Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

PROYECTO DE IMPLEMENTACIÓN DE SERVIDOR PARA LA CONVERSIÓN DE ALERTAS RECIBIDAS MEDIANTE EL PROTOCOLO SMTP A LLAMADA POR VOIP, LLAMADA TELEFÓNICA, ENVÍO DE SMS Y/O ENVÍO DE CORREO

Juan Fernando Montúfar Juarez
Asesorado por el Ing. José Aníbal Silva de los Angeles

Guatemala, julio de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROYECTO DE IMPLEMENTACIÓN DE SERVIDOR PARA LA CONVERSIÓN DE ALERTAS RECIBIDAS MEDIANTE EL PROTOCOLO SMTP A LLAMADA POR VOIP, LLAMADA TELEFÓNICA, ENVÍO DE SMS Y/O ENVÍO DE CORREO

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JUAN FERNANDO MONTÚFAR JUAREZ
ASESORADO POR EL ING. JOSÉ ANIBAL SILVA DE LOS ANGELES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, JULIO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADORA	Inga. Ingrid Salomé Rodríguez de Loukota
EXAMINADOR	Ing. José Aníbal Silva de los Angeles
EXAMINADOR	Ing. Guillermo Antonio Puente Romero
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROYECTO DE IMPLEMENTACIÓN DE SERVIDOR PARA LA CONVERSIÓN DE ALERTAS RECIBIDAS MEDIANTE EL PROTOCOLO SMTP A LLAMADA POR VOIP, LLAMADA TELEFÓNICA, ENVÍO DE SMS Y/O ENVÍO DE CORREO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 25 de febrero de 2019.

Juan Fernando Montúfar Juarez

Guatemala, 31 de julio de 2020

Ingeniero:

JULIO CESAR SOLARES PENATE

Coordinador del Área de Electrónica

Escuela de Ingeniería Mecánica Eléctrica

Facultad de Ingeniería

Universidad de San Carlos de Guatemala

Estimado Ingeniero Solares:

Por este medio tengo a bien informarle que he realizado la revisión técnica del Trabajo de Graduación titulado **"Proyecto de implementación de servidor para la conversión de alertas recibidas mediante el protocolo SMTP a llamada por VoIP, llamada telefónica, envío de SMS y/o envío de correo"**, desarrollado por el estudiante Juan Fernando Montúfar Juarez, número de registro académico 2014-03684 y numero de dpi 2668065030101; por lo cual considero que el trabajo de graduación cumple con el alcance y los objetivos definidos para su desarrollo, habiéndolo encontrado satisfactorio en su contenido y resultados, sometiendo a su consideración la aprobación del mismo,siendo responsables del contenido técnico el estudiante y el suscrito, en calidad de asesor.

Sin otro particular,

Atentamente,

JOSE ANIBAL SILVA DE LOS ANGELES
ING ELECTRONICO
COLEGIADO No 5067

Ing. José Aníbal Silva de los Angeles

No. Colegiado 5067

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

Guatemala, 13 de agosto de 2020

Señor Director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor Director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **PROYECTO DE IMPLEMENTACIÓN DE SERVIDOR PARA LA CONVERSIÓN DE ALERTAS RECIBIDAS MEDIANTE EL PROTOCOLO SMTP A LLAMADA POR VOIP, LLAMADA TELEFÓNICA, ENVÍO DE SMS Y/O ENVÍO DE CORREO,** desarrollado por el estudiante **Juan Fernando Montúfar Juarez**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Julio César Solares Peñate
Coordinador de Electrónica

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA

REF. EIME 62. 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; JUAN FERNANDO MONTÚFAR JUAREZ titulado; PROYECTO DE IMPLEMENTACIÓN DE SERVIDOR PARA LA CONVERSIÓN DE ALERTAS RECIBIDAS MEDIANTE EL PROTOCOLO SMTP A LLAMADA POR VOIP, LLAMADA TELEFÓNICA, ENVÍO DE SMS Y/O ENVÍO DE CORREO, procede a la autorización del mismo.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Armando Alonso Rivera Carrillo

GUATEMALA, 12 DE ABRIL 2,021.

Decanato
Facultad de Ingeniería
24189101 - 24189102

DTG. 318-2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **PROYECTO DE IMPLEMENTACIÓN DE SERVIDOR PARA LA CONVERSIÓN DE ALERTAS RECIBIDAS MEDIANTE EL PROTOCOLO SMTP A LLAMADA POR VOIP, LLAMADA TELEFÓNICA, ENVÍO DE SMS Y/O ENVÍO DE CORREO**, presentado por el estudiante universitario: **Juan Fernando Montúfar Juarez,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Inga. Anabela Cordova Estrada
Decana

Guatemala, julio de 2021

AACE/cc

ACTO QUE DEDICO A:

Dios	Por permitirme alcanzar esta meta y realizarme como profesional.
Mis padres	Claudia Isabel Juárez de Montúfar y Luis Alberto Montúfar Paz por su apoyo incondicional desde pequeño.
Mis hermanos	Karla Ana Isabel Aguirre, María Eugenia, Sofia Montúfar, Alex Mauricio Montúfar Juárez y Luis Alberto Montúfar Hernández por ser una fuente de inspiración para seguir adelante.
José Eduardo Soto Castellanos	Por ser un pilar fundamental para culminar con mi carrera y un excelente amigo.
Mis primos	Víctor Salvador, Sebastián, Luis Pedro Ponce, Hugo y Diego Monzón con los que he convivido experiencias inolvidables.
Mis amigos	A todos los que pude hacer, tanto fuera como dentro de la universidad, por su amistad, enseñanza y convivir experiencias inolvidables.

Mi abuela Ana Salazar de Juárez. Por su cariño incondicional.

Mis tíos Miriam y Julio Juárez. Por ser parte fundamental en mi niñez.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por formar profesionales dispuestos a servir al país.
Facultad de Ingeniería	Por ser la facultad en la cual pude realizar mis estudios y formarme como profesional.
Escuela de Ingeniería Mecánica Eléctrica	Por brindar a los catedráticos correspondientes dispuestos a enseñar y desarrollarse como profesional.
Ing. José Aníbal Silva	Por su apoyo en la elaboración de este trabajo de graduación.
Ing. Juan Estuardo Hernández	Por ser parte fundamental en mi desarrollo como profesional.
Ing. Carlos Humberto Pérez	Por su apoyo en el proceso de revisión de este trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES V
GLOSARIO IX
RESUMEN XXIII
OBJETIVOS XXV
INTRODUCCIÓN XXVII

1. SISTEMAS DE ALERTA TEMPRANA 1
   1.1. Introducción a sistemas de alertas 1
      1.1.1. Factores de calidad 2
      1.1.2. Medios para la recepción de alertas 3

2. DESCRIPCIÓN DEL PROYECTO 5
   2.1. Introducción al proyecto 5
   2.2. Elementos del proyecto 5
      2.2.1. Diagrama general 5
      2.2.2. Equipos monitoreados 6
      2.2.3. Servidor de monitoreo 7
      2.2.4. SMTP Appliance 7
      2.2.5. Módulo GSM 9
      2.2.6. Central telefónica 9
      2.2.7. *Firewall Small Business* 10
      2.2.8. Equipo de destino 10

3. PROCESOS 11
   3.1. Introducción a procesos 11

3.2. Composición del SMTP Appliance 11
3.2.1. Sistema de archivos de la aplicación 12
3.2.2. Receptor 17
3.2.3. Procesador 18
3.2.4. Emisor 21
3.2.4.1. Módulo SIM 22
3.2.4.1.1. Comandos AT para envío de llamada 22
3.2.4.1.2. Proceso de llamada telefónica 24
3.2.4.2. Módulo SMS 25
3.2.4.2.1. Comandos AT para envío de SMS 25
3.2.4.2.2. Proceso de envío de SMS 26
3.2.4.3. Módulo VoIP 27
3.2.4.3.1. RTP 28
3.2.4.3.2. RTCP 34
3.2.4.3.3. SDP 45
3.2.4.3.4. SIP 52
3.2.4.3.5. Proceso de registro 56
3.2.4.3.6. Proceso de llamada por VoIP 58
3.2.4.4. Módulo Mail 61

4. ANÁLISIS DEL SISTEMA 63
4.1. Introducción al análisis del sistema 63
4.2. Análisis financiero 63
4.3. Análisis por factores de calidad 73

4.3.1. Tiempo de generación y envío 74
4.3.2. Latencia en respuesta 75
4.3.3. Calidad de llamada 76
4.3.4. Escalabilidad 77
4.3.5. Estabilidad 78
4.3.6. Seguridad 79
4.4. Pruebas de estrés 80
4.4.1. Pruebas de hardware 80
4.4.1.1. Estrés de CPU 81
4.4.1.2. Estrés de memoria 84
4.4.1.3. Estrés de disco 86
4.4.1.4. Estrés de tarjeta de red 89
4.4.2. Pruebas de software 92
4.5. Productos similares 96

5. FUTURAS IMPLEMENTACIONES 99
5.1. Introducción a futuras implementaciones 99
5.2. Kubernetes 99
5.3. SRTP y SRTCP 100
5.4. TLS en la recepción de mensajes SMTP 100
5.5. Twisted 101
5.6. Django 101
5.7. WhatsApp 102
5.8. Estadísticas de alertas 102
5.9. Sensores 102
5.10. Sistema bidireccional 103

CONCLUSIONES 105
RECOMENDACIONES 107
BIBLIOGRAFÍA 109
APÉNDICES 117

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Elementos del proyecto 6
2. Diagrama SMTP Appliance 12
3. Composición de receptor 17
4. Ejemplo de mensaje de correo entregado al procesador 18
5. Paquete RTP 31
6. Paquete RTCP 35
7. Paquete SR 36
8. Paquete SDES 40
9. Estructura general de elementos SDES 42
10. Paquete BYE 44
11. Proceso de registro 57
12. Validación de agente de usuario 57
13. Proceso de llamada VoIP 60
14. Proyección de costo anual según tipo de nube 73
15. Monitor del procesador con nmon 82
16. Monitoreo de pruebas de estrés del CPU 83
17. Monitor de la memoria RAM con nmon 85
18. Monitoreo de pruebas de estrés de la memoria RAM 86
19. Monitor del disco con nmon 88
20. Monitoreo de pruebas de estrés del disco 89
21. Monitor de la tarjeta de red con nmon 90
22. Monitoreo de pruebas de estrés de la tarjeta de red 91
23. Monitoreo del servidor SMTP Appliance – antes del inicio 93

24. Monitoreo del servidor SMTP Appliance – estado inicial 94
25. Monitoreo del servidor SMTP Appliance – durante prueba..... 95
26. Sensaphone IMS-4000 98

TABLAS

I. Descripción de factores de calidad de generador de alertas 2
II. Datos entregados por el procesador a cada módulo del emisor 20
III. Comandos AT para llamadas 22
IV. Comandos AT para SMS 26
V. Protocolos utilizados para VoIP 28
VI. Codificaciones tradicionales de audio en RTP 29
VII. Estándares en el tipo de carga útil en RTP 32
VIII. Tipos de elementos SDES 43
IX. Atributos de descripción general de la sesión 46
X. Atributos de descripción del tiempo 48
XI. Atributos de descripción de medios 50
XII. Mensajes de petición 53
XIII. Mensajes de respuesta 55
XIV. Nube privada vs nube pública vs nube híbrida 64
XV. Precios según nube 65
XVI. Precios de llamadas y SMS por CPaaS 66
XVII. Precios por servidor físico 67
XVIII. Mantenimiento de servidores 68
XIX. Precios de módulo GSM 69
XX. Tarifas de llamadas y SMS 70
XXI. Precios de Raspberry Pi 71
XXII. Costos de las nubes 72
XXIII. Tiempos de generación y envío de alertas 74

XXIV. Latencia en respuesta promedio por módulo 76
XXV. Medición de calidad de llamada 77
XXVI. Resultados de pruebas de estabilidad 78
XXVII. Aspectos de seguridad 79
XXVIII. Sensaphone IMS-4000 vs SMTP Appliance 97

GLOSARIO

Active directory	Término que utiliza Microsoft para referirse a su implementación de servicio de gestión de usuarios en una red distribuida de computadores.
ADPCM	Codificación adaptativa basada en el tipo de codificación DPCM.
Antivirus	Programa de computadora utilizado para prevenir, detectar y eliminar software malicioso.
Appliance	Dispositivo, con software y firmware integrado, diseñado para tareas específicas.
Asterisk	Programa *open source* que proporciona funcionalidades de una central telefónica PBX.
Baud rate	Número de símbolos transmitidos por segundo.
Bit	Unidad básica de información en teoría de la información, informática y comunicaciones digitales. El nombre es un acrónimo de dígito binario.
Blacklist	Mecanismo básico de control de acceso que deniega acceso a ciertos elementos de una red mencionados explícitamente.

Byte	Conjunto de 8 bits.
Central telefónica	Dispositivo al cual se le puede conectar un número determinado de teléfonos para hacer llamadas entre sí dentro de una misma organización e incluso acceder a comunicaciones fuera de la misma.
Cluster	Grupo de ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador.
Codificación	Proceso en el cual se transforman los datos, o una secuencia dada de caracteres o símbolos, en un formato específico para garantizar la transmisión segura de los datos.
Código ASCII	Código de caracteres de 7 bits donde cada bit representa un carácter único.
Comandos AT	Estándar abierto de comandos para configurar y parametrizar módems o módulos.
Container	Unidad estándar de software que empaqueta el código con todas sus dependencias para que la aplicación se ejecute de manera rápida y confiable de un entorno informático a otro.
CPaaS	Plataforma basada en la nube que permite a los desarrolladores agregar funciones de comunicación

	en tiempo real a sus propias aplicaciones sin necesidad de construir infraestructura.
Debian	Es una distribución del sistema operativo Linux compuesta de software libre y de código abierto.
DHCP	Protocolo de red mediante el cual un servidor asigna una dirección IP, entre otros parámetros de configuración de red, a cada dispositivo en una red para que puedan comunicarse con otras redes IP.
Dirección IP	**E**tiqueta numérica asignada a cada dispositivo conectado a una red informática que utiliza el Protocolo de Internet, IP, para la comunicación.
Dirección MAC	Identificador de 48 bits que corresponde de forma única a una tarjeta o dispositivo de red.
DMZ	Subred física o lógica que contiene y expone los servicios externos de una organización a una red no confiable como Internet.
DNS	Sistema el cual permite la traducción de nombres de dominio, como el caso de www.google.com, en direcciones IP y viceversa.
Dominio	Un grupo de *hosts* en una red que se encuentran dentro de un mismo sistema de administración.

Driver	Programa que controla un dispositivo.
DTMF	Tecnología utilizada para marcar números de teléfono o para emitir comandos a los sistemas de conmutación.
Encriptación	Procedimiento de seguridad que consiste en la alteración, mediante algoritmos, de los datos, con el objetivo de que dichos datos se vuelvan ilegibles en caso de que un tercero los intercepte.
Escalabilidad	Término usado en tecnología para referirse a la propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer el funcionamiento y calidad normales del mismo.
Etiquetas IETF	Conjunto de etiquetas compuesto por las normas ISO 639 para representar nombres de idiomas y las normas ISO 3166-1 para representar códigos de países.
FIFO	Método para la organización y la manipulación de una memoria intermedia de datos, donde la primera entrada se procesa primero.
FQDN	Nombre de dominio completo el cual incluye el nombre del *host* más el nombre del dominio al que pertenece, por ejemplo luis_pc.ejemplo.com.

Firewall	Sistema de seguridad de la red que permite filtrar contenido de una red externa para proteger a los equipos dentro de una red local de cualquier tipo de ataque posible.
Framework	Plataforma base sobre la cual los desarrolladores de software pueden crear programas para una aplicación específica.
FreePBX	Interfaz gráfica de usuario de código abierto que controla y administra Asterisk.
Full rate	Primer tipo de codificación digital de audio utilizado para el sistema de comunicaciones GSM. Utiliza codificación lineal predictiva.
Gateway	Nodo de red que permite que los datos fluyan de una red discreta a otra.
GitHub	Empresa global que ofrece *hosting* para desarrollo de software.
Gmail	Servicio de correo electrónico gratuito proporcionado por la empresa estadounidense Google a partir del 1 de abril de 2004.
GSM	Sistema estándar para la comunicación de dispositivos móviles.

Host	Computadoras u otros dispositivos como tabletas, móviles, portátiles y máquinas virtuales, conectados a una red que proveen y utilizan servicios dentro de ella.
Hostname	Nombre del *host*.
Interfaz	Punto a través del cual dos o más componentes separados de un sistema informático intercambian información.
IPS	Tecnología de prevención de amenazas que examina los flujos de tráfico de red para detectar y prevenir vulnerabilidades.
Imagen ISO	Archivo informático donde se almacena una copia o imagen exacta de un sistema de archivos. Algunos de los usos más comunes incluyen la distribución de sistemas operativos.
Kernel	Software el cual es el principal responsable de facilitar a los distintos programas acceso seguro al hardware del *host*.
LAN	Red informática que interconecta computadoras dentro de un área limitada, como una residencia, escuela, laboratorio, campus universitario o edificio de oficinas.

Ley A	Sistema de cuantificación logarítmica de una señal de audio, usado en el campo de comunicaciones telefónicas de Europa y resto del mundo exceptuando Estados Unidos, Canadá y Japón.
Ley Mu	Sistema de cuantificación logarítmica de una señal de audio, usado en el campo de comunicaciones telefónicas de Estados Unidos, Canadá y Japón.
Máquina virtual	Software que permite emular el funcionamiento de un ordenador dentro de otro ordenador gracias a un proceso de encapsulamiento que aísla a ambos.
Módulo GSM	Módulo de comunicación utilizado para conectarse a la red de telefonía móvil.
Módulo USB a UART	Dispositivo utilizado como interfaz serial, para comunicar un microcontrolador con un ordenador.
MP3	Formato de compresión de audio digital que usa un algoritmo con pérdida para conseguir un menor tamaño de archivo.
Muestreo	Proceso que consiste en tomar muestras del valor de una señal cualquiera, N veces por segundo.
Multicast	Comunicación grupal donde la transmisión de datos se dirige a un grupo de computadoras de destino simultáneamente.

