ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"SERVIDOR DE ACCESO REMOTO SOBRE LINUX"

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del Título de:

LICENCIADO EN REDES Y SISTEMAS OPERATIVOS

Presentado por:

SANCHEZ GRANJA XAVIER ARTURO

SOLORZANO CEDENO MARCOS JOSUE

GUAYAQUIL – ECUADOR

AÑO 2014

AGRADECIMIENTO

A Dios por permitirnos culminar nuestros estudios Universitarios, sin el nada de esto hubiera sido posible.

A nuestros padres que son un pilar indispensable en nuestras vidas y siempre tuvimos su apoyo.

A nuestros profesores que son parte importante en nuestro aprendizaje.

DEDICATORIA

A Dios porque él me ha dado la sabiduría para poder alcanzar mis objetivos

A mis padres que siempre estuvieron en el lugar y momento adecuado para poder ayudarme absolutamente en todo.

A mis compañeros y profesores los cuales me brindaron siempre su apoyo a lo largo de la carrera

XAVIER SANCHEZ GRANJA

DEDICATORIA

A Dios por brindarme la oportunidad y la dicha de la vida, al brindarme los medios necesarios para continuar mi formación como profesional.

A mis padres que me acompañaron a lo largo del camino, brindándome la fuerza necesaria para continuar

MARCOS SOLORZANO CEDENO

TRIBUNAL DE SUSTENTACIÓN

Ing. Fabián Barboza

Profesor de la Materia de Graduación

Ing. Jorge Magallanes

Profesor delegado por la Unidad Académica

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este informe, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)

_____

Xavier Arturo Sánchez Granja

_____

Marcos Josué Solórzano Cedeño

RESUMEN

El proyecto de graduación realizado consiste en desarrollar una herramienta web para facilitar la configuración y administración de enlaces de acceso remoto (VPN) en la empresa PETROGAS, la aplicación fue creada en HTML, CSS y el sistema operativo que utilizamos para realizar el túnel de acceso remoto es Linux fue LINUX con su distribución CENTOS 6.

ÍNDICE GENERAL

GLOSARIO

Middleware: es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias en los sistemas distribuidos. De esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicio, etc.

JavaScript: JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML.

HTML: HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

CSS: Son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada, hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML.

Firma Digital: Una firma digital es un mecanismo criptográfico que permite al receptor de un mensaje firmado digitalmente determinar la entidad originadora de dicho mensaje (autenticación de origen y no repudio), y confirmar que el mensaje no ha sido alterado desde que fue firmado por el originador (integridad).

Autoridad Certificadora: (**AC** o **CA** por sus siglas en inglés ***Certification Authority***) es una entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados, utilizados en la firma electrónica, para lo cual se emplea la criptografía de clave pública.

Script: En Informática, un script o como también se lo conoce, un archivo de órdenes o archivo de procesamiento por lotes, es un programa simple, que se almacena en un archivo de texto plano y cuyo uso fundamental resulta a la hora de tener que realizar diversas tareas como ser la combinación de componentes, la interacción con el usuario o con el sistema operativo en cuestión. Facilita la automatización de tareas a través de la creación de pequeñas utilidades.

OpenVPN: es una solución de conectividad basada en software libre; SSL (Secure Sockets Layer) VPN Virtual Private Network (red virtual privada), OpenVPN ofrece conectividad punto-a-punto con validación jerárquica de usuarios y host conectados remotamente

NAT: La "Traducción de Direcciones de Red", Network Address Translation (NAT), es un método mediante el que las direcciones IP son mapeadas desde un dominio de direcciones a otro, proporcionando encaminamiento transparente a las máquinas finales.

Shell: Es un intérprete de comandos, el cual consiste en la interfaz de usuario tradicional de los sistemas operativos basados en Unix y similares como GNU/Linux.

Protocolos: Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes.

Funciones: Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que

modifiquen su funcionamiento. Las funciones son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código.

CORE: Núcleo de red es la capa encargada de proporcionar conectividad entre los distintos puntos de acceso (router, switch, etc). El Núcleo de Red nos permite enlazar diferentes servicios, como Internet, redes privadas, redes LAN o telefonía entre otros.

ABREVIATURAS

TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet.

ICMP: Protocolo de Mensajes de Control de Internet.

PING: Packet Internet Groper o Buscador de paquetes en redes.

UDP: Protocolo de datagrama de usuario.

VPN: Virtual Prívate Network es utilizada para establecer conexiones remotas desde redes públicas con redes privadas.

SSH: Secure Shell es un protocolo de red que permite el intercambio de datos sobre un canal seguro entre dos computadoras.

SSL: protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

ISP: Proveedor de servicios de Internet, empresa que brinda conexión a Internet

WAN: Red de área amplia

LAN: Red de área local

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCION

En la actualidad las comunicaciones a través de las redes de información resultan de vital importancia para un gran número de empresas y organizaciones. Para llegar a su destino ese tráfico debe atravesar, muy a menudo, una infraestructura de redes públicas (internet), lo que hace vulnerable a los ataques de usuarios mal intencionados. Ante ese peligro potencial, resulta imprescindible poseer herramientas que permitan proteger el contenido de dicho tráfico, para asegurar tanto su privacidad como su integridad, en las comunicaciones extremo a extremo.

La solución más evidente consiste en montar redes privadas, para el uso exclusivo de los propietarios de la red, garantizándose la seguridad en las comunicaciones. Sin embargo esta política de seguridad se está abandonando debido a la baja escalabilidad. El uso de redes privadas supone la implementación de un nuevo enlace cada vez que se quiera unir un nuevo miembro a la red de una organización.

Como solución más eficiente, aparecen las Redes Privadas Virtuales (VPN), un sistema para construir conexiones seguras a través de la

infraestructura de redes públicas, tanto para enlaces punto a punto, como para conectar distintas redes locales entre si o permitir a un tele-trabajador conectarse a la sede de su empresa desde cualquier lugar con acceso a Internet. Este sistema permite aprovechar la infraestructura de comunicaciones existente, apartando los elementos de seguridad necesarios para evitar cualquier intrusión en el contenido del tráfico que protegen. Se consigue así un método de comunicación segura que combina un bajo coste con unos altos niveles de privacidad.

CAPITULO I

MARCO TEORICO

1.1 MPORTANCIA Y USO DEL SOFTWARE LIBRE EN EMPRESAS ECUATORIANAS

1.1.1 IMPORTANCIA DEL SOFTWARE LIBRE

"El uso de Software Libre asegura la soberanía tecnológica, impulsa la innovación nacional, optimiza el gasto estatal fortaleciendo el desarrollo local y facilita la inclusión digital.

Algunos beneficios del uso de software libre son:

- **Estandarización e Integración:** El Software Libre es producido utilizando especificaciones y estándares tecnológicos libres y públicos, también denominados "estándares abiertos". Esto beneficia la integración de sistemas y el intercambio de información, de forma que se garantiza la accesibilidad sin restricciones por parte de la ciudadanía.

- **Seguridad:** El hecho de hacer públicos los códigos de los programas favorece a la seguridad de los mismos. Utilizando Software Libre se puede saber qué está haciendo realmente un programa, qué tipo de información maneja y cómo lo hace.

- **Economía:** Se estima que la compra de un sistema operativo más un paquete de suite de oficina, ambos con una licencia comercial, cuestan entre

300 y 600 dólares por cada computadora, y ese gasto debe renovarse cada dos o tres años debido a la dependencia hacia el fabricante en que se incurre. Los países en vías de desarrollo, con las carencias de recursos que cuentan, pueden ahorrar una gran cantidad de recursos económicos".

1.1.2 USO DEL SOFTWARE LIBRE EN EL ECUADOR

"El Gobierno Constitucional del Economista Rafael Correa Delgado promueve el uso de Software Libre como política de Gobierno ya que le permite al Estado tener mayor seguridad informática, libre acceso a datos y programas, ahorro en costos de licencias y es un generador de empleo para profesionales ecuatorianos".

1.1.3 DECRETO PRESIDENCIAL SOBRE EL USO DEL SOFTWARE LIBRE

"Mediante Decreto Ejecutivo No. 1014 emitido el 10 de Abril de 2008, se dispone el uso de Software Libre en los sistemas y equipamientos informáticos de la Administración Pública de Ecuador.

Previo a la promulgación del Decreto de Software Libre en Abril del 2008, la mayoría de instituciones de la Administración Central utilizaban software privativo en sus sistemas informáticos. Actualmente, todas estas entidades tienen planificado o se encuentran ejecutando procesos de migración y prácticamente todos los nuevos proyectos informáticos consideran la adopción de herramientas de Software Libre

Se estima que la inversión del Gobierno Central alrededor del Software Libre es de $450.000, Esto incluye el desarrollo de los sistemas transversales y portales institucionales, así como la capacitación en herramientas de Software Libre".

1.2 ISO/IEC 27002 E IMPORTANCIA DE APLICARLA EN LAS EMPRESAS

El Estándar Internacional ISO/IEC 27002 es una guía de buenas prácticas que describe los objetos de control y controles recomendables en cuanto a seguridad de la información, es totalmente necesario para cualquier organización que tenga que ver de alguna forma con aspectos relacionados a tecnologías de información ya que ahora se ha agudizado más la importancia de contar con buenos mecanismos de seguridad debido a que los riesgos y amenazas no solamente consisten en que personas que se encuentren en el área geográfica donde están las computadoras, roben información, sino que ahora también existen riesgos de robo o accesos no autorizados a información mediante las diferentes redes que interconectan a las computadoras o a cualquier equipo tecnológico utilizado para transmitir información digital.

La norma ISO/IEC 27002 contiene:

- 11 Dominios
- 39 objetos de control
- 113 Controladores agrupados

Dominios de la ISO 27002:

- Políticas de seguridad.
- Aspectos organizativos de la seguridad de la información.
- Gestión de Activos.
- Seguridad ligada a los Recursos Humanos.
- Seguridad física del entorno.
- Gestión de comunicaciones y operaciones.
- Control de acceso.
- Adquisición, desarrollo y mantenimiento de sistemas de información
- Gestión de incidentes de seguridad de la información.
- Gestión de la continuidad del negocio.
- Cumplimiento.

1.2.1 MEJORES PARACTICAS PARA EL CONTROL DE ACCESO ESTABLECIDOS EN LA ISO/IEC 27002

Entre las mejores prácticas tenemos las siguientes:

- “Se debe contar con una política de control de acceso. Todo acceso no autorizado debe ser evitado y se deben minimizar al máximo las probabilidades de que eso suceda. Todo esto se controla mediante registro de usuarios, gestión de privilegios, autenticación mediante usuarios y contraseñas, etc.”

- “Son necesarios controles de acceso a la red, al sistema operativo, a las aplicaciones y a la información. Para todo esto deben existir registros y bitácoras de acceso”.
- “Deben existir políticas que contemplen adecuadamente aspectos de comunicación móvil, redes inalámbricas, control de acceso a ordenadores portátiles, y teletrabajo, en caso que los empleados de la empresa ejecuten su trabajo fuera de las instalaciones de la organización”.

1.3 USO DE LA ISO/IEC 27002 EN LA EMPRESA PETROGAS

La Empresa PETROGAS hace uso de la Norma ISO/IEC 27002 con el fin de asegurar la seguridad de la información en la empresa, para ello han establecido controles de seguridad que se clasifican en:

- Gestión de Activos
- Seguridad Física y del Entorno
- Control de Acceso
- Control de Acceso a la Red

A continuación detallaremos el control de Acceso a la Red con el cual justificamos nuestro proyecto.

1.3.1 CONTROL DE ACCESO A LA RED

1.3.1.1 Política de usos de Acceso a la Red

- El uso del acceso a la red esta con restricciones para evitar el traspaso de información que no tenga que no interese a la empresa, y solo es uso para cumplir labores investigativos a favor de la empresa

- Se utiliza la restricciones para el uso de internet mediante la implementación de proxy y firewalls utilizando el bloque por puertos y dominios

1.3.1.2 Autenticación de usuarios para conexiones externas

- Como en la empresa existe la administración remota se tiene implementados VPN para acceder a la red interna de la empresa mediante el uso de usuario y contraseña para el acceso seguro a la red y salvaguardar la integridad de la información sensible de la institución.
- Manejo de protocolos de autenticación SSH, SSL, IPSEC.

1.3.1.3 Identificación de los equipos por la red

- Para segura que los equipos que ingresen a la red institucional primero el equipo debe pertenecer al dominio de la empresa, debido a que tiene parches de seguridad antivirus de y los equipos son identificados a través de la red por la dirección MAC.

1.3.1.4 Protección de los puertos de diagnósticos y configuración remota

- Se configuro en firewall de la empresa para que estén habilitados puertos especiales para realizar las configuraciones remotas y también se utiliza el filtrado de ACL por puerto y por IP’s.

- Se utiliza herramientas Open Source para la administración de las VPN's de la empresa y también se utiliza software licenciado.

1.4 JUSTIFICACION

El uso de estándares internacionales emitidos por la ISO hace que las empresas aseguren sus sistemas de información de mejor manera, así como también pueden garantizar la fiabilidad de sus operaciones internas.

Para la creación de este proyecto nos basamos en la política "Autenticación de usuarios para conexiones externas" que aplica PETROGAS para el uso de conexiones externas además del decreto presidencial del uso de software libre, y se propuso desarrollar una herramienta que facilite la instalación, configuración y administración del servicio de acceso remoto.

CAPITULO II

ANALISIS DE LA INFRAESTRUCTURA DE TI

2.1 ANTECEDENTES DE LA EMPRESA

El bloque 6 de la empresa pública de hidrocarburos PETROGAS EP, está encargada de la exploración, perforación y explotación de los recursos no renovables (GAS) ubicados en la subsuelo marino del golfo de Guayaquil. Mejorando la gestión de las actividades asumidas por el estado ecuatoriano en el sector estratégico de los recursos no renovables (GAS), en las fases de exploración y explotación; con patrimonio propio, autonomía presupuestaria, financiera, económica, administrativa y de gestión.

Su matriz está ubicada en Machala y tiene 2 sucursales llamadas locaciones las cuales están ubicadas en BAJOALTO y la PLATAFORMA AMISTAD ubicada a 35 km desde la base logística a mar abierto.

2.2 INFRAESTRUCTURA DE LA RED DE AREA AMPLIA (WAN) DE PETROGAS

La empresa pública de hidrocarburos PETROGAS EP en la actualidad cuenta con una locación principal ubicada en la ciudad de Machala Provincia del Oro, y dos locaciones secundarias la primera es la locación llamada PLATAFORMA AMISTAD ubicadas a 40km mar abierto desde la base logística y la segunda locación llamada

BAJOALTO, ubicada en la comunidad de BAJOALTO que se encuentra a 20km desde la locación principal.

A continuación detallamos los enlaces WAN que tiene la empresa para la comunicación con las sucursales

- **Telconet**: carrier que proporciona 5 mbps de enlace hacia cada sucursal (MPLS).
- **Enlace Raidal** 1.5 megas que apunta para la Plataforma Amistad.
- **Enlace Radial** de 1.5 megas que apunta para la Planta de Gas.

El enlace principal es el que provee Telconet en caso de que este se caiga se aplica la redundancia con los enlaces radiales.

Los equipos de acceso a los enlaces son los siguientes:

- Router 1841 (Telconet)
- Modem Tew lab 8110 (Enlaces Radiales)

Adicionalmente a esos enlaces se tiene conexión con internet a través del proveedor de servicios de internet (ISP) Andinanet con un ancho de banda de 3 Mbps y cuyo proveedor de última milla para esta conexión de internet es Transferdatos, este enlace se encuentra en la matriz.

2.2.1 CARACTERISTICAS DE TRANSMISIÓN

La red de la empresa de Hidrocarburos PETROGAS se enlaza con todos las sucursales a través de 2 protocolos (MPLS) MULTILAYER PROTOCOL y

ENLACES RADIALES. Cada sucursal posee 2 enlaces al mismo tiempo ya que permite salvaguardar cualquier congestión y caída del servicio.

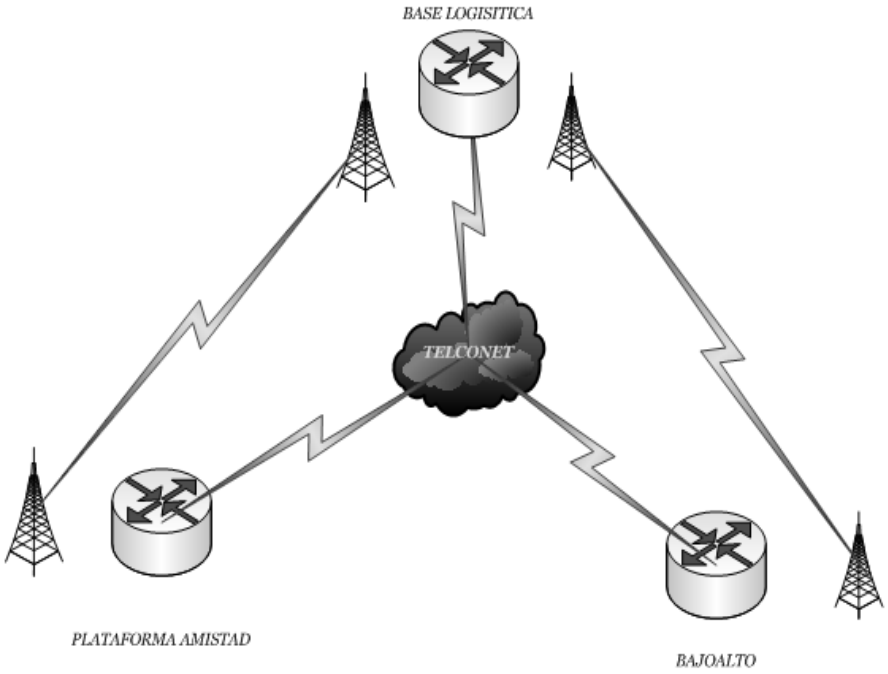


Figura 2.2-1 Red WAN PETROGAS

2.3 INFRAESTRUCTURA DE LA RED DE AREA LOCAL (LAN) DE PETROGAS

La Infraestructura Tecnológica de la Empresa de Hidrocarburos PETROGAS EP ha venido en un proceso de actualización y estandarización desde el 2008, esto con el objetivo de tener una plataforma abierta y de trabajo cooperativo en red, para convertir la infraestructura en un modelo estratégico que apoye efectivamente el negocio.

La red de datos tiene una arquitectura básica, su tráfico esta segmentado y los paquetes de comunicación emitidos por cada uno de los nodos son transmitidos a través de los ROUTERS hacia toda la red.

Esta red de datos está constituida por 15 servidores de datos y aproximadamente 100 nodos interconectados, las estaciones de trabajo poseen sistemas operativos Windows 7, todos estos nodos se comunican mediante SWITCHES a 100/1000 Mega Bytes de velocidad ubicados en las diferentes plantas de la empresa.

La topología de la red como ya se había mencionado es estrella, y está conformada por los siguientes componentes:

- Centro de datos o BACKBONE.
- Rack de comunicación por piso.

El centro de datos o BACKBONE se encuentra en el segundo piso de la MATRIZ donde están instalados los servidores, la central VOIP, un Switrch MULTICAPA marca ENTERASYS N3, un FIREWALL marca CISCO ASA.

Existe 3 RACKS de comunicaciones uno por piso en los que se encuentran 2 SWITCHES marca ENTERASYS C2 y un SWITCH marca ENTERASYS B5 en cada uno de los RACKS.

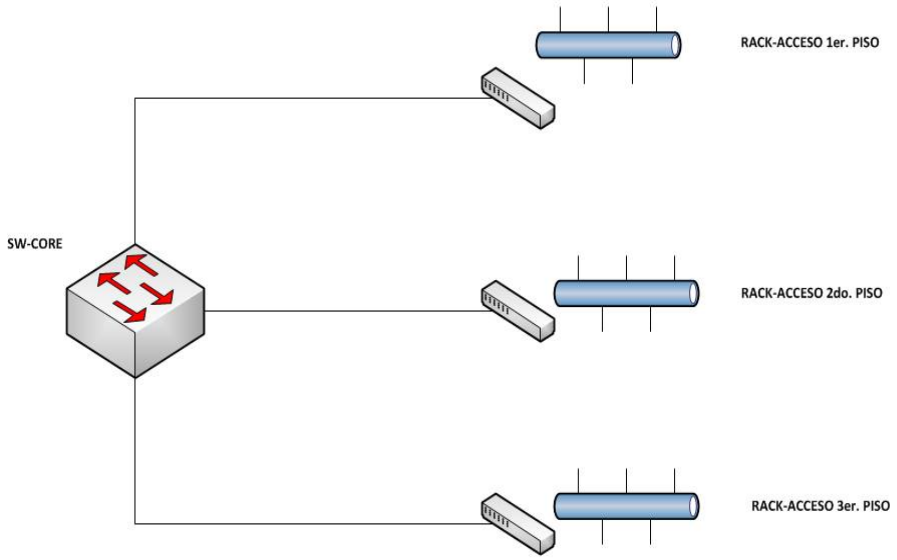


Figura 2.2-2 Topología LAN de la matriz

Cabe destacar, que el área es de acceso restringido solo para los encargados de la de administración de los mismos, mediante el uso de tarjetas de aproximación, y código de acceso además los servidores de datos, los dispositivos se encuentran instalados de forma apropiada y cada puerto en estos dispositivos están correctamente identificados así como los cables que están conectados hacia los dispositivos de datos esta implementado la norma de rotulación EIA/TEIA 606.

A continuación detallamos el direccionamiento utilizado en PETROGAS

Direccionamiento IP			
VLAN	*DESCRIPCION*	*RED*	*MASCARA*
20	*Servidores*	*172.29.97.0*	*255.255.255.0*
30	*Wireless*	*172.29.100.0*	*255.255.255.0*
40	*Telefonia*	*172.29.104.0*	*255.255.254.0*
50	*Usuarios*	*172.29.98.0*	*255.255.254.0*
60	*Dispositivos Estaticos*	*172.29.102.0*	*255.255.255.0*
70	*Camaras CCTV*	*172.29.103.0*	*255.255.255.0*
80	*Invitados*	*172.29.107.0*	*255.255.255.0*

Tabla 1 Direccionamiento aplicado en PETROGAS

2.3.1 SISTEMA DE CABLEADO ESTRUCTURADO

La tecnología de cableado estructurado está basada en topología tipo estrella utilizando la norma TIA/EIA 568 A y B, que permite la conexión de cada una de sus estaciones de trabajo a sitios centrales denominados RACKS de comunicación los cuales a su vez están conectados con el DATACENTER o BACKBONE Principal de Telecomunicaciones, es precisamente esta topología la que da una alta confiabilidad al sistema de cableado, ya que la falla de la conexión en uno de los puntos no afecta el funcionamiento de lo demás.

