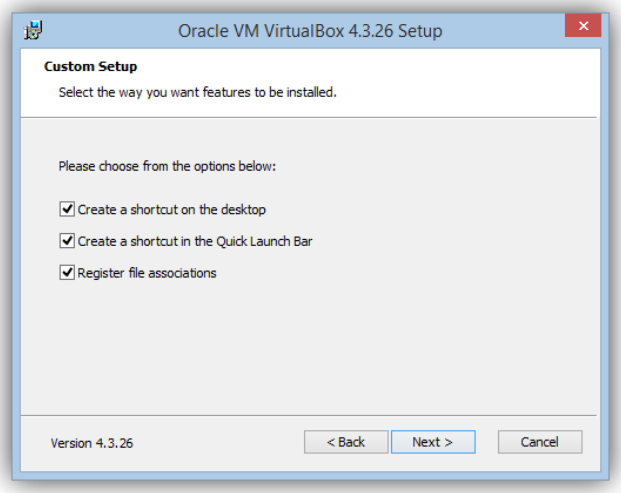

Figura 13 Pantalla de selección de accesos directos.

Fuente: www.VirtualBox.com

La siguiente pantalla advertirá sobre la instalación de varios archivos necesarios y solicitara reiniciar las tarjetas de red. Para finalizar el programa de instalación, se pulsa en "*Yes*" y se espera la confirmación de finalización.

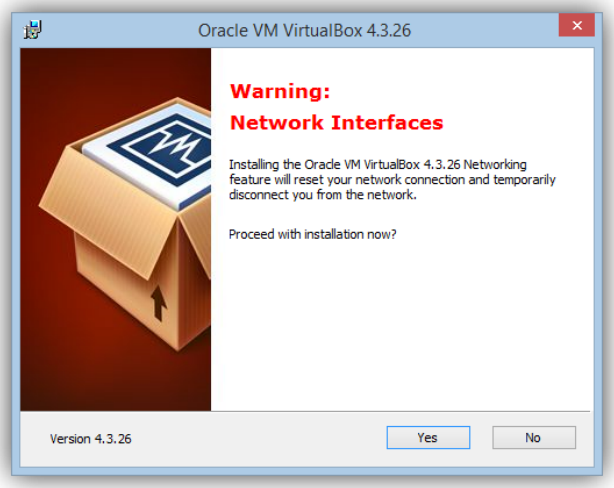

Figura 14 Pantalla advertencia de instalación.

Fuente: www.VirtualBox.com

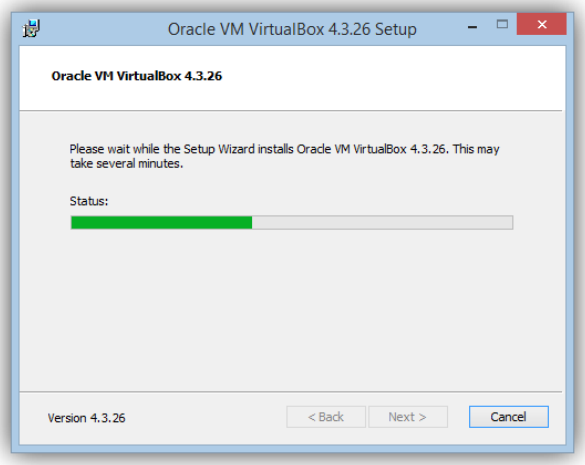

Figura 15 Pantalla status de instalación.

Fuente: www.VirtualBox.com

Una pantalla notificara el éxito de la instalación, y si se desea iniciar Virtual Box, en este momento, se pulsa, "*Finish*", y se continua a la pantalla principal.

Figura 16 Pantalla finalización de instalación.

Fuente: www.VirtualBox.com

4.2.2. Instalación sistema operativo

Virtual Box, posee una interfaz sencilla, que permite instalar un nuevo sistema operativo de manera ágil. Virtual Box brinda un asistente que guiara en el proceso de paso a paso.

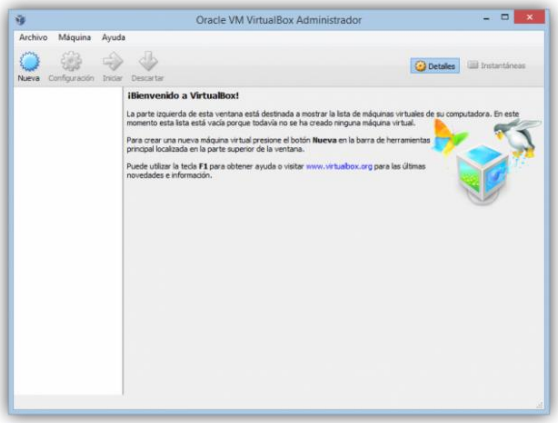

Figura 17 Pantalla principal Virtual Box.

Fuente: www.VirtualBox.com

Para empezar se pulsa en "*Nueva*" y el asistente guiara en la instalación. Se procede a colocar un nombre al sistema que va a instalar y seleccionar el sistema operativo que se va a utilizar. Se continua dando clic en el botón "*Next*".

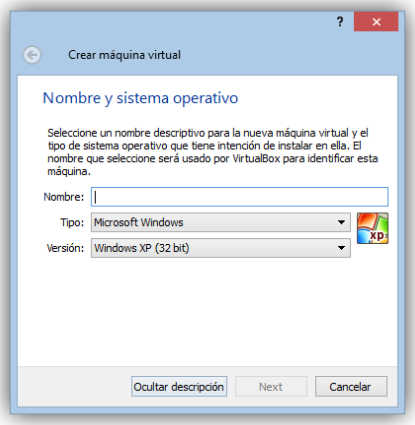

Figura 18 Pantalla selección de versión de sistema operativo.

Fuente: www.VirtualBox.com

La pantalla siguiente permite asignar una cantidad de memoria RAM que se asignara a la máquina virtual. En este caso se ha dejado el tamaño predefinido. El asistente permite crear un disco nuevo o utilizar uno ya existente.

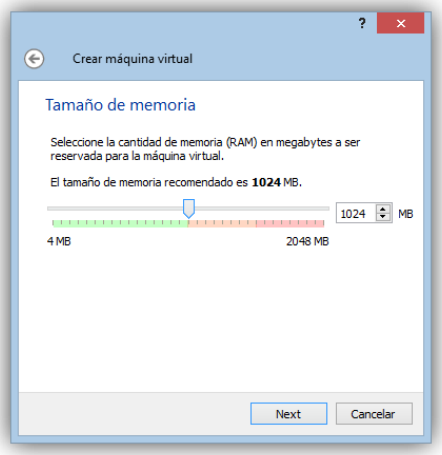

Figura 19 Pantalla asignación de memoria RAM

Fuente: www.VirtualBox.com

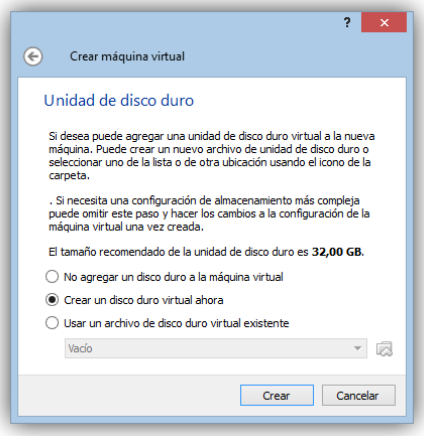

Figura 20 Pantalla Selección de Disco Duro

Fuente: www.VirtualBox.com

Al pulsar en "*Crear*" se brindara un resumen de las características seleccionadas, y estas pueden ser modificadas, o finalizar la creación mediante el botón "*Crear*"

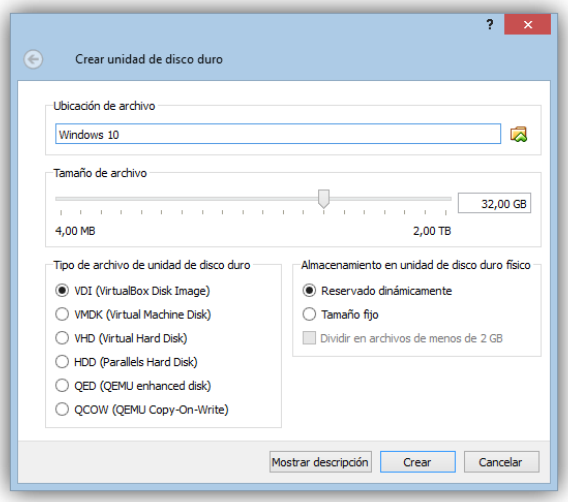

Figura 21 Pantalla Resumen de creación de máquina virtual.