Nethserver	Solución de código abierto de tipo *appliance* para pequeñas y medianas empresas. Cuenta con un set de módulos orientados en temas de administración de redes.
NTP	Protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable.
Open source	Programas informáticos que permiten el acceso a su código de programación o código abierto.
Paquete	Cada uno de los bloques en que se divide la información a enviar a través de la red.
Paravirtualización	Tipo de virtualización en el cual la máquina virtual creada ejecuta cada proceso directamente en el hardware físico del virtualizador en el que reside.
PBX	Cualquier central telefónica conectada directamente a la red pública de telefonía por medio de líneas troncales.
PCM	Procedimiento de modulación utilizado para transformar una señal analógica en una secuencia de bits.

Ping	Utilidad de software para administración de red informática utilizada para probar la accesibilidad de un *host* en una red IP.
PJSIP	Librería de comunicación multimedia de código abierto que implementa protocolos basados en estándares como SIP, SDP, RTP, STUN, TURN e ICE.
Plugin	Aplicación o programa informático que se relaciona con otra para agregarle una función nueva y generalmente muy específica.
Port forwarding	Acción de redirigir un puerto de red de un nodo de red a otro.
Proxmox VE	Plataforma de código abierto para virtualización a nivel empresarial.
Proxy	Servidor que hace de intermediario en las peticiones de recursos que realiza un cliente a otro servidor.
Puerto	Punto de comunicación lógico que identifica un proceso específico o un tipo de servicio de red.
Red	Grupo de computadoras que utilizan un conjunto de protocolos de comunicación comunes a través de interconexiones digitales.

Router	Dispositivo de red que reenvía paquetes de datos entre redes de computadoras, permitiendo así su comunicación.
RTC	Reloj de computadora que realiza un seguimiento de la hora actual.
Serial	Se refiere al tipo de comunicación en la cual se envían bits de forma secuencial.
Sesión	Período temporal ocupado por una cierta actividad. Esto quiere decir que, durante una determina sesión, se llevan a cabo una serie definida de tareas.
Small Business	Suite integrada diseñado para el funcionamiento de la infraestructura de la red de las pequeñas y medianas empresas.
Smartphone	Tipo de ordenador de bolsillo con las capacidades de un teléfono móvil, con llamada telefónica y SMS.
SMS	Servicio de mensajería más utilizado a través de telefonía celular, internet u otro dispositivo móvil. Generalmente se utiliza como acrónimo de mensaje de texto.
SMTP	Protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre dispositivos.

SMTP Relay	Servidor que recibe correos electrónicos de un remitente y los transfiere a un tercero.
Socket	Software que actúa como un punto final el cual establece un enlace de comunicación de red bidireccional entre el extremo del servidor y el programa receptor del cliente.
Spam	Mensajes no solicitados, no deseados o con remitente desconocido enviados por medio de correo electrónico.
SPICE	Protocolo utilizado para emular un ambiente de ordenador de escritorio dentro de una máquina virtual.
TCP	Protocolo de comunicaciones utilizado para la intercambio seguro de datos entre aplicaciones dentro una red.
Terminal	Una de las consolas del sistema proporcionadas en el *kernel* de Linux. La terminal de Linux actúa como el medio para las operaciones de entrada y salida para un sistema Linux.
Tarjeta SIM	Tarjeta desmontable usada en teléfonos móviles y módems HSPA o LTE que almacena de forma segura la clave de servicio del suscriptor usada para identificarse ante la red móvil.

Telefonía celular	Medio de comunicación inalámbrico a través de ondas electromagnéticas. Como cliente de este tipo de redes, se utiliza un dispositivo denominado teléfono móvil, teléfono celular o móvil.
Tupla	Lista finita ordenada de elementos. En el lenguaje de programación Python solo pueden poseer elementos del mismo tipo.
UART	Dispositivo de comunicación serial asíncrona el cual el formato de datos y transmisión son configurables.
Ubuntu Server	Distribución de Linux gratuita y de código abierto basado en Debian, destinada para ser utilizada como servidor multipropósito.
UDP	Protocolo de comunicaciones que se utiliza principalmente para establecer conexiones de baja latencia y tolerancia a pérdidas entre aplicaciones en una red.
USB	Hace referencia a un protocolo de conexión que permite enlazar diversos periféricos a un dispositivo electrónico que frecuentemente es un ordenador, para el intercambio de datos u otras operaciones.
Usuario root	Usuario especial de los sistemas operativos basados en Linux, utilizado para la administración completa del mismo sistema operativo.

UTC	Principal estándar de tiempo por el cual el mundo regula los relojes y el tiempo.
VGA	Estándar de gráficos para controlador de pantalla de video.
Virtual host	Cada sitio web que se ejecuta en una sola máquina.
Virtualización	Consiste en crear una representación basada en software, o virtual, de una entidad física como, por ejemplo, servidores, redes y almacenamiento.
Virtualización añidada	Característica que permite a un equipo virtualizador, crear *hosts* virtuales dentro de un *host* virtual. Su uso generalmente consiste en montar laboratorios de demos y pruebas.
Virtualizador	Equipo capaz de crear y administrar *hosts* virtuales.
VLAN	Red lógica independiente dentro de una misma red física.
VPN	Tecnología de red que se utiliza para conectar una o más computadoras a una red privada por medio de Internet.
WAV	Formato de audio digital con o sin compresión de datos desarrollado por Microsoft e IBM.

Zabbix	Solución de monitoreo de código abierto para monitoreo de red y monitoreo de aplicaciones de millones de métricas.

RESUMEN

Se desarrolló un servidor generador de alertas, al que se le denominó SMTP Appliance, el cual recibe alertas por medio de protocolo SMTP, luego las reenvía por llamada telefónica, llamada por VoIP, mensaje de texto y/o correo electrónico, según se configure dicho equipo.

Para mostrar el funcionamiento del servidor elaborado dentro de una empresa, se montó un entorno virtual, que consta de los siguientes elementos: equipo a monitorear, servidor de monitoreo, equipo generador de alertas, central telefónica, *firewall*, módulo GSM y celular de destino.

En el primer capítulo se describe que es un sistema generador de alertas, sus funcionalidades en diversos ámbitos y que factores de calidad hay que analizar al momento que se requiera uno de estos equipos.

El segundo capítulo describe, de forma general, los elementos del entorno empresarial elaborado, así como la función que cumple cada uno para informar al cliente sobre cualquier alerta que se genere.

En el tercer capítulo se detalla cómo funciona el servidor generador de alertas, desde el momento en que recibe la alerta, hasta que la transmite por los diversos medios mencionados previamente.

El cuarto capítulo consta de un análisis monetario, cualitativo, de funcionamiento general y comparativo con equipos similares en el mercado, del servidor SMTP Appliance.

El quinto capítulo hace una referencia a las futuras implementaciones a ejecutar para que el servidor sea competitivo en el mercado.

OBJETIVOS

General

Desarrollar un servidor para cualquier empresa que cuente con área de servidores, que sea capaz de informar acerca de cualquier alerta de interés para el destinatario, ya sea por llamada telefónica, llamada VoIP, SMS y/o correo electrónico.

Específicos

1. Informar acerca de las funciones que tiene un servidor generador de alertas.

2. Demostrar cómo se debe implementar el servidor creado dentro de un entorno empresarial.

3. Explicar detalladamente el funcionamiento del servidor SMTP Appliance.

4. Efectuar un análisis del equipo generador de alertas en los ámbitos financiero, cualitativo, de funcionalidad y de mercado.

5. Indicar que futuras implementaciones se tienen planificadas para hacer un sistema generador de alertas competitivo en el mercado.

INTRODUCCIÓN

Hoy en día muchas organizaciones como las entidades bancarias, entidades de telefonía y entidades con servicios web, buscan mantener sus servicios disponibles las 24 horas del día. Para estas organizaciones es indispensable que su personal a cargo esté alerta a cualquier fallo que pueda afectar los servicios que ellas ofrecen. Para ello, generalmente, utilizan un servicio de correo para recibir las alertas, sin embargo, suele ser más efectivo que un individuo responda ante una llamada que ante un mensaje de correo, lo que reduciría el tiempo de respuesta del destinatario ante cualquier evento inesperado que acarree un alto riesgo para la empresa. De igual forma es preferible tener un servidor capaz de enviarles alertas por los varios medios: llamada telefónica, llamada por VoIP, envío de SMS o envío de correo, para aumentar la probabilidad de que la alerta sea recibida.

Por lo anterior, se decidió crear un servidor capaz de informar con respecto a cualquier alerta recibida por diversos medios, de tal forma que se reduzca el tiempo de respuesta por parte del receptor ante dicha alerta.

Además, el servidor a implementar pretende ser una solución de bajo costo de implementación, con lo que se busca que cualquier tipo de empresa, ya sea grande, mediana o pequeña, pueda adquirirlo. Este *appliance* será de tipo *open source* para que cualquiera que desee contar con un sistema similar, pueda adaptarlo a las necesidades de su empresa.

1. SISTEMAS DE ALERTA TEMPRANA

1.1. Introducción a sistemas de alertas

En Estados Unidos existe un sistema de alertas por mensajes de texto, que se encarga de enviar un mensaje a todas las personas que se encuentren en una zona que pueda estar afectada por una catástrofe, como un huracán, un tsunami o un atentado terrorista. Comisión Federal de Comunicaciones de Estados Unidos: Alerta de Emergencia Móvil WAE, 2019. Estos tipos de sistemas se han empleado con la principal función de salvar vidas humanas, lo que denota su gran importancia dentro de la sociedad.

A nivel empresarial, los sistemas de generación de alertas se han utilizado para obtener información acerca de una falla en cualquier equipo que les sea relevante o para recibir información de equipos de seguridad.

Un buen sistema de generación de alertas ayuda a reducir el tiempo de respuesta del personal de la empresa a cargo de sus equipos ante una falla de alguno de estos y del personal de seguridad en caso se genere una alerta de seguridad la cual deba ser atendida con prontitud. Si se genera falla o problema de seguridad que no sea atendido con prontitud, se pueden provocar daños permanentes en los equipos, pérdidas de información, pérdidas en servicios e inclusive pérdidas monetarias.

1.1.1. Factores de calidad

Los factores de calidad de un sistema de alertas permiten determinar si el sistema de alertas es adecuado para utilizarse a nivel empresarial.

Los factores de calidad a tomar en cuenta son:

Tabla I. **Descripción de factores de calidad de generador de alertas**

Factor	**Descripción**
Tiempo en generación y envío	Un buen sistema es capaz procesar la información y notificar a los usuarios interesados en el menor intervalo de tiempo posible, desde la detección de la falla o señal de alerta, hasta la aparición del mensaje en los dispositivos finales.
Latencia en respuesta	La latencia en respuesta promedio de un destinatario ante una alerta es información vital para estimar la calidad del sistema.
Calidad de llamada, para equipos generadores de llamadas.	Al utilizar un medio de llamada para envío de alertas, el mensaje de alerta sea entendible por el receptor, por lo cual es importante medir el factor de calidad de llamada que el sistema generador provee.
Escalabilidad	Un sistema escalable es capaz de integrarse con cualquier cantidad de sistemas de monitoreo de los cuales reciba información y, a partir de ella, genere alertas. Además es capaz de integrar diversos protocolos de comunicación e inclusive incrementar sus funcionalidades según la empresa lo requiera.

Continuación de la tabla I.

Estabilidad	Un sistema debe estar diseñado para funcionar todo el tiempo y producir resultados consistentes.
Seguridad	Es necesario que cualquier otro individuo ajeno a los destinatarios sea incapaz de leer la información transmitida en los mensajes de alerta.

Fuente: elaboración propia, empleando Libreoffice v6.1.

1.1.2. Medios para la recepción de alertas

Existen múltiples formas de recibir alertas, desde medios tradicionales, hasta medios personalizados como lo son aplicaciones dedicadas. Algunos de los medios por el cual se puede recibir una alerta son:

- Correo: Es el medio más utilizado. Muchos equipos envían las alertas por correo dado que suele ser el más sencillo de configurar y no requiere de hardware adicional.

  La rapidez respuesta de los usuarios interesados a alertas recibidas por correo suele ser mejor en horarios de oficina que en horarios fuera de oficina, siendo los horarios nocturnos el peor tiempo de respuesta ante este tipo de alertas.

  En caso no esté correctamente configurado el servidor, los mensajes de alerta pueden ser detectados como *spam*.

- SMS: Es un medio que requiere de un módulo GSM con una tarjeta SIM, conectado a la red de telefonía celular, para enviar mensajes de texto. El

costo de este tipo de servicios depende del plan de pago utilizado con las compañías Claro o Tigo para Guatemala, a la que pertenezca la tarjeta SIM.

Al igual que los correos, la rapidez de respuesta de los usuarios interesados a los SMS también depende del horario, siendo las horas fueras de oficina las de mayor latencia en respuesta al momento de ocurrir una alerta.

- Llamada telefónica: Al igual que el SMS, este necesita el uso de un módulo GSM con tarjeta SIM para enviar llamadas telefónicas.

  Suele ser un método más efectivo que el SMS o el correo para obtener respuestas rápidas de los receptores ante cualquier alerta, especialmente en horarios fuera de oficina.

- Llamada por VoIP: A diferencia de la llamada telefónica tradicional, la llamada por VoIP no requiere ningún hardware adicional para efectuar este tipo de llamadas. Para poder recibir este tipo de llamadas afuera de una red VoIP local es requerido una troncal de SIP, un *gateway* SIP o una VPN. No es común encontrar un equipo de alerta que incluya llamadas por VoIP.

La rapidez respuesta a llamadas VoIP es similar que una llamada tradicional.

2. DESCRIPCIÓN DEL PROYECTO

2.1. Introducción al proyecto

El proyecto consiste en crear un *appliance* de generación de alertas el cual sea capaz de transformar la información de alertas la cuales serían destinadas a ser recibidas por correo, a llamada telefónica, llamada por VoIP, SMS o nuevamente correo.

2.2. Elementos del proyecto

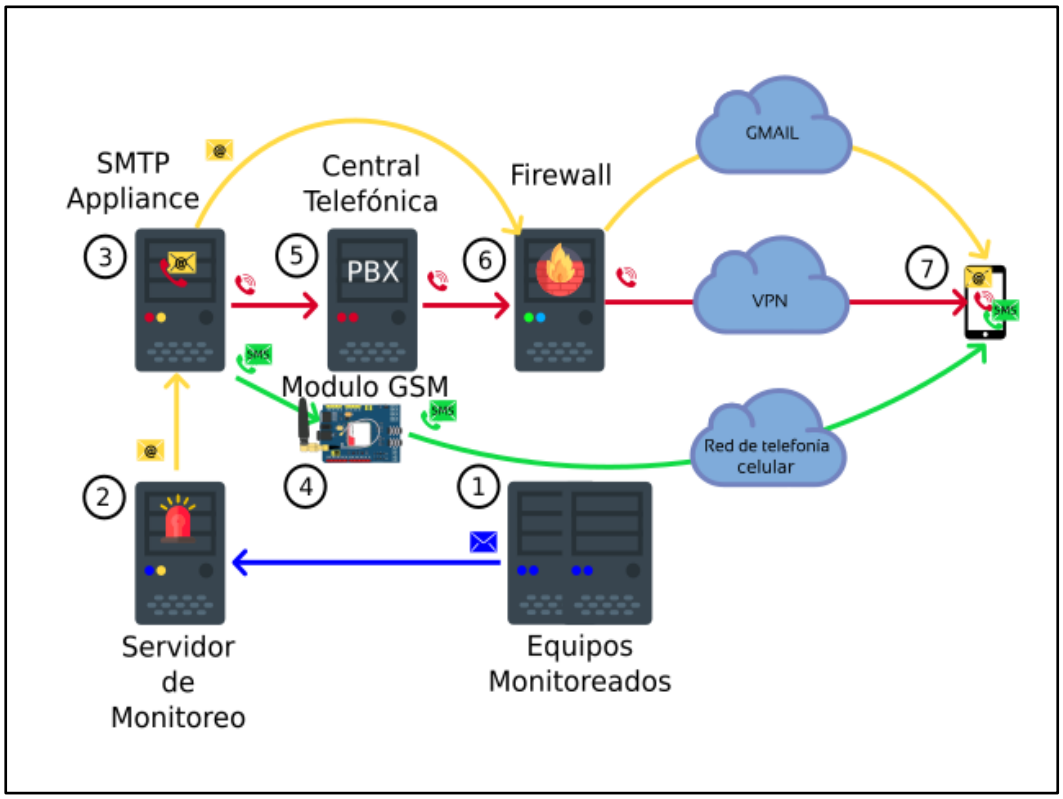
Para este proyecto piloto se utilizaran máquinas virtuales, instalados en un servidor de virtualización Proxmox, los cuales simularan un entorno empresarial en el que se monitorean servidores virtuales. Las máquinas virtuales con las que contará tal entorno son: un servidor de monitoreo Zabbix, el servidor de generación de alerta 'SMTP Appliance', una central telefónica FreePBX y un *firewall* Nethserver. Además se requiere de un módulo GSM SIM900 para efectuar tanto llamadas telefónicas como enviar de SMS.

2.2.1. Diagrama general

Para comprender como es que se coordina el servidor SMTP Appliance se ha realizado un diagrama ilustrativo en el cual se muestran todos los elementos del proyecto, así como los procesos que intervienen en la conversión del mensaje final.

El diagrama se presenta en la siguiente figura:

Figura 1. **Elementos del proyecto**



Fuente: elaboración propia, empleando Inkscape v0.92.4.