El cableado de la red está protegido por canaletas especiales, éste llega la caja de conexión (tipo RJ45) que están colocadas en las paredes y luego mediante un cable blindado se conectan las computadoras a dicha red.

La conexión de las estaciones de trabajo hacia los RACKS de comunicación denominada cableado horizontal se efectúan en cable trenzado de cuatro pares categoría 6 para velocidades de hasta 1GBPS en cable UTP con conectores RJ45 garantizando transportar cualquier tipo de información, y ofreciendo una excelente inmunidad a interferencias y ruidos eléctricos.

La conexión entre el CENTRO DE DATOS o BACKBONE hacia los RACKS de comunicación denominada cableado vertical se efectúa con fibra óptica multimodo de 125/omh, en cada piso encontramos un Rack de comunicaciones con SWITCHES marca ENTERASYS.

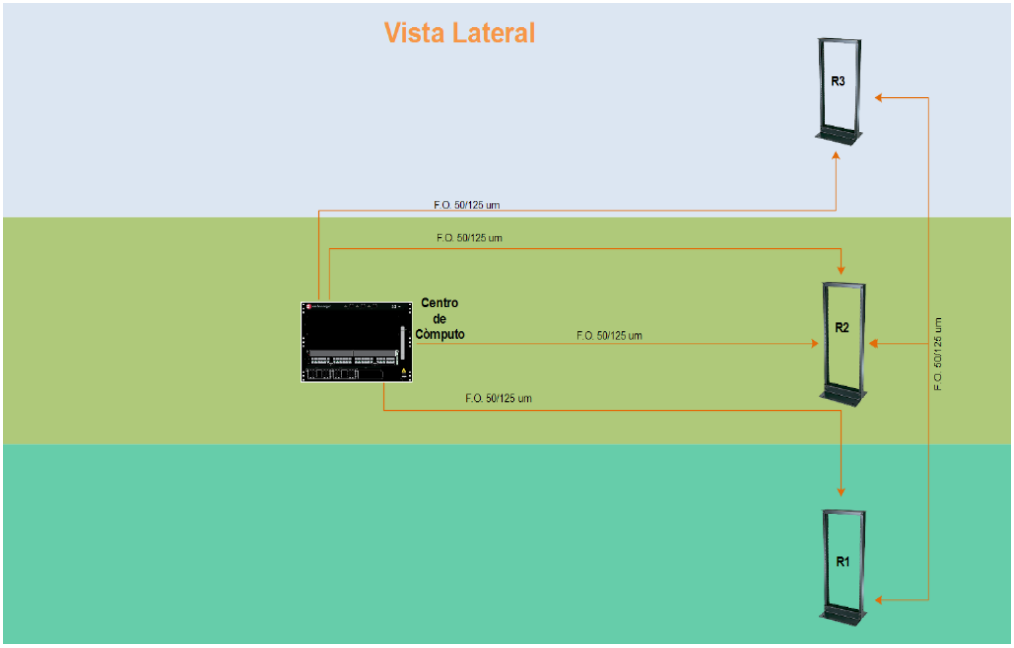


Figura 2.3-1 Arquitectura LAN

2.3.2 SERVIDORES

Los servidores que tiene la empresa se clasifican en 2 grupos: de un lado están los equipos IBM AS/400 que se usan como servidores aplicativos de base de datos, desarrollo de software y prueba de los sistemas del negocio. Estos Servidores tienen sistema operativo CENTOS y trabajan con los manejadores de base de datos de Windows, son equipos con una configuración robusta.

Por otro lado se tiene los equipos de servidores de redes locales que son tecnología Intel; para automatización de oficinas se tiene un servidor de correo electrónico trabajando Microsoft Exchange, un servidor de domino en donde se validan todos los usuarios de la red, un servidor ELASTIK el cual se encarga de gestionar la telefonía IP en la empresa, además de un servidor que controla las cámaras de seguridad de la empresa.

A continuación aparece un resumen con descripción detallada de los servidores.

Servidores			
Ubicación	*Equipo*	*S.O*	*Descripción*
MATRIZ	*HP*	*WINDOWS SERVER 2003*	*Cámaras de seguridad*
MATRIZ	*HP*	*WINDOWS SERVER 2008*	*Servidor de Dominio Y DHCP*
MATRIZ	*AVAYA*	*LINUX*	*Telefonía IP*
MATRIZ	*IBM*	*LINUX*	*BASE DE DATOS / APLICATIVOS*

Tabla 2 Servidores implementados

2.3.2.1 ESPECIFICACIONES TECNICAS DE LOS SERVIDORES

A continuación detallamos las especificaciones técnicas de los servidores de la empresa PETROGAS.

Servidor de Dominio ML10 HP ProLiant	
Procesador	Intel Xeon Quad Core E5540 2.o Ghz,
Memoria Ram	4Gb maxima capacidad 36 Gb (con dos procesadores usuando memoria tipo UDIMM
Capacidad de disco	Soporte un maximo de 4 x 1 TB disco SATA o SAS, en forma expandible de 8 TB
Tarjeta de red	Tarjeta Express Gigabit NIC 10/100/1000 WOL (Wake on LAN)

Tabla 3 Servidor de Dominio

Servidor de Telefonía ML10 HP ProLiant	
Procesador	Intel 1.06 GHz Core Duo
Memoria Ram	8Gb de RAM
Capacidad de Estacione	Hasta 450 Estaciones, IP´S y Analogicas
Tarjeta de red	Tarjeta Express Gigabit NIC 10/100/1000 WOL (Wake on LAN)

Tabla 4 Servidor de telefonía

Servidor Cámaras Hp Proliant ML110 G6	
Procesador	Intel Xeon Quad Core E5540 2.o Ghz, 1x4MB Level 3 cache - 550 series
Memoria Ram	estandar 2Gb maxima capacidad 24 Gb (con dos procesadores usuando memoria tipo UDIMM
Tarjeta de Red	Tarjeta integrada HP NIC P NC107i PCI Express Gigabit NIC 10/100/1000 WOL (Wake on LAN)
Capacidad de disco	Soporte un maximo de 4 x 1 TB disco SATA o SAS, en forma expandible de 8 TB

Tabla 5 Servidor Cámaras IP

Servidor Base de Datos	
Procesador	1 x Intel Xeon E5620 / 2.4 GHz (Quad-Core)
Memoria RAM	4 GB (instalados) / 128 GB (máx.) - DDR3 SDRAM
Memoria Cache	12 MB L3
Conexión de red	Adaptador de red - Ethernet : 2 x Gigabit

Tabla 6 Servidor de Base de Datos

2.3.2.2 SERVICIOS DE RED

Los servicios básicos de la red local permiten que la empresa pueda realizar sus actividades utilizando los siguientes servicios:

Correo Electrónico. Permite el envío y recepción de información entre todos los usuarios de la red local además del envío y recepción de correo a través de internet utilizando la aplicación web (OWA) Outlook Web Access.

File Server. Se cuenta con un servidor de automatización de espacio en disco el cual permite que varios usuarios compartan información almacenada allí. Adicionalmente cada usuario tiene implementado el sistema de disco de red para el manejo de la información departamental, documental y archivos temporales que están compartidos para todos los usuarios del domino.

WSUS. Este servicio presenta una mejor forma de actualizar el sistema operativo y productos Microsoft a cada uno de los usuarios de la red local.

INTRANET. Este servicio es utilizado por todos los usuarios de la red local para poder acceder a las aplicaciones que utiliza la empresa como: Directorio telefónico, sistema de Recursos Humanos Buxis para revisar roles de pagos y acreditaciones, sistema Oracle para realizar adquisiciones de bienes que llegase a necesitar la empresa, etc.

INTERNET. Los empleados de PETROGAS acceden al servicio de internet a través de un servidor PROXY que cumple las funciones del control de ancho de banda y restricción de páginas web que no sean para beneficios estrictamente laborales.

FIREWALL desempeña la función de filtrado de paquetes provenientes de todo el tráfico entrante de internet utilizando el mecanismo IDS (Intrusion Detection System) que es un programa utilizado para detectar accesos no autorizados a un computador o a una red en general, utilizando sensores virtuales o Sniffer que capturan paquetes para analizarlos y detectando anomalías que pueden ser inicio de la presencia de ataques o de falsas alarmas.

DHCP los dispositivos finales para acceder a los recursos de la red interna utilizan el servicio de DHCP que otorga direccionamiento de forma automática a todos los equipos de cómputo.

2.3.3 EQUIPOS DE ESCRITORIO

La empresa, basa su plataforma tecnológica de computadoras en equipos PC compatibles. Actualmente se cuenta con equipos con procesadores INTEL-PENTIUM I5 y superiores tal como se detalla en la tabla número7

Desktop				
Ubicación	*Equipo*	*Procesador/Memoria/HD*	*Sistema Operativo*	*Cantidad*
MATRIZ	*HP*	*Core i5/8GB RAM/250 Gb*	*W7 64 bits*	*40*
PLATAFORMA AMISTAD	*HP*	*Core i5/8GB RAM/250 Gb*	*W7 64 bits*	*15*
BAJOALTO	*HP*	*Core i5/8GB RAM/250 Gb*	*W7 64 bits*	*20*

Tabla 7 Equipos de Escritorio

2.3.4 LAPTOPS

La empresa, cuenta actualmente con Computadoras portátiles de tecnología PENTIUM, estos equipos son utilizados básicamente por personal de nivel directivo y personal que por motivos de las operaciones tienen que desplazarse a las distintas sucursales.

Laptops				
Ubicación	*Equipo*	*Procesador/Memoria/HD*	*Sistema Operativo*	*Cantidad*
MATRIZ	*Latitude E6430*	*Core i5/8GB RAM/250 Gb*	*W7 64 bits*	*22*
PLATAFORMA AMISTAD	*Latitude E6410*	*Core i5/8GB RAM/250 Gb*	*W7 64 bits*	*12*
BAJOALTO	*Latitude E6430*	*Core i5/8GB RAM/250 Gb*	*W7 64 bits*	*10*

Tabla 8 Portátiles

2.3.5 IMPRESORAS

La Empresa, cuenta 7 impresoras a laser de las cuales 2 son a colores utilizadas básicamente para la impresión de documentos importantes como contratos vigentes, circulares etc.

Las impresoras de PETROGAS están distribuidas de la siguiente manera:

Impresoras			
MARCA	*TIPO*	*UBICACIÓN*	*DEPARTAMENTOS*
Xerox WorkCentre 6400	*Multifuncional a colores*	*MATRIZ*	*Finanzas, Relaciones Comunitarias, Recursos Humanos*
Xerox WorkCentre 6400	*Multifuncional a colores*	*MATRIZ*	*TI, Mantenimiento, Materiales, Operaciones, Administración, Perforación*
Xerox WorkCentre 5330	*Multifuncional B/N*	*MATRIZ*	*TI, Mantenimiento, Materiales, Operaciones, Administración, Perforación*
HP DESINGJET 2300	*Plotter*	*PLATAFORMA AMISTAD*	*Exploración, Perforación*
Xerox Phaser 3635 MFP	*Multifuncional B/N*	*BAJOALTO*	*Operaciones*
Xerox Phaser 3635 MFP	*Multifuncional B/N*	*BAJOALTO*	*Operaciones*
Xerox Phaser 3635 MFP	*Multifuncional B/N*	*MATRIZ*	*Seguridad Salud y Ambiente, Seguridad Fisica*

Tabla 9 Impresoras

CAPITULO III

DISEÑO DEL CANAL DE ACCESO REMOTO

3.1 DISEÑO DE UNA TOPOLOGIA DE RED PARA EL CANAL DE ACCESO REMOTO

De acuerdo al análisis de la red realizado en la empresa PETROGAS, para poder implementar el servicio de acceso remoto es necesario el uso de un servidor centralizado.

Hemos diseñado una topología en la cual el servidor estará ubicado en la zona perimetral de la red con el fin de permitir las conexiones remotas, optamos por esta ubicación debido a que es el punto indicado donde entran y salen paquetes dirigidos hacia internet.

3.2 IMPLEMENTACION DE VPN PARA EL CANAL DE ACCESO REMOTO

Las siglas VPN significan: virtual personal network. VPN es una técnica que permite enlazar 2 o más redes, de tal manera que simulan ser una sola red privada, permitiendo así la conexión entre ellas como si los usuarios de ambas redes estuviesen dentro de una misma red. VPN generalmente es usado por usuarios

remotos para acceder a redes LAN.

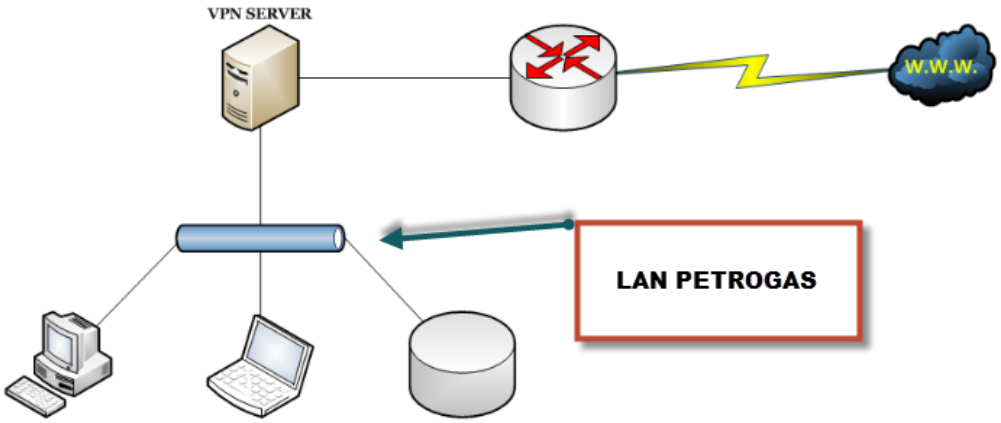


Figura 3.1-1 Topología para la implementación

Como deseamos crear una VPN dentro de la empresa debemos saber que ventajas que nos puede proporcionar esta tecnología, entre las principales tenemos:

- **Seguridad:** Por medio de una VPN podemos crear túneles en los cuales pasan la información encriptada entre los clientes por lo cual existe una integridad segura de los datos.
- **Autenticación y Autorización:** Solo se permiten conectarse a los equipos o dispositivos móviles autorizados, por medio de certificados de autenticación, llaves encriptadas y usuarios/contraseñas.
- **Velocidad:** Cuando enviamos o solicitamos información por medio de una red VPN es comprimida y descomprimida entre los 2 clientes de la VPN, esto hace que la VPN funcione más veloz en la transferencia de información.
- **Costos:** Un VPN nos ahorra en costo de los equipos y otros servicios que se estén ofreciendo dentro de la red local.

3.2.1 TIPOS DE VPN

Para poder implementar la red VPN tendremos que saber qué tipos de VPN existe para poder instalarla y configurarle en nuestra red local, existen 3 tipos de redes VPN:

- **Host to Host:** Nos permite la comunicación entre dos máquinas de las cuales solo tienen conexión entre ellos, esto quiere decir que solo exista la comunicación por medio de la VPN entre estos 2 equipos y pueden estar dentro de una red local o en internet.

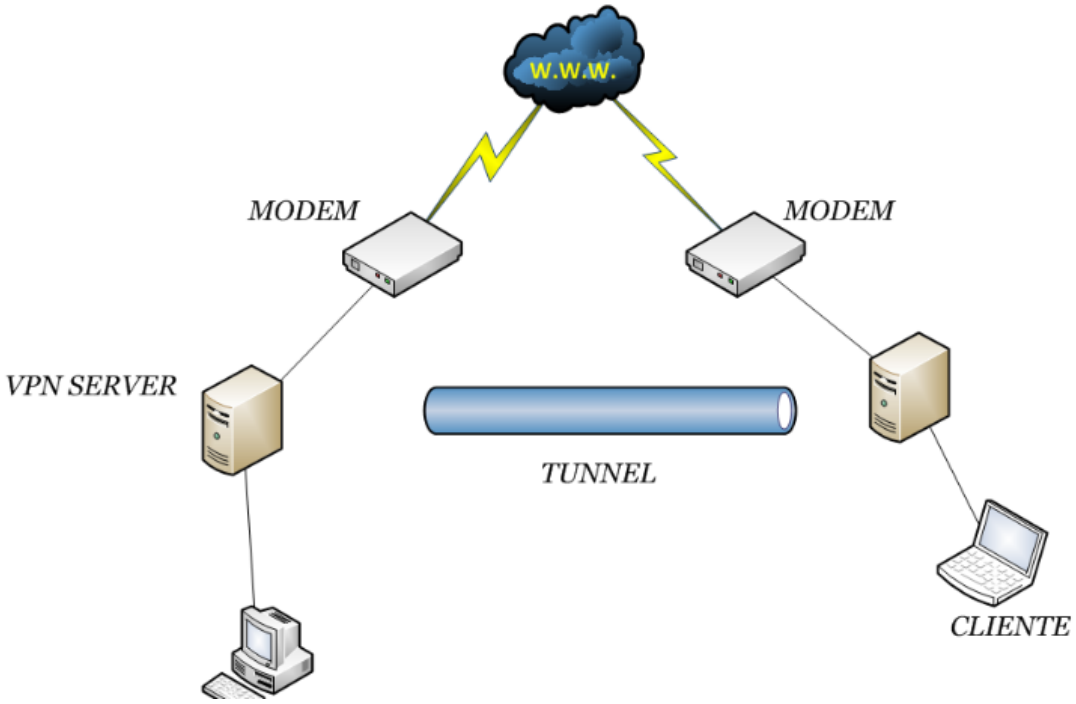


Figura 3.2-1 VPN Host to Host

- **Road Warrior:** Esta una de las formas más solicitadas y la que vamos a utilizar ya que permite que un conjunto de máquinas ya sean de la red local o de internet se conecten dentro de la red VPN, existiendo un servidor en el controle las conexiones, todas estas conexiones se realizan mediante certificados de autenticación.

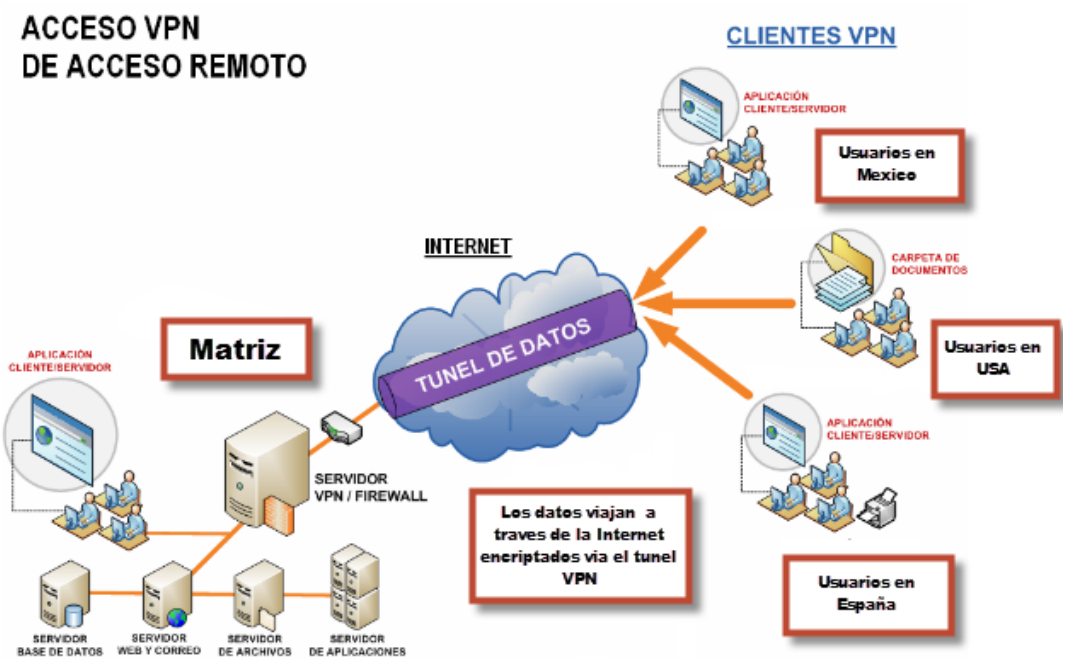


Figura 3.2-2 VPN Road Warrior

- **Net to Net:** Mediante esta forma conectamos 2 a varias redes LAN en lugares física apartados, la conexión entre las redes viajara encriptada, con esto podremos acceder a cualquier recurso de la red que se encuentre en el otro extremo de la VPN.

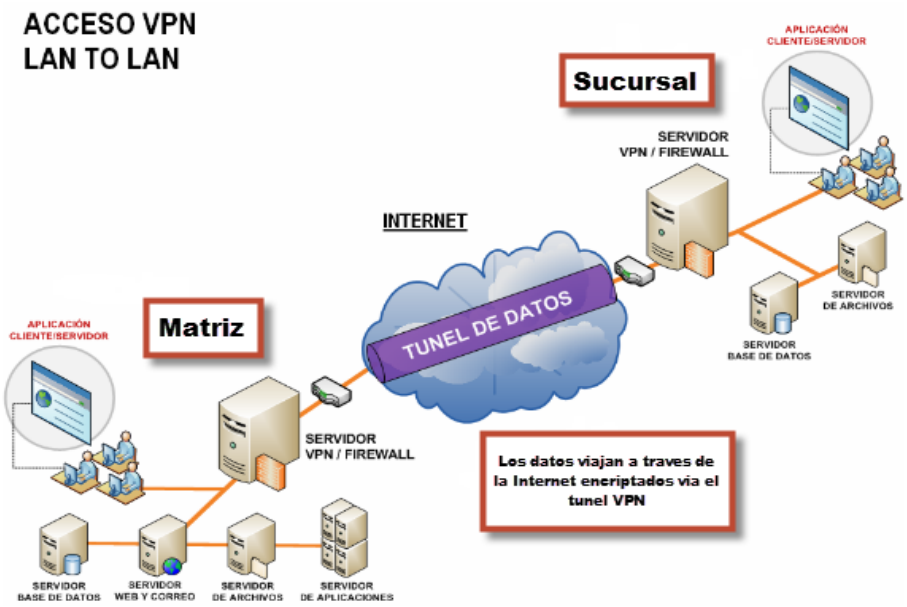


Figura 3.2-3 VPN Net to Net

3.2.2 SERVIDORES VPN

Tenemos varios servicios de los cuales nos permite crear estos túneles en GNU/Linux, los más conocidos son:

- **OpenVPN:** Es una solución completa de conexión de redes VPN, contiene validación de usuario, enviado de informaron encriptada.
- **PPTP:** Este el protocolo que realiza conexiones punto a punto, su principal habilidad la conexión múltiple de protocolos dentro del servicio.
- **Openswan:** Es una implementación de IPSec, son varios protocolos cuya función es garantizar la comunicación sobre el protocolo de internet, permite la autenticación y cifrado.