Fuente: www.VirtualBox.com

Una vez finalizada la creación, se agregara el icono correspondiente, y en el botón Iniciar se oprime *ejecutar,* para iniciar nuestra máquina virtual previamente instalada.

Figura 22 Pantalla Virtual Box con la máquina virtual instalada.

Fuente: www.VirtualBox.com

4.2.3. Configuración de Red

Virtual box permite escoger entre los siguientes modos de conexión:

- ***No conectado***: Virtualbox muestra un adaptador de red pero sin conexión. (Cable desconectado).
- **Network Address Translation (NAT):** Permite funcionalidad básica desde el sistema operativo Huésped. Navegar por internet acceder al correo, descargar ficheros.
- **Adaptador puente:** Simula una conexión física real a la red, asignando una IP al sistema operativo huésped. Esta IP se puede obtener por DHCP o directamente configurándola en el Sistema Operativo huésped.
- **Red interna:** Similar al Adaptador puente, se puede comunicar directamente con el mundo exterior con la salvedad de que ese mundo exterior está restringido a las máquinas virtuales conectadas en la misma red interna. Esta limitación viene justificada por seguridad y velocidad.
- **Adaptador sólo-anfitrión:** Es una mezcla entre los tipos "Adaptador puente" e "interna". Por defecto se tiene configurado el modo NAT.

Para el caso, esta configuración no es la más adecuada, ya que no se tiene "visibilidad" del sistema operativo huésped desde el sistema operativo anfitrión y resulta muy útil tener una IP diferente para cada uno. Se elige la 2ª opción (Adaptador puente).

En el campo nombre se selecciona la interfaz que se va a utilizar: Ethernet, Wifi,etc. Es importante establecer que este dispositivo debe estar conectado para el correcto funcionamiento de la red entre el sistema operativo anfitrión y el huésped.

Figura 23 Pantalla configuración red Virtual Box

4.2.4. Configurar red de las máquinas virtuales

Para configurar el acceso que el Sistema operativo virtualizado tiene sobre red, se lo selecciona en la pantalla principal del Virtual Box y se pulsa sobre Configuración.

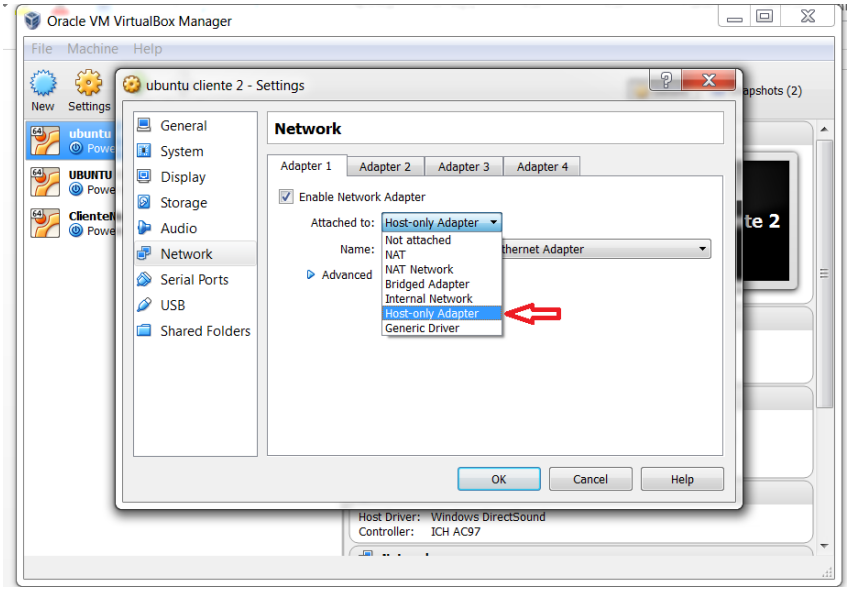

Figura 24 Pantalla configuración red Ubuntu

En la ventana de configuración se selecciona la pestaña Red. En Adaptador 1, se selecciona *Habilitar adaptador de red.*

4.2.5. Configuración PC físico - PC Virtual

Para comprobar que las máquinas virtuales y el computador se encuentren en la misma red se utiliza Host Only-Adapter, que sirve para manejar una red interna solamente entre la maquina física y la máquina virtual. Al seleccionar esta opción se debe poner IPs a las máquinas para que estén en una misma red. Por ejemplo a la máquina virtual se coloca la IP: 192.168.0.10. y a la maquina física se coloca la IP: 192.168.0.11, Esta IP debe ser colocada la tarjeta de red del virtual box.

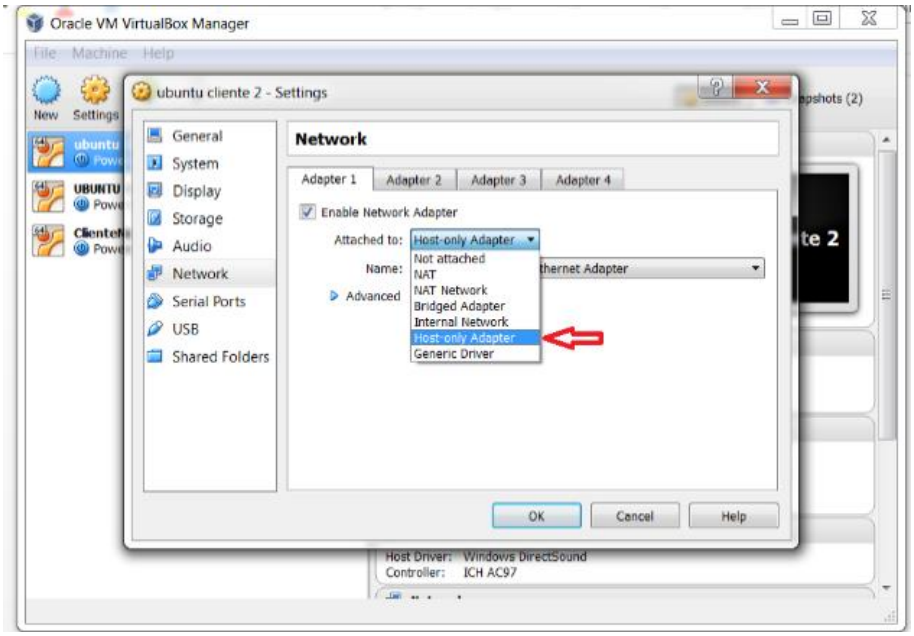

Figura 25 Configuración red PC Física

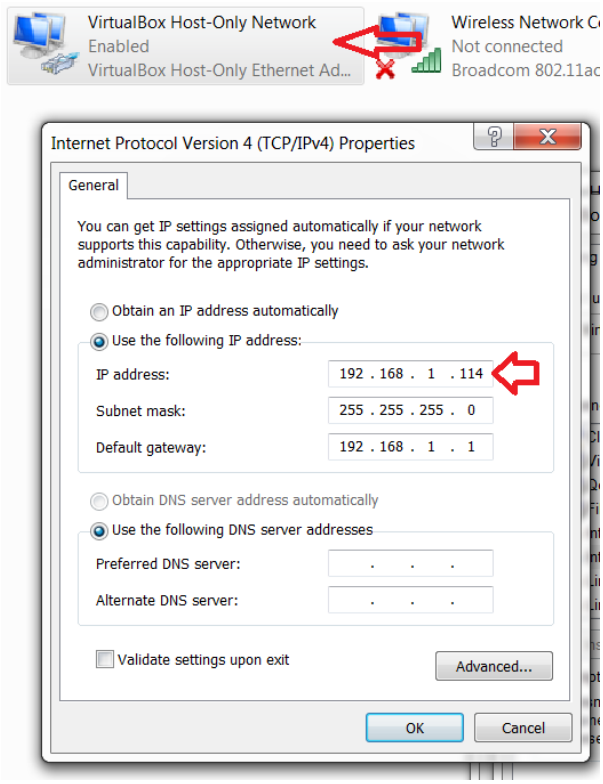

Figura 26 Configuración red PC Virtual

Esto brinda conectividad entre los dos equipos, y se puede replicar cuantas veces sea necesario. Una vez completada la configuración se continúa con la instalación en cada uno de los computadores virtuales, y trabajar directamente con ellos. Por facilidad de trabajo se ha utilizado una herramienta de acceso remoto llamada *MobaXter,* para el control remoto vía SSH de los nodos clientes y servidores, este no necesita instalación, y simplemente se necesita descargar de la página oficial *http://mobaxterm.mobatek.net/download-home-edition.html.*

La facilidad que brinda este software se acomoda perfectamente a las necesidades del proyecto, dado que el mantener el contro de varios computadores virtuales en una única pantalla de procesos agilitara los siguientes procedimientos que se detallan a continuación.