2.2.2. Equipos monitoreados

Estos pueden ser cualquier tipo de equipo de interés de la empresa. Generalmente son servidores o equipos de seguridad. Estos equipos transmiten su información a los servidores de monitoreo a través de protocolos de monitoreo como lo son SNMP, WMI, IPMI o Modbus, dependiendo de los protocolos que maneje el servidor de monitoreo. Ciertos servidores de monitoreo se comunican con los equipos a monitorear con su propio protocolo, como lo son los agentes de Zabbix instalados en los equipos a monitorear.

En este proyecto se eligió un solo equipo a monitorear el cual será un servidor web Apache con Wordpress como gestor de contenido bajo sistema operativo Ubuntu Server 18.04, sistema operativo seleccionado arbitrariamente. El motivo de esta elección es que tanto Apache como Wordpress son los más utilizados en sus categorías a nivel empresarial. Hosting Diario: Tipos de Servidores Web, 2019 y Blogthinkbig.com: Gestores de contenidos, 2017.

2.2.3. Servidor de monitoreo

El servidor de monitoreo extrae toda la información de los equipos a monitorear y verifica si existe alguna falla o hay alguna información importante que deba ser notificada. En este esquema, el servidor de monitoreo enviará la notificación de la alerta al servidor SMTP Appliance mediante el protocolo SMTP, que es el protocolo estándar para envío de correos.

Algunos servidores de monitoreo utilizados son: Zabbix, Solarwinds NPM, PRGT Network Monitor, Nagios, OpenNMS, entre otros. Para este proyecto se utilizará un servidor Zabbix como servidor de monitoreo dado que es una solución *open source* que es muy utilizada por varias empresas a nivel productivo. *IT Central Station*: *Network Monitoring Software*, 2019. Además, se tiene el conocimiento para administrar de dicho producto.

2.2.4. SMTP Appliance

Es el encargado de convertir la alerta del servidor de monitoreo recibida vía SMTP, en llamada telefónica, llamada por VoIP, SMS o relevar el correo.

Para efectuar tanto llamadas telefónicas como enviar SMS, este *appliance* utiliza un módulo USB a UART el cual intercambia información con el módulo

GSM mediante el uso de comandos AT. En los SMS solo intervienen el uso de comandos AT, mientras que en el proceso de llamadas telefónicas se utilizan los comandos AT para marcar el teléfono indicado y monitorear el estado de la llamada. Además dentro del procedimiento de llamada se requiere conectar un cable de audio desde el servidor SMTP Appliance hacia el módulo GSM para transmitir la información del mensaje de alerta.

Para realizar llamadas por VoIP, el *appliance* intercambia información con la central telefónica mediante el uso de los protocolos SIP, SDP, RTP y RTCP. En el procedimiento de llamada los protocolos intervienen de la siguiente forma:

- SIP: Se encarga de establecer la comunicación entre ambos puntos.
- SDP: Indica cómo es que cada usuario de va a comunicar.
- RTP: Interviene para transferir los datos de audio codificados una vez iniciada la comunicación. El tipo de codificación depende del sistema de telefonía IP.
- RTCP: Envía datos para el control de paquetes RTP.

En el caso de correo, el SMTP Appliance actual como un *SMTP Relay* que modifica la fuente del mensaje. Dicho mensaje pasa por el *firewall*, luego por el servidor de Gmail y finaliza la bandeja de entrada del destinatario.

Es importante mencionar que este *appliance* es el principal del proyecto, porque es en sí el servidor de generación de alertas. Este equipo debe ser integrable con cualesquiera otros servidores de distinto sistemas operativos o marcas, siempre que cumplan con la misma función mencionada en esta sección.

Para este *appliance* se utilizará como sistema operativo un Ubuntu Server 18.04 en el que se instalarán diversos paquetes para su funcionamiento. El

sistema operativo seleccionado debido a que se posee un buen manejo de este en su versión 18.04.

Cabe mencionar que el lenguaje de programación seleccionado para ejecutar las labores previamente mencionadas fue Python en su versión 2.7.

2.2.5. Módulo GSM

El módulo GSM utiliza una tarjeta SIM con la que se conecta a la red de telefonía celular para poder enviar SMS o realizar llamadas telefónicas. Para que este módulo pueda hacer llamadas o enviar SMS se requiere que la tarjeta SIM utilizada cuente con el saldo suficiente. El Módulo GSM utilizado es el SIM900 con tarjeta SIM de la compañía Claro.

2.2.6. Central telefónica

Este equipo puede actuar como servidor proxy o como servidor redirector con la función de conectar a los equipos de telefonía IP. Si este equipo actúa como proxy en el proceso de llamada, toda la información de VoIP pasa a través de este servidor, se recodifica en el caso de ser necesario y redirige hacia el destino final. Si este equipo actúa como redirector, solo tendrá la función de conectar ambos puntos al iniciar el proceso de llamada, sin intervenir el en tráfico al momento de que la llamada sea contestada por el destinatario.

Una central telefónica también cumple con la función de servidor de registro, cuya función es asignar números de extensión a los equipos conectados a él, para que puedan ser contactados.

En este esquema se utilizará como central telefónica un servidor virtual con el sistema operativo de FreePBX, que actuará como servidor proxy. En la llamada por VoIP este servidor se conectará a una VPN, utilizando el protocolo OpenVPN, medio por el que transmitirá la información de la alerta desde la SMTP Appliance hacia el equipo de destino el cual debe estar conectado a la misma VPN.

2.2.7. *Firewall Small Business*

Dado que esta es una simulación de un entorno empresarial, es requerido que se cuenta con un *firewall*. Dicho *firewall* añadirá protección a los servidores previamente mencionados y brindará acceso limitado a internet.

Como *firewall* se utilizará una máquina virtual con sistema operativo Nethserver, el cual también funcionará como un servidor *Small Business* brindando también los servicios de DHCP, DNS, Antivirus e IPS al área de servidores.

2.2.8. Equipo de destino

Como equipo de destino se tendrá un *smartphone* convencional. Para la recepción de correos tendrá instalado la aplicación de Gmail, mientras que para la recepción de llamadas por VoIP contará con las aplicaciones OpenVPN – Connect, para conectarse a la VPN a la cual está conectada FreePBX, y MizuDroid, para recibir llamadas por VoIP.

3. PROCESOS

3.1. Introducción a procesos

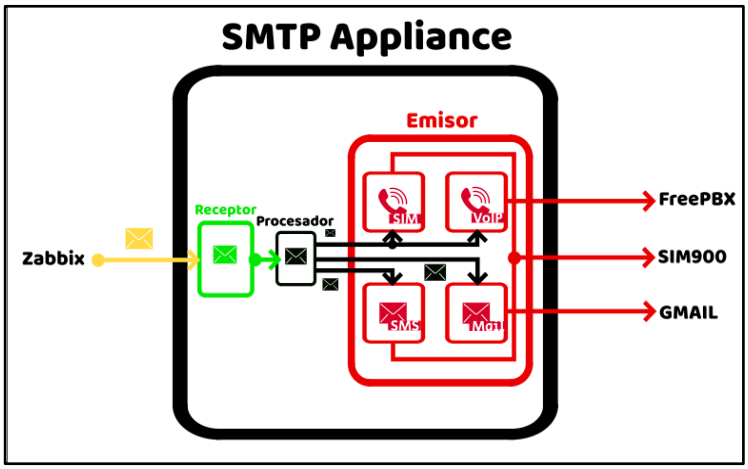
En esta sección se detallarán todos los procesos a nivel de protocolo y conexiones que intervienen para enviar el mensaje de alerta, desde su generación en el servidor de monitoreo hasta llegar al destino final.

La trama existente entre el equipo monitoreado y el servidor de monitoreo no será descrita, debido a que existen diversos protocolos de comunicación entre ambos.

3.2. Composición del SMTP Appliance

Para comprender el desarrollo del proyecto, hay que comprender como es que el servidor SMTP Appliance está compuesto. Para ello se ha realizado el siguiente diagrama:

Figura 2. **Diagrama SMTP Appliance**



Fuente: elaboración propia, empleando Inkscape v0.92.4.

En síntesis, el receptor del SMTP Appliance se encarga de recibir los correos enviados por el servidor de monitoreo Zabbix. Estos datos son entregados al procesador, el cual cumple con la función de adaptar la información recibida a cada uno de los módulos del emisor: VoIP, SIM, SMS y Mail, quienes transmiten su información recibida a través del medio que cada uno maneja.

3.2.1. Sistema de archivos de la aplicación

Para que pueda funcionar este equipo como servidor de alertas se creó el siguiente sistema de archivos:

- *Softphone:* Directorio inicial. Se puede colocar en cualquier parte del sistema de archivos del sistema operativo Ubuntu Server 18.04.
  - *code:* Directorio donde están los archivos escritos en Python 2.7.
    - *myip.py:* Librería que permite encontrar la dirección IP asignada al servidor.

- serverlib.py: Librería designada para crear servidores mediante la programación por ‘hilos’. Dentro de esta librería se encuentra la clase *SMTPServer* la cual es el receptor del SMTP Appliance e incluye el uso del puerto 25 que es utilizado por defecto para el servicio de SMTP. Esta librería también maneja los usuarios capaces de recibir mensajes de alerta.
- *SMTP_Appliance.py:* Programa principal. Acopla al receptor, ubicado en *serverlib.py*, al procesador, ubicado internamente en su estructura, y al emisor, ubicado en *Phone/__init__.py*. Además, tiene variables de ingreso a la base de datos del servidor de monitoreo Zabbix.

- *Phone:* Directorio donde se encuentran los módulos requeridos para la transmisión.
  - ✓ *__init__.py:* Emisor del *appliance*. Aquí se transmite la alerta por los medios de: correo regular, usando la librería *smtplib* que viene por defecto en Python 2.7, llamada VoIP, administrando las librerías *sip.py, sdp.py* y *rtp.py*, llamada convencional, manejando la librería *SIM900.py*, y mensaje de texto, utilizando nuevamente la librería *SIM900.py*.

    Dentro de esta librería se definen todos los parámetros requeridos: números de teléfono, correos de los destinatarios, extensiones, dirección IP de la central telefónica, correo del servidor, contraseña de correo, entre otros, para la administración de los diversos medios de comunicación.

- ✓ *SIM900.py:* Dicha librería envía los comandos AT al módulo GSM, utilizando como intermedio el módulo USB a UART. Para el envío de comandos AT se requiere instala la librería *serial*. Mediante tales comandos se puede enviar SMS, leer SMS, marcar algún número telefónico, monitorear el estatus de la llamada, colgar una llamada, entre otras funciones.

  Para el envío de voz durante una llamada, utiliza la librería *TTSDevice.py*, ubicado en la carpeta *Audio*, que transforma una porción del texto del correo recibido a un archivo de audio y se reproduce una vez el usuario al cual se le ha marcado conteste la llamada. Para la transmisión del audio se utiliza un cable auxiliar conectado desde el servidor SMTP Appliance al módulo GSM.
- ✓ *sip.py:* Esta librería opera el protocolo SIP, empleado para registrarse en la central telefónica con las credenciales de extensión y contraseña asignados en la central para este *appliance*, realizar llamadas por VoIP, colgar llamadas, entre otras funciones.
- ✓ *sdp.py:* Maneja el protocolo SDP el cual sirve para indicar el tipo de codificación en la que los datos de audio o video serán trasmitidos en la llamada, la duración de la llamada, la dirección IP a la que se debe conectar el destinatario para comunicarse, entre otros parámetros de llamada.
- ✓ *rtp.py:* Esta librería usa el protocolo RTP, que se encarga de transmitir, en paquetes, los datos de audio

codificados a enviar. Los datos a enviar se crean mediante el empleo de la librería *Audio/Device.py*.

- ✓ *rtcp.py:* Aplicando el protocolo RTCP, dicha librería trasmite datos estadísticos de los paquetes de RTP recibidos del otro dispositivo involucrado en la llamada, permitiendo al otro dispositivo modificar sus parámetros de transmisión de paquetes RTP. De igual forma, recibe datos estadísticos de los paquetes RTP enviados, para que permita modificar la transmisión de paquetes RTP de este servidor.
- ✓ *Audio:* Directorio donde se encuentran todas las librerías de audio.
  - ❖ *__init__.py:* librería en el cual se declaran todas las constantes del paquete Audio.
  - ❖ *Device.py:* Librería principal del paquete Audio. En este archivo se define cual va a ser el modo de operación de las llamadas por VoIP: utilizando el dispositivo de audio, enviando un archivo de audio o transformando un texto a voz. También define el tipo de codificación a utilizar, PCMA o PCMU, dentro de los paquetes de RTP.
  - ❖ *TTSDevice.py:* Librería en la cual transforma el texto a un archivo de audio. En el archivo *processed_text.txt* se almacena el texto ya procesado y en la carpeta *saved* se almacenan los audios correspondientes a cada texto, de tal forma que cada texto que pasé más de una vez por este módulo, no se procesará nuevamente.

- *FileDevice.py:* Este permite procesar archivos de audio de tipo MP3 o WAV para enviarlos a través de VoIP.
- *OSDevice.py:* Se utiliza para enviar datos de la entrada de audio del sistema operativo vía VoIP.
- *processed_text.txt:* Archivo que almacena los textos ya procesados. Cada texto tiene un archivo de audio asignado el cual depende del número de línea en la que se encuentre el texto.
- *saved:* directorio en el que se almacena los audios ya procesados en formato WAV.

Para propósitos de esta tesis, se ha creado un repositorio en github con el cual se puede descargar la versión de desarrollo aquí o con el siguiente comando en la terminal de Linux:

```
$ git clone https://github.com/chunfer/SMTP-Appliance.git
```

Para ponerlo a funcionar hay que abrir una terminal del servidor, ubicarse en el directorio '*Softphone/code*' y utilizar los siguientes comandos:

```
$ sudo python
>> from SMTP_Appliance import SMTP_Softphone
>> server = SMTP_Softphone()
>> server.stop() #Detiene el emisor
>> server.close() #Detiene el receptor
```

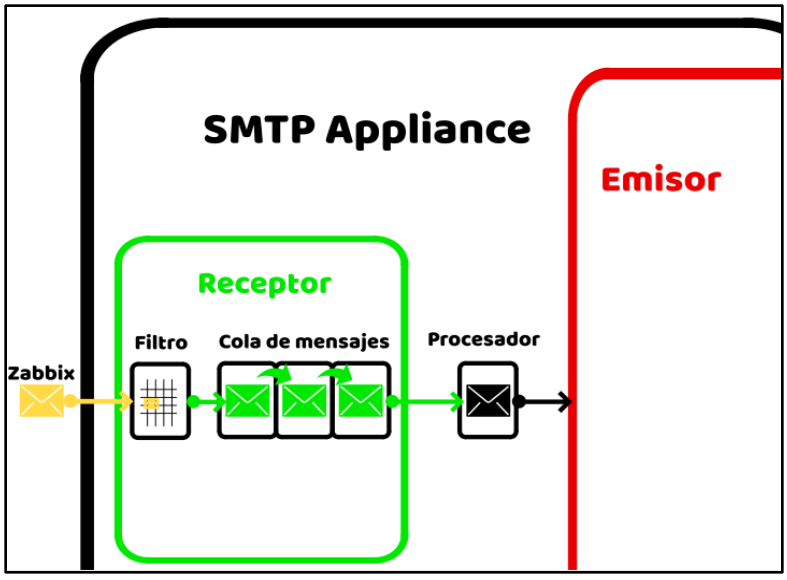
3.2.2. Receptor

El receptor se compone de dos elementos, estos son:

- Filtro: El receptor al recibir el mensaje del servidor de monitoreo Zabbix, hace que este pase por un filtro. El filtro cumple con la función de eliminar todos aquellos mensajes que no son de interés del cliente. Además elimina todos aquellos mensajes de alerta repetidos recibidos dentro de un intervalo de tiempo designado, así como controla la cantidad máxima de mensajes que puede ingresar a la cola.
- Cola de mensajes: Una vez pasa el mensaje por el filtro, se almacena en una cola de mensajes de tipo FIFO. Esta cola contiene todos aquellos mensajes que no han sido procesados y emitidos a los usuarios de interés.

En general el receptor se compone de la siguiente forma:

Figura 3. **Composición de receptor**



Fuente: elaboración propia, empleando Inkscape v0.92.4.

3.2.3. Procesador

Los mensajes entregados del receptor al procesador se dividen en encabezados y cuerpo.

Un mensaje de correo tradicional se compone de la siguiente forma:

Figura 4. **Ejemplo de mensaje de correo entregado al procesador**

----------Encabezados----------
From: <zabbix@smtp.appliance>
To: <juanfmontufarjuarez@gmail.com>
Date: Wed, 26 Dec 2019 22:39:49 -0600
Subject: Problem: Zabbix agent on Wordpress Server is unreachable for 5 minutes
MIME-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: base64

----------Cuerpo----------
UHJvYmxlbSBzdGFydGVkIGF0IDIyOjM5OjQ2IG9uIDIwMTguMTIuMjYNClB
yb2JsZW0gbmFtZTog
WmFiYml4IGFnZW50IG9uIFdvcmRwcmVzcyBTZXJ2ZXIgaXMgdW5yZWFja
GFibGUgZm9yIDUgbWlu
dXRlcw0KSG9zdDogV29yZHByZXNzIFNlcnZlcg0KU2V2ZXJpdHk6IEF2ZXJ
hZ2UNCg0KT3JpZ2lu
YwwgcHJvYmxlbSBJRDogNzgNCg==

Fuente: elaboración propia, empleando Inkscape v0.92.4.

Cada uno de los encabezados de la figura 4 significa lo siguiente:

- *From*: Dirección de correo de donde proviene el mensaje.
- *To*: Dirección de correo del destinatario
- *Date*: Fecha y hora creada del mensaje.
- *Subject*: Asunto del correo.