3.3 USO DE OPENVPN COMO HERRAMIENTA DE ACCESO REMOTO.

3.3.1 INTRODUCCION

OpenVPN es una solución de conectividad basada en software libre: SSL (Secure Sockets Layer) Openvpn ofrece conectividad punto-a-punto con validación jerárquica de usuarios y host conectados remotamente, soporta una amplia configuración, entre ellas balanceo de cargas. Está publicado bajo la licencia GPL, de software libre.

3.3.2 CARACTERISTICAS PRINCIPALES

El componente Principal es el driver tun/tap utilizada para simular interfaces de red, que se encarga de levantar el túnel y encapsular los paquetes a través del enlace virtual, además tiene las siguientes características:

- Encriptación y autenticación con OpenSSL
- Utiliza un único puerto TCP p UDP
- Multiplataforma
- Comprensión de datos LZO

3.3.3 MODOS DE FUNCIONAMIENTO

Openvpn VPN puede trabajar en dos modos: modo “TUN” (modo túnel), o modo “TAP” (modo bridge). Ambos modos utilizan el adaptador TUN/TAP detallado en el glosario, en concreto, utilizando el adaptador TUN para transmitir trafico IP a lo largo del túnel o, por el contrario, utilizando el adaptador TAP para transmitir trafico Ethernet por el túnel.

- **Modo Túnel:** Como ya se ha comentado el modo TUN o modo Túnel establece un enlace IP punto a punto virtual mediante la utilización del dispositivo virtual TUN que proporciona OPENVPN.
- **Modo Puente:** El modo Puente o modo TAP es capaz de unir, mediante un puente, dos redes locales Ethernet (separadas por Internet, por ejemplo), mediante el uso del dispositivo virtual TAP que ofrece OPENVPN. Es decir, el modo bridge es una técnica para crear redes de área local (LAN) virtuales.

3.3.4 AUTENTICACION

OpenVPN soporta diferentes métodos de autenticación desde cifrado convencional usando llaves secretas pre-compartidas (Static Key mode) o métodos de autenticación basada en llaves públicas (SSL/TLS mode) usando certificados X.509 para el servidor y clientes VPN.

- **Autenticación basada en llaves estáticas pre compartida.-** OpenVPN soporta cifrado convencional usando llaves secretas pre-compartidas usando el modo Static Key, las llaves estáticas son usadas tanto para la autenticación y la autorización.

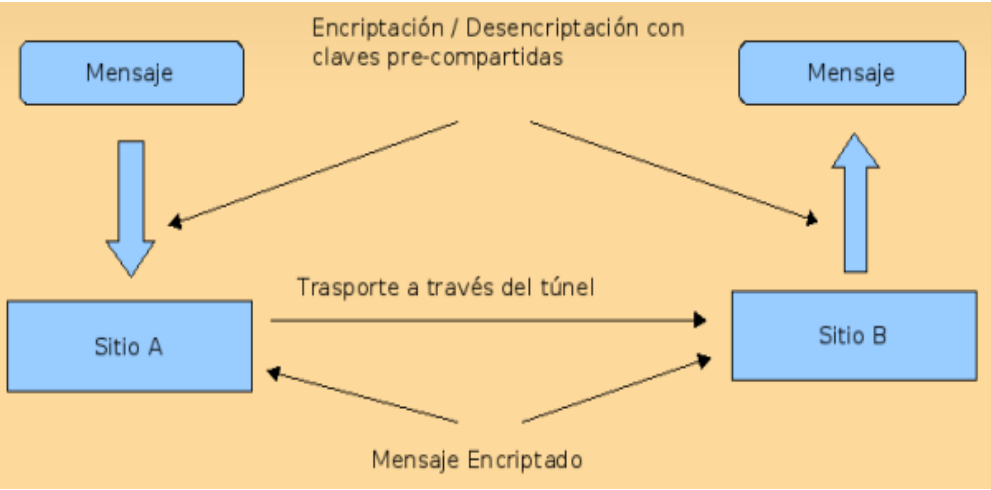


Figura 3.3-1 Autenticación basada en llaves pre compartidas

- **Autenticación basada en certificados X.509.-** En la autenticación basada en certificados SSL, OpenVPN, el certificado raíz es usado para validar la autenticidad del certificado del servidor OpenVPN y de los clientes VPN, es decir, se realiza una autenticación mutua, el cliente valida la autenticidad del certificado con el que se identifica el servidor y el servidor valida la autenticidad del certificado con el que se identifica el cliente.

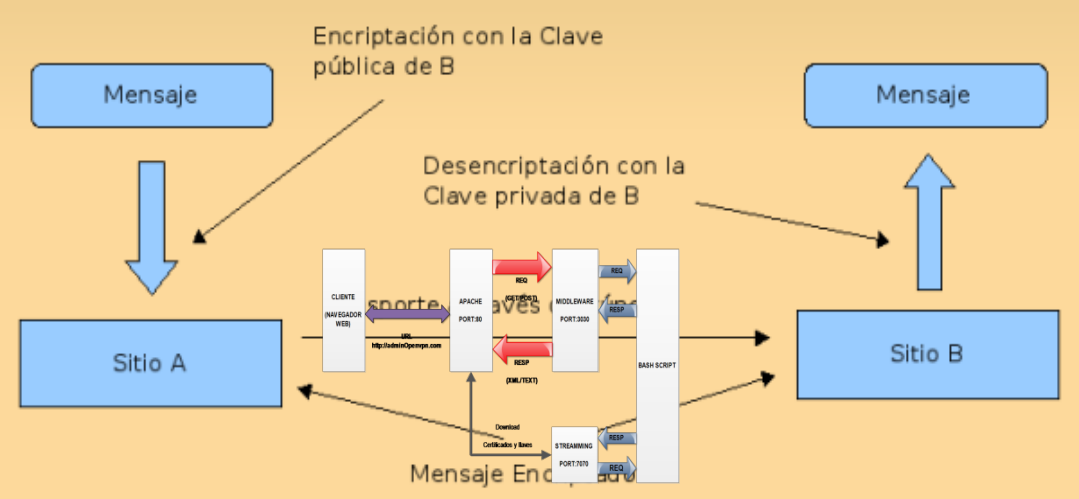


Figura 3.3-2 Autenticacion basada en certificado X.509

3.4 DISEÑO DE HERRAMIENTAS TECNOLOGICAS PARA FACILITAR LA INSTALACION, CONFIGURACION Y ADMINISTRACION DEL OPENVPN

Para facilitar la administración del servidor de acceso remoto, se desarrolló un método automatizado que permite realizar la instalación, configuración y administración del servicio VPN (OPENVPN).

Como parte del proceso de diseño a continuación mostramos el esquema del diagrama de arquitectura en la figura 3.4-1, usado para la herramienta de administración de acceso remoto.

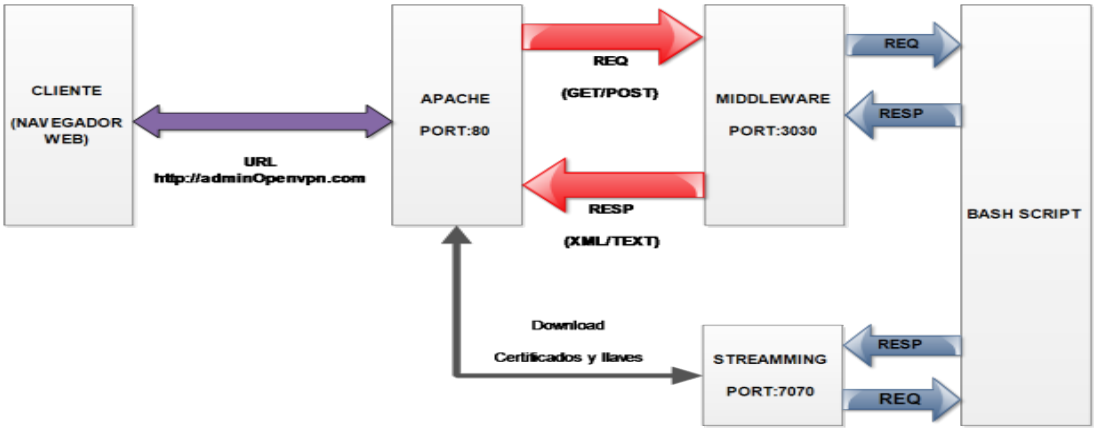


Figura 3.4-1 Arquitectura de la herramienta

La arquitectura que utilizamos es cliente servidor en donde el cliente realiza peticiones al servidor a través de un agente MIDDLEWARE que asiste a una aplicación para interactuar y comunicarse con el servidor.

El cliente envía peticiones hacia MIDDLEWARE; éste recibe la petición, los procesa y los reenvía hacia el servidor a través de un puerto específico previamente configurado.

El Streaming nos permite realizar la descarga de los certificados en el cliente.

Además desarrollamos scripts con la capacidad de instalar y configurar los parámetros de la VPN/LINUX y una herramienta web que facilita la administración del servicio.

A continuación se listan los archivos que se configuraron para la automatización de la herramienta:

- Script: **Instalar.sh**
- Script: **init.sh**
- **Script: clear.sh**
- Script: **ca.sh**
- Script: **gcert.sh**
- Script: **scert.sh**
- Script: **dfile.sh**
- Script: **datacert.sh**
- Archivo de configuracion: **vars**
- Archivo de configuración: **new 2.html**
- Archivo de configuración: **setup.html**
- Archivo de configuración: **prueba1**

3.4.1 DETALLES DE LOS ARCHIVOS DE CONFIGURACION

- Instalar.sh: El Script Instalar.sh realiza la configuración inicial de la herramienta. Es el que se encarga de descargar los packetes necesarios para la implementación del túnel de acceso remoto. Además se encarga de crear los directorios donde irán alojados los archivos de configuración:

- **/MIDDLEWARE.-** Ruta de ubicación de los archivos de configuración e inicialización de los servicios (middleware, httpd, streaming), cuanta con los siguientes archivos:

  Init.sh: Este script se encarga de iniciar los servicios que utiliza la herramienta de administración del OPENVPN.

  Clear.sh: Borra cualquier tipo de certificado existente

Ca.sh: Permite la creación de la Autoridad Certificadora necesaria en la configuración del túnel.

Gcert.sh: Permite la creación de los certificados para los clientes OPENVPN.

Scert.sh: Permite la creación del certificado del servidor openvpn (único por servidor).

Dfile.sh: Comprime los certificados creados en un único archivo.

Middleware.js: Permite la interacción entre los scripts y la herramienta web.

Stream.js: Permite descargar los certificados comprimidos desde la aplicación web.

- **/var/www/httpd/:** Carpeta que contiene los archivos de configuración HTML y CSS de la herramienta web.

3.4.2 DISEÑO DE LA HERRAMIENTA WEB

La función principal de la herramienta es facilitar la configuración del túnel de acceso remoto al permitir administrarlo con una interfaz gráfica, está basada en HTML5 Y CSS e interactúa con el servidor gracias al middleware explicado anteriormente.

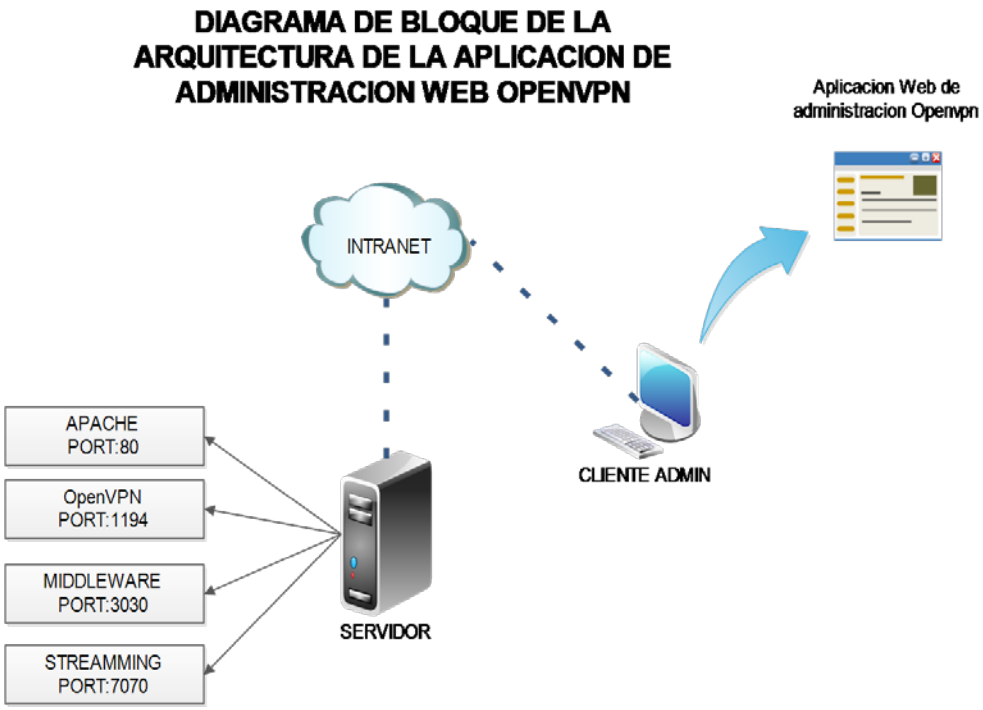


Figura 3.4-2 Arquitectura de la aplicación

Los archivos creados son los siguientes:

- **Index.html:** Menú principal de la herramienta, nos muestra un resumen del estado del servicio

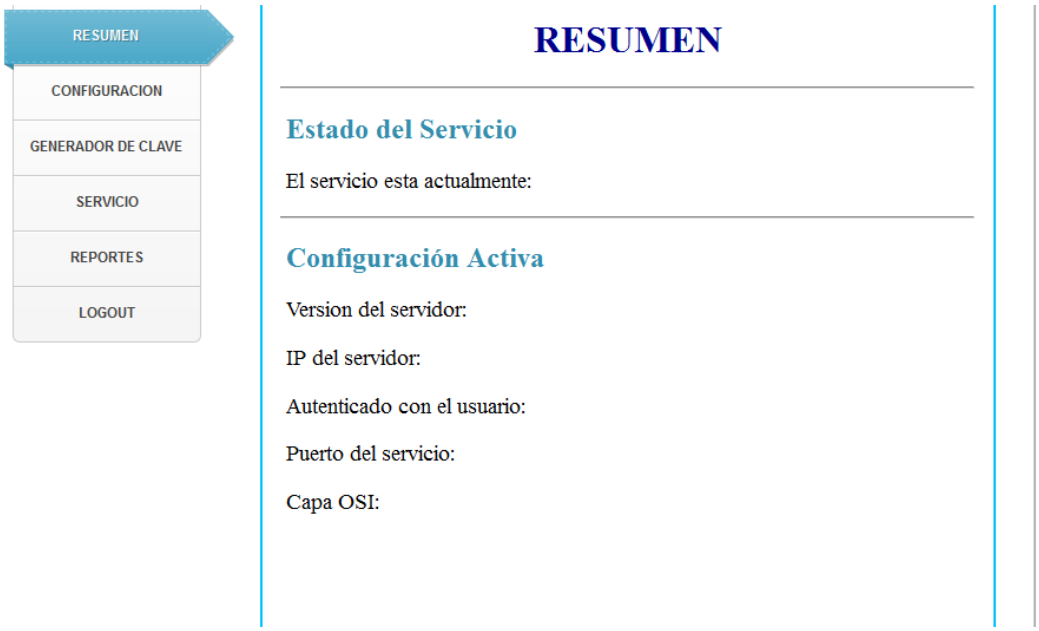


Figura 3.4-3 Botón "RESUMEN" de la página web

- **Setup.html:** Nos permite configurarlos parámetros del túnel VPN de acceso remoto.

Figura 3.4-4 Botón "CONFIGURACION" de la página web

- **Keygen.html:** Nos permite generar los certificados digitales para el cifrado d

Figura 3.4-5 Botón "GENERADOR DE LLAVE" de la pagina web

Serv.html: Nos permite dar de alta, baja y muestra un status del servicio

Figura 3.4-6 Botón "SERVICIO" de la página web

Logs.html: Nos muestra los reportes del servicio una establecida el túnel VPN.

Figura 3.4-7 Botón “REPORTES” de la página web

CAPITULO IV

CONFIGURACION DEL CANAL DE ACCESO REMOTO

4.1 REQUERIMIENTOS DE HARDWARE Y SOFTWARE

Para poder realizar implementación del servidor de acceso remoto se utilizaron varios componentes que detallamos a continuación:

Hardware:

El hardware recomendado para el servidor que tendrá implementado el canal de acceso remoto es:

- Memoria RAM de 2Gb
- Procesador para tener un óptimo rendimiento utilizar un Intel Quad Core i3
- Espacio de disco duro: de 250 Gb.

Software:

- Paquete OpenVpn de Linux.
- Nodejs para la Aplicación Middleware.
- Página basado en condigo HTML v5.
- APACHE de Linux para el servicio de la página web

4.2 CONFIGURACION MANUAL DE OPENVPN

A continuación se detalla los pasos para configurar el servicio de forma manual

1. Desde el root, se creó el fichero /etc/yum.repos.d/AL-Server.repo:

#vi /etc/yum.repos.d/AL-Server.repo, y se ingresó el siguiente contenido:

```
[AL-Server] name=AL Server para Enterprise Linux
mirrorlist=http://www.alcancelibre.org/al/el$releasever
/al-server gpgcheck=1
gpgkey=http://www.alcancelibre.org/al/AL-RPM-KEY
```

2. Se importó la firma digital de Alcance Libre ejecutando lo siguiente desde el root:

```
#rpm --import http://www.alcancelibre.org/al/AL-RPM-KEY
```

3. Se instaló desde los depósitos yum, los paquetes RPM de OpenVPN, vim-enhanced

```
#yum -y install openvpn shorewall vim-enhanced
```

Los siguientes procedimientos se realizaron sin salir del directorio: /etc/openvpn/

4. Dentro del directorio /etc/openvpn/ se copió los ficheros openssl.cnf, whichopensslcnf, pkitool y vars, localizados en /etc/openvpn/easy-rsa/2.0/:

```
cp /usr/share/openvpn/easy-rsa/2.0/openssl.cnf ./
cp/usr/share/openvpn/easy-rsa/2.0/whichopensslcnf ./
cp /usr/share/openvpn/easy-rsa/2.0/pkitool ./
cp /usr/share/openvpn/easy-rsa/2.0/vars ./
```

5. Se editó las últimas líneas del fichero /etc/openvpn/vars, que corresponden a lo siguiente:

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL=me@myhost.mydomain
Reemplazandolo por:
export KEY_COUNTRY="ECU"
export KEY_PROVINCE="Guayas"
```

```
export KEY_CITY="Guayaquil"
export KEY_ORG="hotmail"
export KEY_EMAIL=jeyvane@hotmail.com
```

Datos que corresponden a la ubicación del servidor VPN y el dominio o departamento de la organización o Empresa.

6. Para que se carguen las variables de entorno configuradas, se ejecutó la siguiente línea de comando:

source /etc/openvpn/./vars

7. Se ejecutó el fichero /usr/share/openvpn/easy-rsa/ 2.0/clean-all con sh.

sh /usr/share/openvpn/easy-rsa/2.0/clean-all

Esta línea de comando realiza una eliminación recursiva sobre el directorio: /etc/openvpn/keys, lo que significa que elimina todos los certificados y firmas digitales que hubieran existido con anterioridad.

8. Se creó el certificado del servidor:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-ca
```

9. Se creó el fichero dh1024.pem, el cual contiene los parámetros del protocolo Diffie-Hellman, de 1024 bits:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-dh
```

El protocolo Diffie-Hellman permite el intercambio secreto de claves entre dos partes. Este protocolo es usado para el cifrado de una sesión.

10. Se generó la firma digital con la siguiente línea de comando:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-key-server server
```

11. Se creó los certificados para ambos usuarios de la PBX (para los usuarios 200.126.13.215 y 200.126.23.219). Con las líneas de comando:

```
sh /usr/share/openvpn/easy-rsa/2.0/build-key cliente1
sh /usr/share/openvpn/easy-rsa/2.0/build-key cliente2
```

12. Se hizo uso de los certificados creados y las configuraciones realizadas, en el fichero:

vi /etc/openvpn/servidorvpn-udp-1194.conf, editándolo con lo siguiente:

```
port 1194
proto udp
dev tun #---- Seccion de llaves -----
ca keys/ca.crt
cert keys/server.crt
key keys/server.key
dh keys/dh1024.pem
#----------------------------
server 10.10.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status-servidorvpn-udp-1194.log
verb 3
```

Se ingresó la IP 10.0.8.0, porque es recomendable usar una red privada para evitar conflictos entre los host del Sistema VoIP cuando el túnel se encuentre

activo. Con mascara 255.255.255.0 porque permitirá a 253 clientes conectarse a la VPN.

13. Se usó el mandato restorecon sobre el directorio /etc/openvpn a fin de asignar los contextos adecuados.

```
restorecon -R /etc/openvpn/
```

14. Se crearon los ficheros ipp.txt y openvpn-status-servidorvpn-udp-1194.log:

```
cd /etc/openvpn/

touch ipp.txt
touch openvpn-status-servidorvpn-udp-1194.log
```

15. Se aplicó contextos de lectura y escritura (openvpn_etc_rw_t) a los ficheros que contiene el directorio /etc/openvpn:

```
cd /etc/openvpn/
chcon -u system_u -r object_r -t openvpn_etc_rw_t
ipp.txt chcon -u system_u -r object_r -t openvpn_etc_rw_t
openvpn-status-servidorvpn-udp-1194.log
```

Esto cambia los contextos a usuario de sistema (system_u), rol de objeto (object_r) y tipo configuración de OpenVPN de lectura y escritura (openvpn_etc_rw_t).