4.2.6. Pre requisitos Globus ToolKit

La distribución de Globus ToolKit se distribuye como un RPM[9], y paquetes Debian para los sistemas Linux. Para instalar la versión correspondiente a nuestra versión de Ubuntu, se debe ejecutar el comando:

```
root@protel-VirtualBox:/# wget http://
www.globus.org/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repository-
5.2-stable-precise_0.0.3_all.deb
```

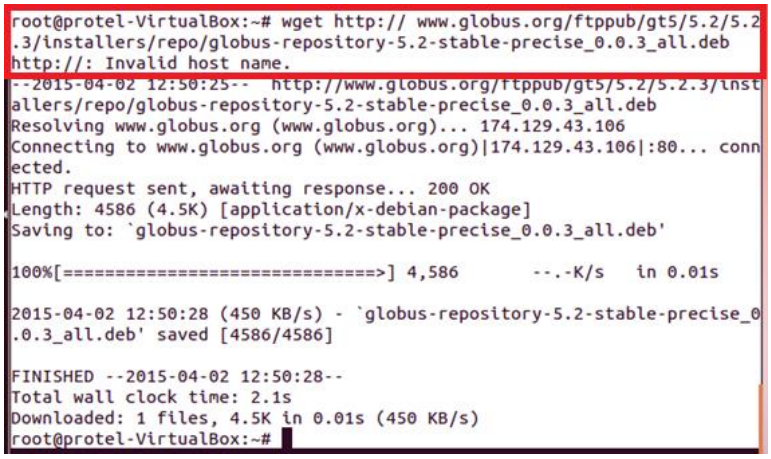

Figura 27 Comando para obtener Globus Toolkit del repositorio

```
root@protel-VirtualBox:/# dpkg -i globus-repository-5.2-stable-
precise_0.0.3_all.deb
```

[9] RPM, originalmente llamado Red Hat Package Manager, pero se convirtió en acrónimo recursivo

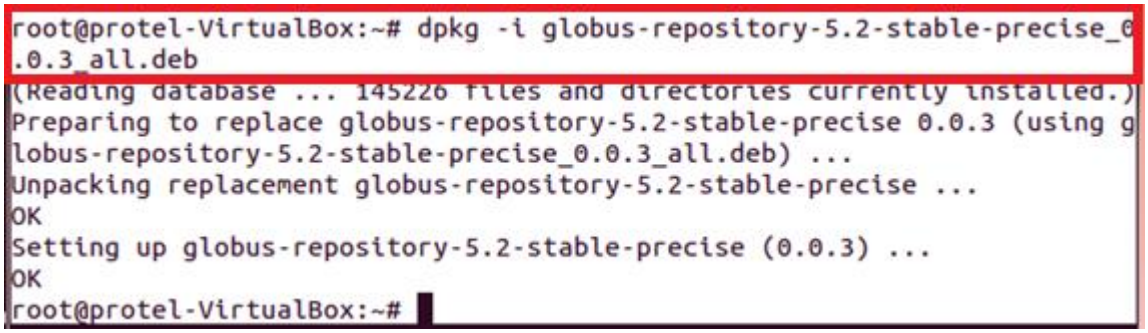

Figura 28 Instalación de paquete GlobusToolkit

Antes de realizar las actualizaciones se procede a entrar en la ruta **`/etc/apt/source.list`** y se comenta las líneas del documento que inician con la palabra **`deb-src`**, solamente agregando el símbolo # al principio de las líneas.

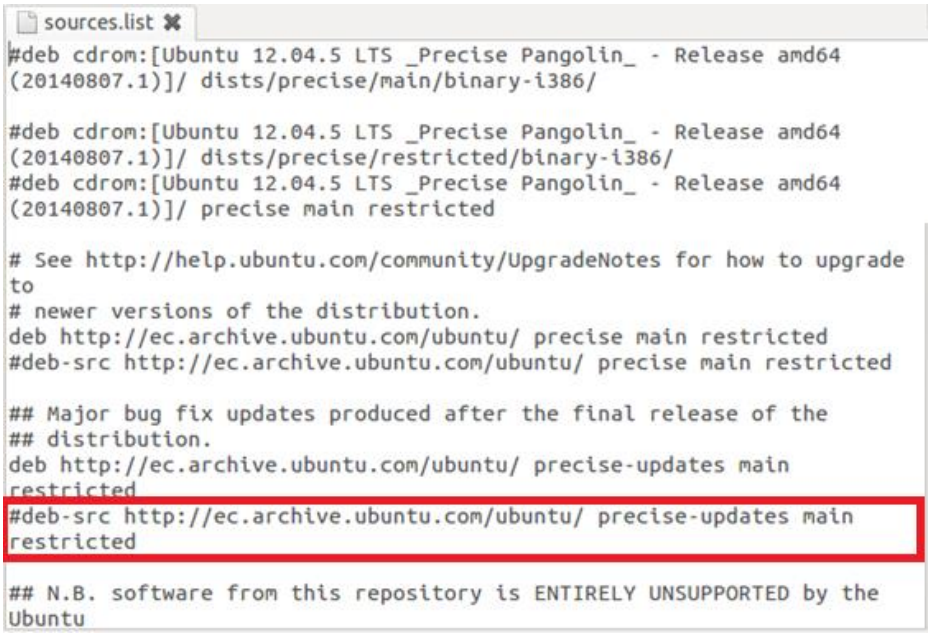

Figura 29 Archivo source.list

Luego se procede a realizar una actualización completa con el comando:

```
root@protel-VirtualBox:/# apt-get update
```

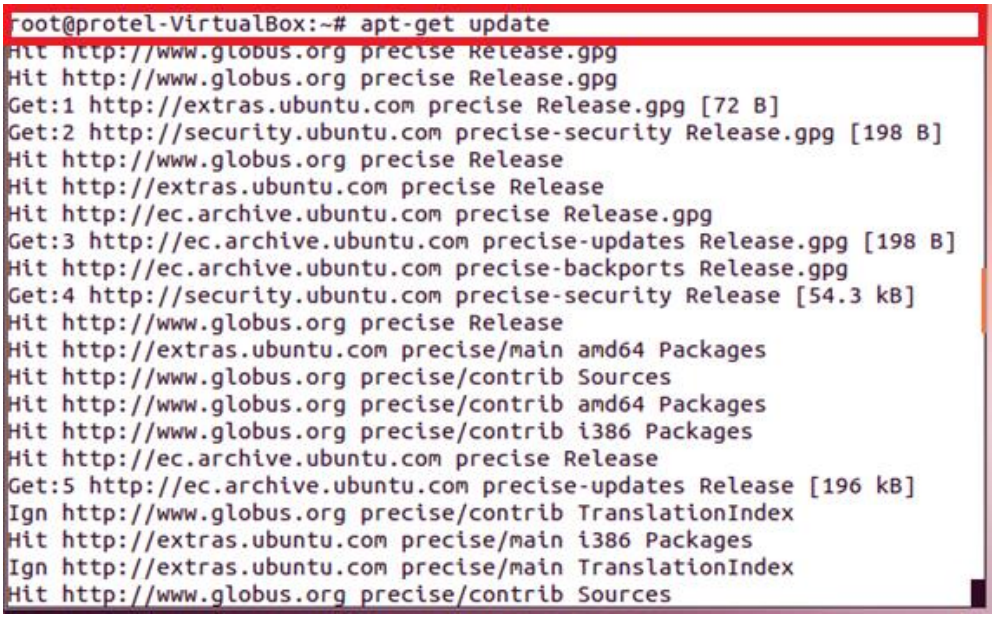

Figura 30 Comando de actualización Ubuntu.

Una vez actualizado el nodos, se modifica el archivo de host, ubicado en la ruta, **`/etc/hosts`**, agregando las direcciones IP, de todas las máquinas que se deseen incluir en la Grid.