- *MIME-Version*: Versión de MIME. MIME, extensión multipropósito de correo de internet, es un conjunto de parámetros los cuales definen el contenido del cuerpo del mensaje. MIME permite enviar textos en distintos idiomas y alfabetos. El encabezado de versión indica que las características del cuerpo del mensaje son descritas por los encabezados MIME en su versión 1.0. Actualmente no existe una versión alterna a la 1.0.
- *Content-Type*: Tipo de contenido en el cuerpo del mensaje. El contenido puede ser algún archivo, audio, video, texto o cualquier otro tipo de información. El formato que sigue es:

  Content-Type: <tipo>/<subtipo>; <parámetro>

  En Content-Type la sección '; <parámetro>' es opcional. En este caso del tipo de contenido en el cuerpo del mensaje es texto plano, text/plain, y utiliza el conjunto de caracteres 'UTF-8', charset="UTF-8". Este encabezado es parte de MIME.
- *Content-Transfer-Encoding*: Indica el tipo de codificación utilizada para el cuerpo del mensaje. Esta sección es parte de MIME y se utiliza para enviar cualquier tipo de texto o información a través de la red.

  Los tipos de codificación utilizados son: 7bit, quoted printable, base64, 8bit y binary. En este caso el cuerpo del mensaje es enviado con codificación base64.

El procesador tiene como función, en base al mensaje recibido, generar la información requerida cada módulo del emisor.

La información entregada a cada módulo se muestra en la siguiente tabla:

Tabla II. **Datos entregados por el procesador a cada módulo del emisor**

Módulo	**Datos a entregar**	**Información entregada en base a figura 4**
VoIP	Asunto del correo.	Problem: Zabbix agent on Wordpress Server is unreachable for 5 minutes
SIM	Asunto del correo.	Problem: Zabbix agent on Wordpress Server is unreachable for 5 minutes
SMS	Mensaje recibido sin los encabezados de MIME con cuerpo de correo decodificado. Para la decodificación se utiliza la librería base64, que viene por defecto en Python 2.7.	From: zabbix@smtp.appliance To: juanfmontufarjuarez@gmail.com Date: Wed, 26 Dec 2018 22:39:49 -0600 Subject: Problem: Zabbix agent on Wordpress Server is unreachable for 5 minutes Problem started at 22:39:46 on 2018.12.26 Problem name: Zabbix agent on Wordpress Server is unreachable for 5 minutes Host: Wordpress Server Severity: Average Original problem ID: 78

Continuación tabla II.

Mail	Mensaje completo con encabezado ‘From’ reemplazado por correo del servidor SMTP Appliance.	From: <jmsoftphone@gmail.com> To: <juanfmontufarjuarez@gmail.com> Date: Wed, 26 Dec 2019 22:39:49 -0600 Subject: Problem: Zabbix agent on Wordpress Server is unreachable for 5 minutes MIME-Version: 1.0 Content-Type: text/plain; charset="UTF-8" Content-Transfer-Encoding: base64 UHJvYmxlbSBzdGFydGVkIGF0IDIyOjM5OjQ2IG9uIDIwMTguMTIuMjYNClByb2JsZW0gbmFtZTog WmFiYml4IGFnZW50IG9uIFdvcmRwcmVzcyBTZXJ2Z IgaXMgdW5yZWFjaGFibGUgZm9yIDUgbWludXRlcw0KSG9zdDogV29yZHByZXNzIFNlcnZlcg0KU2V2ZXJpdHk6IEF2ZXJhZ2UNCg0KT3JpZ2luYwgcHJvYmxlbSBJRDogNzgNCg==

Fuente: elaboración propia, empleando Libreoffice v6.1.

3.2.4. Emisor

Esta parte del SMTP Appliance es la encargada de transmitir el mensaje a través de los diversos medios que maneja, a cada uno de los destinatarios correspondientes. Se compone de varios módulos los cuales son: SIM, para el envío de llamadas, SMS, para el envío de mensajes de texto, VoIP, para el manejo de llamadas VoIP, y Mail, para el envío de correo.

Dentro del emisor se puede seleccionar el orden en el cual se enviará el mensaje por los diversos medios o inhabilitar cualesquiera módulos.

3.2.4.1. Módulo SIM

El módulo SIM del emisor se encarga de realizar llamadas telefónicas. Este módulo se comunica con el módulo GSM SIM900 por medio de comandos AT. Los comandos AT son los comandos utilizados para el manejo del módulo GSM y se envían a través del módulo USB a UART conectado al *appliance*.

Para el envío de la alerta audible a través del módulo GSM, el texto recibido de la alerta pasa por una transformación de texto a voz. En tal transformación se genera un archivo de audio en formato WAV, utilizando la librería TTSDevice.py. Este archivo es reproducido por el módulo SIM una vez el destinatario conteste la llamada.

3.2.4.1.1. Comandos AT para envío de llamada

Los comandos utilizados en el proceso de envío de llamada son:

Tabla III. **Comandos AT para llamadas**

Comando	**Función**	**Respuestas**
ATD<número de teléfono>;	Sirve para cualquier número de teléfono con el cual se desea comunicar. Por ejemplo: ATD44116689;	OK: La operación ha sido procesada correctamente. NO CARRIER, BUSY o NO ANSWER: Fallo en la conexión. ERROR: Falla general. OPERATION NOT ALLOWED: Fallo por motivo de seguridad como por ejemplo que no posea tarjeta SIM. UNKNOWN CALLING ERROR: Fallo por razones desconocidas.

Continuación de la tabla III.

AT+CLCC	Brinda información del estado de la llamada.	OK: No existe llamada activa. +CLCC: <id>,<dir>,<est>,<mod>,<conf[,<num>,<tipo>] Los parámetros son: • id: Identifica el número de línea de la llamada. • dir: indica la dirección de la llamada. 0 → saliente, 1 → entrante. • est: indica el estado de la llamada. 0 → activa, 1 → retenida, 2 → marcando, 3 → alerta, 4 → entrando, 5 → esperando. • mod: modo. 0 → voz, 1 → datos, 2 → fax. • conf: modo conferencia. 0 → inactivo, 1 → activo. • num: número con el cual se está comunicando. • tipo: formato de num Ejemplo de respuesta: +CLCC: 1,0,2,0,0,"42781722", 129. Esto representa una llamada saliente en la línea 1 y se está marcando el teléfono en modo de voz.
ATH	Sirve para colgar la llamada	OK: llamada colgada correctamente. NO CARRIER: sin conexión.

Fuente: AGNIHOTRI, Nikhil. *AT Commands, GSM AT command set*. https://www.engineersgarage.com/tutorials/at-commands-gsm-at-command-set. Consulta: 22 de mayo de 2019.

Dichos comandos son enviados desde el módulo SIM del servidor SMTP Appliance hacia el módulo GSM.

3.2.4.1.2. Proceso de llamada telefónica

Los pasos del proceso son los siguientes:

- Recibir la información destinada al módulo SIM.

- Transformar la información destinada a un archivo de audio.

  La librería ‘gTTS’, que se debe instalar, permite generar un archivo MP3 en base a un texto recibido y a un lenguaje indicado. Este archivo es reformateado a WAV para luego ser reproducido. En este paso también existe un proceso almacenamiento de los archivos WAV en la carpeta *saved,* así como los textos ya procesados en el archivo *processed_text.txt*, lo que evita el reprocesamiento del texto.

- Marcar el número del equipo de destino utilizando el comando ATD.

- Verificar el estado de la llamada.

  Enviando continuamente el comando AT+CLCC permite revisar si la llamada está activa o no. Para proseguir al siguiente paso la llamada se debe encontrar en estado activo.

  Para que la llamada esté activa se debe obtener un ‘0’ en el parámetro de estado ‘est’, por lo que se debe recibir como respuesta al comando AT+CLCC algo similar a lo siguiente:

  +CLCC: 1,0,**0**,0,0,"44551321",129

- Reproducir archivo de audio WAV.

  Para reproducir el archivo de audio se utiliza el siguiente comando en la terminal de Linux:

  $ aplay **<archivo de audio>**

- Colgar la llamada mediante el comando ATH.

3.2.4.2. Módulo SMS

Como se mencionó, para el envío de SMS solamente se utilizan comandos AT enviados al módulo GSM mediante el módulo USB a UART.

3.2.4.2.1. Comandos AT para envío de SMS

Los comandos utilizados en el proceso de envío de SMS son:

Tabla IV. **Comandos AT para SMS**

Comando	Función	Respuestas
ATZ	Se utiliza para regresar a todas las configuraciones de estado a sus valores predeterminados	OK: La operación ha sido procesada correctamente.
AT+CMGF=x	Configura el módulo GSM para enviar el SMS en modo texto o PDU. x = 1 → modo texto, x = 0 → modo PDU	OK: La operación ha sido procesada correctamente.
AT+CMGS=" <número>"	Sirve para indicar a que número se enviara el SMS. Este comando inicia el proceso de envío del SMS.	OK: La operación ha sido procesada correctamente.

Fuente: AGNIHOTRI, Nikhil. *AT Commands, GSM AT command set*. https://www.engineersgarage.com/tutorials/at-commands-gsm-at-command-set. Consulta: 22 de mayo de 2019.

3.2.4.2.2. Proceso de envío de SMS

Los pasos del proceso son los siguientes:

- Recibir la información destinada al módulo SMS.
- Restablecer los parámetros de estado del módulo GSM utilizando en comando ATZ.
- Configura el módulo GSM para enviar el SMS en modo texto enviando el comando AT+CMGF=1.

- Asignar el número al cual se le desea enviar el SMS mediante el comando AT+CMGS.

- Enviar el texto que se desea transmitir.

- Hay que tomar en cuenta que solamente se puede enviar una cantidad máxima de 160 caracteres en un mensaje, por lo que si el texto supera a este valor, es recomendable repetir este proceso nuevamente por cada 160 caracteres del texto.

- Enviar caracter de finalización de mensaje de texto.

El caracter es el número 26 del código ASCII, el cual representa ‘Ctrl+Z’.

3.2.4.3. Módulo VoIP

VoIP, Voz sobre IP, es un conjunto de tecnologías que permiten el envío de audio y diversos tipos de datos multimedia, como video, a través de redes IP.

Los protocolos utilizados de VoIP fueron:

Tabla V. **Protocolos utilizados para VoIP**

Protocolo	**Descripción**
SIP, *Session Initiation Protocol*	Se utiliza para la inicializar sesiones. Con este protocolo se pueden administrar las llamadas de cualquier teléfono IP. También se puede utilizar para administrar mensajes de texto, video, juegos en línea, entre otros. Este protocolo se combina con otros tres protocolos los cuales son SDP, RTP y RTCP. Otros protocolos también utilizados en vez de SIP son: IAX, IAX2, SCCP, H.323, MGCP y Skype.
SDP, *Session Description Protocol*	Este describe la información de la sesión. En este protocolo se envía la información de cómo es que se va a transmitir la información: formato de audio, formato de video, usuario de origen, y demás.
RTP, *Real-time Transport Protocol*	Es el encargado de enviar datos de audio y/o video codificados. Los tipos de codificación son PCMU, PCMA, GSM, DVI4, ADPCM, L16, etcétera.
RTCP, *Real-time Transport Control Protocol*	Es el encargado de enviar datos estadísticos del protocolo RTP. En base a esta información, los parámetros de la transmisión: tiempo de envío del paquete, cantidad de muestras de voz por paquete, entre otros, de paquetes de RTP puede ser modificada.

Fuente: ROSENBERG, Jonathan. *SIP: Session Initiation Protocol*. https://tools.ietf.org/html/rfc3261. Consulta: 10 de enero de 2019.

3.2.4.3.1. RTP

Para el envío de audio a través de VoIP mediante los paquetes RTP, se requiere que el audio pase por un proceso de muestreo y codificación. El audio

puede provenir de un micrófono o un archivo de audio el cual se reproduce durante la llamada.

Los parámetros a tomar en cuenta en el procesamiento del audio son: frecuencia de muestreo y tipo de codificación. La siguiente tabla muestra los tipos de codificaciones comunes con su frecuencia de muestreo correspondiente:

Tabla VI. **Codificaciones tradicionales de audio en RTP**

Codificación	**Frecuencia de muestreo**	**Descripción**
L8	44 100	Representa un muestreo lineal de una señal PCM de 8 bits de precisión sin signo.
L16	44 100	Representa un muestreo lineal de una señal de 16 bits PCM de precisión con signo.
PCMU	8 000	Representa a la codificación PCM el escalamiento logarítmico ley Mu.
PCMA	8 000	Representa a la codificación PCM el escalamiento logarítmico ley A.
GSM	8 000	Representa a la codificación *Full Rate* utilizado en el estándar mundial GSM.
DVI4	8 000	Dicha codificación representa a la codificación por modulación de impulso diferencia adaptativa, ADPCM.
CN	8 000	Ruido de confort. Sirve para reemplazar el ruido en el audio.

Fuente: SCHULZRINNE, Henning. *RTP Profile for Audio and Video Conferences with Minimal Control*.https://tools.ietf.org/html/rfc3551. Consulta: 20 de mayo de 2019.

El texto que proviene del Procesador pasa por un proceso en el que se empaqueta en forma de audio codificado. Tal proceso es:

- Recibir la información en forma de texto destinada al módulo VoIP.

- Analizar si el texto ya fue procesado con anterioridad. De ser así, debe estar almacenado en el archivo *processed_text.txt*, con un archivo de WAV asignado en la carpeta *saved*, con el que se puede proseguir al paso 6.

- Transformar el texto a voz con la librería ‘gTTS’. Tal librería generará un archivo de tipo MP3 con una frecuencia de muestreo de 44 100 y codificación L16.

- Transformar dicho archivo MP3 a WAV con la librería de Linux ‘lame’.

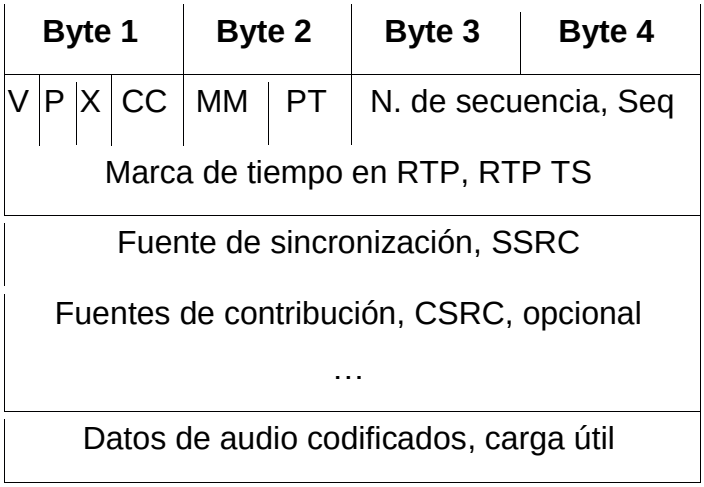
- Utilizar la librería de Linux ‘sox’. Esta librería se encarga de disminuir la frecuencia de muestreo en el archivo WAV de 44 100 a 8 000 y de almacenar dicho archivo en la carpeta *saved*.

- Con la librería por defecto de Python ‘wave’ se abre el archivo WAV y se extraen sus muestras.

- Una vez extraído, estas muestras se separan en porciones de 160 muestras, lo que representa 20 ms del audio, y se recodifican a PCMU o PCMA con la librería ‘audioop’, lo que reduce la cantidad de bits de 16 a 8 por muestra, así como el ancho de banda requerido para la transmisión del paquete.

Un paquete RTP posee la siguiente estructura:

Figura 5. **Paquete RTP**



Fuente: SCHULZRINNE, Henning. *RTP: A Transport Protocol for Real-Time Applications*. https://tools.ietf.org/html/rfc3550. Consulta: 07 de junio de 2019.

Los primeros doce bytes están presentes en cualquier paquete de RTP, desde V hasta SSRC, formando parte del 'encabezado fijo'. La descripción de la figura 5 es la siguiente:

- Número de versión de RTP; V, 2 bits: Es la versión de RTP la cual actualmente es la 2.
- Relleno; P, 1 bit: Si este bit es igual a 1, significa que existe uno o más bytes al final del paquete RTP que no pertenecen a los datos de audio. Se utiliza para algoritmos de cifrado. En este caso se asignó como 0.
- Extensión; X, 1 bit: Este bit indica que existe un encabezado seguido después del encabezado fijo, como son los encabezados de CSRC. En este proyecto tiene un valor de 0.
- Conteo de CSRC; CC, 4 bits: Indica la cantidad de fuentes contribuyentes. Dicho conteo se utiliza en un mezclador el cual permite transmitir varias fuentes de paquetes RTP en conjunto, como la transmisión de audio estéreo, la transmisión conjunta de audio y video, entre otros. Como solo

se cuenta con una fuente generadora de paquetes RTP, este parámetro es 0 por lo que la trama de ‘fuentes de contribución’ no existe.

- Marcador; M, 1 bit: Si este bit es igual a 1, representa un evento importante en la parte de la trama de envío de paquetes RTP, por ejemplo, el envío de tonos DMTF por RTP indica que se ha presionado un tono del teclado. En el envío de audio se mantiene en 0.
- Tipo de carga útil; PT, 7 bits: Según el número indicado en el tipo de carga útil, es la codificación que se utilizó en el audio del paquete. Algunos de los estándares de tipo de carga útil según tipo de codificación son:

Tabla VII. **Estándares en el tipo de carga útil en RTP**

Codificación	Tipo de carga útil
L16	11
PCMU	0
PCMA	8
GSM	3
DVI4	5
CN	13

Fuente: SCHULZRINNE, Henning. *RTP Profile for Audio and Video Conferences with Minimal Control*. https://tools.ietf.org/html/rfc3551. Consulta: 25 de mayo de 2019.

Hay codificaciones cuyo tipo de carga útil es variable como L8, GSM-EFR, entre otras. Para estos, el tipo de carga se debe especificar en el protocolo SDP. En este caso solamente se utilizarán las cargas de tipo PCMU con posibilidad de usar PCMA y CN.