16. Se inició el servicio openvpn:

```
service openvpn Start
```

17. Para que el servicio de OpenVPN esté activo en el siguiente inicio del sistema, se utilizó el mandato chkconfig de la siguiente forma:

```
Chkconfig openvpn on
```

4.3 AUTOMATIZACION DEL SERVICIO OPENVPN EN UNA HERRAMIENTA WEB

Para realizar la configuración del servidor OpenVpn de forma dinámica se procedió a crear varios scripts que realizan la configuración completa desde la herramienta web. Los scripts se detallan en el siguiente subcapítulo.

4.3.1 SCRIPT DE INSTALACION DE LOS SERCICIOS NECESARIOS

EL script instalar.sh es el que realiza las descargas de los paquetes necesarios para levantar el servidor VPN. En la figura xxx se muestra el contenido y la descripción del script de instalación **instalar.sh**

instalar.sh

```
#!/bin/bash
yum update -y
yum install nano wget -y
yum install gcc make rpm-build autoconf.noarch zlib-devel pam-devel openssl-devel -y
cd /root
rm -r *.rpm
wget http://openvpn.net/release/lzo-1.08-4.rf.src.rpm
wget http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.i686.rpm
pwd
rpmbuild --rebuild lzo-1.08-4.rf.src.rpm
rpm -Uvh lzo-*.rpm
rpm -Uvh rpmforge-release*
yum install openvpn -y
cp -R /usr/share/doc/openvpn-2.2.2/easy-rsa/ /etc/openvpn/
sed -i '29c\export KEY_CONFIG=\/etc\/openvpn\/easy-rsa\/2.0\/openssl-1.0.0.cnf' /etc/openvpn/easy-rsa/2.0/vars
rm -r /etc/openvpn/server.conf
touch /etc/openvpn/server.conf
echo 'port 1194 #- port' >> /etc/openvpn/server.conf
echo 'proto udp #- protocol' >> /etc/openvpn/server.conf
echo 'dev tun' >> /etc/openvpn/server.conf
echo 'tun-mtu 1500' >> /etc/openvpn/server.conf
echo 'tun-mtu-extra 32' >> /etc/openvpn/server.conf
echo 'mssfix 1450' >> /etc/openvpn/server.conf
echo 'reneg-sec 0' >> /etc/openvpn/server.conf
echo 'ca /etc/openvpn/easy-rsa/2.0/keys/ca.crt' >> /etc/openvpn/server.conf
echo 'cert /etc/openvpn/easy-rsa/2.0/keys/server.crt' >> /etc/openvpn/server.conf
echo 'key /etc/openvpn/easy-rsa/2.0/keys/server.key' >> /etc/openvpn/server.conf
echo 'dh /etc/openvpn/easy-rsa/2.0/keys/dh1024.pem' >> /etc/openvpn/server.conf
echo 'plugin /usr/share/openvpn/plugin/lib/openvpn-auth-pam.so /etc/pam.d/login' >> /etc/openvpn/server.conf
echo 'client-cert-not-required' >> /etc/openvpn/server.conf
echo 'username-as-common-name' >> /etc/openvpn/server.conf
echo 'server 10.8.0.0 255.255.255.0' >> /etc/openvpn/server.conf
echo 'push "redirect-gateway def1"' >> /etc/openvpn/server.conf
```

instalar.sh

```
echo 'push "dhcp-option DNS 8.8.8.8"' >> /etc/openvpn/server.conf
echo 'push "dhcp-option DNS 8.8.4.4"' >> /etc/openvpn/server.conf
echo 'keepalive 5 30' >> /etc/openvpn/server.conf
echo 'comp-lzo' >> /etc/openvpn/server.conf
echo 'persist-key' >> /etc/openvpn/server.conf
echo 'persist-tun' >> /etc/openvpn/server.conf
echo 'status 1194.log' >> /etc/openvpn/server.conf
echo 'verb 3' >> /etc/openvpn/server.conf
echo 0 > /selinux/enforce
sed -i '7c\SELINUX=disabled' /etc/selinux/config
sed -i '7c\net.ipv4.ip_forward = 1' /etc/sysctl.conf
sysctl -p
cd /etc/openvpn/easy-rsa/2.0/
chmod 755 *
source ./vars
./vars
./clean-all
chkconfig openvpn on
yum -y groupinstall "Development Tools"
cd /usr/src
wget http://nodejs.org/dist/v0.10.4/node-v0.10.4.tar.gz
tar zxf node-v0.10.4.tar.gz
cd node-v0.10.4
make
make install
npm -g install express supervisor
cd /var/
mkdir middleware
cd middleware/
mkdir downloads
mkdir scripts
cd /var/middleware/
npm install shelljs
npm install ssh2
npm install url
npm install express
yum install httpd -y
chkconfig --levels 235 httpd on
```

Figura 4.3-1 Contenido del script “Instalar sh”

1.- se realiza las instalaciones de los paquetes necesarios:

- Nano
- OpenVpn

2-3.- Creacion del archivo “server.conf”. Que contiene los parámetros necesarios para la creación del túnel que son:

- ✓ Port
- ✓ Proto
- ✓ Dev tun
- ✓ Ip Server
- ✓ Dns Server

4-5-6.- se procede a realizar la desactivación del SELINUX, ubicado en el directorio /etc/selinux/config luego se modifica el bit de 0 a 1 para que permita el reenvió de paquetes. Luego se procede a instalar el Nodejs, y a crear los directorios donde se van alojar los archivos de configuración y finalmente se realiza la descarga de los paquetes restantes:

- Shh2.
- Httpd.
- Shell js.

4.3.2 SCRIPT´S DE CONFIGURACION

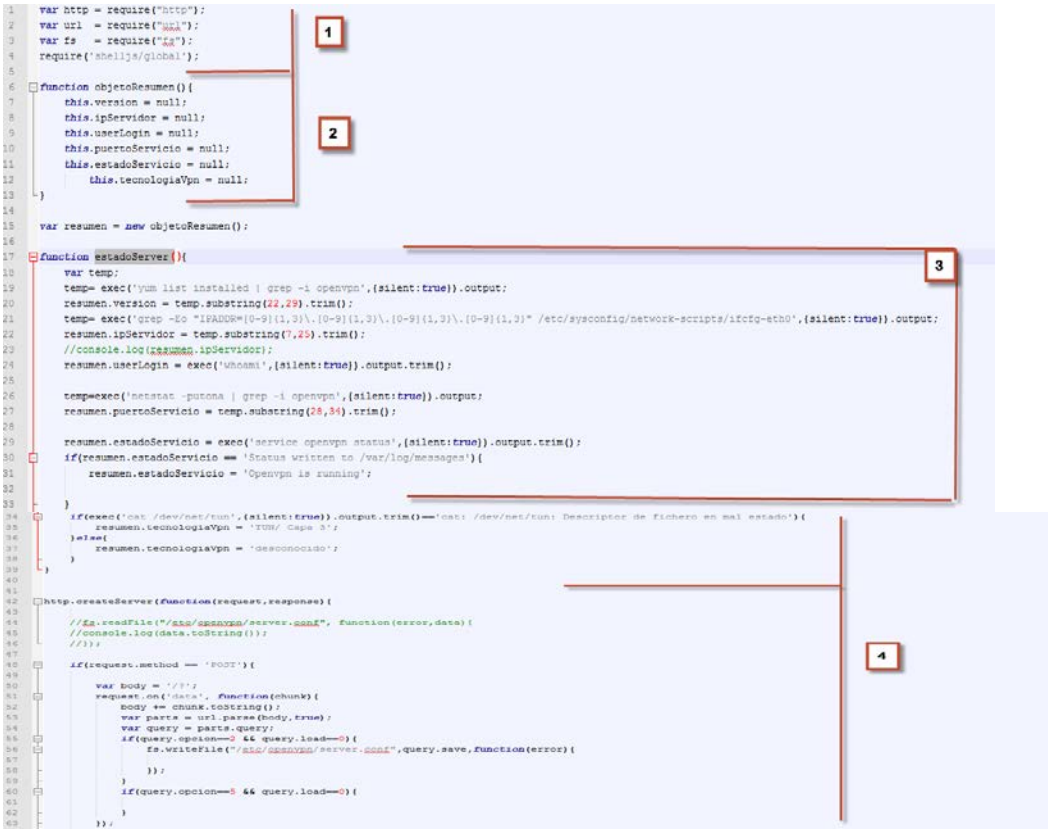
En los directorios creados se van a alojar los siguientes scripts de configuración :

- /var/middleware

- /var/middleware/scripts
- /var/downloads

La carpeta/var/middleware contiene los siguientes scripts:

- **/var/middleware/middleware.js:** Aplicación que se ejecutara en segundo plano y que escuchara a través del puerto 3030 todas las peticiones para luego por medio métodos y procedimientos ejecutar bloques de código que contienen sentencias bash, e interactuar directamente con los ficheros y directorios del servidor.



```
            var parts = url.parse(body,true);
            var query = parts.query;
            var dataFile  = 'error !!';
            response.writeHead(200, "OK" , {'Content-Type': 'text/plain',
                'Access-Control-Allow-Origin' : '*',
                'Access-Control-Allow-Methods': 'GET,PUT,POST,DELETE'
            });
            if(query.opcion==2 && query.load==1){
                dataFile = fs.readFileSync("/etc/openvpn/server.conf","utf8");
                response.write(dataFile);

            }
            if(query.opcion==5 && query.load==1){
                dataFile = fs.readFileSync("/etc/openvpn/1194.log","utf8");
                response.write(dataFile);
            }
            response.end();
        });
    }
    else{

        var parts = url.parse(request.url,true);
        var query = parts.query;
        //console.log(query.opcion);
        //console.log(request.url);


        response.writeHead(200, {
            'Content-Type': 'text/xml',
            'Access-Control-Allow-Origin' : '*',
            'Access-Control-Allow-Methods': 'GET,PUT,POST,DELETE'
        });

    if(query.opcion==1){
        estadoServer();
        response.write("<dataResumen>");
        response.write("<statusService>"+resumen.estadoServicio+"</statusService>");
        response.write("<versionService>"+resumen.version+"</versionService>");
        response.write("<ipService>"+resumen.ipServidor+"</ipService>");
        response.write("<userLogin>"+resumen.userLogin+"</userLogin>");
        response.write("<portService>"+resumen.puertoServicio+"</portService>");
        response.write("<tecnoService>"+resumen.tecnologiaVpn+"</tecnoService>");
        response.write("</dataResumen>");
    }else if(query.opcion==3){
        var str = query.datos;
        var element = str.split('#');

        if(query.save=='1'){
            exec('/var/middleware/scripts/./clear.sh',{silent:true});
            response.write("<estado>TODAS LAS LLAVES HAN SIDO  ELIMINADAS !!</estado>");
        }else if(query.save=='2'){
            exec('/var/middleware/scripts/./datacert.sh '+element[0]+' '+element[1]+' '+element[2]+' '+element[3]+' '+element[4]+' '+element[5]+' '+element
            exec('/var/middleware/scripts/./ca.sh',{silent:true});
            response.write("<estado>SE GENERO CORRECTAMENTE CA !! </estado>");
        }else if(query.save=='3'){
            exec('/var/middleware/scripts/./datacert.sh '+element[0]+' '+element[1]+' '+element[2]+' '+element[3]+' '+element[4]+' '+element[5]+' '+element
            if(element[6]=="server"){
                exec('/var/middleware/scripts/./scert.sh',{silent:true});
            }else{
                exec('/var/middleware/scripts/./ucert.sh '+query.load,{silent:true});
            }
            response.write("<estado>SE GENERO CORRECTAMENTE EL CERT !!</estado>");
        }else if(query.save=='4'){
            //console.log(query.load);
            exec('/var/middleware/scripts/./dfile.sh');
            response.write("<estado>DESCARGA CORRECTA DEL CERT !!</estado>");
        }else{

        }
    }else if(query.opcion==4){
        if(query.save=='1'){
            exec('service openvpn start',{silent:true});
            response.write("<estado>SERVICIO INICIADO !!</estado>");
        }else if(query.save=='2'){
            exec('service openvpn restart',{silent:true});
            response.write("<estado>SERVICIO REINICIADO !!</estado>");
        }else if(query.save=='3'){
            exec('service openvpn stop',{silent:true});
            response.write("<estado>SERVICIO DETENIDO !!</estado>");
        }else if(query.save=='4'){
            exec('service openvpn status',{silent:true});
            resumen.estadoServicio = exec('service openvpn status',{silent:true}).output.trim();
            if(resumen.estadoServicio == 'Status written to /var/log/messages'){
                resumen.estadoServicio = 'Openvpn is running';
            }else{
            resumen.estadoServicio = 'Openvpn is Stop';
            }
            response.write("<estado>"+resumen.estadoServicio+" !!</estado>");
        }else{
        }

    }else{
```

Figura 4.3-2 Contenido de script “middleware.js”

1.- Definición de módulos y librerías externas.

2.- Crea un objeto con todas las propiedades y métodos necesario, que contendrán la información relevante del servicio.

3.- esta parte del script realiza las siguientes funciones:

- Obtiene la informacion de la versión del openvpn
- Obtiene la IP del servidor.
- Obtiene el usuario que inicio sesión.
- Obtiene el puerto por el cual escucha el openvpn.
- Obtiene el tipo de tecnología para realizar el túnel.

4.- Método que permite escuchar peticiones a través del puerto 3030 por medio de requerimientos http (POST/GET) realiza las llamadas a los demás procedimientos y métodos dependiendo de la petición enviada desde el cliente.

- **/var/middleware/stream.js:** Aplicación que que se ejecutara en segundo plano y que escuchara a través del puerto 7070 todas las peticiones de streamming de manera que ejecute los métodos apropiadas para serializar lo objetos y enviarlos al cliente, para que puedan ser descargados

Figura 4.3-3 Contenido de script stram.js

- **/var/middleware/nat.sh:** Script que realiza la configuración de las reglas de NAT y el direccionamiento estático utilizado.


```
#!/bin/bash
cd /etc/sysconfig/network-scripts/
rm -Rf route-eth0
rm -Rf route-eth1
touch route-eth0
echo 'ADDRESS0=10.0.0.0' >> route-eth0
echo 'NETMASK0=255.255.255.252' >> route-eth0
echo 'GATEWAY0=10.0.0.5' >> route-eth0
echo ' ' >> route-eth0
echo 'ADDRESS1=11.0.0.0' >> route-eth0
echo 'NETMASK1=255.255.255.252' >> route-eth0
echo 'GATEWAY1=10.0.0.5' >> route-eth0
echo ' ' >> route-eth0
echo 'ADDRESS2=191.8.8.0' >> route-eth0
echo 'NETMASK2=255.255.255.0' >> route-eth0
echo 'GATEWAY2=10.0.0.5' >> route-eth0
touch route-eth1
echo 'ADDRESS0=172.21.8.0' >> route-eth1
echo 'NETMASK0=255.255.255.252' >> route-eth1
echo 'GATEWAY0=192.168.30.1' >> route-eth1
echo ' ' >> route-eth1
echo 'ADDRESS1=192.168.10.0' >> route-eth1
echo 'NETMASK1=255.255.255.0' >> route-eth1
echo 'GATEWAY1=192.168.30.1' >> route-eth1
echo ' ' >> route-eth1
echo 'ADDRESS2=192.168.20.0' >> route-eth1
echo 'NETMASK2=255.255.255.0' >> route-eth1
echo 'GATEWAY2=192.168.30.1' >> route-eth1
echo ' ' >> route-eth1
echo 'ADDRESS2=192.168.30.0' >> route-eth1
echo 'NETMASK2=255.255.255.252' >> route-eth1
echo 'GATEWAY2=192.168.30.1' >> route-eth1
 echo ' ' > /etc/sysconfig/iptables
 iptables -F
 iptables -X
 iptables -Z
 iptables -t nat -F
 iptables -P INPUT ACCEPT
 iptables -P OUTPUT ACCEPT
 iptables -P FORWARD ACCEPT
 iptables -t nat -P PREROUTING ACCEPT
 iptables -t nat -P POSTROUTING ACCEPT
 iptables -A INPUT -i lo -j ACCEPT

 iptables -A INPUT -i eth0 -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
 iptables -A INPUT -i eth0 -p tcp -m tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
 iptables -A INPUT -i eth0 -p tcp -m tcp --dport 7070 -m state --state NEW,ESTABLISHED -j ACCEPT
 iptables -A INPUT -i eth0 -p tcp -m tcp --dport 3030 -m state --state NEW,ESTABLISHED -j ACCEPT
 iptables -A INPUT -i eth0 -p udp -m udp --dport 1194 -m state --state NEW,ESTABLISHED -j ACCEPT


 iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

 iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
 iptables -A FORWARD -s 192.168.30.0/30 -j ACCEPT
 iptables -A FORWARD -s 192.168.20.0/30 -j ACCEPT
 iptables -A FORWARD -s 192.168.10.0/30 -j ACCEPT
 iptables -A FORWARD -s 10.8.0.0/24 -j ACCEPT
 iptables -A FORWARD -j REJECT --reject-with icmp-port-unreachable

 iptables -A OUTPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT

 iptables -t nat -A POSTROUTING -s 192.168.30.0/30 -o tun0 -j MASQUERADE
 iptables -t nat -A POSTROUTING -s 192.168.20.0/30 -o tun0 -j MASQUERADE
 iptables -t nat -A POSTROUTING -s 192.168.10.0/30 -o tun0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth1 -j MASQUERADE
iptables -t nat -A POSTROUTING -j SNAT --to-source 192.168.30.2
iptables -t nat -A POSTROUTING -j SNAT --to-source 10.8.0.1

service iptables save
service iptables restart
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -L -n
reboot
```

Figura 4.3-4 Contenido de script “nat.sh”

1.- Se añaden rutas estáticas.

2.- Se borra las reglas de NAT existentes y se define que las zonas del firewall (IPTABLES) acepten todo el tráfico.

3.- Se establece regla de apertura de puertos, para los servicios configurados: SSH, APACHE, MIDDLEWARE, STREAMMING, OPENVPN; y se acepta el paquete ICMP entrante.

4.- Se acepta todo el trabajo que haya sido establecido en las reglas anteriores, se reenviar el trafico entrante de la red 10.8.0.0/24 y se rechaza cualquier ping que no haya alcanzado la redes establecidas.

5.- Se acepta el ICMP saliente.

6.- Se establece la regla NAT de la red 10.8.0.0/24, luego se enruta por la interfaz eth1, luego se acepta todo tráfico para NAT que venga de la interfaz tun0

7.- Se guarda la configuración, se activa el reenvió de paquetes, y se reinicia el servidor.

- **/var/middleware/scripts/clear.sh:** Script que sirve para eliminar todos los certificados existentes y asi evitar inconvenientes de certificados duplicados.

ca.sh

```
#!/bin/bash
cd /etc/openvpn/easy-rsa/2.0/
source ./vars
./vars
./clean-all
cd keys/
openssl req -days 3650 -nodes -new -x509 -keyout ca.key -out ca.crt -config $KEY_CONFIG -batch
chmod 0600 ca.key
```

Figura 4.3-5 Contenido script “clears.sh”

- **/var/middleware/scripts/ca.sh:** En este script se procede a configurar los parámetros para la unidad que emite los certificados con cifrado SSL

ca.sh

```
#!/bin/bash
cd /etc/openvpn/easy-rsa/2.0/
source ./vars
./vars
./clean-all
cd keys/
openssl req -days 3650 -nodes -new -x509 -keyout ca.key -out ca.crt -config $KEY_CONFIG -batch
chmod 0600 ca.key

```

Figura 4.3-6 Contenido de script "ca.sh"

- **/var/middleware/scripts/gcert.sh:** Este script contiene los parámetros de configuración para el certificado del cliente con cifrado SSL.

gcert.sh

```
#!/bin/bash
echo $1
cd /etc/openvpn/easy-rsa/2.0/
source ./vars
./vars
cd keys/
openssl req -config $KEY_CONFIG -new -nodes -keyout $1.key -out $1.csr -days 3650 -batch
openssl ca -batch -config $KEY_CONFIG -policy policy_anything -out $1.crt -infiles $1.csr
chmod 0600 $1.key
cd ..
./build-dh

```

Figura 4.3-7 Contenido de script "gcert.sh"

- **/var/middleware/scripts/scert.sh:** Este script contiene los parámetros de configuración para el certificado del servidor con cifrado SSL.

scert.sh

```
#!/bin/bash
cd /etc/openvpn/easy-rsa/2.0/
source ./vars
./vars
cd keys/
openssl req -config $KEY_CONFIG -new -nodes -keyout server.key -out server.csr -days 3650 -batch
#openssl ca -batch -config $KEY_CONFIG -policy policy_anything -out $1.crt -infiles $1.csr
openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out server.crt
chmod 0600 server.key
```

Figura 4.3-8 Contenido de script “scert.sh”

- **/var/middleware/scripts/dfile.sh**: Este script realiza la tarea de comprimir los certificados creados y los guarda en la ruta /var/middleware/downloads con el nombre de “filekeys.tar.gz” para que puedan ser instalados en la maquina cliente y se efectué la conexión

dfile.sh

```
#!/bin/bash
cd /var/middleware/downloads/
rm -rf filekeys.tar.gz
cd /etc/openvpn/easy-rsa/2.0/
tar -zcvf filekeys.tar.gz keys/
cp filekeys.tar.gz /var/middleware/downloads
rm -rf filekeys.tar.gz
```

Figura 4.3-9 Contenido de script “dfile.sh”

4.3.3 ESTRUCTURA DE LA PAGINA HTML

La herramienta web está conformada por 5 páginas web que se ejecutan mediante el accionar de los botones de la página principal, a continuación se listan las páginas web creadas:

- Index.html
- Setup.html
- Keygen.html
- Logs.html
- New.html
- Serv.html

Las paginas esta configuradas en un ambiente jerárquico, en donde la página principal hace que se ejecuten las demás páginas con el accionar de los botones desde la página principal, como se muestra en la figura 4.3-10.

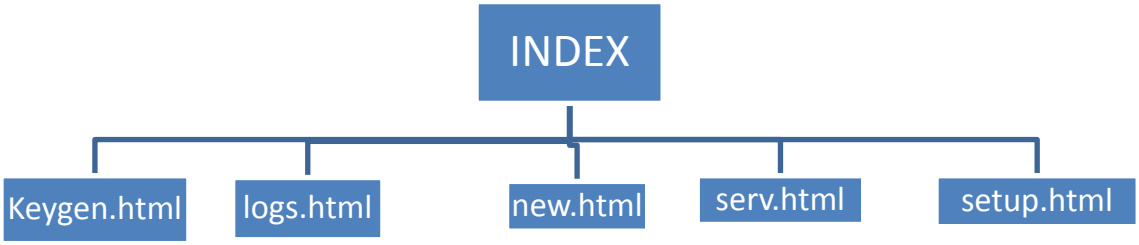


Figura 4.3-10 Estructura de página web

4.3.3.1 DETALLE DE LAS PAGINAS QUE CONFORMAN LA HERRAMIENTA WEB

Figura 4.3-11 Detalles de la página principal

En la figura 4.3-11 se muestra el cuerpo de la página como está configurada. La página Index.html es la página principal de la herramienta y en ella se puede ver toda la configuración del servidor que tiene en ese momento, mostrando:

- El estado del servicio si está activo/desactivo.
- versión del servidor.
- IP configurada del servidor VPN.
- Usuario autenticado.
- Puerto configurado del servicio.
- Capa del modelo OSI en la que se trabaja el servidor.