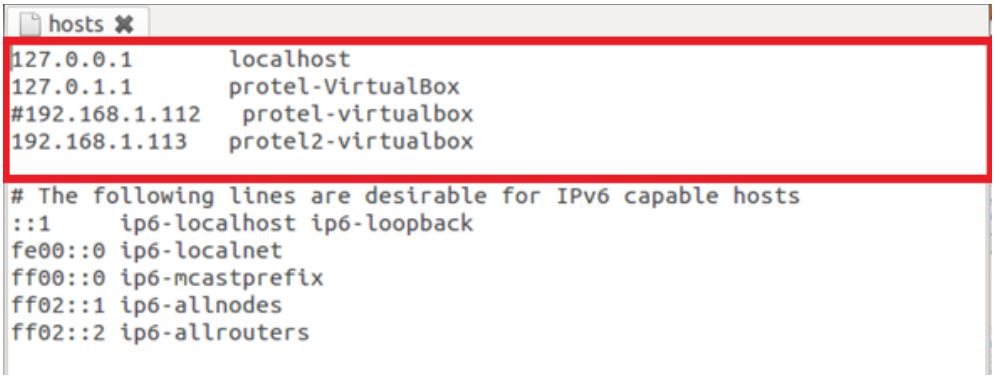

Figura 31 Archivo Hosts

Si se desea verificar la dirección IP de nuestra esquipo, se lo puede realizar mediante el comando `ifconfig`

4.3. Configuración de Servidor

Primero se debe iniciar con la instalación de los paquetes necesario para el manejo de servicios Myproxy, GSI, y GridFTP, usando el siguiente comando:

```
root@protel-VirtualBox:/# apt-get install -y globus-data-management-client globus-gridftp globus-gram5 globus-gsi myproxy myproxy-server myproxy-admin libperl4-corelibs-perl
```

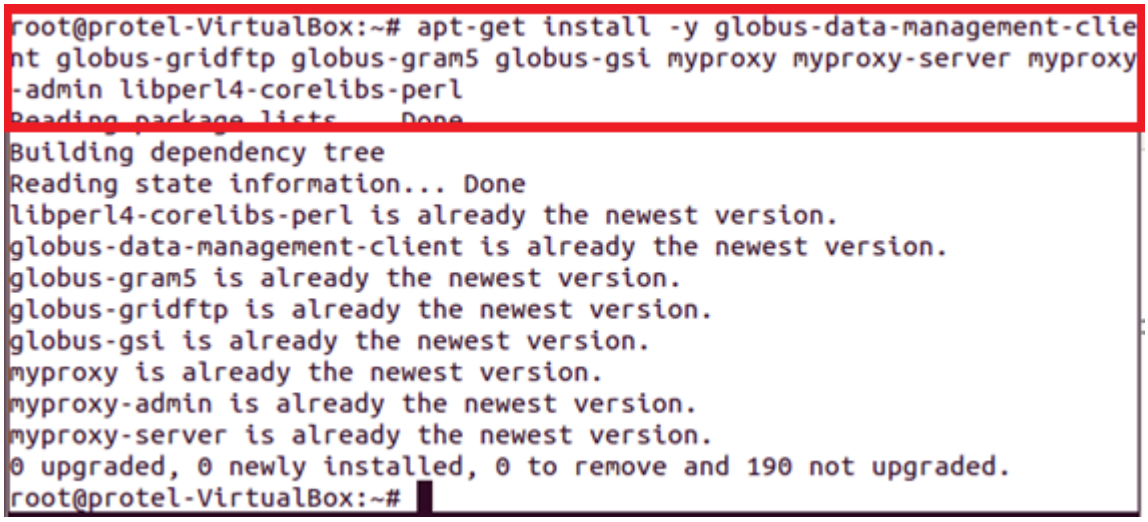

Figura 32 Comandos de instalación Myproxy y GridFTP

Con el anterior comando, también se incluye la instalación por defecto de SimpleCA, que es una entidad certificadora propia del Globus Toolkit, y con ella la creación de credenciales de seguridad que se utilizaran los usuarios para ejecutar los servicios de Globus Toolkit.

Al momento de instalar automáticamente se creara una nueva entidad certificadora, y generara un certificado público en el directorio de confianza del Globus, esto también generara una llave que permitirá utilizar los servicios de Globus Toolkit.

SE coloca los certificados del servidor y la llave en el directorio, para que el servicio Myproxy pueda usarlos, usando las siguientes líneas de comando:

```
root@protel-VirtualBox:/# install -o myproxy -m 644 /etc/grid-security/hostcert.pem /etc/grid-security/myproxy/hostcert.pem
```

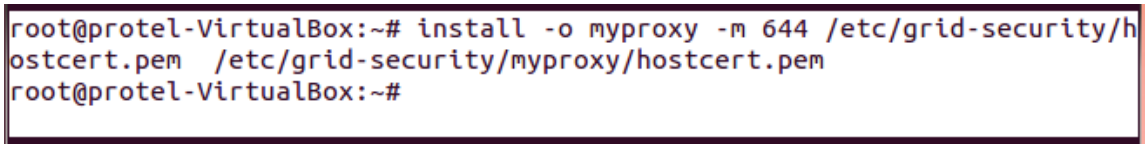

Figura 33 Comando generador de llaves publicas 644

```
root@protel-VirtualBox:/# install -o myproxy -m 600 /etc/grid-security/hostkey.pem /etc/grid-security/myproxy/hostkey.pem
```

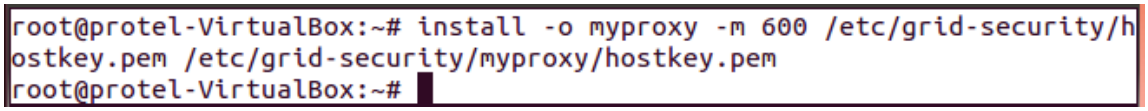

Figura 34 Comando generador de llaves publicas 600

4.4. Creación del servidor Myproxy

Una vez realizado los procesos anteriores se encuentra listo para crear el servidor Myproxy, atravez de un simple comando, el cual debe ser ejecutado en nuestra maquina servidor `protel-VirtualBox`

```
root@protel-VirtualBox:/#apt-get install myproxy-admin
```

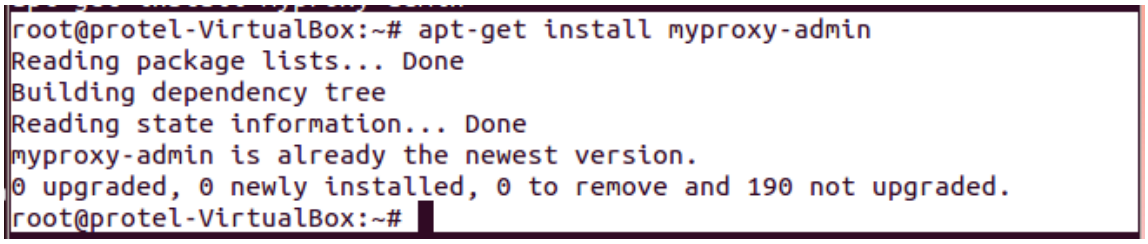

Figura 35 Comando Instalador Servidor MyProxy

MyProxy será el encargado de almacenar los certificados de los usuarios. Para que este servicio utilice SimpleCA, se debe modificar el archivo ubicado en la ruta `/etc/myproxy-server.config`

Se debe retirar el símbolo `#`, de las líneas de sección “Ejemplos de Política”, debiendo quedar de la siguiente manera:

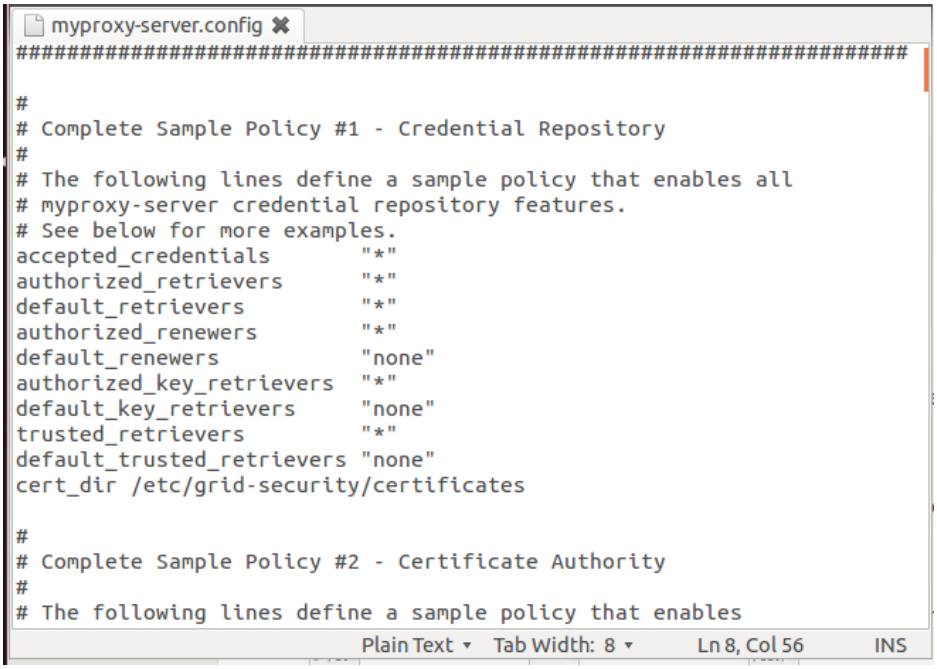

Figura 36 Archivo myproxy-server.config

Luego, se agrega el usuario myproxy al grupo de la SimpleCA, así el servidor myproxy podrá crear certificados registrados a la Grid.

```
root@protel-VirtualBox:/# usermod -a -G simpleca myproxy
```

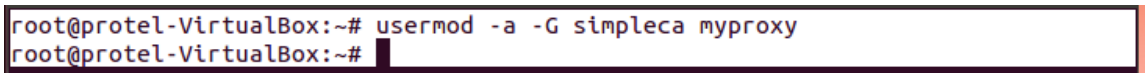

Figura 37 Comando registro de grupo SimpleCA

Una vez ejecutado el comando anterior, si servidor fue instalado correctamente, se iniciar el servicio mediante el siguiente comando:

```
root@protel-VirtualBox:/# service myproxy-server start
```

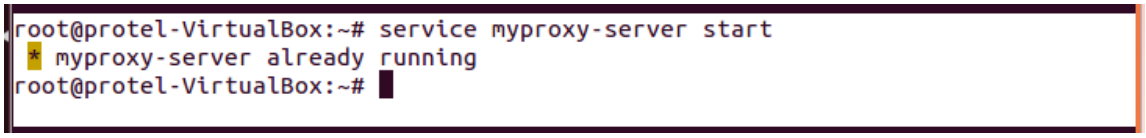

Figura 38 Comando inicio de servicio MyProxy

El servidor brinda un comando especial, con el cual se pode revisar el estado de ejecución: `service myproxy-server status`

La última verificación que se realizara al servidor myproxy es el estado y los puertos que están siendo utilizados, esto se lo realiza usando el comando:

```
root@protel-VirtualBox:/# netstat -an | grep 7512
```

Figura 39 Comando Verificación de Estado MyProxy

4.5. Creación de Credenciales

Al momento de realizar las configuraciones se debe especificar el nombre completo y el nombre de usuario para las credenciales de usuario que se va a crear, para este proceso se utilizó el nombre `protel`, como nombre de cuenta de usuario, dado que fue el usuario administrador del equipo servidor, `protel-virtualbox`, en la cual se esta instalando los servicios principales pero si fuera el caso necesario se puede utilizar en otro usuario local.

Primero se debe cambiar al prompt del usuario `myproxy`, y ejecutar el comando: `$ myproxy-admin-adduser`

En esta pantalla se necesita una frase de contraseña o "***passphrase***", para el usuario que va a acceder a la nueva credencial. Dicho usuario puede cambiar la frase, mediante el comando: `$ myproxy-change-passphrase`

Se procede a cambiar el promt al usuario myproxy

```
root@protel-VirtualBox:/# su - -s /bin/sh myproxy
```

Se agrega la variable del path:

```
$ PATH=$PATH:/usr/sbin
```

Se agrega el usuario al proxy

```
$ myproxy-admin-adduser -c "QuickStart User" -l protel
```

```
root@protel-VirtualBox:~# su - -s /bin/sh myproxy

-su: 1: $: not found
$ PATH=$PATH:/usr/sbin
$ myproxy-admin-adduser -c "QuickStart User" -l protel
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

Revoking previous certificate

Signing new certificate

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/09.pem

using storage directory /var/lib/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-protel-virtualbox/OU=local/CN=QuickStart User
$
```

Figura 40 Comandos de autentificación

4.6. Mapa de Credenciales

Se necesita crear el archivo “mapa de Grid”, el cual permite ingresar las credenciales de los usuarios, estas serán usadas para acceder a los servicios de Globus

```
root@protel-VirtualBox:/# grid-mapfile-add-entry -dn
"/O=Grid/OU=GlobusTest/OU=simpleCA-protel-
VirtualBox/OU=local/CN=QuickStart User" -ln protel
```

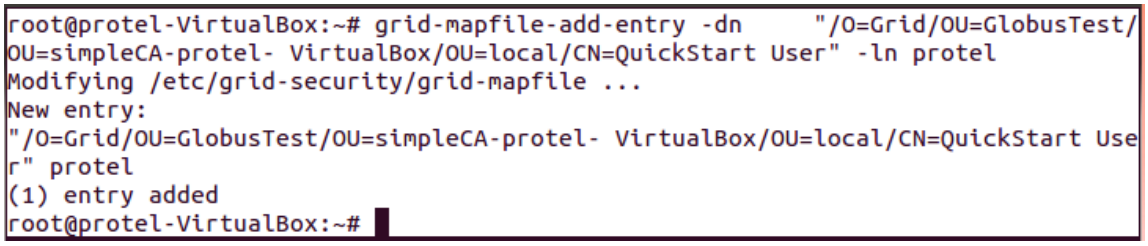

Figura 41 Comando creación de mapas Grid

El resultado de este comando, serán las credenciales de usuario, y la cuenta del usuario local, además de un mensaje de confirmación de éxito al agregar una nueva entrada al archivo.

4.7. Arquitectura Single Sign-On

Globus Toolkit, utiliza una Infraestructura de Seguridad para Grid conocida como GSI, la cual establece una autentificación y comunicación segura en una red abierta, además provee servicios que incluyen autentificaciones certificadas mutuas del tipo Single Sign On.

Esto basado en una encriptación de llave publica, con certificados X.509[10], y protocolos de comunicación SSL[11].

Para observar adecuadamente el funcionamiento de las autentificaciones certificadas se utiliza el servicio GridFTP, propio de Globus Toolkit, el cual permitirá compartir archivos de manera segura, entre dos nodos certificados y autenticados de la Grid.

4.8. Instalación de GridFTP

Una vez que el servidor posee las credenciales de usuario en su lugar se procede con los servicios de Globus. Se inicia el servidor GridFTP, mediante el siguiente comando:

```
root@protel-VirtualBox:/# service globus-gridftp-server start
```

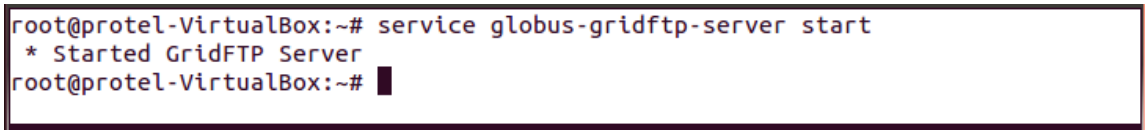

Figura 42 Comando Inicio de servicio GridFTP

Una vez que el servidor GridFTP se encuentre en ejecución se procede a verificar el estado del mismo, y el puerto en el cual se encuentra escuchando las peticiones con el comando:

```
root@protel-VirtualBox:/# service globus-gridftp-server status
```

---

[10] En criptografía, X.509 es un estándar UIT-T para infraestructuras de claves públicas

[11] Secure Sockets Layer es un protocolo criptográfico que proporcionan comunicaciones seguras por una red, comúnmente Internet.