- Número de secuencia; Seq, 16 bits: Tal número identifica la posición del paquete en la secuencia de paquetes. Este parámetro incrementa una unidad con cada paquete enviado. Su valor inicial es aleatorio y se regresa a 0 cuando llega al valor de 2 ^ 16.
- Marca de tiempo RTP; RTP TS, 32 bits: Indica el instante de muestreo la primera muestra de los datos codificados. Tal valor incrementa según la cantidad de muestras enviadas en el paquete anterior. En esta ocasión se envían 160 muestras por paquete, por lo el valor de RTP TS incrementa por 160 cada vez que se envía un paquete. Su valor inicial, al igual que Seq, es aleatorio.
- Fuente de sincronización; SSRC, 32 bits: Permite identificar de que fuente proviene el paquete. Puede que dos equipos posean la misma dirección IP, sin embargo la fuente es distinguida por su SSRC. El SSRC de una fuente es generado de manera aleatoria.
- Fuentes de contribución; CSRC, 32 bits c/u: Indica cuales son las fuentes que contribuyen a la carga útil, es decir conjunto de SSRC contribuyentes. Se utiliza en mezcladores. Dado que el valor de CC utilizado es 0, estos encabezados no existen en el paquete.
- Datos de audio codificados; carga útil: Son los datos obtenidos del proceso de muestreo y codificación. Se extraen las 160 muestras codificadas, luego se agregan al paquete de RTP. Para el envío de paquete de tipo CN, las 160 muestras poseen un valor de 0.

La transmisión de paquetes RTP generada sigue la siguiente secuencia:

- Iniciar la transmisión con el envío un paquete de tipo CN cada 20 ms durante 2 segundos. Esto da un lapso de tiempo para que el usuario responda al teléfono IP.

- Enviar un paquete RTP con el audio codificado con PCMU cada 20 ms hasta terminar la reproducción del audio.
- Una vez finaliza la reproducción del audio, se continúa enviando paquetes de tipo CN hasta que el usuario en la otra línea finalice la llamada o hayan transcurrido 2 segundos.

3.2.4.3.2. RTCP

RTCP está compuesto por un conjunto de paquetes los cuales son:

- Reportes de transmisor; sender report o SR: Tal paquete cumple con la función de enviar estadísticas de transmisión y recepción de los participantes considerados como transmisores activos, es decir todo aquel equipo que transmite paquetes RTP.
- Reportes de receptor; receiver report o RR: Se utiliza para el envío de estadísticas de recepción de aquellos participantes que no son considerados transmisores activos, que son equipos que se utilizan únicamente para el control de la transmisión.
- Descriptor de la fuente; Source descriptor o SDES: Indica las características de la fuente: ubicación geográfica, correo, número telefónico, nombre de dominio o CNAME, entre otras.
- Finalización de transmisión; BYE: Se utiliza para indicar que la fuente de paquetes RTCP ya no se encuentra activo. Es utilizado en caso de encontrar una falla en la transmisión, con lo que se corta la transmisión y recepción de paquetes RTP, o en caso se desee deliberadamente finalizar la comunicación.
- Aplicación; APP: Paquete que es utilizado para aplicaciones específicas o experimentales.

Cada uno de los paquetes cuenta son su propia forma de empaquetado. Sin embargo, un paquete RTCP se compone de agrupar a estos paquetes de la siguiente forma:

Figura 6. **Paquete RTCP**

Trama 1	**Trama 2**	**Trama 3**	**Trama 4**
SR o RR, según la fuente	SDES	BYE, opcional	APP, opcional

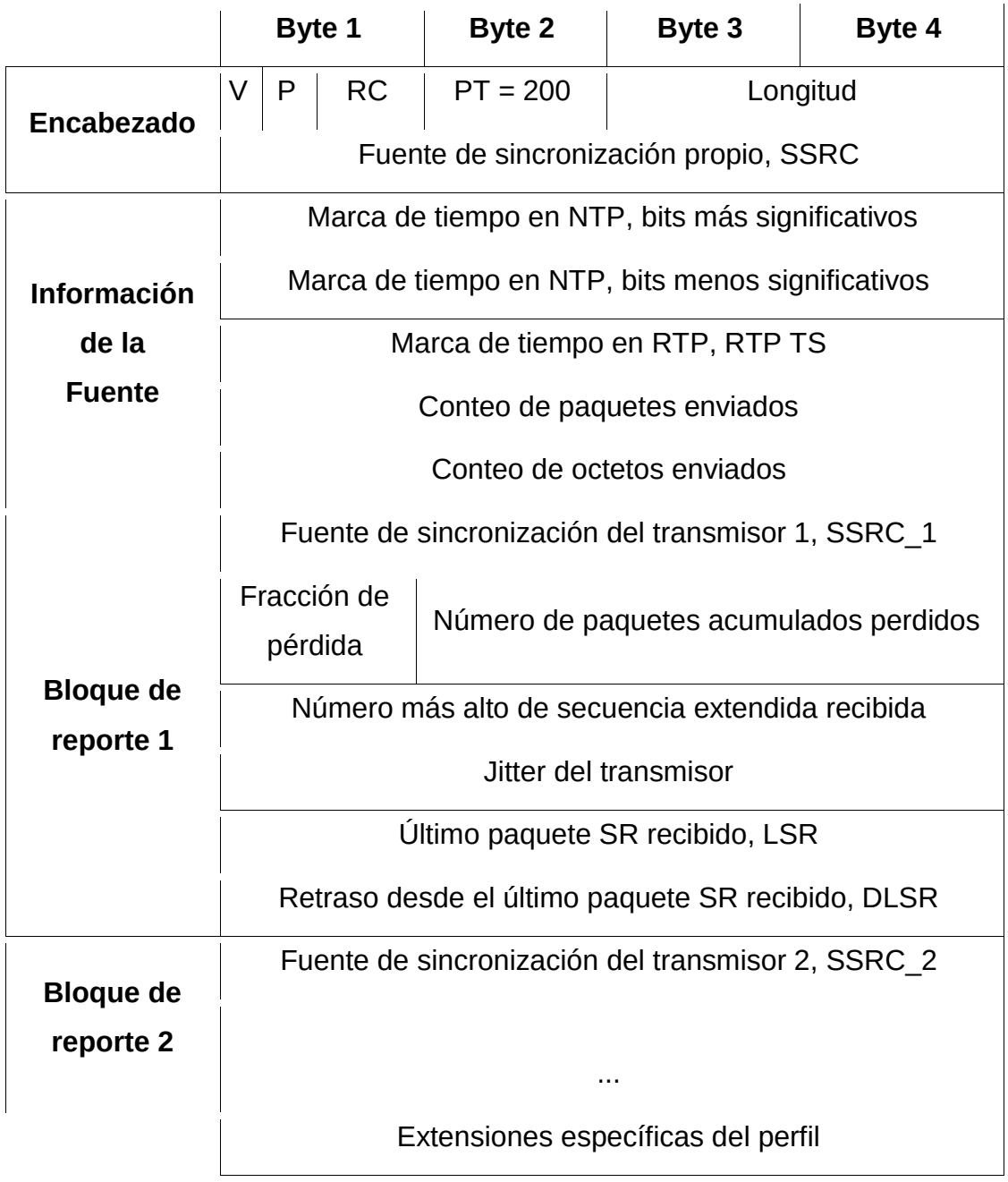
Fuente: SCHULZRINNE, Henning. *RTP: A Transport Protocol for Real-Time Applications*. https://tools.ietf.org/html/rfc3550. Consulta: 14 de junio de 2019.

Durante la transmisión de paquetes de voz RTP, estos paquetes RTCP son enviados en intervalos de 5 segundos.

Los tipos de paquetes RTCP manejados por el SMTP Appliance son: SR, dado que este equipo es considerado como un transmisor activo, SDES y BYE. Dado que tanto los paquetes de tipo RR como los de tipo APP no son utilizados, su composición no será mostrada.

Los paquetes SR están representados en la siguiente figura:

Figura 7. **Paquete SR**



Fuente: elaboración propia, empleando Libreoffice v6.1.

La descripción de cada bloque es:

- Versión; V, 2 bits: Indica la versión de los paquetes RTP utilizados. Al igual que en RTP, este valor es 2.
- Relleno; P, 1 bit: Este bit, al igual que en RTP, indica que existo por lo menos uno o más bits al final del paquete que no pertenecen a la carga útil. Se utiliza para algoritmos de cifrado. En este caso se le asignó 0 dado que no se utilizan bits de relleno.
- Conteo de bloques de reporte de recepción; RC, 5 bits: Demuestra cuantos bloques de reporte de recepción hay en el paquete. Por cada transmisor que envía paquetes RTP a la fuente debe existir un reporte de recepción.
- Tipo de paquete; PT, 8 bits: Se utiliza para distinguir que tipo de paquete RTCP se está transmitiendo. Para los paquetes de tipo SR dicho valor es de 200.
- Longitud; *length*, 16 bits: Cuenta la cantidad de tramas de 32 bits existentes en el paquete SR y le resta una unidad por el primer encabezado.
- Fuente de sincronización propia; SSRC, 32 bits: Es el mismo valor del encabezado de fuente de sincronización SSRC transmitido en los paquetes RTP.
- Marca de tiempo en NTP; NTP TS, 64 bits: La marca de tiempo en NTP mide la cantidad de segundos transcurridos desde las 12:00 a.m. del 1 de enero de 1900 en horario UTC. Como la marca de tiempo es un valor irracional, la parte entera forma los primeros 32 bits, los cuales son los bits más significativos, mientras que la parte fraccional se multiplica por 2^32 formando los bits menos significativos. Un ejemplo de la separación de bits es el siguiente:

  Marca de tiempo en UTC = 3 768 677 463,120 127 7

Bits más significativos = 3 768 677 463

Bits menos significativos = 0,1201277 * 2 ^ 32 = 515 944 542

- Marca de tiempo en RTP; RTP TS, 32 bits: Indica el instante temporal de muestreo en el que se envía el paquete RTCP. Su valor inicial es el mismo que el valor inicial del encabezado RTP TS del paquete RTP. Dado que se envía un paquete RTCP cada 5 segundos y la frecuencia de muestreo utilizada es de 8 000 muestras por segundo, lo que representa 40 000 muestras, este valor incrementa cada vez por 40 000.
- Conteo de paquetes enviados; 32 bits: Indica cuantos paquetes RTP se han enviado en el transcurso de la comunicación.
- Conteo de octetos enviados; 32 bits: Envía datos acerca de cuantos octetos de bits, denominados bytes, de carga útil se han enviado en los paquetes RTP.
- Fuente de sincronización del transmisor n; SSRC_n, 32 bits: Es el SSRC de cada uno de transmisores activos con los cuales se está comunicando.
- Fracción de pérdida; 8 bits: Representa al porcentaje de paquetes perdidos del transmisor SSRC_n desde el último reporte SR enviado.
- Cantidad acumulada de paquetes perdidos; 24 bits: Cantidad total de paquetes perdidos por el transmisor SSRC_n desde el último reporte SR enviado.
- Número más alto de secuencia extendida recibida; Ext Seq, 32 bits: La secuencia extendida toma en cuenta el valor de la secuencia actual más las veces que la secuencia se reinició multiplicado por 2 ^ 16 (Seq actual + Reset_Seq * 2 ^ 16). En esta porción se toma el número más alto de la secuencia extendida recibida en los paquetes RTP del transmisor SSRC_n justo antes de enviar el paquete RTCP.
- Jitter del transmisor; 32 bits: El Jitter representa a una varianza en la recepción de los paquetes RTP. Es decir, que indica un retraso o adelanto

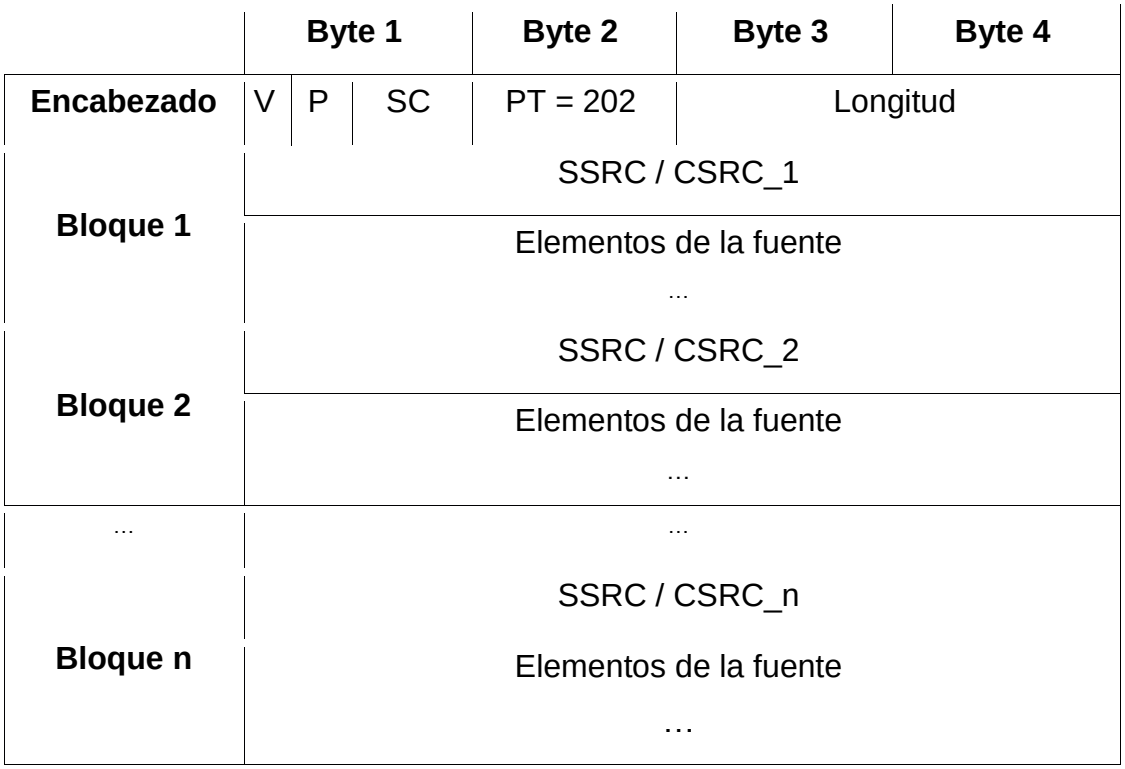
en la entrega de paquetes RTP del transmisor SSRC_n a la fuente. El jitter efectúa una operación matemática entre los valores de RTP TS del transmisor SSRC_n y de esta fuente.

- Último paquete SR recibido; LSR, 32 bits: Se utilizan los 32 bits de en medio marca de tiempo NTP, mitad inferior de los bits más significativos y mitad superior de los bits menos significativos, del último reporte SR recibido del transmisor SSRC_n.
- Retraso desde el último paquete SR recibido; DLSR, 32 bits: Toma en cuenta el tiempo transcurrido desde el último paquete SR recibido del transmisor SSRC_n.
- Extensiones específicas del perfil: Se utiliza en caso de que se requiera enviar cierta información específica de la fuente o de los transmisores.

Las estadísticas enviadas permiten determinar la calidad del servicio que se está proveyendo durante la llamada, así como modificar los parámetros de transmisión de paquetes RTP: tipo de codificación, cantidad de muestras por paquete, intervalo de transmisión de paquetes u otro, para mejorar la calidad de dicho servicio.

Los paquetes de tipo SDES poseen la siguiente estructura:

Figura 8. **Paquete SDES**



Fuente: elaboración propia, empleando Libreoffice v6.1.

La descripción de cada sección es la siguiente:

- Versión; V, 2 bits: Versión de RTP. Equivale a 2.
- Relleno; P, 1 bit: Se utiliza para algoritmos de cifrado.
- Conteo de fuentes; SC, 5 bits: Número de bloques SSRC/CSRC.
- Tipo de paquete; PT, 8 bits: Para los paquetes SDES, este posee un valor de 202.
- Longitud; *length*, 16 bits: Cuenta la cantidad de tramas de 32 bits existentes en el paquete SDES y le resta la unidad por el primer encabezado.

- SSRC / CSRC_n; 32 bits: Fuente de sincronización o contribución a la cual se está describiendo en el Bloque n. Los paquetes SDES describen tanto fuentes de sincronización como fuentes de contribución.
- Elementos de la fuente; múltiplo de 32 bits: Son los elementos que describen a la fuente SSRC / CSRC_n del Bloque n. En esta sección la cantidad de bits total de los elementos debe ser un múltiplo de 32 bits, por lo que se pueden utilizar bytes de ceros de relleno para llegar a esta cantidad.

  Los elementos de la fuente son:

  - CNAME: El nombre canónico está compuesto por un nombre de dominio o dirección IP y un usuario en caso de existir. Algunos ejemplos de nombres canónicos para sistemas multi-usuario pueden ser ‘juan@dominio.com’, ‘juan@1.0.0.2’ o ‘juan@2201::1111’. Para sistemas en los que no existe nombre usuario podrían ser ‘alpha.patitos.com’, ‘10.120.120.2’ o ‘2004::BBBB:1’. Es de destacar que se puede utilizar tanto un dominio como una dirección IPv4 o IPv6 en este elemento.

    Dado que los SSRC y CSRC son valores generados de forma aleatoria, existe una probabilidad de que dos equipos posean el mismo valor para identificador de la fuente, aunque la probabilidad es muy baja, por lo que el CNAME permite identificar colisiones entre las fuentes con el mismo valor de SSRC o CSRC. Cuando se detecta una colisión entre fuentes, ambos valores de SSRC o CSRC se deben modificar inmediatamente, enviar un paquete RTCP con BYE incluido, para luego reiniciar la transmisión de paquetes.
    El elemento CNAME es el único elemento obligatorio en los paquetes SDES.