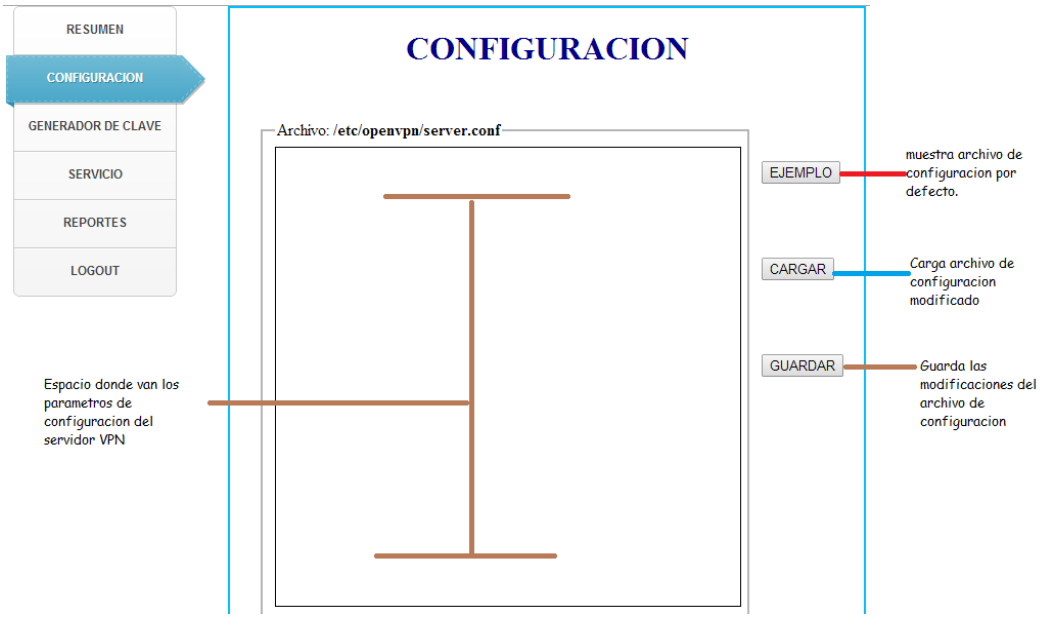


Figura 4.3-12 Detalles de la página de "CONFIGURACION"

En la figura 4.3-12 se muestra la página donde se procede a ingresar los datos necesarios para la configuración del servidor VPN que se muestran en el fichero /etc/openvpn/servidor/vpn-udp1194.conf.

Figura 4.3-13 Detalles de la página que configura las llaves

En la figura 4.3-13 se muestra la página donde se ingresan los datos que se necesita para crear el certificado, luego se procede a generarlos y descargarlos del servidor en un archivo comprimido. También se muestra el estado de servicio y se puede realizar borrado de los certificados creados.

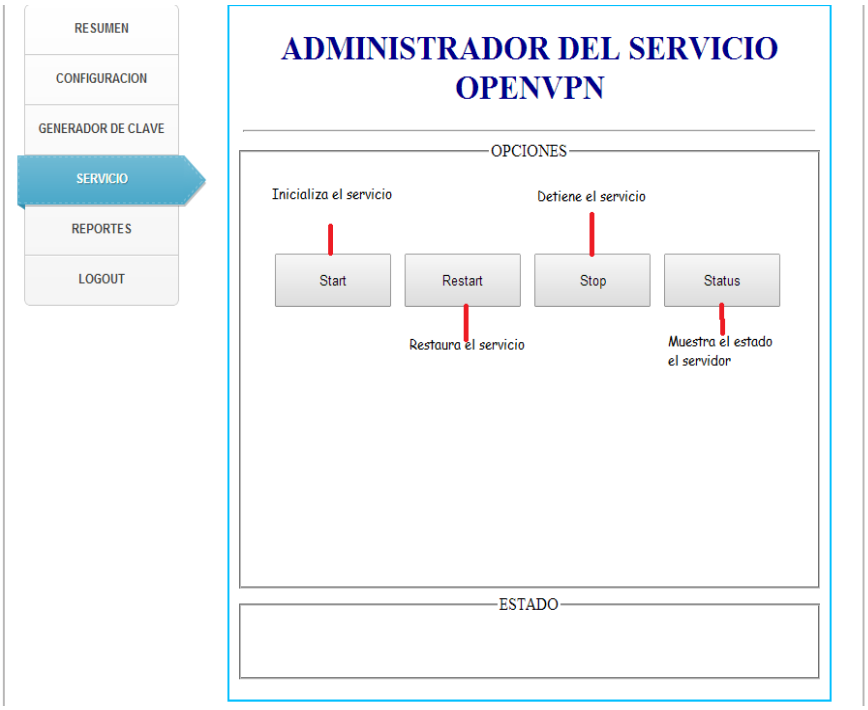


Figura 4.3 14 Detalles de la página que controla los estados del servidor

En la figura 4.3-14 se muestra la página que controla al servidor, debido a que desde esta página se puede realizar la función de:

- Ejecución del servicio.
- Restaurar el servicio.
- Detener el servicio.
- Ver el estatus del túnel.

4.4 INTERACCION DE LA PAGINA WEB CON EL MIDDLEWARE

Para que la página web pueda realizar todas las funciones configuradas en los botones de acción, es necesario hacer una interacción con el Middleware para que se pueda realizar la configuración que requiere el servidor desde la herramienta web. Para realizar esta interacción fue necesario crear los siguientes archivos:

- Conection.js
- Middleware.js

4.4.1 CONTENIDO DE SCRIPTS PARA INTERACION CON MIDDLEWARE

Archivo Conection: realiza los requerimientos al middleware mediante IP:PORT (10.0.0.6:3030) y PortStream = 7070 como se muestra en la figura 4.4-1

```
var IP = "10.0.0.6";
var portMiddleware = "3030";
var portStream = "7070";

function peticionSERVER(tab, metodo, data, tipoData, carga, params, url, async) {
    var req = $.ajax({
        type: metodo,
        async: async,
        url: url,
        dataType: tipoData,
        data: {
            opcion: tab,
            load: carga,
            save: data,
            datos: params
        }
    });
    req.done(function(msg) {
```

Figura 4.4-1 Contenido de archivo "Conection"

- En la figura 4.4-2 se realizan varias funciones dentro de una sentencia Switch con 5 casos:

```
switch (tab) {
    case 1:
        var a = $(msg).find('statusService').text();
        var b = $(msg).find('versionService').text();
        var c = $(msg).find('ipService').text();
        var x = $(msg).find('userLogin').text();
        var y = $(msg).find('portService').text();
        var z = $(msg).find('tecnoService').text();
        showResumen(a, b, c, x, y, z);
        break;
   case 2:
        if (carga === 1) {
            $("textarea#fileConf").val(msg.toString());
            $("#respuestaServer").text("CARGADO CORRECTAMENTE!!");
        }
        if (carga === 0) {
            $("#respuestaServer").text("GUARDADO CORRECTAMENTE!!");
        }
        break;
   case 3:
        if (data === "1") {
            $("#respuestaServerKey").text($(msg).find('estado').text());
        } else if (data === '2') {
            $("#respuestaServerKey").text($(msg).find('estado').text());
        } else if (data === '3') {
            $("#respuestaServerKey").text($(msg).find('estado').text());
        } else if (data === '4') {
            $("#respuestaServerKey").text($(msg).find('estado').text());
        } else {

        }
        break;
case 4:
    if (data === "1") {
        $("#respuestaServerSrv").text($(msg).find('estado').text());
    } else if (data === '2') {
        $("#respuestaServerSrv").text($(msg).find('estado').text());
    } else if (data === '3') {
        $("#respuestaServerSrv").text($(msg).find('estado').text());
    } else if (data === '4') {
        $("#respuestaServerSrv").text($(msg).find('estado').text());
    } else {

    }
    break;
case 5:
    showDataLog(msg.toString());
    $("#respuestaServerLog").text("OK !!");
    break;
default :
    alert("Error !!");
```

Figura 4.4-2 Contenido de archivo “Conection” sentencia switch

Case 1: Se realiza el requerimiento de resumen del servicio openvpn.

Case 2: Realiza **el** requerimiento del fichero **server.conf**. en donde se configura los parámetros del servidor como IP del servidor, Puerto, etc.

Caso 3: requerimiento de creación de llaves y certificados openvpn es donde se crean las llaves para poder hacer conexiones de redes remotas.

Case 4: Requerimiento de administración de servicio, esta función controla al servidor ya que puede iniciar, restaurar, detener al servidor y ver el estado del servidor en ese momento.

- En la figura 4.4-3 se muestra la función que permite mostrar en la página web el resumen de toda la configuración del servidor. Es decir Estado del servicio; versión del servicio; IP del servidor; Login del usuario; Puerto del servicio configurado;

```
function showResumen(estadoService, versionService, ipService, userLogin, puertoService, tecnoService) {
    $("#estadoS").text(estadoService);
    $("#versionS").text(versionService);
    $("#ipS").text(ipService);
    $("#userS").text(userLogin);
    $("#portS").text(puertoService);
    $("#layerS").text(tecnoService);
}
```

Figura 4.4-3 Contenido de archivo "Conection" estado del servidor

- La figura 4.4-4 muestra la función de requerimiento para guardar el archivo de configuración por defecto "**server.conf"** y poderlo mostrar como ejemplo en la página web accionando el botón "EJEMPLO"

```
function showDefaultConfig() {
    $("textarea#fileConf").val(confDefault());
    $("#respuestaServer").text("");
}
```

Figura 4.4-4 Contenido de archivo "Conection" botón "EJEMPLO"

- En la figura 4.4-5 muestra la Función de requerimiento que al momento de ejecutar el botón “CARGAR” de la página web muestra la configuración modificada del archivo **server.conf**

```
function seleccionarTipo() {
    var tipo = "";
    $("select").change(function() {
        tipo = $("select option:selected").text();
        if (tipo === "Servidor") {
            $("#name_cert").attr("disabled", "");
            $("#name_cert").val("server");
            $("#KEY_CN").val("server");
        } else if (tipo === "Cliente") {
            $("#name_cert").removeAttr("disabled");
            $("#name_cert").val("");
            $("#KEY_CN").val("");
        }
        else {
            $("#name_cert").removeAttr("disabled");
            $("#name_cert").val("");
            $("#KEY_CN").val("");

        }
    }).trigger("change");
}
function cambiarKEY_CN() {
    var nombre = "";
    $("#name_cert").keyup(function() {
        nombre =  $("#name_cert").val();
        $("#KEY_CN").val(nombre);



    });
}
```

Figura 4.4-5 Contenido archivo “Conection” botón “CARGAR”

- En la figura 4.4-6 muestra la función de requerimiento para limpiar certificados anteriores accionando el botón “BORRAR” de la página web.

```
function deleteCert() {
    $("#respuestaServerKey").text(' ');
    var opt = "1";
    peticionSERVER(3, "GET", opt, 'xml', 0, '', 'http://' + IP + ':' + portMiddleware, false);
}
```

Figura 4.4-6 Contenido de archivo “Conection” Botón “BORRAR”

- La figura 4.4-7 muestra la función de requerimiento para generar la unidad certificadora.

```
function genCA() {
    $("#respuestaServerKey").text(' ');
    var opt = "2";
    var KEY_SIZE = $("#KEY_SIZE").val();
    var KEY_COUNTRY = $("#KEY_COUNTRY").val();
    var KEY_PROVINCE = $("#KEY_PROVINCE").val();
    var KEY_CITY = $("#KEY_CITY").val();
    var KEY_ORG = $("#KEY_ORG").val();
    var KEY_EMAIL = $("#KEY_EMAIL").val();
    var KEY_CN = $("#KEY_CN").val();
    var parametros = KEY_SIZE + "#" + KEY_COUNTRY + "#" + KEY_PROVINCE + "#" + KEY_CITY + "#" + KEY_ORG + "#" + KEY_EMAIL + "#" + KEY_CN;
    peticionSERVER(3, "GET", opt, 'xml', 0, parametros, 'http://' + IP + ':' + portMiddleware, false);
}
```

Figura 4.4-7 Contenido de archivo “Conection” Genera certificado para CA

- La figura 4.4-8 muestra la función de requerimiento para crear certificados al momento que se ejecuta el botón “Generar Cert” de la página web.

```
function genCert() {
    $("#respuestaServerKey").text('ESPERE POR FAVOR !!');
    var name = $("#name_cert").val();
    var opt = "3";
    var KEY_SIZE = $("#KEY_SIZE").val();
    var KEY_COUNTRY = $("#KEY_COUNTRY").val();
    var KEY_PROVINCE = $("#KEY_PROVINCE").val();
    var KEY_CITY = $("#KEY_CITY").val();
    var KEY_ORG = $("#KEY_ORG").val();
    var KEY_EMAIL = $("#KEY_EMAIL").val();
    var KEY_CN = $("#KEY_CN").val();
    var parametros = KEY_SIZE + "#" + KEY_COUNTRY + "#" + KEY_PROVINCE + "#" + KEY_CITY + "#" + KEY_ORG + "#" + KEY_EMAIL + "#" + KEY_CN;
    peticionSERVER(3, "GET", opt, 'xml', name, parametros, 'http://' + IP + ':' + portMiddleware, false);
}
```

Figura 4.4-8 Contenido de archivo “Conection” Boton “Generar Cert”

- En la figura 4.4-9 muestra la función de requerimiento descargar el certificado creado y poderlo llevar hacia el cliente para poder realizar la autenticación.

```
function certFiles() {
    var opt = "4";
    peticionSERVER(3, "GET", opt, 'xml', 0, '', 'http://' + IP + ':' + portMiddleware, false);
    window.location.href = 'http://' + IP + ':' + portStream + '/vpns/keygen.html';
}
```

Figura 4.4-9 Contenido de archivo “Conection” Descarga de llaves comprimidas

- En la figura 4.4-10 se muestran las funciones de requerimiento que hacen que el servicio se inicie, se detenga o se restaure.

```
function initSrv() {
    var opt = "1";
    peticionSERVER(4, "GET", opt, 'xml', 0, '', 'http://' + IP + ':' + portMiddleware, false);
}


function restSrv() {
    var opt = "2";
    peticionSERVER(4, "GET", opt, 'xml', 0, '', 'http://' + IP + ':' + portMiddleware, false);
}
function stopSrv() {
    var opt = "3";
    peticionSERVER(4, "GET", opt, 'xml', 0, '', 'http://' + IP + ':' + portMiddleware, false);
}
```

Figura 4.4-10 Contenido de archivo "Conection" Administración del Servicio

- En la figura 4.4-11 se muestra la función de requerimiento que permite mostrar los reportes generados en el servidor. Al momento de ejecutar el botón "REPORTES" de la página web se muestran la siguiente información:

  - ✓ Nombre del certificado del cliente.
  - ✓ IP del equipo cliente.
  - ✓ Bits enviados y recibidos.
  - ✓ Fecha y hora de la conexión.
  - ✓ Dirección IP virtual.

```
function showReports() {
    var save = "0";
    peticionSERVER(8, "POST", save, 'text', 1, '', 'http://' + IP + ':' + portMiddleware, false);
}
function showDataLog(prueba) {
    var tam1 = 0;
    var tam2 = 0;
    var data1;
    var data2;
    var dataShow;
    var dataRoutes;
    data1 = prueba.match(/\w+\,[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\:\d+\,\d+\,\d+\,\w+\s+\w+\s+\d+\s+\d\d\:\d\d\:\d\d\s+\d\d\d\d/g);
    data2 = prueba.match(/[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\,\w+\,[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\:\d+\,\w+\s+\w+\s+\d+\s+\d\d\:

    if (data1 === null) {
        tam1 = 0;
    } else {
        tam1 = data1.length;
    }
    if (data2 === null) {
        tam2 = 0;
    } else {
        tam2 = data2.length;
    }
    $("#tabla1").text("");
    $("#tabla2").text("");
    $("#tabla1").append("<h4>Common Name</h4>");
    $("#tabla1").append("<h4>Real Address</h4>");
    $("#tabla1").append("<h4>Bytes Received</h4>");
    $("#tabla1").append("<h4>Bytes Sent</h4>");
    $("#tabla1").append("<h4>Connected Since</h4>");
    for (i = 0; i < tam1; i++) {
        dataShow = data1[i].split(",");
        $("#tabla1").append("<br/><b>" + dataShow[0] + "</b>");
        $("#tabla1").append("<b>" + dataShow[1] + "</b>");
        $("#tabla1").append("<b>" + dataShow[2] + "</b>");
        $("#tabla1").append("<b>" + dataShow[3] + "</b>");
        $("#tabla1").append("<b>" + dataShow[4] + "</b><br/>");
        //alert(data1[i]);
    }
    $("#tabla2").append("<h4>Virtual Address</h4>");
    $("#tabla2").append("<h4>Common Name</h4>");
    $("#tabla2").append("<h4>Real Address</h4>");
    $("#tabla2").append("<h4>Last Ref</h4>");
    for (i = 0; i < tam2; i++) {
        dataRoutes = data2[i].split(",");
        $("#tabla2").append("<br/><b>" + dataRoutes[0] + "</b>");
        $("#tabla2").append("<b>" + dataRoutes[1] + "</b>");
        $("#tabla2").append("<b>" + dataRoutes[2] + "</b>");
        $("#tabla2").append("<b>" + dataRoutes[3] + "</b>");
    }
}
```

Figura 4.4-11 Contenido del archivo “Conection” Botón Reportes

CAPITULO V

PRUEBAS DE ACCESO REMOTO Y DIAGNOSTICO A ERRORES

5.1 EMULACION DE UNA RED DE DATOS VIRTUALIZADA PARA PRUEBAS

Con el fin de verificar la funcionalidad de las herramientas creadas, decidimos diseñar una red virtualizada en GNS3 y Virtual Box para el ambiente de pruebas.

Como parte del proceso de diseño a continuación mostramos la topología de la red en la figura 5.1-1, usada para el ambiente de pruebas.

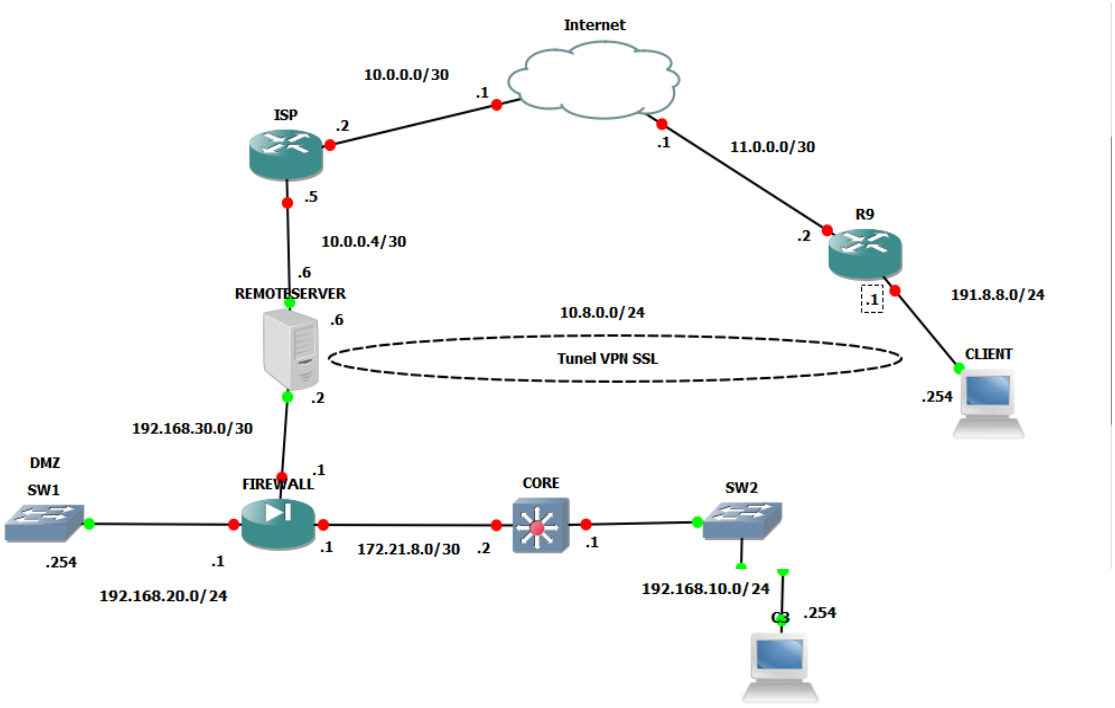


Figura 5.1-1 Diagrama de la red emulada

La red de datos que podemos observar en el grafico es la sugerida a aplicar en PETROGAS, ya que gracias a este diseño tenemos mejor seguridad al momento de acceder a la VPN ya que el tráfico interno es filtrado por el FIREWALL.

5.1.1 DISPOSITIVOS QUE CONFORMAN LA RED VIRTUALIZADA

Hemos dividido los dispositivos en 3 grupos para una mejor comprensión de la red: Dispositivos Internos, Dispositivos INTERNET, Dispositivos del Cliente.

- Dispositivos Internos: SWITCHES DE ACCESO,ROUTER CORE, FIREWALL, SERVIDOR VPN (REMOTESERVER), PC (WINDOWS XP)
- Dispositivos INTERNET: ROUTER ISP, NUBE WAN (SIMULANDO INTERNET), ROUTER R9
- Dispositivos del Cliente: PC-CLIENTE (WINDOWS XP)

5.1.2 DETALLES DEL DIRECCIONAMIENTO UTILIZADO

A continuación detallamos en la tabla #10 el esquema de direccionamiento utilizado por el ambiente virtualizado.