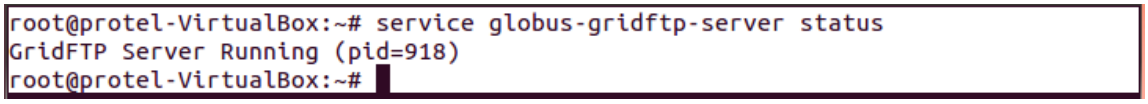

Figura 43 Comando estado de servicio GridFTP

El servidor se encuentra esperando por las peticiones y está listo para ser utilizado. Ahora es necesario iniciar las autentificaciones mediante el proxy. Desde el servidor myproxy, se utiliza el comando `myproxy-logon` para realizar nuestra autentificación. Esta será posible ya que previamente se ha colocado las llaves necesarias para dicho proceso. Para comprobar que nuestra autentificación fue exitosa se utiliza el comando `globus-url-copy`, este comando permite enviar un archivo entre nodos de la Grid. Aun no se ha registrado ningún otro nodo, pero este comando es el mismo con el cual, una vez registrado un cliente, se puede enviar archivos entre ellos.

Se debe recordar que previamente se registro una “frase de contraseña” para nuestra credencial, la misma que será requerida para la autentificación.

```
protel@protel-VirtualBox% myproxy-logon -l protel -s protel-VirtualBox
```

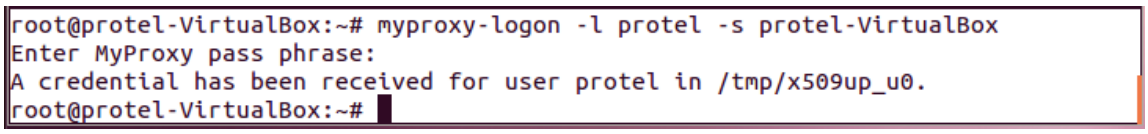

Figura 44 Comando de logeo

```
protel@protel-VirtualBox% globus-url-copy gsiftp://protel-VirtualBox/etc/group file:///tmp/protel.test.copy
protel@protel-VirtualBox% diff /tmp/protel.test.copy /etc/group
```

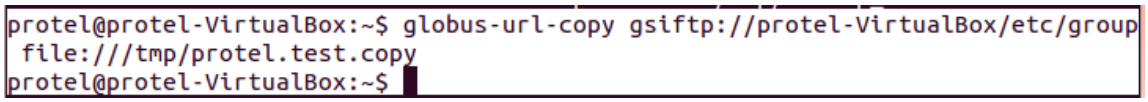

Figura 45 Comando envío de archivo GridFTP

4.9. Instalación de Clientes Grid

Una vez concluida las configuraciones del servidor myProxy y GridFTP se pueden utilizar credenciales seguras de autentificación en el servidor Grid, las cuales permiten transferir archivos luego de realizar las autentificaciones necesarias entre nodos de nuestra SmartGrid. Por el momento solo se ha configurado el servidor principal. Ahora es necesario agregar un nodo cliente a nuestra Grid, el cual consumirá los servicios del servidor.

4.10. Requisitos previos de los Clientes Grid

El primer requisito necesario en cada uno de los nodos clientes de nuestra grid es instalar Globus Toolkit, el cual se procede a realizar utilizando el siguiente comando:

```
root@protel2-VirtualBox:/#                                                wget
http://www/ftppub/gt5/5.2/5.2.3/installers/repo/globus-repository-5.2-
stable-precise_0.0.3_all.deb
```

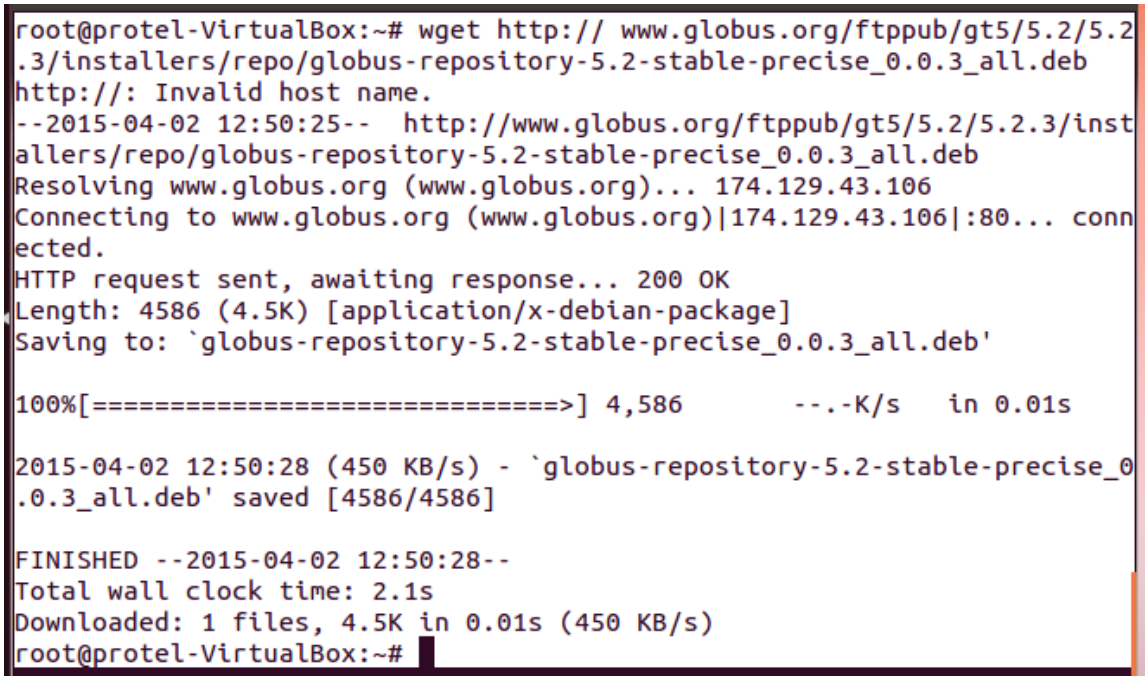

Figura 46 Comando para obtener Globus Toolkit del repositorio

```
root@protel-VirtualBox:/#    dpkg    -i    globus-repository-5.2-stable-
precise_0.0.3_all.deb
```

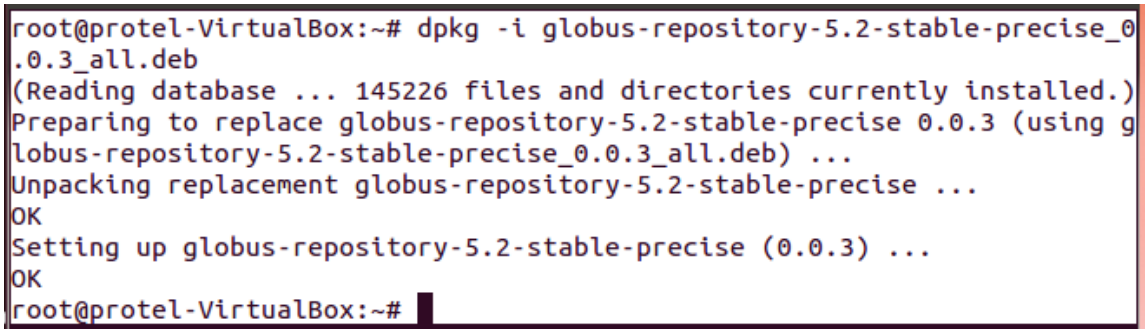

Figura 47 Instalación de paquete GlobusToolkit

Antes de realizar las actualizaciones, se debe entrar en la ruta **`/etc/apt/source.list`** y comentar las líneas del documento que inician con la palabra **`deb-src`**, solamente agregando el símbolo # al principio de las líneas.