- NAME: Es el nombre verdadero del usuario. Por ejemplo ‘Juan Pérez Leal’.
- EMAIL: Es el correo del usuario. Por ejemplo ‘abc@hotmail.com’.
- PHONE: Teléfono del usuario. Siempre debe iniciar con un símbolo ‘+’ e incluyendo el código de acceso internacional. Por ejemplo ‘+502 2200 2000’.
- LOC: Ubicación geográfica. Dependiendo de la aplicación, se pueden utilizar coordenadas o direcciones. Por ejemplo ‘15 avenida 2-95 zona 5’.
- TOOL: Indica el nombre de la aplicación o herramienta utilizada. Por ejemplo ‘Python 2.7’.
- NOTE: Es utilizado para indicar el estado actual de la fuente. Por ejemplo ‘En llamada’.
- PRIV: Utilizado para aplicaciones experimentales.

La estructura general de cada elemento, con la excepción de los elementos de tipo PRIV, es la siguiente:

Figura 9. **Estructura general de elementos SDES**

Trama 1	**Trama 2**	**Trama 3**	**Trama 4**
Tipo de elemento	Longitud	Valor	Fin

Fuente: elaboración propia, empleando Libreoffice v6.1.

- Cada trama representa lo siguiente:

- Tipo de elemento; 8 bits: Indica el valor del tipo de elemento utilizado. El valor asignado a cada tipo de elemento se presenta en la siguiente tabla:

Tabla VIII. **Tipos de elementos SDES**

Tipo de elemento	**Valor asignado**
CNAME	1
NAME	2
EMAIL	3
PHONE	4
LOC	5
TOOL	6
NOTE	7

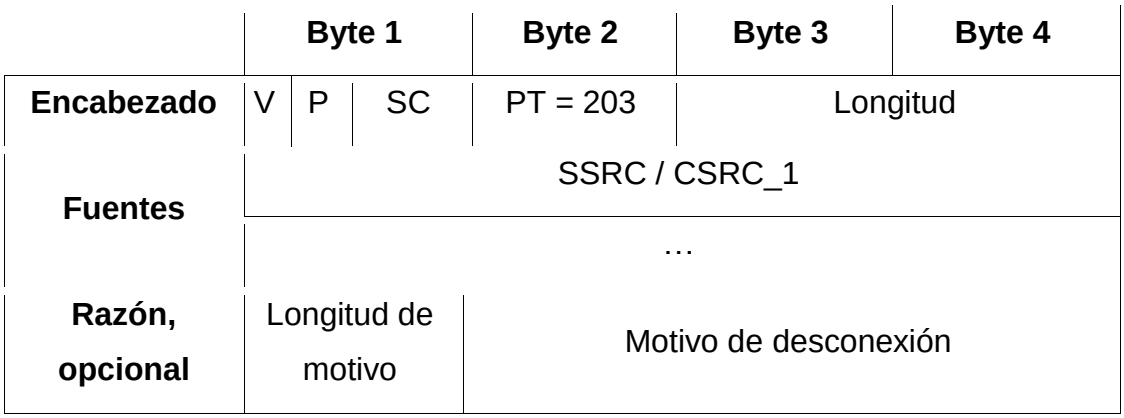
Fuente: elaboración propia, empleando Libreoffice v6.1.

- Longitud; 8 bits: Indica la cantidad de bytes que posee el valor del elemento.
- Valor: Es el valor asignado al elemento. Su longitud es variable.
- Fin: Representa a uno o más bytes de valor 0. Se utiliza para indicar cuando se ha finalizado el elemento y de relleno para que el empaquetado del elemento forme una cantidad de bits múltiplo de 32. En los elementos, siempre debe existir al menos un byte con valor de 0.

Dado que los elementos de tipo PRIV son experimentales, su estructura no es de vital importancia para motivos de este documento.

El paquete BYE posee la siguiente estructura:

Figura 10. **Paquete BYE**



Fuente: elaboración propia, empleando Libreoffice v6.1.

La descripción de cada sección es:

- Encabezado: Esta sección es similar a la del paquete SDES, con la diferencia que el conteo de fuentes, SC, representa a todas las fuentes que se desligan de la conexión y el tipo de paquete, PT, tiene el valor de 203.
- Fuentes: Es el conjunto de fuentes que se desligan de la conexión. Es de recalcar nuevamente que cada fuente se identifica mediante su valor de SSRC o CSRC.
- Razón: Aquí se define el motivo de la desconexión de las fuentes, así como la longitud, 8 bits, en bytes del motivo de la desconexión. Dicha sección

siempre debe completar que su cantidad de bits sea un múltiplo de 32, por lo que, de ser necesario, se pueden agregar bytes de 0 como relleno.

Media vez el envío de paquetes RTP funcione con buena calidad de servicio, el SMTP Appliance envía en la trama RTCP solamente paquetes de tipo SR y SDES.

En dado caso que exista un problema en la transmisión de paquetes RTP, por ejemplo, que haya un jitter muy alto, una colisión en fuentes de sincronización dentro de la transmisión, muchos paquetes perdidos u otro, se envía la trama RTCP con paquete de tipo BYE incluido, con lo que concluye la transmisión.

3.2.4.3.3. SDP

Este protocolo se encarga de describir una sesión mediante una serie de atributos del participante. Solamente puede haber un atributo por línea.

Los atributos se dividen en tres tipos: descripción general de la sesión, descripción del tiempo y descripción de medios. Cada uno de los atributos asignados debe estar seguido por un símbolo ‘=’ más su valor respectivo.

Los atributos de descripción general de la sesión indican datos de la comunicación, así como información ordinaria de la sesión.

Los atributos principales son:

Tabla IX. **Atributos de descripción general de la sesión**

Atributo	**Descripción**	**Ejemplos de línea**
v	Versión del protocolo SDP. La versión actual es la 0.	v=0
o	Origen de la sesión.	o=fmon 2890844526 2890844526 IN IP4 10.20.248.47 Formato: o=<us> <id-ses> <ver-ses> <tipo-red> <ver-ip> <ip> • us: Representa al usuario. • Id-ses: Es el identificador único de la sesión. Generalmente se usa la marca de tiempo NTP. • Ver-ses: Número de versión del SDP. Se recomienda usar la marca de tiempo NTP. • Tipo-red: Actualmente solo existe un tipo de red para SDP que es IN de Internet. • Ver-ip: Si es IPv4 o IPv6. • Ip: Dirección IP de donde se origina la sesión
s	Nombre de la sesión. Puede ser aleatorio. No debe estar vacío.	s=Sesión de prueba

Continuación de la tabla IX.

c	Información de conexión. Sirve para indicar a que dirección IP se debe conectar. Tiene mucho uso en *multicast* para conferencias.	c=IN IP4 223.231.65.3 Formato: c=<tipo-red> <ver-ip> <ip> • Tipo-red: Actualmente solo existe un tipo de red para SDP que es IN de Internet. • Ver-ip: Si es IPv4 o IPv6. Ip: Dirección IP de donde se debe conectar.

a	Define ciertos parámetros de la sesión. Tal atributo también se utiliza para la descripción de los medios. Entre sus funciones principales en la descripción de la sesión están: indicar el modo de transmisión del participante, indicar el tiempo de transmisión por paquete de audio en ms, así como el tiempo máximo de transmisión por paquete en ms. Los modos existentes de transmisión son sendonly, solo envía paquetes, recvonly, solo recibe paquetes, y sendrecv, puede tanto enviar como recibir.	a=sendrecv # modo de transmisión
		a=ptime:50 # tiempo del paquete
		a:maxptime:100 #tiempo de paquete máximo
		
		Formato:
		a=<parámetro>
		a=<parámetro>:<valor>

Fuente: HANDLEY, Mark. *SDP: Session Description Protocol*. https://tools.ietf.org/html/rfc2327. Consulta: 23 de junio de 2019.

Los atributos de *descripción del tiempo* se utilizan para el control temporal de la sesión. Estos son:

Tabla X. **Atributos de descripción del tiempo**

Atributo	**Descripción**	**Ejemplos de línea**
t	Tiempo de duración de la sesión.	t=2890844526 2900844526 Formato: t=<NTP_inicio> <NTP_fin> • NTP_inicio: Valor de inicio de la sesión en formato NTP medido desde el 1 de Enero de 1900. • NTP_fin: Valor de finalización de la sesión en formato NTP. Si posee el valor de 0 significa que la sesión no posee un final e inicia en el tiempo NTP_inicio. NOTA: Si tanto NTP_inicio y NTP_fin valen 0, significa que la sesión no posee un control temporal.

Continuación de la tabla X.

r	Repeticiones de la sesión. Este parámetro es opcional	r=3600 60 0 120 Formato: r=<int_rep> <dur> <offsets> • int_rep: Intervalo, en segundos, en el que se repite la sesión entre el NTP_inicio y el NTP_fin. • dur: Duración, en segundos, de la sesión. • Offsets: Indican los tiempos, en segundos, en los que se realiza una repetición después de haber transcurrido el int_rep. NOTA: para este ejemplo, se generan repeticiones de sesión cada 3 600 s, con duración de 60 s, e iniciando la primera sesión inmediatamente, y la segunda 120 s después de haber iniciado.

Fuente: HANDLEY, Mark. *SDP: Session Description Protocol*. https://tools.ietf.org/html/rfc2327. Consulta: 23 de junio de 2019.

Los atributos de descripción de medios se utilizan para especificar los métodos soportados para transmisión de datos. Tales atributos indican los tipos de codificación, frecuencia de muestreo y tipo carga útil de los paquetes RTP que es capaz de manejar el participante.

Los atributos de descripción de medios principales son:

Tabla XI. **Atributos de descripción de medios**

Atributo	**Descripción**	**Ejemplos de línea**
m	Describe los medios disponibles en el participante.	m=audio 45520 RTP/AVP 0 3 13 Formato: m=<med> <puerto> <proto> <fmts> • med: Es el tipo de dato que se va a transmitir. Los tipos disponibles son: audio, video, texto, mensaje y aplicación. • puerto: Es el puerto utilizado para la transmisión de datos. Para RTP se establece que los puertos deben ser pares y para RTCP debe ser una unidad mayor al puerto RTP. • proto: Indica el protocolo utilizado. Para RTP se utiliza RTP/AVP para SRTP, se utiliza RTP/SAVP, y para cualquier otro protocolo UDP, se usa UDP. • fmts: Tipos de carga útil disponibles. Para RTP y SRTP se utilizan los estándares de tipos de carga útil.

Continuación de la tabla XI.

a	En los atributos de descripción de los medios, se utiliza para mapear todos los tipos de carga útil indicados en el atributo 'm', con sus respectivos tipos de codificación y tasa de muestreo.	a=rtpmap:0 PCMU/8000 a=rtpmap:3 GSM/8000 a=rtpmap:13 CN/8000 Formato: a=rtpmap:<tipo carga> <cod>/<frec> • tipo carga: Tipo de carga útil a describir. • Cod: codificación utilizada según el tipo de carga descrita. • frec: Frecuencia de muestreo.

Fuente: HANDLEY, Mark. *SDP: Session Description Protocol*. https://tools.ietf.org/html/rfc2327. Consulta: 28 de junio de 2019.

Una descripción de sesión completa sería:

```
v=0
o=jfer 3210844526 3210844526 IN IP4 10.10.10.4
s=sesion del SMTP Appliance
c=IN IP4 10.10.10.4
a=sendrecv
a=ptime:20
a=maxptime:150
t=0 0
m=audio RTP/AVP 0 8 13
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:13 CN/8000
```

3.2.4.3.4. SIP

Dentro de protocolo SIP existen las siguientes entidades:

- Agentes de usuario: Estas entidades representan a cualquier dispositivo terminal que maneja el protocolo SIP. Estos dispositivos pueden ser: videoteléfono, teléfono IP, un cliente de software, entre otros. Los agentes de usuario de pueden comportar como clientes, denominados UAC, cuando realizan peticiones, o como servidores, denominados UAS, cuando las reciben.
- Servidor de registro: Permite determinar los puntos de red en los que se encuentran conectados los agentes de usuario. El servidor de registro asigna a cada agente una dirección lógica invariable respecto a su dirección IP, la cual puede ser variable. Las direcciones lógicas poseen el mismo formato que una dirección de correo la cual es: 'usuario@dominio'. Para la autenticación de los usuarios existe un proceso de registro, en el cual el servidor hace un enlace entre la dirección lógica y su dirección IP. Los servidores de registro permiten conectar a varios agentes indistintamente de su ubicación física.
- Servidor proxy: Este tipo de servidor cumple con la función de intermediar la comunicación entre un UAC con un UAS. Los servidores proxy generalmente cumple también con la función de servidor de registro, permitiendo la localización de tanto el UAC como el UAS, haciendo su comunicación posible. El servidor proxy recibe los paquetes de los protocolos pertenecientes a VoIP como si fueran dirigidas hacia él, para luego encaminarlas hacia su destino. El servidor proxy forma parte constante del proceso de comunicación entre el UAC y el o los UAS.
- Servidor redirector: También cumple con la función de comunicar al UAC con el o los UAS. Dicho servidor generalmente cumple de igual forma con

la de ser un servidor de registro. El servidor redirector indica, al inicio de la comunicación, al UAC como encaminar el mensaje hacia el o los UAS, luego ya no interviene en el proceso de comunicación.

- Servidor de localización: Este tipo de servidor solo indica al UAC como es posible localizar al o a los UAS. Dicho servidor no interviene en el proceso de comunicación.

Como se ha mencionado antes, el servidor FreePBX tiene la función de un servidor proxy con registro. Además, el SMTP Appliance y el equipo de destino se categorizan como agentes de usuario.

Para la comunicación entre entidades, el protocolo SIP utiliza varios mensajes de petición y respuesta. De todos ellos, los mensajes utilizados por el SMTP Appliance son los siguientes:

Tabla XII. **Mensajes de petición**

Mensaje	**Función**
REGISTER	Permite al agente de usuario registrarse en el servidor de registro.
INVITE	Sirve para iniciar el proceso de llamada, también denominada sesión, por VoIP. En su contenido se debe incluir el protocolo SDP para indicar el o los métodos de comunicación.

Continuación tabla XII.

CANCEL	Sirve para cancelar el proceso de intento de comunicación de una llamada.
BYE	Una vez iniciada la llamada permite colgar la llamada.
NOTIFY	Permite al servidor de registro identificar que agentes de usuario se encuentran activos.
OPTIONS	Permite al servidor de registro determinar las funciones que los agentes de usuario poseen, por ejemplo transferir, realizar llamadas y registrarse.
MESSAGE	Se utiliza para enviar mensajes de texto a través de VoIP.
ACK	Sirve para indicar que se ha recibido un mensaje de finalización de un proceso, como cuando se finaliza el proceso de intento de comunicación de una llamada, ya sea porque esta fue respondida o fue cancelada.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Tabla XIII. **Mensajes de respuesta**

Mensaje	**Función**
100 Trying	En el proceso de llamada, indica al UAC que se ha recibido el mensaje INVITE para iniciar la llamada y que existe un proceso interno en el UAS, como la carga de la base de datos, para establecer la comunicación.
180 Ringing	Esto indica que el UAS se encuentra alertando al individuo o bot que responderá la llamada.
200 OK	Indica que la petición ha sido procesada correctamente.
400 Bad Request	Significa que existe un problema de sintaxis en la petición.
401 Unauthorized	Le dice al UAC que se requiere de una autenticación del usuario.
403 Forbidden	Es utilizado para declarar que la petición es comprendida, pero no es permitida.
407 Proxy Authentication Required	Indica que el agente debe autenticarse en el servidor proxy para luego registrarse en el mismo.
487 Request Terminated	Declara que el proceso se ha terminado por una petición de CANCEL o BYE.

Fuente: elaboración propia, empleando Libreoffice v6.1.

3.2.4.3.5. Proceso de registro

Antes de poder efectuar una llamada por VoIP entre el SMTP Appliance y el equipo de destino, estos deben registrarse dentro de la central telefónica FreePBX. Para tal proceso solo se requiere la utilización del protocolo SIP.

El proceso se describe de la siguiente manera:

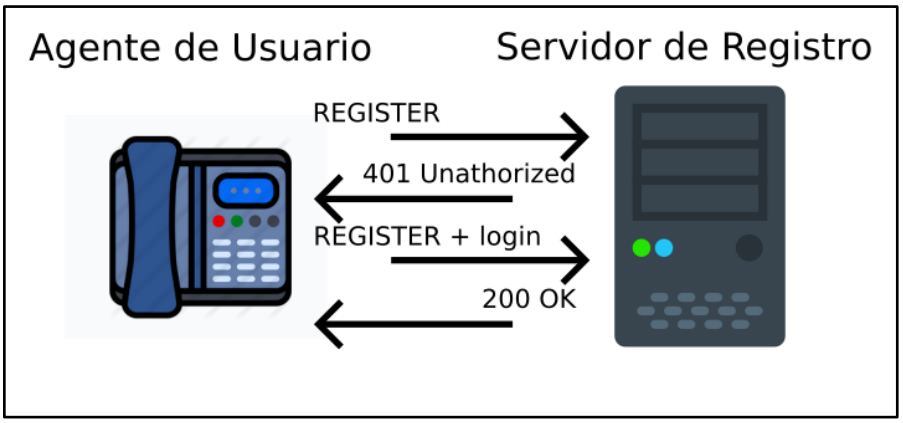
- El agente de usuario, ya sea el SMTP Appliance o el equipo de destino, realiza una petición SIP de 'REGISTER' al servidor de registro FreePBX.

- El servidor de registro responde con un mensaje de '401 Unathorized' solicitando credenciales al agente e incluye el algoritmo de autenticación a utilizar en su respuesta.

- El agente de usuario responde con otra petición de 'REGISTER' en la cual incluye las credenciales con el algoritmo de autenticación indicado.

- El servidor de registro responde con un '200 OK', en caso de que las credenciales sean aceptadas, o con otro '401 Unathorized' en caso dichas credenciales no lo sean.