DIRECCIONAMIENTO DE LAN INTERNA		
Direccionamiento	*Especificación*	*IP'S UTILIZADAS*
192.168.10.0/24	*Segmento privado de los funcionarios*	*192.168.10.1 (Interfaz del Gateway) 192.168.10.254 (PC de la red Interna)*
192.168.20.0/24	*Segmento de la DMZ*	*192.168.20.1 (Interfaz DMZ del FIREWALL)*
192.168.30.0/30	*Segmento utilizado en la conexión del FIREWALL y el VPNSERVER*	*192.168.30.1 (Interfaz del FIREWALL hacia VPNSERVER)192.168.30.2 (INTERFAZ DEL VPNSERVER HACIA EL FIREWALL)*
172.21.8.0/30	*Segmento utilizado en la conexión del FIREWALL y el CORE*	*172.21.8.1 (Interfaz del FIREWALL hacia CORE) 172.21.8.2 (INTERFAZ DEL CORE HACIA EL FIREWALL)*
10.0.0.4/30	*Segmento utilizado en la conexión del VPNSERVER y el ROUTER ISP*	*10.0.0.6 (INTERFAZ DEL VPNSERVER HACIA EL ISP)10.0.0.5 (Interfaz del ISP hacia VPNSERVER)*

Tabla 10 Direccionamiento LAN interna

DIRECCIONAMIENTO DEL ISP		
Direccionamiento	*Especificación*	*IP'S UTILIZADAS*
10.0.0.0/30	*Segmento utilizado en la conexión del ISP y LA NUBE INTERNET*	*10.0.0.2 (INTERFAZ DEL ISP HACIA LA NUBE INTERNET) 10.0.0.0.1 (INTERFAZ DE LA NUBE INTERNETHACIA ISP)*
11.0.0.0/30	*Segmento utilizado en la conexión de la NUBE ISP hacia ROUTER CLIENTE (R9)*	*11.0.0.1 (INTERFAZ DE LA NUBE INTERNET HACIA R9)11.0.0.2 (INTERFAZ DEL R9 HACIA LA NUBE INTERNET)*
191.8.8.0/24	*SEGMENTO DEL CLIENTE*	*191.8.8.1 (DIRECCION DEL GATEWAY)191.8.8.254 (DIRECCION DEL PC-CLIENTE)*

Tabla 11 Direccionamiento del ISP

5.2 MANUAL DE USO DE LAS HERRAMIENTAS DESARROLLADAS

A continuación detallamos los pasos necesarios a seguir para el correcto funcionamiento de las herramientas desarrolladas, para ello aclaramos que el sistema operativo que debe estar corriendo en el servidor debe ser CENTOS 6 ya que los scripts fueron desarrollados específicamente para esta versión del S.O, además el quipo cliente debe tener instalada la aplicación OPENVPN GUI que la encontramos en www.openvpn.net

1. **EJECUTAR EL SCRIPT DE INSTALACION DE PAQUETES NECESARIOS (INSTALAR.SH)**

Para hacerlo copiamos el archivo instalar.sh en la raíz de nuestro CENTOS y lo ejecutamos, algo importante que debemos recalcar es que el servidor debe tener acceso a internet.

Figura 5.1-2 Ejecucion del script “init.sh”

sistema, además de descargar todos los paquetes necesarios para la integración del OPENVPN en ambiente web, entere los más importantes tenemos:

- OPENVPN
- NODEJS
- OPENSSL
- APACHE

Luego de descargar los archivos y configurar parámetros de inicialización crea el directorio /var/middleware y dentro de este crea los subdirectorios

/var/middleware/downloads y /var/middleware/scripts, en estos directorios se van a ubicar los scripts necesarios para la integración.

2. UBICAR LOS ARCHIVOS DENTRO DE LOS DIRECTORIOS CREADOS

Luego de haber ejecutado el script de instalación ubicamos el resto de scripts en los directorios creados tal como se detalla a continuación.

DIRECTORIO	SCRIPT
/var/middleware/scripts	ca.sh, clear.sh, datacert.sh,dfile.sh,gcert.sh,scert.sh
/var/middleware	middleware.js, stream.js, init.sh
/var/middleware/downloads	Ruta reservada para generar archivos con la herramienta web

Tabla 12 Ubicación de los archivos creados

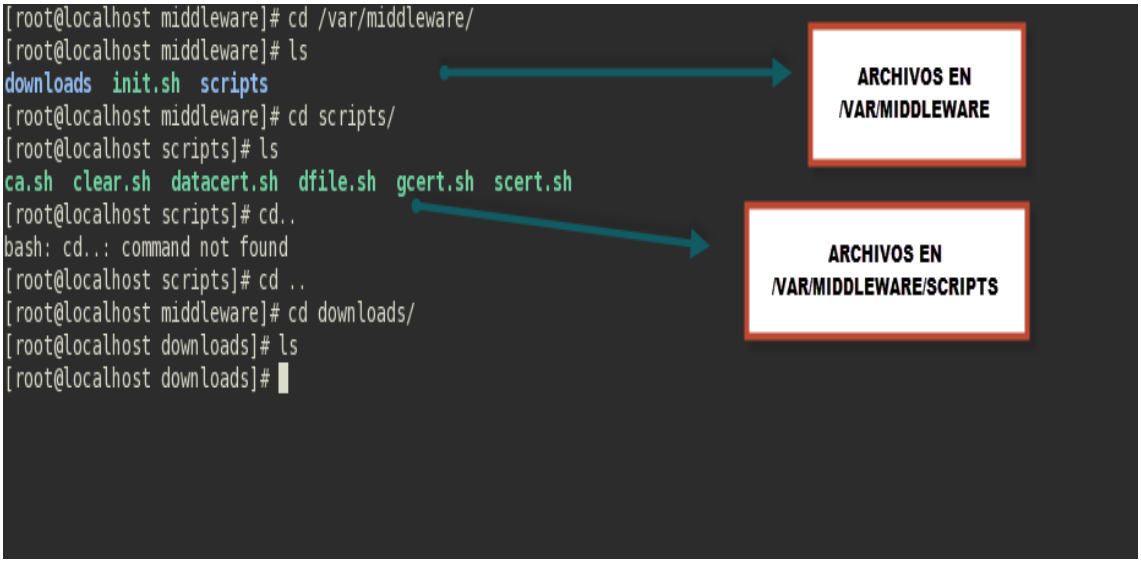


Figura 5.2-1 Ubicación de los archivos de configuración

Función de cada script

ca.sh: Se encarga de generar el certificado de la Autoridad certificador, único por servidor.

clear.sh: Se encarga de eliminar todos los certificados digitales existentes.

datacert.sh: Se encarga de

dfile.sh: Se encarga de

gcert.sh: Se encarga de

scert.sh: Se encarga de

middleware.js:

stream.js:

init.sh:

3. **Ubicar los archivos de la Pagina Web dentro de la ruta correspondiente**

La ruta donde debemos ubicarla es la ruta por default del apache web server /var/www/html, todos los archivos generados en la creación de la página web deben ir en esa ruta con el fin de que cuando inicialicemos el servicio apache nuestra página este activa.

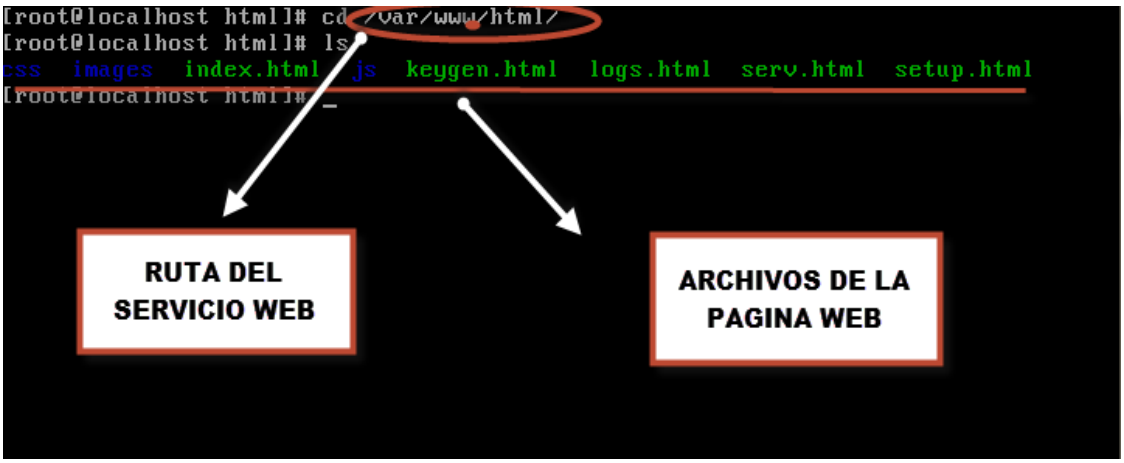


Figura 5.2-2 Ubicación de los archivos de la página web

4. **Ejecutar el Script de Nateo**

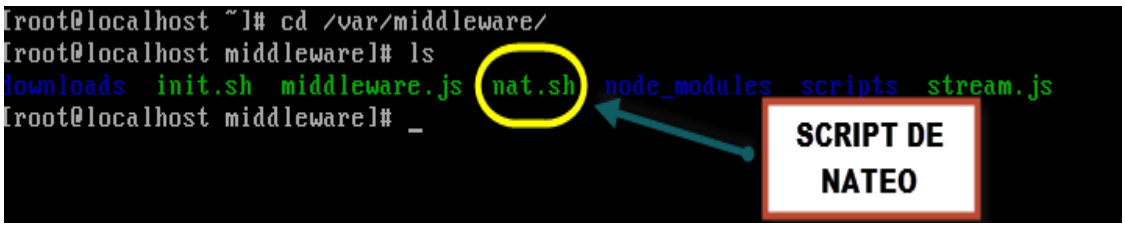


Figura 5.2-3 Ejecución del script de NAT

Con este script procederemos a configurar las reglas NAT en IPTABLES con el fin de intercambiar paquetes entre la red del cliente y la red LAN por medio del túnel virtual, luego de ejecutar este script se reiniciara el servidor.

5. **Ejecutar el Script de inicialización de los servicios init.sh**

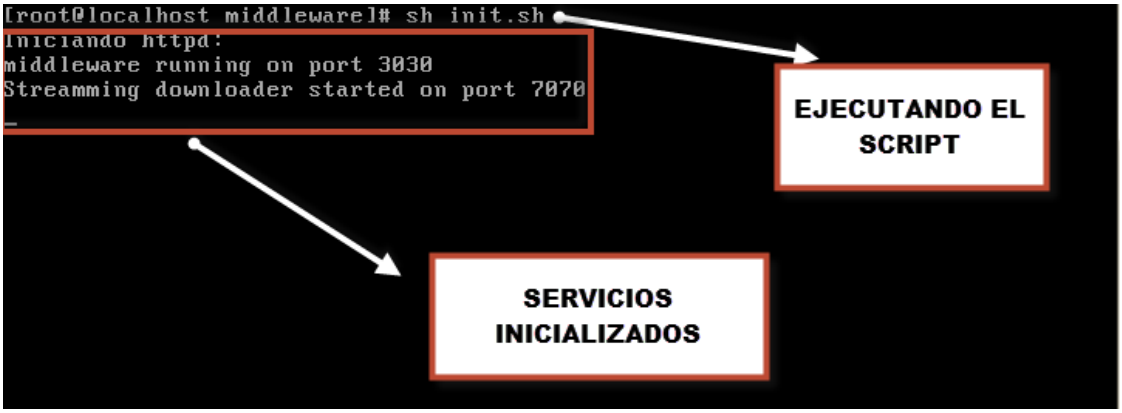


Figura 5.2-4 Ejecución de script “init.sh”

6. Iniciar la aplicación Web desde el cliente

Para ello ingresamos con nuestro navegador a la dirección IP del servidor configurado 10.0.0.6

Figura 5.2-5 Página principal de la aplicación

El menú principal de la herramienta nos muestra un resumen de la conexión en la que encontramos versión de OPENVPN que utilizamos, IP del servidor, nombre del usuario con el cual se inicializo los servicios en el servidor, puerto utilizado para el servicio VPN.

7. Configurar parámetros del túnel

Para ello nos dirigimos al botón CONFIGURACION y damos clic en el botón EJEMPLO, se cargara un ejemplo de configuración del túnel con todos los parámetros de configuración (detallados en el Anexo 1), del túnel OPENVPN, estos

parámetro podemos modificarlos según nuestros requerimientos, luego de haber modificado los parámetros damos clic en el botón GUARDAR de esta forma guarda los parámetros de configuración en el servidor.

El Botón CARGAR tiene la función de cargar el archivo de configuración que viene por default en el OPENVPN

Figura 5.2-6 Ejecución del botón "Configuración"

8. Generar los Certificados

Para generar los certificados nos dirigimos al botón GENERADOR DE CLAVE, escogemos el certificado a generar y le damos un nombre para ello nos dirigimos a la sección PARAMETROS , luego especificamos los datos del certificado tal como se muestra en el ejemplo de configuración, y por ultimo damos clic en generar certificado, solo para generar el certificado de la autoridad certificadora CA damos clic en Generar CA.

En total son 3 certificados que tenemos que generar:

- CERTIFICADO DE LA AUTORIDAD CERTIFICADORA (CA)

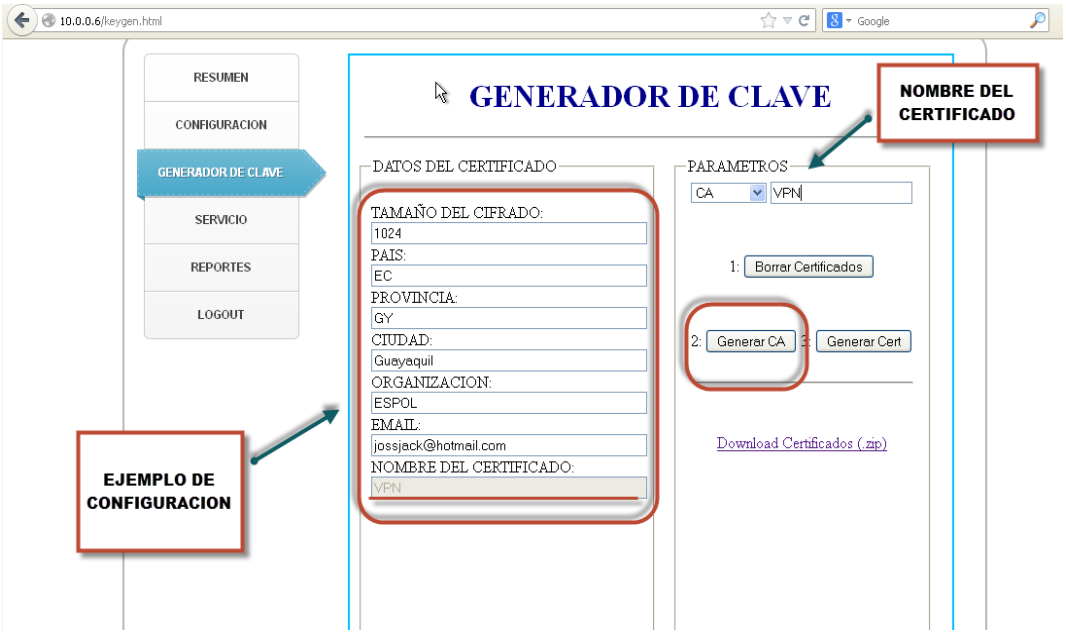


Figura 5.2.7 Generando certificado para CA

- CERTIFICADO DEL SERVIDOR

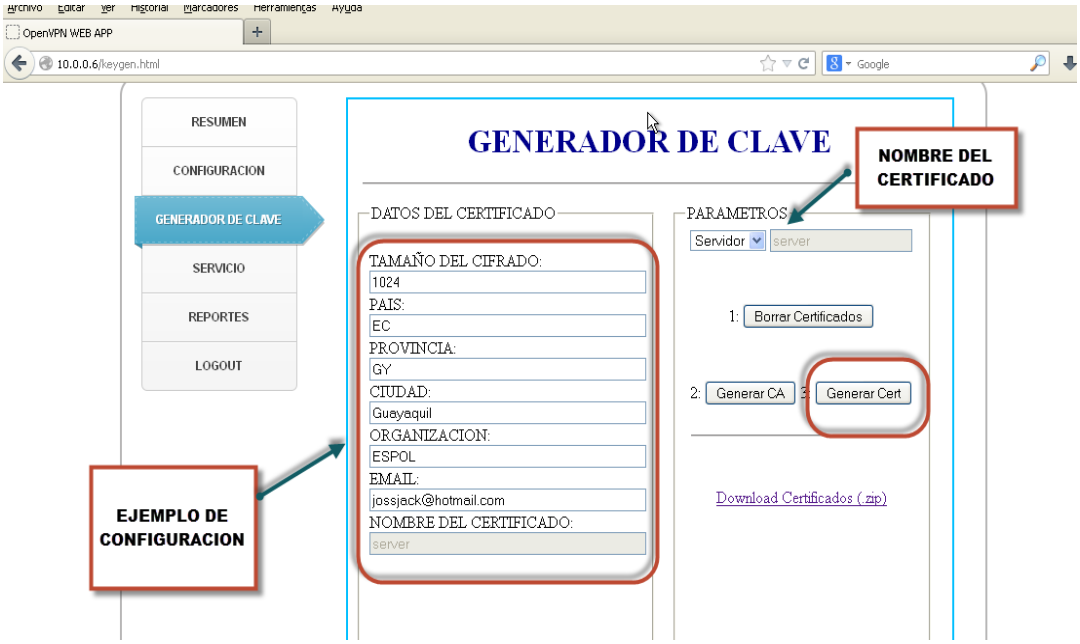


Figura 5.2-8 Generando certificado para el servidor

- CERTIFICADO DEL CLIENTE

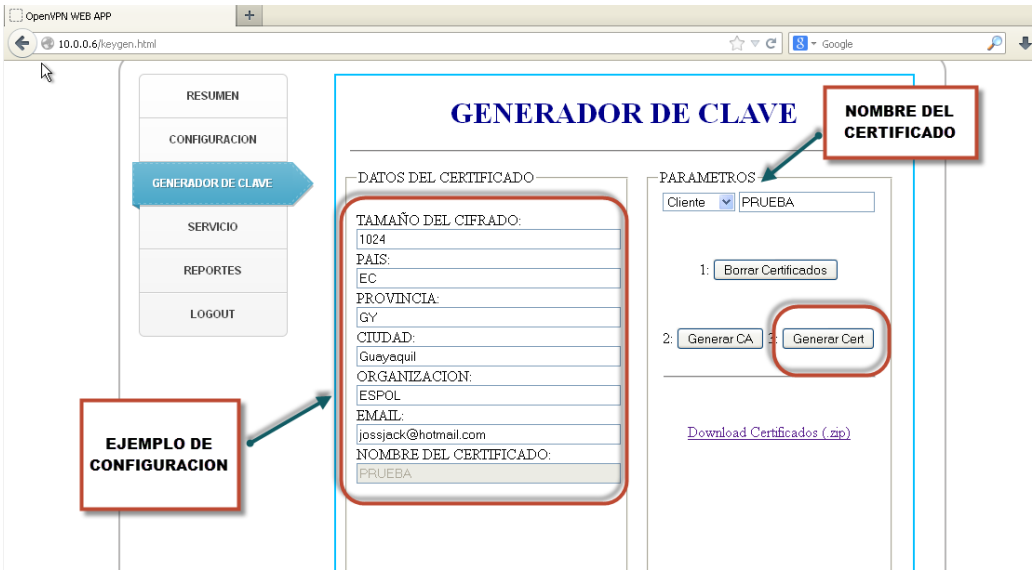


Figura 5.2 9 Generando certificado del cliente

Al concluir este proceso damos clic en Download Certificados (.zip) y ubicamos el archivo comprimido en el directorio deseado.

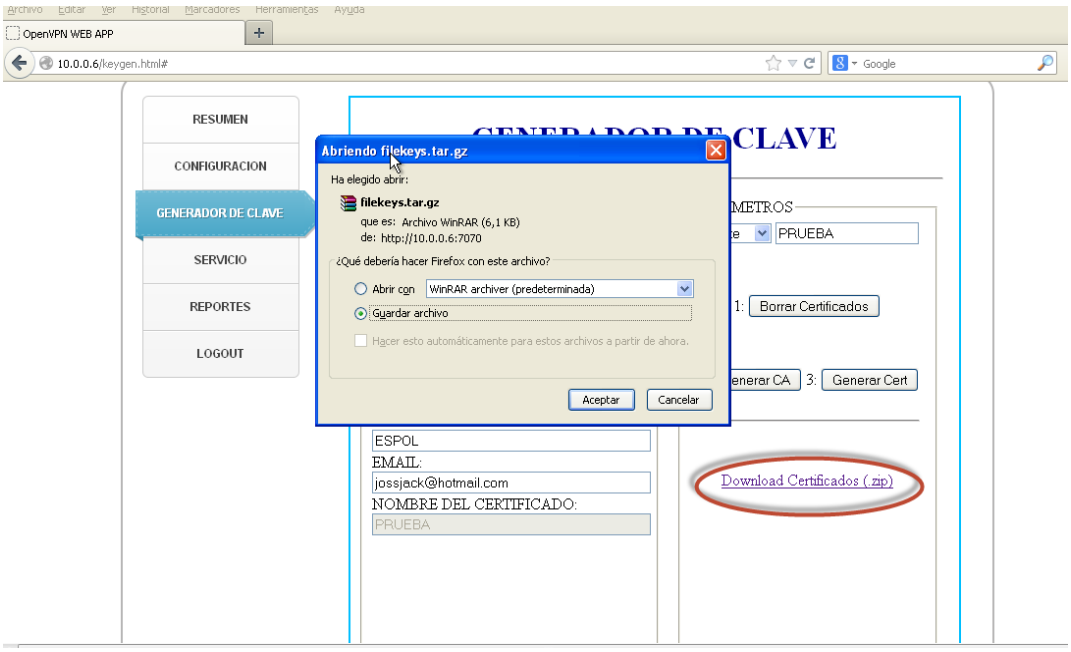


Figura 5.2-10 Descarga de los certificados comprimidos

9. Ubicar los archivos descargados en la ruta del cliente OPENVPN

Para ello descomprimimos el archivo y nos genera la carpeta keys ingresamos a la carpeta y copiamos los archivos “ ca, PRUEBA, PRUEBA.key “ (nombre que le dimos según nuestro ejemplo), y los ubicamos en el directorio C:\OpenVpn\config

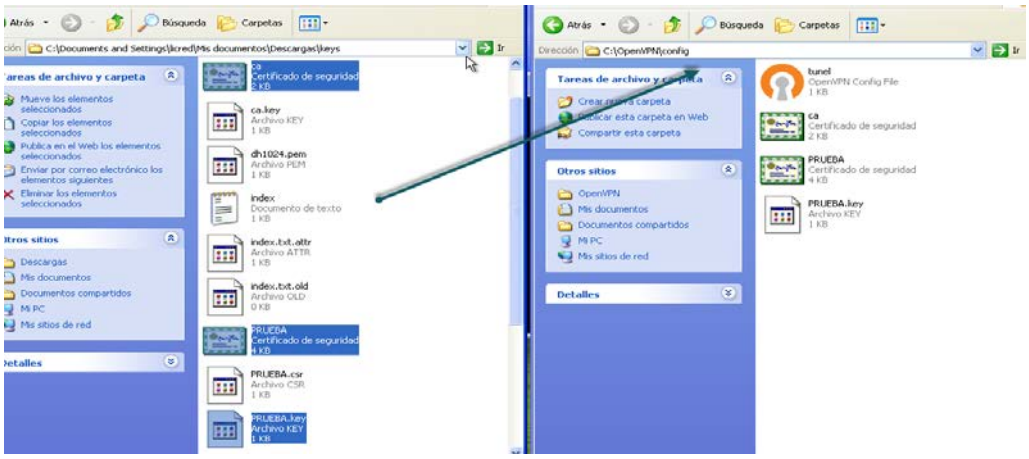


Figura 5.2-11 Ubicación de los certificados del cliente

El archivo Túnel se crea al momento de instalar el OpenVpn GUI y tenemos que configurarlo con nuestros parámetros de configuración para ello lo abrimos con un editor de texto plano (notepad) y especificamos el nombre de nuestros archivos de configuración.