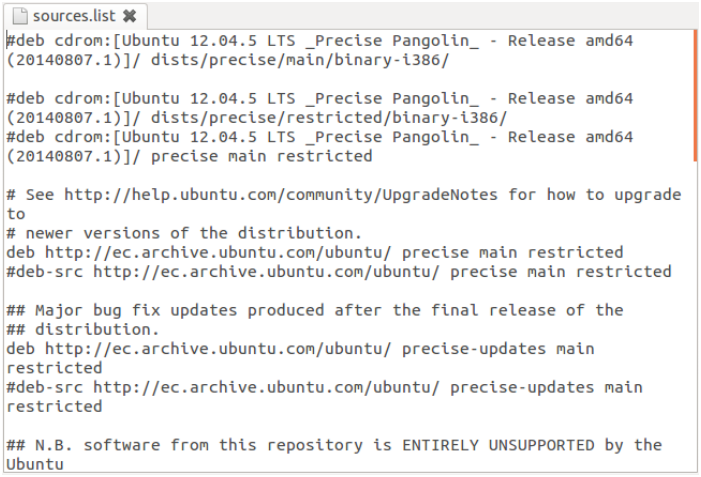

Figura 48 Archivo source.list

Luego se realiza una actualización completa con el comando detallado:

```
root@protel2-VirtualBox:/# apt-get update
```

```
root@protel-VirtualBox:~# apt-get update
Hit http://www.globus.org precise Release.gpg
Hit http://www.globus.org precise Release.gpg
Get:1 http://extras.ubuntu.com precise Release.gpg [72 B]
Get:2 http://security.ubuntu.com precise-security Release.gpg [198 B]
Hit http://www.globus.org precise Release
Hit http://extras.ubuntu.com precise Release
Hit http://ec.archive.ubuntu.com precise Release.gpg
Get:3 http://ec.archive.ubuntu.com precise-updates Release.gpg [198 B]
Hit http://ec.archive.ubuntu.com precise-backports Release.gpg
Get:4 http://security.ubuntu.com precise-security Release [54.3 kB]
Hit http://www.globus.org precise Release
Hit http://extras.ubuntu.com precise/main amd64 Packages
Hit http://www.globus.org precise/contrib Sources
Hit http://www.globus.org precise/contrib amd64 Packages
Hit http://www.globus.org precise/contrib i386 Packages
Hit http://ec.archive.ubuntu.com precise Release
Get:5 http://ec.archive.ubuntu.com precise-updates Release [196 kB]
Ign http://www.globus.org precise/contrib TranslationIndex
Hit http://extras.ubuntu.com precise/main i386 Packages
Ign http://extras.ubuntu.com precise/main TranslationIndex
Hit http://www.globus.org precise/contrib Sources
```

Figura 49 Comando de actualización Ubuntu.

Una vez actualizados los nodos, se modifica el archivo de host, ubicado en la ruta, **`/etc/hosts`**, agregando las direcciones IP, de todas las máquinas que se deseen incluir en la Grid

hosts

```
127.0.0.1       localhost
127.0.1.1       protel-VirtualBox
#192.168.1.112   protel-virtualbox
192.168.1.113   protel2-virtualbox

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Figura 50 Archivo Hosts

Si se desea verificar la dirección IP de nuestra esquipo, se lo puede realizar mediante el comando `ifconfig`

4.11. Instalación de Servicios

Ahora el cliente está listo para iniciar los procesos de configuración de GridFTP y MyProxy.

Se ejecuta el comando para instalar los paquetes de myProxy y GridFTP como se detalla a continuación:

```
root@protel2-VirtualBox:/# apt-get install -y globus-gridftp myproxy
globus-gram5
```

Figura 51 Comandos de instalación Myproxy y GridFTP

4.12. Configuración de Seguridad

El cliente debe estar certificado por medio de una entidad certificadora, para este caso particular se está utilizando SimpleCA, instalado en el servidor. No es necesario crear uno nuevo, aunque pudiera ser implementado según el caso práctico, pero para este efecto se ha realizado una grid simple. Lo primero que se debe ejecutar, es un bootsrap al servicio simpleCA que se encuentra ejecutándose en el servidor protel-VirtualBox

```
root@protel2-VirtualBox:/# export
MYPROXY_SERVER_DN="/O=Grid/OU=GlobusTest/OU=simpleCA-protel-
virtualbox/CN=protel-VirtualBox"
```

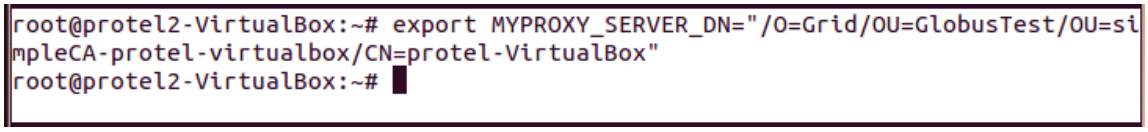

Figura 52 Comando envío de Certificados

```
root@protel2-VirtualBox:/#  myproxy-get-trustroots  -b  -s  protel-
VirtualBox
```

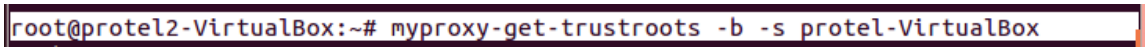

Figura 53 Comando aceptación de las certificaciones SimpleCA

Estos comandos permitirán a los clientes y servicios de los nodos confiar en los certificados firmados por la entidad certificadora que se encuentra en la servidor principal. Los usuarios en el cliente pueden adquirir sus credenciales usando el comando `myproxy-logon` y permitir la transferencia de archivos mediante `globus-url-copy` **y** `globus-job-run`**.** Las configuraciones entre los nodos servidores y clientes son similares, aunque difieren solamente con respecto a las entidades certificadora.

Para continuar con la configuración se procede a crear el certificado de host para el cliente pero este debe ser creado desde el servidor. Esto evita copiar los archivos entre las maquias, aunque GlobusToolkit permite crearlo en cada uno de los nodos para luego ser enviado al servidor.

El comando `myproxy-admin-addservice` solicita una “frase de contraseña” como credencial, la cual será utilizada desde el cliente al momento de la autentificación.

```
$ myproxy-admin-addservice -c "protel2-VirtualBox" -l protel
```

Una vez ejecutado el comando anterior, se crean las credenciales en el cliente como un usuario de root.

```
root@protel2-VirtualBox:/# myproxy-retrieve -s protel-VirtualBox -k protel2-VirtualBox -l protel
```

```
root@protel2-VirtualBox:~# myproxy-retrieve -s protel-VirtualBox -k protel2-VirtualBox -l protel
/etc/grid-security/hostcert.pem exists.
root@protel2-VirtualBox:~#
```

Figura 54 Comando creación de credenciales

Creadas las credenciales, ya no es necesario el certificado de host en el servidor, así que se lo elimina.

```
root@protel2-VirtualBox:/# myproxy-destroy -s protel-VirtualBox -k protel2-VirtualBox -l protel
```

```
root@protel2-VirtualBox:~# myproxy-destroy -s protel-VirtualBox -k protel2-VirtualBox -l protel
```

Figura 55 Comando destrucción de certificados

Y finalmente se agrega la credencial del cliente en el grid-mapfile, ubicado en el nodo cliente

```
root@protel2-VirtualBox:/# grid-mapfile-add-entry -dn "/O=Grid/OU=GlobusTest/OU=simpleCA-protel- VirtualBox/OU=local/CN=QuickStart User" -ln protel
```

```
root@protel2-VirtualBox:~# grid-mapfile-add-entry -dn "/O=Grid/OU=GlobusTest/OU=simpleCA-protel- VirtualBox/OU=local/CN=QuickStart User"
```

Figura 56 Comando creación de mapas Grid

Esto permite definir las seguridades del nodo cliente, y confiar en las certificaciones del servidor. Se debe crear un certificado de host para el cliente, y así poder ejecutar los servicios del Globus desde cualquier nodo agregado. Al final únicamente se debe iniciar los servicios para poder empezar a utilizarlos.

4.13. Pruebas de certificación digital

Las configuraciones del servidor GridFTP son similares en cualquier nodo de la Grid, pudiendo ser estos clientes o servidores. Esto permite la comunicación entre cualquier punto de la Grid, además sirve como respaldo de interconexión si uno de los puntos es interrumpido. También se puede replicar en tantos nodos como sean necesarios y estos llevaran sus propias credenciales para las autentificaciones.

Para realizar la prueba de comunicación y validación de credenciales, se procede a levantar los servicios en el cliente con el comando:

```
root@protel2-VirtualBox:/# service globus-gridftp-server start
```

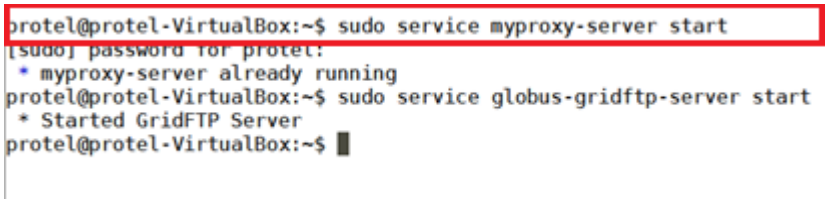

Figura 57 Comando de inicio de Servidor GridFTP

Ahora el cliente está listo a recibir las credenciales del servidor myproxy. Esto se lo realiza utilizando el siguiente comando:

```
protel@protel2-VirtualBox%                                    export
MYPROXY_SERVER_DN="/O=Grid/OU=GlobusTest/OU=simpleCA-protel-
virtualbox/CN=protel-VirtualBox"
```

Y el nodo cliente ya puede interactuar y autenticarse en la Grid, utilizando la misma frase de contraseña utilizada para generar la credencial

```
protel@protel2-VirtualBox% myproxy-logon –l protel -s protel-VirtualBox
```

Para verificar el funcionamiento de nuestra grid, se crea un archivo en el servidor, en la ruta **`/home/protel/ORIGEN/test`**

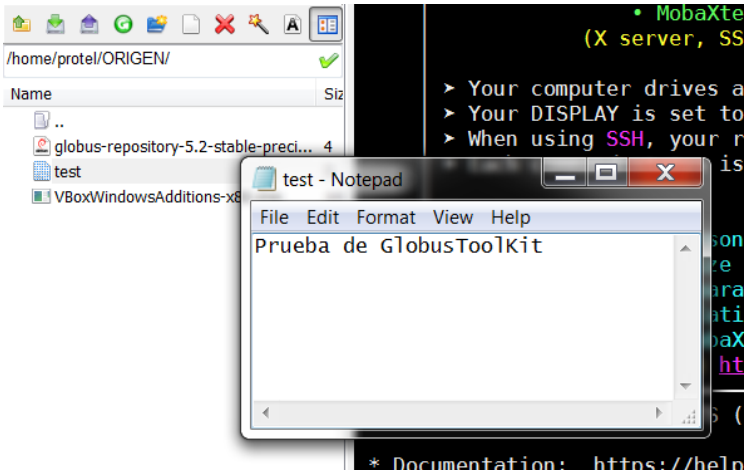

Figura 58 Archivo de prueba en el servidor.

Luego de tener el archivo de prueba listo, se ejecuta el comando de envío. Este comando no tiene un mensaje de finalización exitosa, pero para comprobarlo al acceder a la carpeta de destino.

```
globus-url-copy      gsiftp://protel-VirtualBox/home/protel/ORIGEN/test
gsiftp://protel3-VirtualBox/home/protel3/DESTINO/pruebacopiada
```

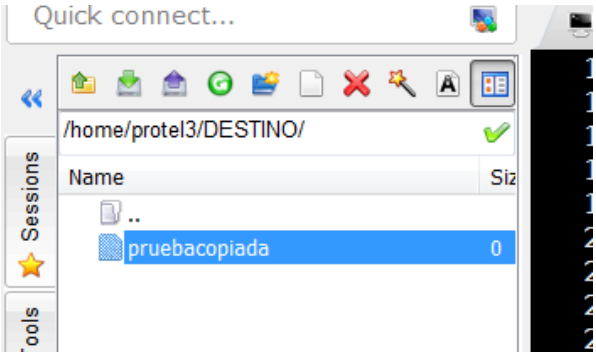

Figura 59 Archivo en la carpeta de destino.

Si se ejecuta el comando anterior, desde un usuario no autorizado este responderá con un mensaje el cual informa que no posee credenciales de

usuario, además de una verificación de los certificados, esto se debe a un error al momento de realizar la creación de certificados y la exportación al MyProxy server, se recomiendo repetir el proceso de creación de credenciales.

```
usutest@protel3-VirtualBox:~$ globus-url-copy gsiftp://protel-VirtualBox/home/prot

l3/DESTINO/ bus_ftp_control: gss_init_sec_context failed
globus_gsi_gssapi: Error with gss credential handle
globus_credential: Valid credentials could not be found in any of the possible loc
Valid credentials could not be found in any of the possible locations specified by
Attempt 1
globus_credential: Error reading host credential
globus_sysconfig: Error with certificate filename
globus_sysconfig: Error with certificate filename
globus_sysconfig: File is not owned by current user: /etc/grid-security/hostcert.p
Attempt 2
globus_credential: Error reading proxy credential
globus_sysconfig: Could not find a valid proxy certificate file location
globus_sysconfig: Error with key filename
globus_sysconfig: File does not exist: /tmp/x509up_u1001 is not a valid file
Attempt 3
globus_credential: Error reading user credential
globus_sysconfig: Error with certificate filename: The user cert could not be foun
1) env. var. X509_USER_CERT
2) $HOME/.globus/usercert.pem
3) $HOME/.globus/usercred.p12
```

Figura 60 Error de validación de credenciales.

Esta configuración como puedo ser apreciada, fue ligeramente más complicado que en el capítulo 4.2.1, ya que existe una transferencia con terceras partes de confianza entre los dos servidores GridFTP. Si funciono correctamente significa que la configuración entre estas fue realizada con éxito y la credencia fue aceptada sin ningún inconveniente. Si llegara a existir problemas, pero la configuración fue realizada acorde al manual de instalación, pudiera deberse a los firewalls entre ellas, además GridFTP utiliza algunos puertos de datos para la transferencia, no solamente el Puerto 281. Configurando adecuadamente los permisos de puertos y firewalls, GridFTP funcionara con normalidad.

CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- En la implementación de un SmartGrid fue necesario la utilización de un middleware, mismo que permitió simplificar los procesos y servicios para ofrecer a los clientes mayor acceso a los datos de las operaciones presentes y sus aplicaciones.

- La utilización de certificados digitales otorgados por entidades propias o particulares de certificación y aplicadas en un ambiente SmartGrid, garantizan parámetros de seguridad para la autenticación de los usuarios.

- La herramienta de software libre Globus Tool Kit permite la implementación de una SmartGrid y la creación de una entidad certificadora privada, integrando ambas tecnologías en un ambiente estable y seguro. El Globus Tool Kit tiene grandes beneficios para instituciones públicas o privadas que deseen crear una SmartGrid, ya que provee un gran número de servicios que se actualizan periódicamente y de gran calidad, así como aplicativos de seguridad y transferencia de archivos sin costo.

5.2. Recomendaciones

- Utilizar middlewares para la implementación de un SmartGrid ya que estos permiten simplificar los procesos y servicios para así ofrecer a los usuarios finales un ambiente simple y sencillo de usar.

- Los certificados digitales que son emitidos por entidades certificadoras, se deben utilizar en ambientes SmartGrid para la autenticación de los usuarios, ya que garantizan legal y tecnológicamente los procesos de autenticación.

- Implementar las herramientas provistas en el paquete *Globus Tool Kit* para el desarrollo de ambientes SmartGrid ya que esta permite integrar entidades certificadoras en Grids privadas. Además se puede acceder a los diversos servicios y soportes que GLOBUS ALLIANCE provee con regularidad.

BIBLIOGRAFIA

Sotomayor, B., & Childers, L. (2006). *Globus® Toolkit 4: Programming Java Services.* San Francisco: Mongan Kaufmann.

Virtual Box. (1 de 01 de 2006). Recuperado el 01 de 05 de 2015, de https://www.virtualbox.org/

Gartner Group. (8 de Diciembre de 2005). Defining Gartner Total Cost of Ownership. Stamford, Estados Unidos.

Hernández, V., & Moltó, G. (2006). *GMarte: Grid middleware to abstract remote task execution.* Madrid: Inpress.

Huaxiong, W., Pieprzyk, J., & Varadharajan, V. (2008). *Information Security and Privacy: 9th Australasian Conference.* Sydney: Springer.

Knapp, E. D., & Samani, R. (2013). *Applied Cyber Security and the Smart Grid: Implementing Security Controls.* Waltham: Syngress.

Rediris. (s.f.). Recuperado el 20 de Marzo de 2014, de https://www.rediris.es/cert/doc/unixsec/node14.html

Sun Grid Engine. (n.d.). Retrieved 01 23, 2015, from http://gridengine.sunsource.net.

Wilkinson, B. (2013). *Grid Computing: Techniques and Applications.* Boca Raton : CRC Press.

Zavoral, F., Jung, J. J., & Costin , B. (2013). *Intelligent Distributed Computing VII: Proceedings of the 7th International.* New York: Springer.

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

CANDO PENAHERRERA ESTEBAN EDUARDO

DARIO JAVIER VELA ZAMBRANO

DIRECTOR DE CARRERA DE INGENIERIA EN SISTEMAS E INFORMATICA

ING. MAURICIO CAMPANA

SANGOLQUÍ, AGOSTO 2015