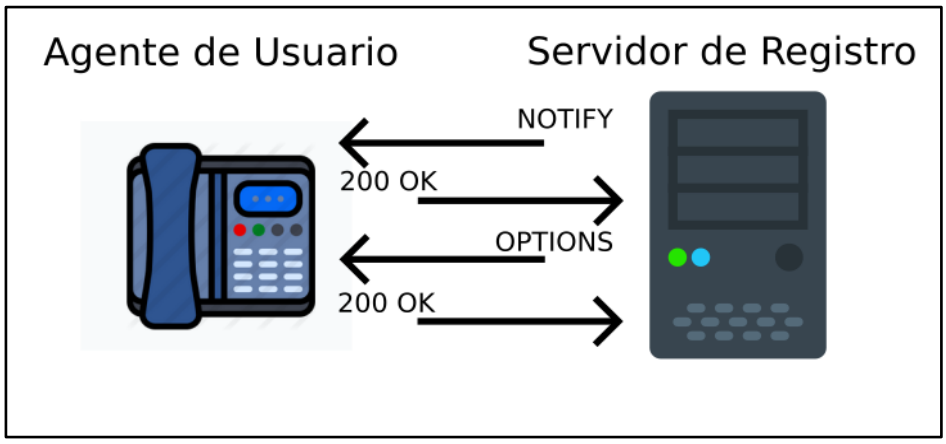
Figura 11. **Proceso de registro**



Fuente: elaboración propia, empleando Inkscape v0.92.4.

Después de que el registro se haya procesado exitosamente, el servidor de registro validará constantemente que el agente de usuario esté activo con las peticiones de OPTIONS y NOTIFY, en las cuales el agente debe responder un mensaje de ‘200 OK’ para validar que se encuentra activo.

Figura 12. **Validación de agente de usuario**



Fuente: elaboración propia, empleando Inkscape v0.92.4.

Es de tomar en cuenta que la comunicación entre FreePBX y el equipo de destino se realiza mediante la conexión de ambos a una VPN pública utilizando el protocolo OpenVPN.

3.2.4.3.6. Proceso de llamada por VoIP

En este proceso pueden intervenir los siguientes grupos de entidades SIP:

- Un agente de usuario cliente y un agente de usuario servidor.
- Un agente de usuario cliente, un agente de usuario servidor y un servidor proxy.
- Un agente de usuario cliente, un agente de usuario servidor y un servidor redirector.

La ventaja de contar con un servidor proxy o un servidor redirector es que hace posible la ubicación de N cantidad de agentes de usuario para realizar las llamadas, de lo contrario se requeriría una base de datos interna dentro de cada agente de usuario para hacer su comunicación posible o en todo caso utilizar un servidor localizador.

De los grupos descritos, el utilizado es el segundo grupo en el cual el agente de usuario cliente es el servidor SMTP Appliance, el agente de usuario servidor es el equipo de destino y el servidor proxy es el equipo FreePBX. También es de notar que durante este proceso intervienen todos los protocolos de VoIP previamente descritos.

El procedimiento utilizado para efectuar una llamada por VoIP es:

- El UAC envía una petición SIP de ‘INVITE’, incluyendo su propio SDP, al servidor proxy. El servidor proxy responde con un ‘401 Unathorized’, en caso la petición de ‘INVITE’ no posea credenciales por lo será necesario reenviar la petición con las credenciales correspondientes, o con un ‘100 Trying’, indicando se está tratando de contactar con el UAS y que la petición si posee credenciales.

- Una vez aceptada la petición de ‘INVITE’, el servidor proxy hace la misma petición de ‘INVITE’, con su propio SDP, hacia el UAS el cual responde con un ‘100 Trying’ indicando que existe un proceso interno dentro del UAS antes de empezar a alertar al destinatario.

- El UAS envía una respuesta de ‘180 Ringing’ al proxy indicando que se encuentra alertando al destinatario, es decir, el teléfono VoIP se encuentra sonando, luego el proxy transmite la misma respuesta hacia el UAC.

- Este proceso puede ser repetir por cada vez que se encuentre alertando al usuario hasta que conteste la llamada, o sea cancelada por parte de cualquier agente por medio de la petición ‘CANCEL’, siendo respondida con un ‘200 OK’, seguido por un ‘487 Request Terminated’, indicando la finalización del proceso. El mensaje de respuesta ‘487 Request Terminated’ es aceptado por parte su destinatario con un ‘ACK’.

- Una vez el usuario en la otra línea contesta la llamada, el UAS responde con un ‘200 OK’ en el cual incluye su propio SDP. Tal mensaje es replicado por el proxy hacia el UAC, solo que en el mensaje enviado se envía el SDP del proxy. El UAC responde con un ‘ACK’ indicando que se ha aceptado la conexión, siendo esta respuesta replicada por el proxy hacia el UAS.

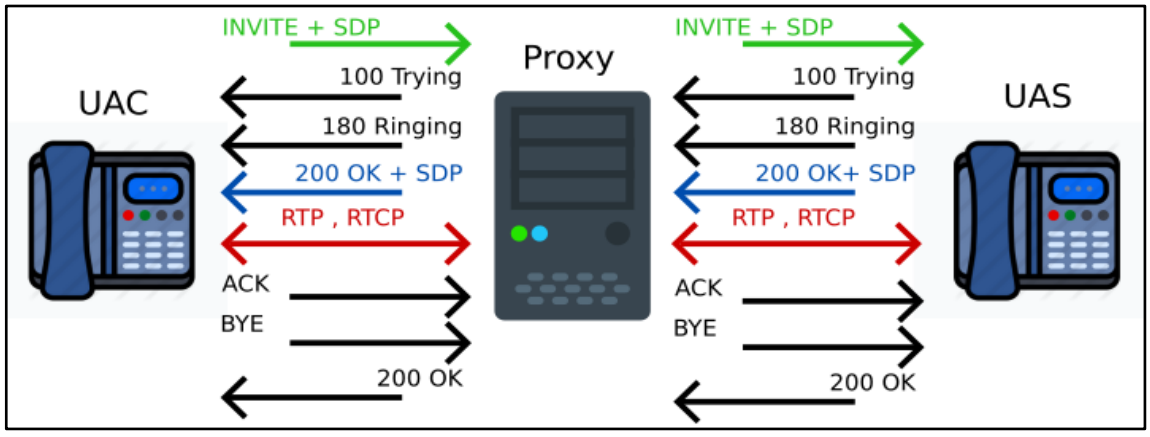
- Iniciada la sesión, se empiezan a transmitir datos de audio, por medio del protocolo RTP, y datos de control de estos paquetes, por medio del protocolo RTCP entre agentes de usuario. El servidor proxy, de ser necesario, recodifica los datos de audio entrantes a su propia codificación estándar que este maneja, después los transmite al destinatario correspondiente.

- En este punto es donde el SMTP Appliance informa sobre la alerta al equipo de destino.

- Para la finalización de la llamada, cualquier agente envía una petición de 'BYE', la cual debe ser respondida con un '200 OK' indicando la finalización del proceso.

- Para esta circunstancia, el *appliance* está configurado de tal forma que, una vez finalice el envío de la alerta por paquetes RTP, espere alrededor de 2 segundos para enviar la petición de 'BYE'.

Figura 13. **Proceso de llamada VoIP**



Fuente: elaboración propia, empleando Inkscape v0.92.4.

3.2.4.4. Módulo Mail

El procedimiento de reenvió de correo por parte de este módulo es:

- Establece una conexión con el servidor de correos de Gmail a través del puerto 587, siendo el *firewall* Nethserver el puente entre el *appliance* y el servidor Gmail.

- Modifica la conexión para que esta sea cifrada por medio de TLS.

- Inicia sesión con las credenciales de correo y contraseña asignados al servidor.

- Envía el correo correspondiente con la información recibida del procesador.

4. ANÁLISIS DEL SISTEMA

4.1. Introducción al análisis del sistema

Este capítulo trata acerca del estudio del *appliance* para que este sea un servidor productivo dentro de una empresa. Los análisis efectuados fueron: financiero, de los factores de calidad vistos en el capítulo 1, de funcionalidad y comparativo con productos similares.

4.2. Análisis financiero

Para una organización de bajo recurso, requiere que el sistema en general sea lo más barato posible y con el mejor rendimiento. Un punto crucial para la empresa es de la selección entre una nube privada, como lo son los servidores internos de la empresa, una nube pública, por ejemplo es AWS, Azure y Google Cloud, y una nube híbrida, que es la combinación de las dos anteriores.

Un cuadro comparativo los tipos de nube es el siguiente:

Tabla XIV. **Nube privada vs nube pública vs nube híbrida**

Nube privada	**Nube pública**	**Nube híbrida**
Toda la información manejada está protegida detrás de un *firewall*. Mantenimiento requerido por parte de la compañía. Altos niveles de control y seguridad. Escalabilidad limitada. La empresa depende de sus recursos disponibles para poder suplir sus necesidades. Alto costo. Mucho del costo se basa en mantenimiento y compra de equipo caro.	Bajo costo en comparación a lo generalmente utilizado en las nubes privadas. El mantenimiento no es requerido. Escalabilidad ilimitada. Se puede utilizar cualquier cantidad de recursos. Alta disponibilidad. Se utilizan varios de servidores a través de red que garantizan el control contra fallos en hardware.	Permite tener el control de la información privada de la empresa. Permite utilizar los recursos de la nube pública cuando sea necesario. Permite establecer un coste de recursos necesarios de forma efectiva. Facilidad de migración entre nubes.

Fuente: elaboración propia, empleando Libreoffice v6.1.

En los precios obtenidos siguientes, se tomó como moneda estándar el dólar estadounidense, dado que es la moneda que manejan las compañías internacionales y en la mayoría de elementos se pueden indicar precios no sujetos a tasa de cambio.

De parte de las nubes, los precios por máquina virtual varían por el tipo de sistema operativo, el tipo de instancia a usar, la ubicación física de los servidores, el tiempo de uso, la cantidad de almacenamiento requerido, entre otros.

Las nubes cotizadas fueron: Azure, AWS y Google Cloud. De dichas nubes se obtuvieron los precios más baratos encontrados por nube para la realización del proyecto entero:

Tabla XV. **Precios según nube**

Nube	**VMs**	**Tipo de instancia**	**Disco**	**Ubicación**	**Precio**
AWS	5	1 vCPU, 2 GB RAM	200 GB	Virginia del Norte - US	$ 110,35/mes
Google Cloud	5	1 vCPU, 2 GB RAM	200 GB	Iowa - US	$ 138,60/mes
Azure	5	1 vCPU, 2 GB RAM	200 GB	Este - UK	$ 91,68/mes

Fuente: elaboración propia, empleando Libreoffice v6.1.

Dentro de la nube, se requiere un servicio que sea capaz de satisfacer la necesidad de realizar llamadas tradicionales y enviar SMS a teléfonos locales en Guatemala. Para ello existen las Plataformas de Comunicación como Servicio, CPaaS por sus siglas en inglés.

Las CPaaS cotizados fueron:

Tabla XVI. **Precios de llamadas y SMS por CPaaS**

Plataforma	**Precios para llamadas realizadas**	**Precio por SMS**
Twilio	$ 0,1900/min a celular $ 0,2250/min a línea fija	$ 0,0498
Plivo	$ 0,1035/min - $ 0,2185/min, dependiendo de la numeración inicial del teléfono.	$ 0,0380 a Tigo $ 0,0490 a Claro
Nexmo	$ 0,1700 / min a celular $ 0,1400 / min a línea fija	$ 0,0480

Fuente: elaboración propia, empleando Libreoffice v6.1.

Como recomendación de nube pública se deben escoger los servicios en la nube de Azure con Nexmo. Se estima que para satisfacer a organización de cualquier tamaño, se requeriría por lo menos utilizar de 1 000 minutos y 3 000 SMS al mes, cantidades que pueden variar en base a la cantidad de servidores que esta posea, por lo que se estima que el costo total mensual por el proyecto en una nube pública sería de $ 405,68.

Para poder montar dicho entorno virtualizado en un servidor físico, nube privada, se pueden utilizar servidores de las marcas HP, DELL, Fujitsu o cualquier otra. Para ello se cotizaron servidores de las marcas mencionadas que sean capaces de satisfacer las necesidades requeridas de hardware.

De los servidores cotizados se escogieron los siguientes por ofrecer un precio barato y buen rendimiento:

Tabla XVII. **Precios por servidor físico**

Marca	**Servidor**	**RAM**	**Disco**	**CPUs**	**Sitio web**	**Precio**
DELL	PowerEdge T30	8 GB	1 TB	4	DELL	$ 619,00
MINIATX	PC MINIATX RYZEN 3 3200	8 GB	240 GB	4	Gesbyte	$ 300,88
Fujistu	PRIMERGY TX1320 M3	8 GB	2 TB	4	Amazon	$ 754,06

Fuente: elaboración propia, empleando Libreoffice v6.1.

Es de notar que la cantidad total de procesadores de estos equipos no satisface la cantidad total necesaria para las máquinas virtuales, la cual sería de 5, sin embargo, dado que el virtualizador Proxmox se encarga de compartir los recursos de manera eficiente, la cantidad de 4 procesadores de cada servidor físico es más que suficiente para suplir la necesidad en términos de procesamiento de estas máquinas.

Para las empresas es de vital importancia que el servidor físico a comprar esté siempre activo y en buen funcionamiento, por lo cual es necesario contar con servicios de mantenimiento que permitan prevenir o solventar cualquier fallo en el equipo en cualquier momento. La siguiente tabla muestra a compañías que ofrecen servicios de mantenimiento quienes, a consideración personal, ofrecen una mejor relación costo-beneficio:

Tabla XVIII. **Mantenimiento de servidores**

Empresa	Precio mensual	Paquete de servicios considerado
CODETEC	$ 61,91/servidor	Mantenimiento profesional: • Diagnóstico inicial de la instalación • Asistencia por control remoto • Monitoreo en servidores • Mano de obra en sitio • Actualizaciones del sistema operativo
Mantenimiento digital	$ 109,44/servidor	Mantenimiento platinium: • Soporte por correo, telefónico y remoto. • Labores de mantenimiento, prevención de fallos, revisiones de copias de seguridad durante jornada laboral. • Reparaciones e instalaciones incluidas, con coste $ 0,00 en mano de obra. Piezas facturables aparte. • Ofertas especiales en suministro de piezas. • Visitas técnicas mensuales
Informática Mayes	$ 94,03 por 1 servidor y $ 26,55 por servidor adicional	Mantenimiento informático superior: • Asistencia técnica presencial ilimitada. • Protección incluida de 5 ordenadores + 1 Servidor, ampliable hasta 25 unidades. • Desplazamientos a sitio gratuitos. • Controles preventivos mensuales: copias de seguridad, virus y estabilidad de red. • Facturación mensual, trimestral, semestral o anual, según preferencia del cliente.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Existen otros tipos de paquetes de servicios más baratos u otras organizaciones, las cuales deberían ser evaluadas en base a las necesidades y presupuesto de la empresa con la que se trabaje.

Para realizar llamadas es necesario la compra de un módulo GSM SIM. Los precios por dicho módulo son:

Tabla XIX. **Precios de módulo GSM**

Módulo GSM	**Sitio web**	**Precio**
SIMCOM SIM900	Ebay	$ 14,27
Xia Fly SIM808	Amazon	$ 29,26
SIM800L	Altronics	$ 19,85

Fuente: elaboración propia, empleando Libreoffice v6.1.

Aparte de comprar el módulo GSM, es requerido pagar un servicio de telefonía. Para tal servicio se debe utilizar una tarjeta SIM de una compañía de telefonía local. Los precios más baratos cotizados, a una tasa de cambio de 7,67 quetzales por dólar (Banco de Guatemala, agosto de 2019), de las compañías locales en Guatemala son los siguientes:

Tabla XX. **Tarifas de llamadas y SMS**

Compañía	**Plan**	**Tipo de plan**	**Precio mensual**
Tigo	8 000 minutos a Tigo, 3 000 minutos a otras compañías y 10 000 SMS	Postpago sin contrato	$ 29,34
Claro	Llamadas ilimitadas y SMS ilimitados	Postpago	$ 38,99

Fuente: elaboración propia, empleando Libreoffice v6.1.

Las mejores opciones de compra de nube privada para satisfacer los requisitos de cualquier compañía serían: el servidor PC MINIATX RYZEN 3 3200, el servicio de Informática Mayes, el módulo GSM SIMCOM SIM900 y la compañía Claro. El precio inicial de la nube privada sería de $ 394,91 con una tarifa mensual de $ 113,02.

Como alternativa se plantea la utilización de una nube híbrida en el que el único servidor físico sea el servidor SMTP Appliance. Dicho servidor puede ser montado en una Raspberry Pi.

Los precios cotizados de esta Raspberry son:

Tabla XXI. **Precios de Raspberry Pi**

Tipo de Raspberry	Sitio web	Precio
Raspberry Pi 3 Modelo A+	Ebay	$ 28,53
Raspberry Pi 3 Modelo B+	Amazon	$ 60,99

Fuente: Elaboración propia, empleando Libreoffice v6.1.

Como selección de nube híbrida se utilizarían los siguientes componentes: la Raspberry Pi 3 Modelo A+, Azure para 4 máquinas virtuales, el módulo GSM SIMCOM SIM900 y la compañía Claro. Azure para 4 máquinas virtuales posee un precio de $ 74,2 mensuales, lo que agregado al pago mensual de la compañía Claro, el precio mensual por la nube híbrida sería de $ 113,51. Como precio inicial de la nube híbrida sería de $ 42,80 que incluiría la Raspberry Pi con el módulo GSM.