Procedemos a generar el archivo túnel.ovpn con los parámetros de configuración que se muestra la figura 5.2-12.

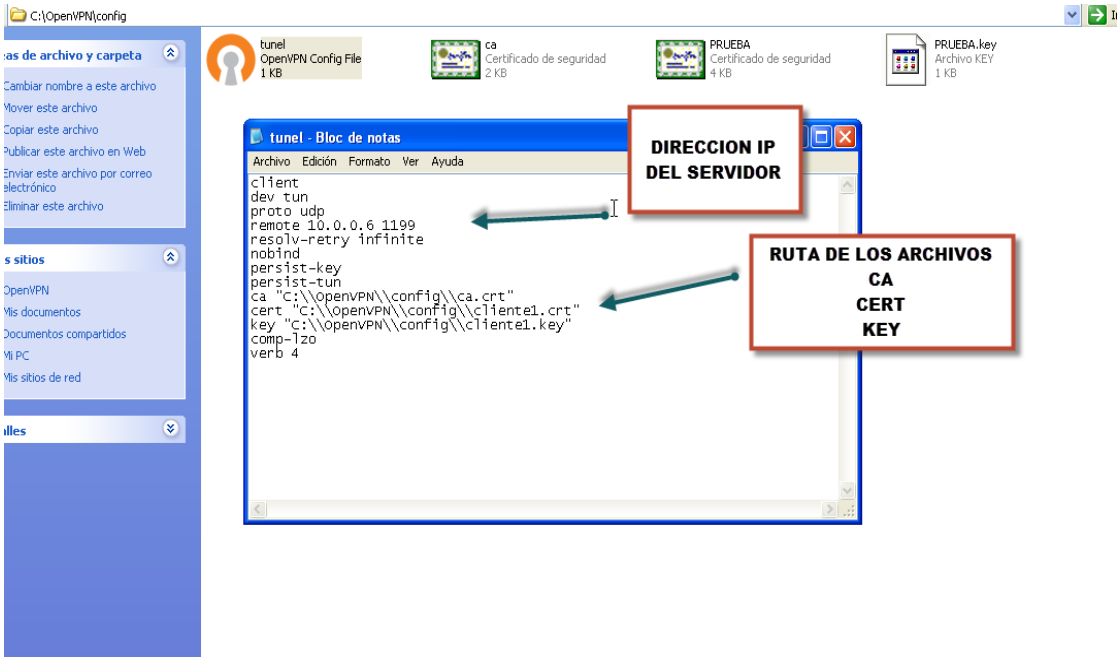


Figura 5.2-12 Generando .ovpn

10. Inicializar el túnel VPN

Abrimos la aplicación OpenVpn GUI, nos dirigimos al task manager, damos clic derecho sobre la aplicación y escogemos Connect con eso conseguimos crear el túnel virtual.

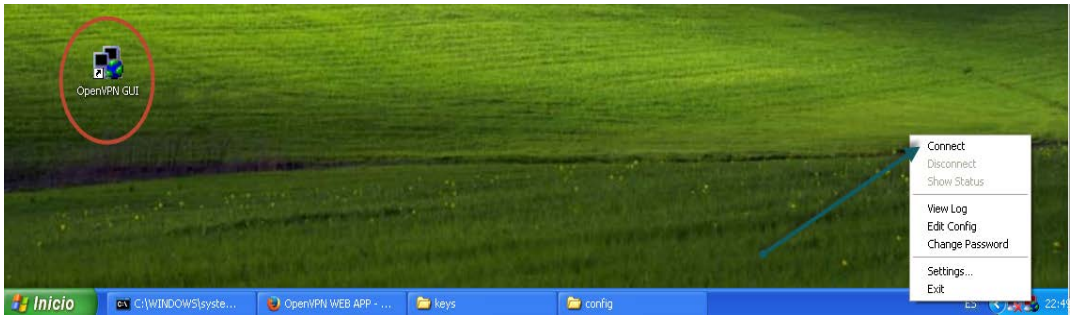


Figura 5.2-14 Ejecutando OpenVPN GUI

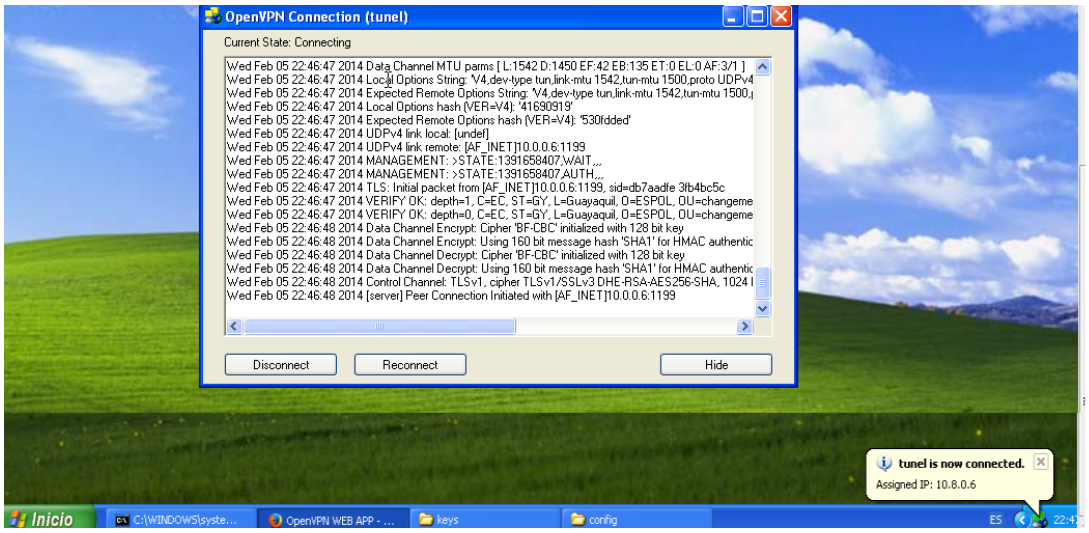


Figura 5.2-13 Conexión al túnel VPN

11. Reportes de Conectividad

Para ver los reportes de conectividad nos dirigimos al botón REPORTES de la herramienta WEB, a continuación damos clic en mostrar reporte y obtendremos una lista de LOGS de conectividad.

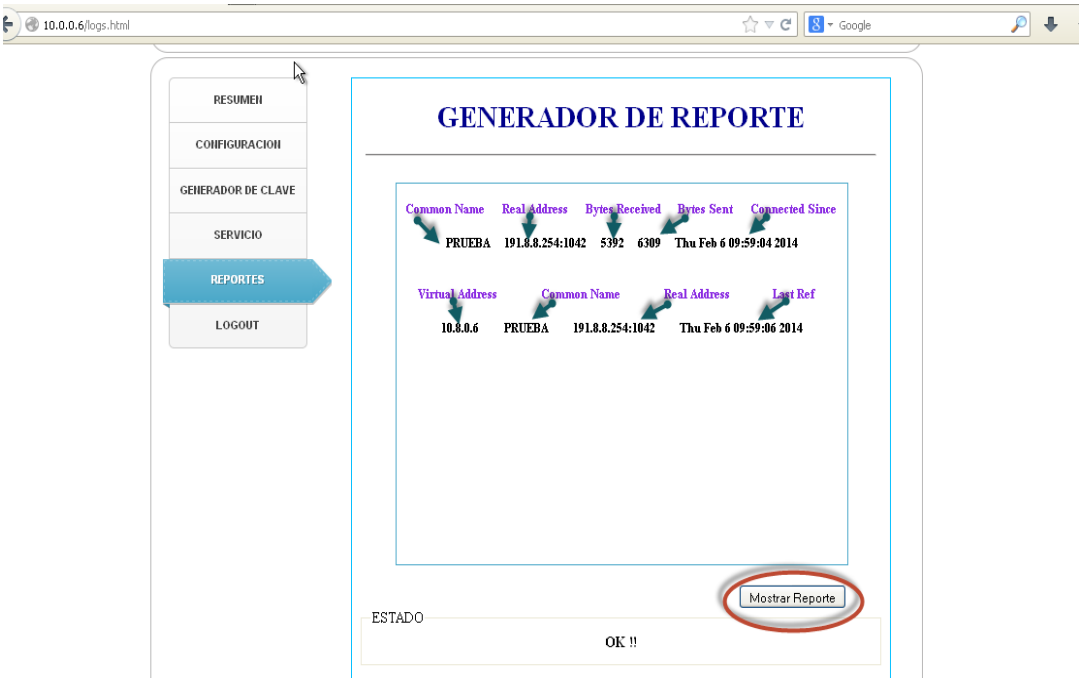


Figura 5.2-15 Reportes de conectividad

Entre los reportes encontramos:

Common Name: Nombre del certificado del Cliente

Real Address: Dirección IP del Cliente

Bytes Received: Cantidad de Bytes recibidos por el túnel OpenVpn

Bytes Sent: Cantidad de Bytes enviados por el túnel OpenvPN

Connected Since: Fecha y hora que se establece la conexión

Virtual Address: Dirección IP virtual del túnel OpenVpn

Last Ref: Fecha y hora de la última conexión establecida

12. Administrar el Servicio

Podemos hacerlo desde el Botón Servicio de nuestra Herramienta WEB en el cual encontramos las opciones de INICIAR, REINICIAR, PARAR Y VER EL STATUS del Servicio.

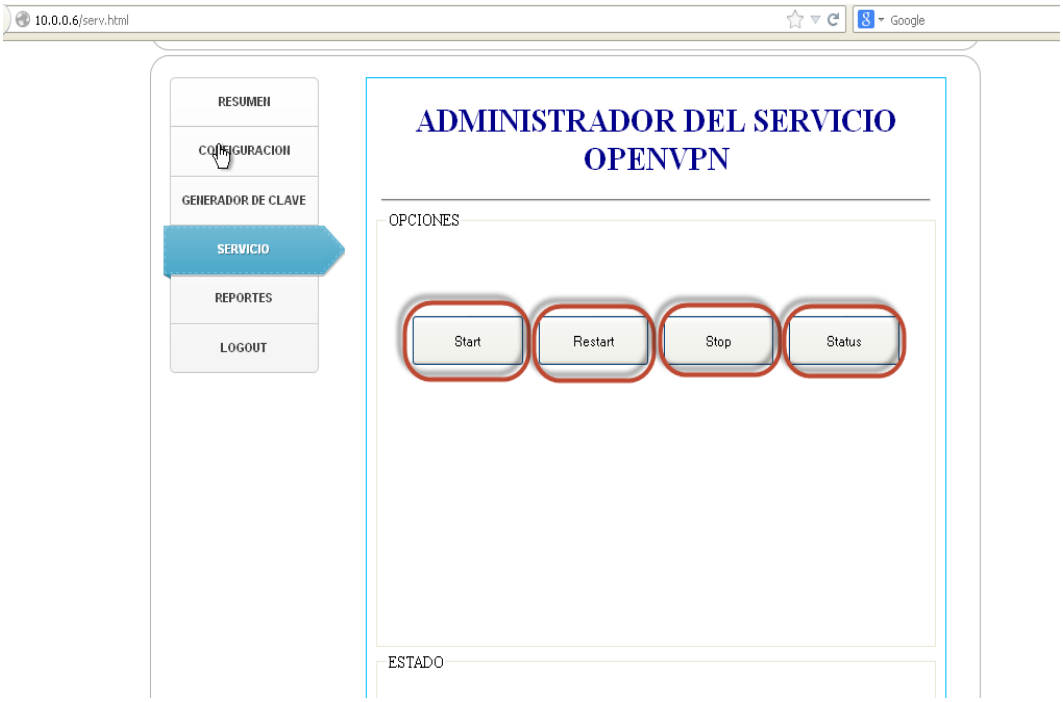


Figura 5.2 16 Administración del servicio

5.3 PRUEBAS DE CONECTIVIDAD

5.3.1 PRUEBAS DE CONECTIVIDAD SIN CONEXIÓN AL TUNEL

A continuación se muestra la ruta que toma la PC-CLIENTE para llegar al servidor VPN al enviar datos y viceversa:

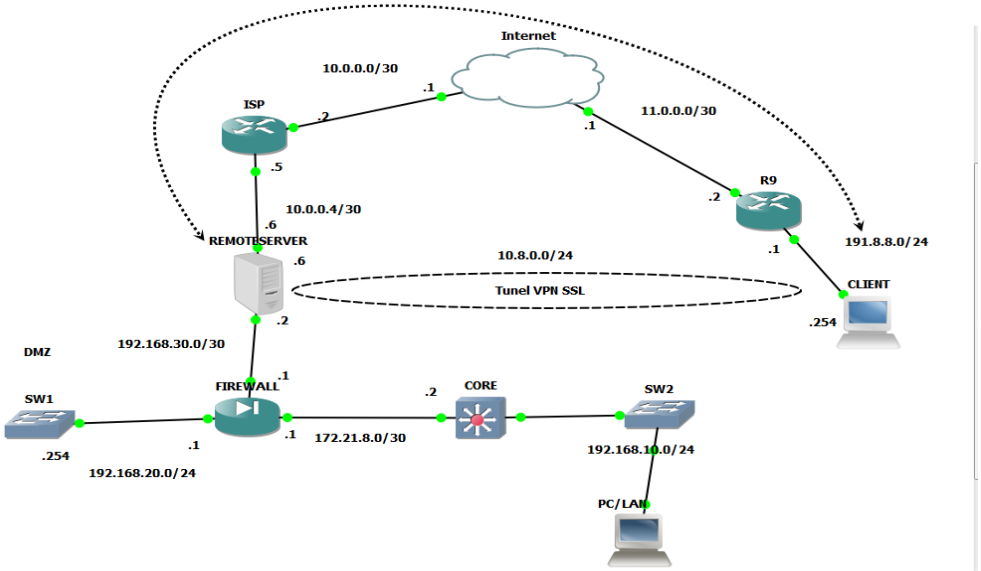


Figura 5.2-17 Ruta del PC-Cliente al servidor

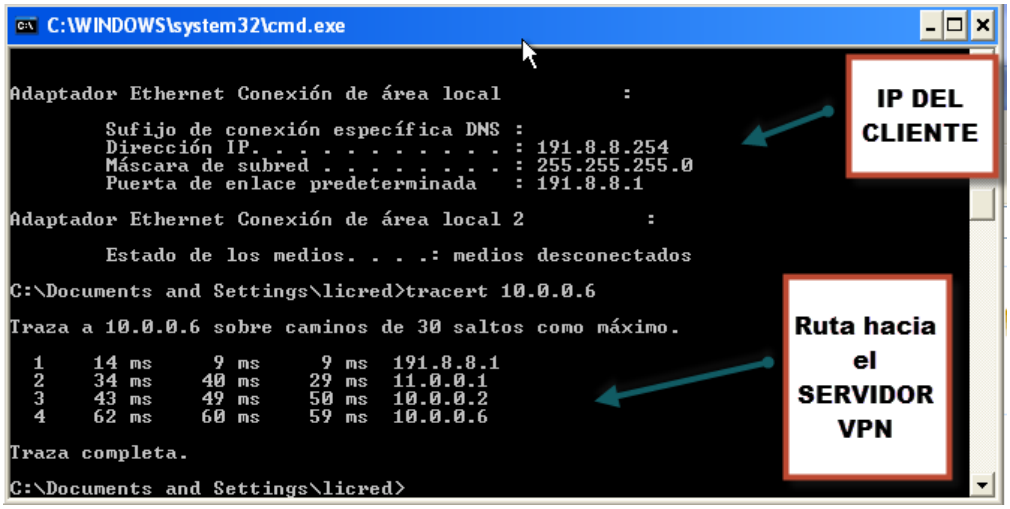


Figura 5.2-18 Tracert desde el cliente al servidor

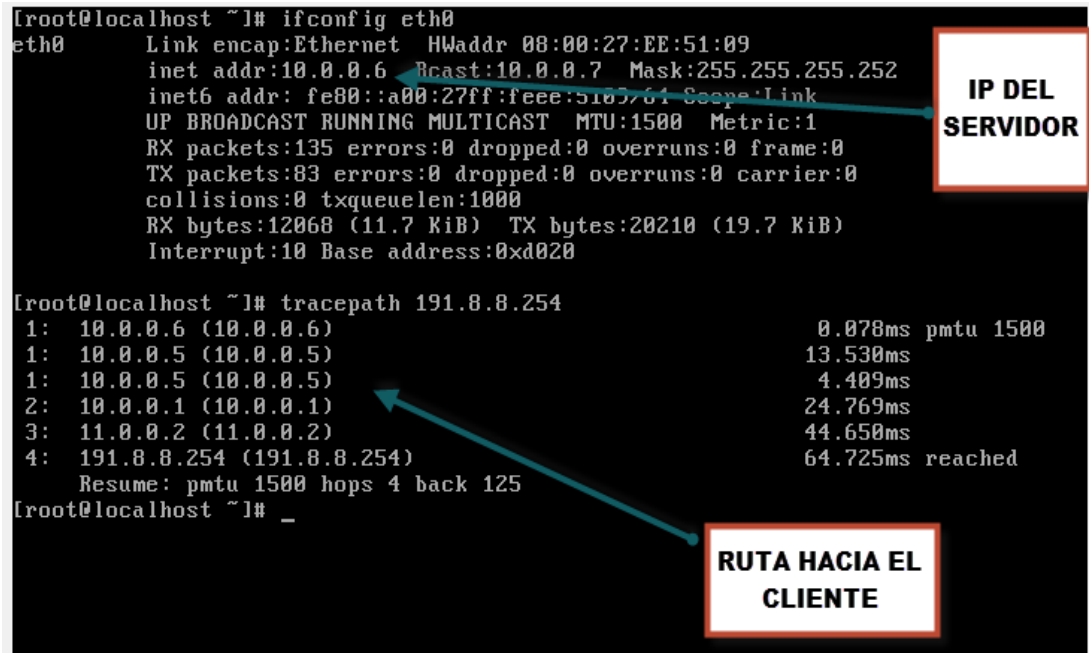


Figura 5.2-19 Tracert del servidor al cliente

Al momento de generar un requerimiento del PC-CLIENTE hacia la PC que se encuentra en la LAN tendremos error de comunicación ya que no existe ruta para llegar al destino de la Red LAN INTERNA, debido a que solo se enruto las redes desde el cliente hasta el VPNSERVER como se pudo observar en la traza mostrada en las imágenes 5.2-18-19.

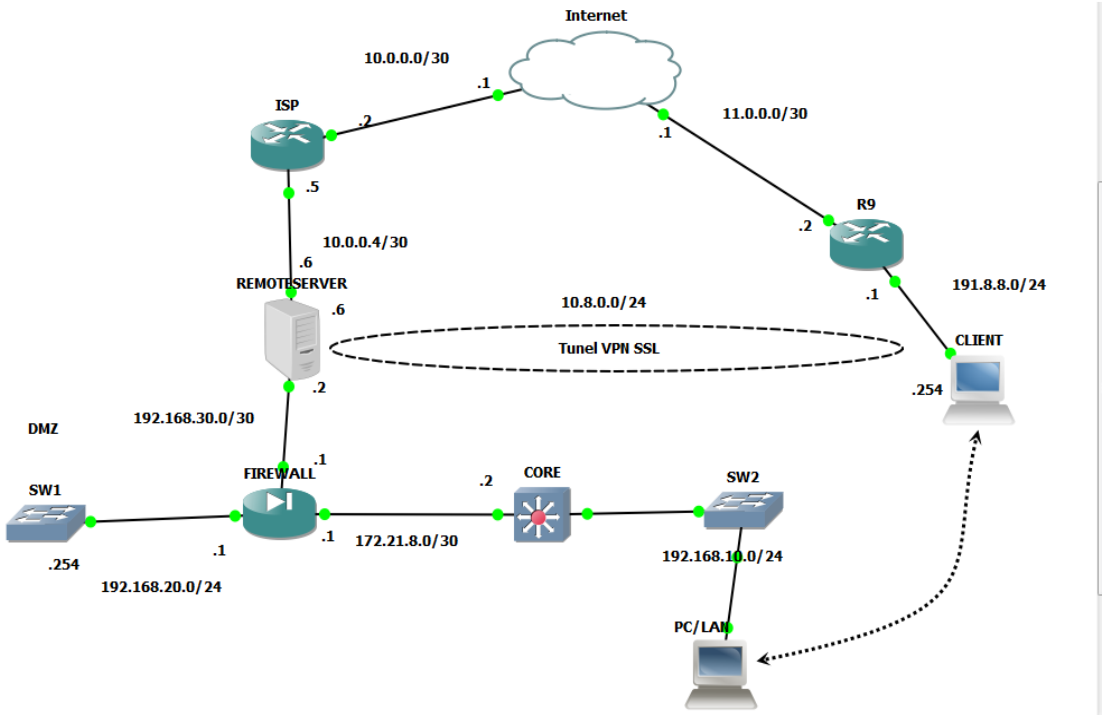


Figura 5.2-20 PC-Cliente a PC-LAN interna

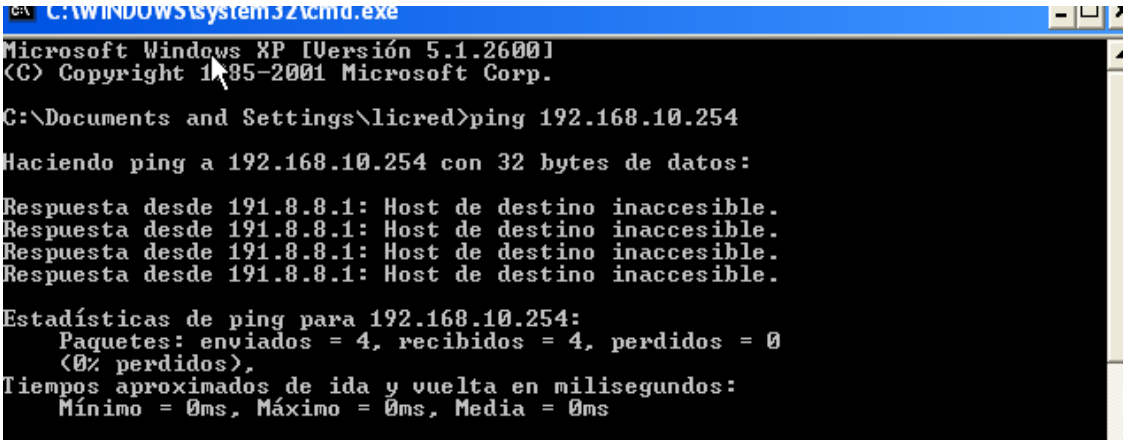


Figura 5.2-21 PING de PC'Cliente a PC LAN

5.3.2 PRUEBAS DE CONECTIVIDAD CON CONEXIÓN AL TUNEL

A continuación se muestra la ruta que toma la PC-CLIENTE para llegar a la PC-LAN al enviar datos y viceversa:

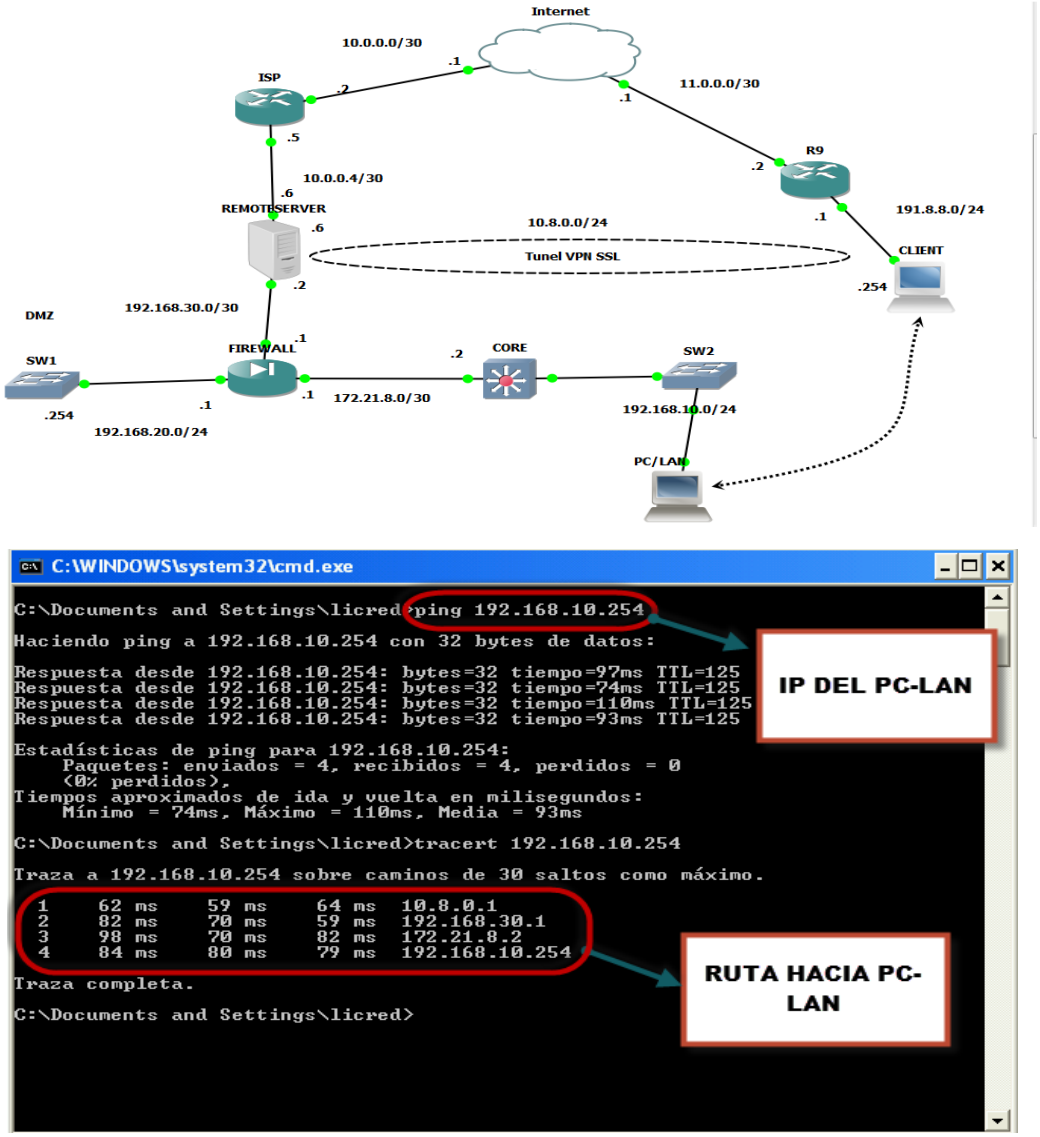


Figura 5.3-1 Conectividad de PC-Cliente a PC-LAN interna

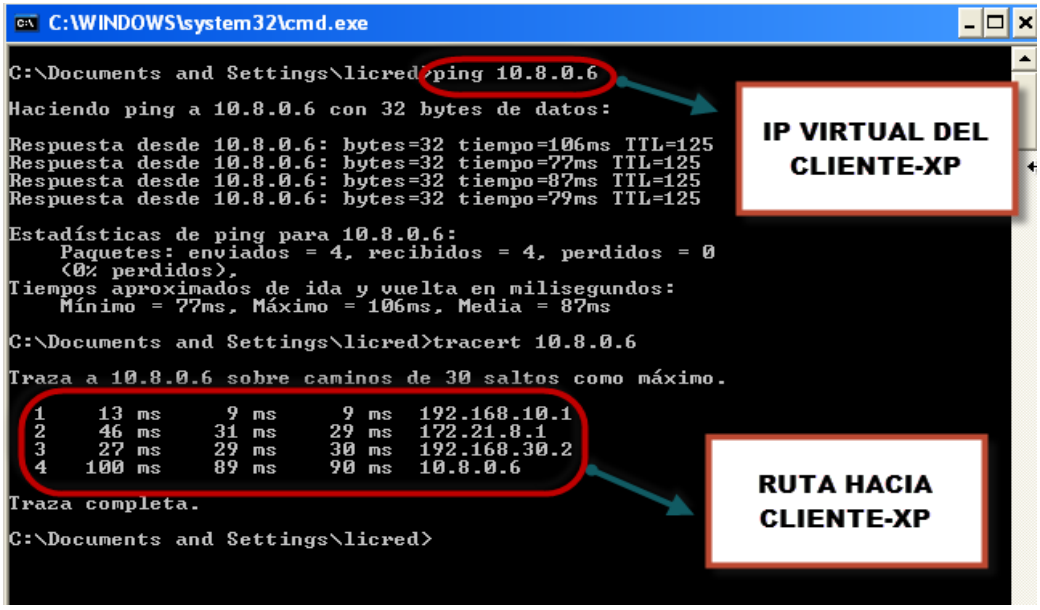


Figura 5.3-¡Error! Secuencia no especificada. **Conectividad de PC.-LAN interna a PC-Cliente**

5.4 DIAGNOSTICO A ERRORES

A continuación mostramos los posibles errores de la herramienta WEB con su respectiva solución

ERROR AL ESTABLECER LA CONEXIÓN CON EL OPENVPN GUI

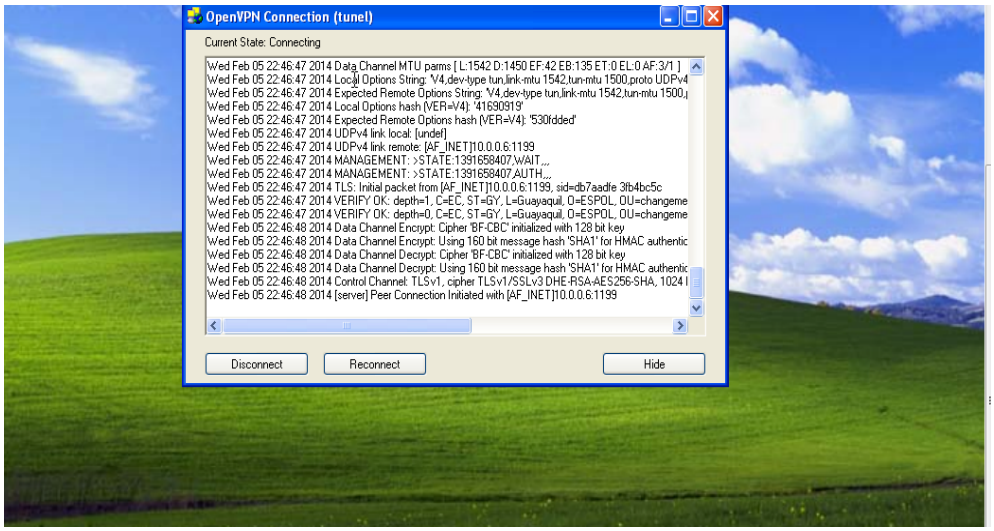


Figura 5.4-1 Error al iniciar el túnel VPN

Al momento de dar clic en connect en el OpenVpn GUI se genera un bucle infinito al tratar de establecer la comunicación para solucionar este error tenemos que reiniciar el servicio OpenVpn desde nuestra Herramienta WEB y esperamos la confirmación tal como se muestra a continuación.

Figura 5.4-2 Reinicio del servicio VPN

Luego volvemos a conectar el túnel mediante el OpenVpn GUI y estableceremos la Conexión.

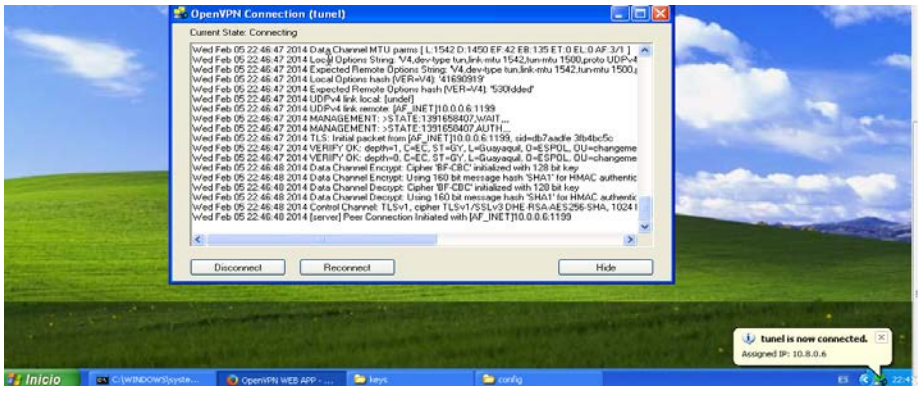


Figura 5.4-3 Conectividad del túnel VPN

CONCLUSIONES

El intercambio de información entre empleados, clientes y socios comerciales y la disponibilidad de información en el ámbito nacional e internacional con un nivel de servicio razonable, son características indispensables para garantizar la satisfacción de las necesidades empresariales, que pueden ser brindadas a través de VPN.

Las soluciones basadas en hardware resultan, en la mayoría de los casos, más costosas y con una mayor cantidad de características técnicas que las basadas en software, sin embargo, las últimas poseen también sólidas características de manejo de la seguridad que hacen factible su utilización.

La utilización de protocolos efectivos para el establecimiento de túneles y encripción y el empleo de adecuadas técnicas de autenticación garantizan la seguridad de una solución VPN. Además, el hecho de que la VPN maneje un medio como el firewall proporciona seguridad adicional.

A través de de nuestras herramientas facilitamos la instalación, configuración y administración del servicio OPENVPN en la oficina principal, con clientes VPN basados en software en las oficinas de clientes y socios comerciales, se establece una combinación económica y segura. Todo esto combinado con los adecuados protocolos para el establecimiento de túneles y encripción.

RECOMENDACIONES

Considerar y evaluar el diseño presentado como una solución para el acceso remoto en el ámbito corporativo.

Ubicar al(los) proveedor(es) de servicio de Internet que otorgará(n) la conexión a fin de llegar a un acuerdo sobre el ancho de banda, disponibilidad del servicio y tiempo de espera para la conexión para cada una las localidades. Ubicar al personal calificado para la instalación, configuración y/o adiestramiento de la solución propuesta.

Determinar segmentos que van a ser visibles por el túnel y los que van a reservarse para el uso exclusivo de la red dentro de la empresa.

Diseñar el plan para la implantación de la red tomando en cuenta el recurso humano involucrado y el cronograma de actividades a seguir.

Extender progresivamente el alcance de la red a los funcionarios de PETROGAS tomando en cuenta las necesidades de la organización.

ANEXOS

ANEXO A

Las siguientes descripciones contenidas en el anexo A de este proyecto han sido redactadas por el Sr. William López Jiménez en el manual: “VPN en servidor Linux y clientes Windows/Linux con OpenVPN” que se encuentra dividido en 2 partes en los enlaces:

http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-winlinuxshorewall-P1

http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-winlinuxshorewall-P2

Respectivamente.

Descripción de Parámetros del archivo de configuración del túnel *.conf

- **Port:** Especifica el puerto que será utilizado para que los clientes vpn puedan conectarse al servidor.
- **Proto:** tipo de protocolo que se empleará en a conexión a través de VPN.
- **dev:** Tipo de interfaz de conexión virtual que se utilizará el servidor openvpn.
- **ca:** Especifica la ubicación exacta del fichero de Autoridad Certificadora [.ca].

- **cert:** Especifica la ubicación del fichero [.crt] creado para el servidor.
- **key:** Especifica la ubicación de la llave [.key] creada para el servidor openvpn.

- dh: Ruta exacta del fichero [.pem] el cual contiene el formato de Diffie Hellman (requerido para --tls-server solamente).
- **server:** Se asigna el rango IP virtual que se utilizará en la red del túnel VPN.
- **Ifconfig-pool-persist:** Fichero en donde quedarán registrado las direcciones IP de los clientes que se encuentran conectados al servidor OpenVPN.
- **Keepalive 10 120:** Envía los paquetes que se manejan por la red una vez cada 10 segundos; y asuma que el acoplamiento es abajo si ninguna respuesta ocurre por 120 segundos.
- **comp-lzo:** Especifica los datos que recorren el túnel vpn será compactados durante la trasferencia de estos paquetes.
- **persist-key:** Esta opción soluciona el problema por llaves que persisten a través de los reajustes SIGUSR1, así que no necesitan ser re leídos.
- **Persist-tun:** Permite que no se cierre y re-abre los dispositivos TAP/TUN al correr los guiones up/down. 59
- **status:** fichero donde se almacenará los eventos y datos sobre la conexión del servidor [.log]
- **verb:** Nivel de información (default=1). Cada nivel demuestra todo el info de los niveles anteriores. Se recomienda el nivel 3 si usted desea un buen resumen de qué está sucediendo.
- **0 --**No muestra una salida excepto errores fatales. 1 to 4 –Rango de uso normal. 5 --Salida R y W caracteres en la consola par los paquetes de lectura y escritura, mayúsculas es usada por paquetes TCP/UDP minúsculas es usada para paquetes TUN/TAP. 60

Descripción de Parámetros del archivo de los clientes *.ovpn

- **Client:** Especifica el tipo de configuración, en este caso tipo cliente OpenVPN.
- **Port:** Especifica el puerto que será utilizado para que los clientes VPN puedan conectarse al servidor.
- **Proto:** tipo de protocolo que se empleará en a conexión a través de VPN
- dev: Tipo de interfaz de conexión virtual que se utilizará el servidor openvpn.
- **remote:** Host remoto o dirección IP en el cliente, el cual especifica al servidor OpenVPN. El cliente OpenVPN puede tratar de conectar al servidor con host: port en el orden especificado de las opciones de la opción --remote.
- **float:** Este le dice a OpenVPN aceptar los paquetes autenticados de cualquier dirección, no solamente la dirección cuál fue especificado en la opción --remote.
- **resolv-retry:** Si la resolución del nombre del anfitrión (hostname) falla para --remote, la resolución antes de fallar hace una re-comprobación de n segundos.
- **nobind:** No agrega bind a la dirección local y al puerto. 61
- **ca:** Especifica la ubicación exacta del fichero de Autoridad Certificadora [.ca].
- **cert:** Especifica la ubicación del fichero [.crt] creado para el servidor.
- **key:** Especifica la ubicación de la llave [.key] creada para el servidor OpenVPN.
- **remote:** Especifica el dominio o IP del servidor así como el puerto que escuchara las peticiones para servicio VPN.

- **comp-lzo:** Especifica los datos que recorren el túnel VPN será compactados durante la trasferencia de estos paquetes.
- **persist-key:** Esta opción soluciona el problema por llaves que persisten a través de los reajustes SIGUSR1, así que no necesitan ser re leídos.
- **Persist-tun:** Permite que no se cierre y re-abre los dispositivos TAP/TUN al correr los guiones up/down.
- **verb:** Nivel de información (default=1). Cada nivel demuestra toda la Información de los niveles anteriores. Se recomienda el nivel 3 si usted desea un buen resumen de qué está sucediendo.
- **0 --** No muestra una salida excepto errores fatales. **1 to 4** –Rango de uso normal. **5** -- Salida R y W caracteres en la consola par los 62 paquetes de lectura y escritura, mayúsculas es usada por paquetes TCP/UDP minúsculas es usada para paquetes TUN/TAP.

ANEXO B

HERRAMIENTAS DE DISEÑO Y SIMULACION

Netbeans

Netbeans es un proyecto de código abierto con una comunidad de muchos usuarios y de constante crecimiento. La empresa Sun Microsystems fundó el proyecto de código abierto Netbeans en junio del año 2000. Actualmente hay disponibles dos productos Netbeans IDE, Netbeans Plataform, amos productos son de código abierto y gratuito para uso tanto comercial como no comercial.

En nuestro proyecto de Materia de Graduación usamos la plataforma NetBeans IDE en conjunto con el lenguaje de programación JAVA, para desarrollar nuestro analizador de red cuyo nombre es 'ERPA', por las iniciales de nuestros nombres [# de Bibliografía].

Netbeans IDE

Es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en lenguaje Java, pero puede trabajar en cualquier otro lenguaje de programación. Existen además muchos módulos para extender el Netbeans IDE. Cabe recalcar que Netbeans IDE es un producto libre y gratuito sin restricciones de uso [# de Bibliografia].

GNS3

GNS3 es un simulador grafico de redes que permite diseñar fácilmente topologías de red y luego ejecutar simulaciones en el. Hasta este momento GNS3 soporta el IOS de routers, ATM/Frame Relay/switchs, Ethernet y PIX firewalls.

Podemos extender nuestra propia red, conectándola a la topología virtual. Para realizar esta magia, GNS3 está basado en Dynamips, PEMU (incluyendo el encapsulador) y en parte en Dynagen, fue desarrollado en python a través dePyQt la interfaz grafica (GUI) confeccionada con la poderosa librería Qt, famosa por su uso en el proyecto KDE. GNS3 también utiliza la tecnología SVG (ScalableVector Graphics) para proveer símbolos de alta calidad para el diseño de las topologías de red [3].

Virtual Box

Es un software desarrollado actualmente por Oracle el cual nos permite la creación de máquinas virtuales dentro de una maquina real, que como ya dijimos anteriormente lo hace tomando recursos de nuestra maquina real [4].

ANEXO C

COMANDOS UTILIZADOS PARA LA CONFIGURACION DE LA HERRAMIENTA

COMANDO	DESCRIPCION
cd	Nos permite cambiar de directorio
mkdir	Crear directorios
rm	Borrar archivos
cp	Copiar directorios
vim	Editor de texto
nano	Editor texto
Rpm -i	Para instalar paquetes de Linux
Rpm -qa	Para realizar consulta de paquetes instalados
Yum	Para bajar paquetes desde internet
ifconfig	Para configuración la interfaz de red del equipo
iptables	Modifcar reglas del firewall de Linux
Iptables –A	Añade una cadena, la opción –i define una interfaz de tráfico entrante
Iptables –o	Define una interfaz para el tráfico saliente
Iptables –j	Establece una regla de destino de tráfico, que puede ser ACCEPT, DROOP o REJECT
Iptables --state	Define una lista separada por comas, de destino tipos de estado de las conexiones (INVALID, ESTABLISHED, NEW, RELATED)
Iptables – to-source	Define que IP reportar al trafico externo

Iptables –s	Define tráfico de origen
Iptables -d	Define trafico destino
Iptables --source-port	Define el puerto desde el que se origina la conexión
Iptables –destination-port	Define el puerto hacia el que se dirige la conexión
route ADD IP MASK GATEWAY	Crear rutas estáticas en Linux
Service “servicio” [start] [stop][restart]	Inicializa, reinicia y detiene servicios en Linux
Tar	Empaquetar directorios
Chmod	Modificar permisos de ejecución de archivos
./	Comando utilizado para la ejecución de Shell scripts
echo	Muestra la cadena dada en la entrada en el output estándar
Sed	Editor de emisiones para procesar texto en archivos

BIBLIOGRAFIA

Manual de configuración de Servidores Linux.pdf

[3] http://es.scribd.com/doc/11840950/GNS3-Simulador-de-Redes-Grafico

[4] http://es.scribd.com/doc/39691787/Virtualizacion-Con-Virtual-Box

http://www.monografias.com/trabajos95/configuracion-openvpn/configuracion-openvpn.shtml

http://recursostic.educacion.es/usuarios/web/es/ayudas/54-conexiones-a-internet-bis

http://www.alcancelibre.org/staticpages/index.php/openvpn-clientes-win-linux-shorewall-P1

http://openvpn.net/

http://pkgs.repoforge.org/

http://www.pello.info/filez/firewall/iptables.html

http://es.wikipedia.org/wiki/OpenVPN