La siguiente tabla posee un comparativo de precios entre los tres tipos de nubes cotizadas:

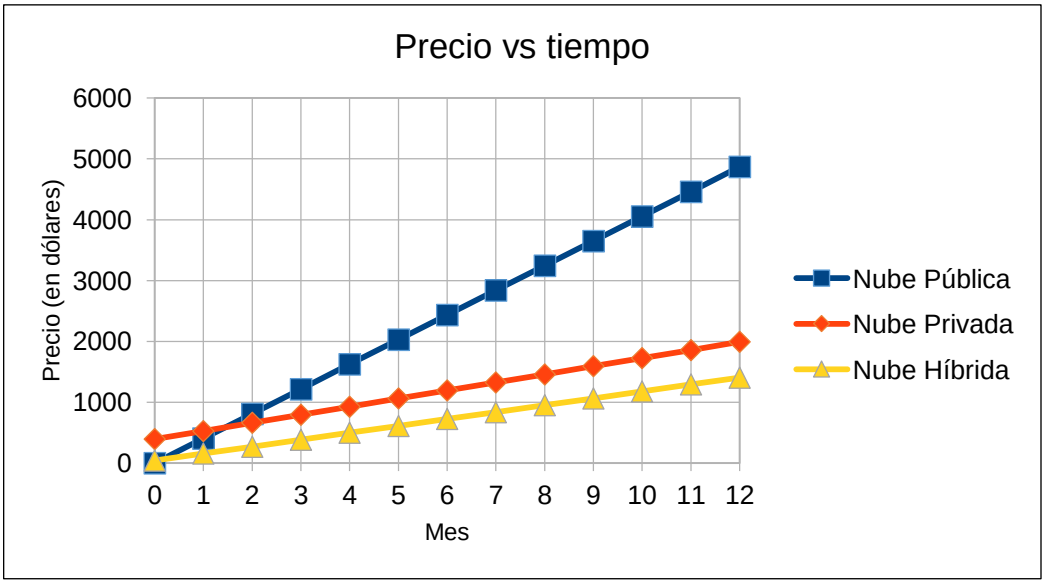
Tabla XXII. **Costos de las nubes**

Tipo de nube	**Componentes**	**Costo inicial**	**Costo mensual**
Pública	Azure, con 5 máquinas virtuales, y Nexmo con 1 000 minutos con 3 000 SMS	$ 0,00	$ 405,68
Privada	PC MINIATX, Informática Mayer, módulo GSM SIMCOM SIM900 y Claro	$ 394,91	$ 133,02
Híbrida	Raspberry Pi 3 Modelo A+, Azure, con 4 máquinas virtuales, el módulo GSM SIMCOM SIM900 y Claro	$ 42,80	$ 113,51

Fuente: elaboración propia, empleando Libreoffice v6.1.

Una proyección anual de los servicios cotizados es:

Figura 14. **Proyección de costo anual según tipo de nube**



Fuente: elaboración propia, empleando Libreoffice v6.1.

Se puede notar que las mejores opciones son la nube privada y la nube híbrida, siendo la nube híbrida la que posee la proyección anual más barata. Lo que eleva el precio enormemente en la nube pública es el requerimiento de un CPaaS, debido a que sus servicios son costosos.

4.3. Análisis por factores de calidad

En esta sección se mostraran los resultados obtenidos del análisis de los factores de calidad mencionados en la sección 1.1.1.

4.3.1. Tiempo de generación y envío

Para el análisis de este factor de calidad se requiere estudiar cada módulo del sistema de generación de alertas de manera individual. Los tiempos a evaluar inician desde el proceso de recepción del mensaje de alerta hasta la notificación de este mensaje a los usuarios correspondientes. Hay que notar que dicho tiempo es dependiente de la longitud del mensaje, por lo que se valuó los tiempos a las longitudes de 20, 50, 100 y 200 caracteres. Los tiempos obtenidos fueron:

Tabla XXIII. **Tiempos de generación y envío de alertas**

Módulo	Tiempo a los 20 caracteres	Tiempo a los 50 caracteres	Tiempo a los 100 caracteres	Tiempo a los 200 caracteres
SIM	2,37 s	2,48 s	3,13 s	3,45 s
SMS	1,60 s	1,60 s	1,60 s	1,60 s
VoIP	2,37 s	2,48 s	3,13 s	3,45 s
Correo	2,30 s	2,30 s	2,30 s	2,30 s

Fuente: elaboración propia, empleando Libreoffice v6.1.

Se puede observar que tanto para SMS como para correo, la longitud del mensaje puede que interfiera pero a escalas mínimas casi imperceptibles por lo que el efecto de longitud de mensaje es casi nulo.

Se puede notar que los tiempos en los módulos SIM y VoIP son iguales, esto es debido a que utilizan el mismo módulo de audio. También es de destacar

que, al almacenar los textos ya procesados, este parámetro se mantiene constante a menos de 1 ms.

4.3.2. Latencia en respuesta

La latencia en respuesta es dependiente de las actividades que realicen los destinatarios, así como la reacción que tienen ellos hacia las ciertas formas en las cuales les es notificada la alerta. Para medir el tiempo en respuesta promedio, es conveniente realizar varias pruebas con distintos usuarios en la cual se generen alertas de forma aleatoria durante el día. Para este evento se ha escogido realizar la prueba de alertas aleatorias durante 1 semana. En la prueba existieron dos individuos, los cuales son:

- Usuario de destino: Es el encargado de recibir la alerta.
- Agente de control: Es el encargado de generar alertas de forma aleatoria para el usuario de destino. Una segunda función del agente es medir el tiempo transcurrido desde que le llega la notificación al usuario de destino hasta que este lee el mensaje de alerta, para alertas por texto, o atiende la llamada, para sistemas de voz.

Para alertas por texto, el usuario de destino recurrirá a informarle al agente de control la hora en la cual él recibió el mensaje de alerta, por lo que dará resultados más precisos en la medición de la alerta. Las alertas por llamada son más sencillas de medir para el agente de control porque el sistema de generación utilizado permite determinar el tiempo exacto en el que el usuario de destino atiende la llamada.

Por módulo se realizaron 30 pruebas en total. De dichas pruebas se obtuvieron los siguientes resultados:

Tabla XXIV. **Latencia en respuesta promedio por módulo**

Módulo	Latencia promedio
SIM	0 min 10 s
SMS	4 min 12 s
VoIP	0 min 13 s
Correo	13 min 17 s

Fuente: elaboración propia, empleando Libreoffice v6.1.

Es de tomar en cuenta que ante cualquier evento aleatorio es más sencillo responder ante alertas sonoras.

4.3.3. Calidad de llamada

Para la medición de calidad de llamada para sistemas de generación de alertas por voz existen diversos factores. Por criterio personal se estima que los factores que determinan la calidad de una llamada son los siguientes:

- Continuidad del mensaje
- Claridad de la voz
- Cantidad de ruido en la línea
- Comprensión del mensaje por el destinatario

Estos parámetros están sujetos a la subjetividad del destinatario, por lo que es una evaluación cualitativa. Esta evaluación se medirá entre el rango del 1 al 10, siendo 10 la nota máxima alcanzada. Las notas obtenidas fueron:

Tabla XXV. **Medición de calidad de llamada**

Módulo	Continuidad del mensaje	Claridad de voz	Ruido en la línea	Comprensión del mensaje
SIM	10	9	1	9
VoIP	8	8	3	7

Fuente: elaboración propia, empleando Libreoffice v6.1.

Se puede estimar que la mejor línea para la transmisión de datos de audio es en este caso la del módulo SIM.

4.3.4. Escalabilidad

El objetivo de realizar un sistema de generación de alertas escalable es que sea capaz de integrarse a varios equipos de monitoreo. Por lo general los sistemas de monitoreo conocidos son capaces de enviar alertas por correo, por lo que este equipo se puede acoplar perfectamente a dichos sistemas.

Se tiene contemplado que se pueda acoplar a otros protocolos para recibir información de varias fuentes, para realizar un sistema más escalable.

Otra forma de medir la escalabilidad que tiene un sistema de generación de alertas, es por medio de la cantidad de usuarios a los cuales le puede enviar dicha alerta. La cantidad de usuarios a quienes se les puede enviar la alerta es dependiente de la cantidad de espacio de almacenamiento que se tiene, así como la cantidad de memoria RAM. Este parámetro no es posible medirse ya que se

requiere de una cantidad considerable de usuarios a los que se les pueda enviar la alerta y no se dispone de tal recurso.

4.3.5. Estabilidad

La estabilidad del sistema se puede medir en base al porcentaje éxito que se tiene en el envío de alertas. Para cada módulo se obtuvieron los siguientes resultados:

Tabla XXVI. **Resultados de pruebas de estabilidad**

Módulo	**Cantidad de pruebas**	**Cantidad de envíos exitosos**	**Cantidad de envíos fallidos**	**Porcentaje de éxito**
SIM	30	30	0	100 %
SMS	30	30	0	100 %
VoIP	30	27	3	90 %
Correo	30	30	0	100 %

Fuente: elaboración propia, empleando Libreoffice v6.1.

Los únicos fallos en el módulo VoIP se dieron debido a la mala conexión a través de VPN, aunque, por lo general, es un sistema bastante estable al igual que el resto de los módulos.

4.3.6. Seguridad

En la siguiente tabla se describe como se implementó un sistema de seguridad en base a cada módulo:

Tabla XXVII. **Aspectos de seguridad**

Módulo	**Aspectos de seguridad**
SIM	GSM es la red más segura de todas las telecomunicaciones disponibles hoy en día, por lo que este módulo es bastante seguro. GSM utiliza encriptación de tipo RAND con algoritmo de autenticación A3.
SMS	Al igual que el módulo SIM, este también se encuentra protegido por la red GSM.
VoIP	Posee varias capas de protección. Una se da por el protocolo SIP que exige proceso de autenticación mediante usuario y contraseña. Otra es que se encuentra protegido mediante conexiones por VPN. La otra es por la implementación del *firewall* tanto del servidor FreePBX, como Nethserver.
Correo	Utiliza algoritmos de autenticación y conexiones cifradas con TLS hacia el servidor de Gmail.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Basado en las pruebas anteriores, se puede indicar que el mejor método para recepción de alertas es por medio de llamada telefónica.

4.4. Pruebas de estrés

Las pruebas de estrés nos permiten determinar cual será el comportamiento del servidor en un entorno productivo. Para esta etapa es necesario probar como es el comportamiento del hardware y el software bajo sobrecargas del servicio.

4.4.1. Pruebas de hardware

En esta sección se realizaran pruebas de estrés de hardware sobre el servidor SMTP Appliance. Se pretende que esta sección sea una guía demostrativa de cómo debe se deben realizar las pruebas de estrés sobre servidores Linux, los parámetros a tomar en cuenta e interpretar de los resultados obtenidos.

Al evaluar un servidor a nivel de hardware es importante ver el rendimiento de este en los siguientes elementos:

- CPU
- Disco
- RAM
- Red

Al realizar pruebas de estrés, se puede determinar la rapidez con la que un servidor puede responder ante una sobrecarga de datos.

Para estas pruebas se utilizarán las herramientas, las cuales pueden ser descargadas del repositorio de Linux con el comando ‘apt’, siguientes:

- *lshw*: imprime la información del hardware.
- *stress-ng*: permite realizar pruebas de estrés sobre CPU, disco, RAM y red
- *nmon*: se utiliza para monitorear el CPU, disco, memoria RAM y red.

4.4.1.1. Estrés de CPU

Los pasos a realizar para ejecutar la prueba de estrés sobre el procesador son los siguientes:

- Obtener la información de procesador con el cual se estará trabajando con el comando ‘lshw’ con la opción ‘-C cpu’:

```
$ sudo lshw - C cpu
  *-cpu
       description: CPU
       product: AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
       vendor: Advanced Micro Devices [AMD]
       physical id: 400
       bus info: cpu@0
       version: pc-i440fx-3.0
       slot: CPU 0
       size: 2 GHz
       capacity: 2GHz
       width: 64 bits
       capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8
       apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht
       syscall nx mmxext fxsr_opt pdpe1gb rdtscp x86-64 rep_good nopl
       extd_apicid pni pclmulqdq ssse3 fma cx16 sse4_1 sse4_2 x2apic
       movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand
```

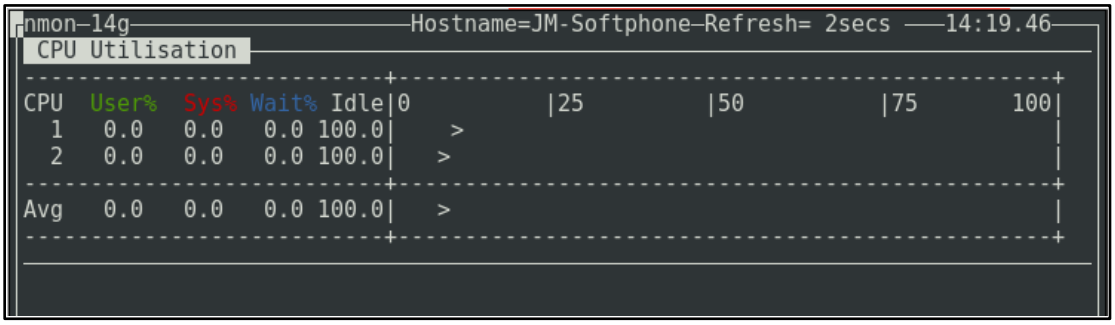
hypervisor lahf_lm cmp_legacy svm cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw perfctr_core ssbd ibpb vmmcall fsgsbase tsc_adjust bmi1 avx2 smep bmi2 rdseed adx smap clflushopt sha_ni xsaveopt xsavec xgetbv1 virt_ssbd arat npt nrip_save
configuration: cores=2 enabledcores=2 threads=1

- Abrir una segunda terminal para ejecutar el comando de monitoreo ‘nmon’.

- Ejecutar el comando ‘nmon’ y se teclea ‘c’ para indica el monitoreo del CPU.

$ nmon

Figura 15. **Monitor del procesador con nmon**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y nmon v16i.

- Ejecutar el comando ‘stress-ng’:

La estructura utilizada para el comando stress-ng fue:

stress-ng --cpu <N> --cpu-ops <M>

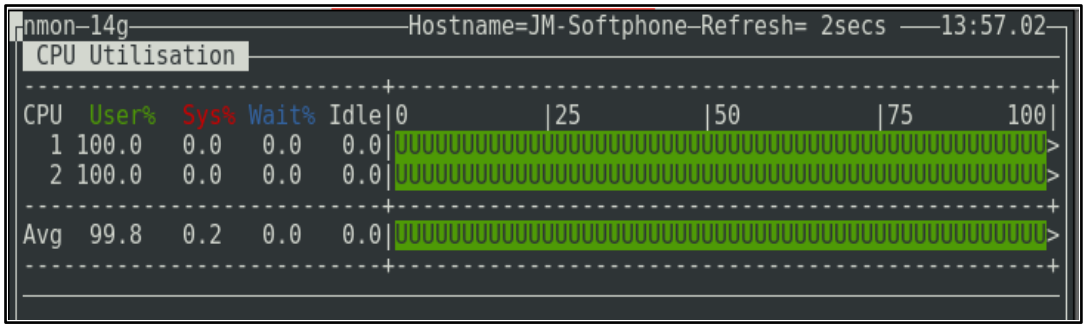
Donde los parámetros son:

- --cpu <N>: Asigna una cantidad N de estresores, o procesos, de CPU.
- --cpu-ops <M>: Indica que se harán una cantidad M de operaciones tipo flotante las cuales acceden directamente al CPU.

Durante la prueba se usaron arbitrariamente 16 estresores y 20 000 operaciones, obteniendo los siguientes resultados:

```
$ sudo stress-ng --cpu 16 --cpu-ops 20000
stress-ng: info:  [21089] defaulting to a 86400 second run per stressor
stress-ng: info:  [21089] dispatching hogs: 16 cpu
stress-ng: info:  [21089] cache allocate: default cache size: 16384K
stress-ng: info:  [21089] successful run completed in 41.02s
```

Figura 16. **Monitoreo de pruebas de estrés del CPU**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6,

En síntesis el procesador utilizado en el servidor es un AMD Ryzen 5 2500U, con velocidad de reloj promedio de 2,0 GHz, de dos núcleos, con arquitectura de 64 bits, el cual puede ejecutar 487,57 instrucciones directas al CPU por segundo, con una capacidad de utilización máxima del 100 %.

4.4.1.2. Estrés de memoria

Los pasos a realizar para ejecutar la prueba de estrés sobre la RAM son:

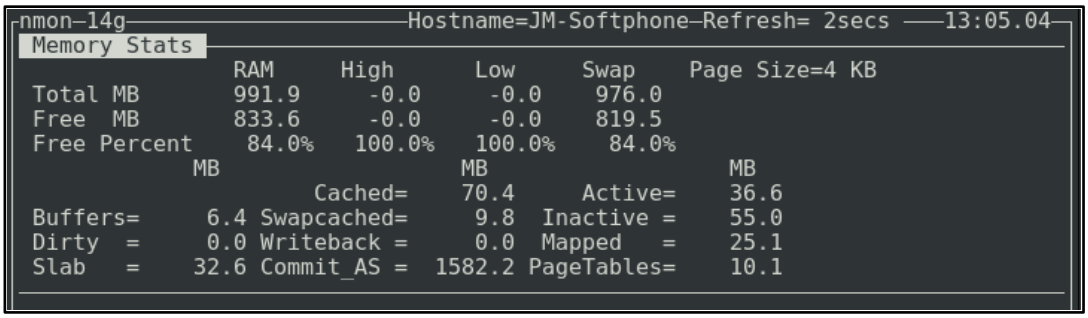
- Desplegar la información de la memoria con el comando ‘lshw’ con la opción ‘-C memory’:

```
$ sudo lshw - C memory
   *-firmware
      description: BIOS
      vendor: SeaBIOS
      physical id: 0
      version: rel-1.11.2-0-gf9626ccb91-prebuilt.qemu-project.org
      date: 04/01/2014
      size: 96KiB
  *-memory
      description: System Memory
      physical id: 1000
      size: 1GiB
      capacity: 1GiB
    *-bank
         description: DIMM RAM
         vendor: QEMU
         physical id: 0
         slot: DIMM 0
         size: 1GiB
```

- Abrir una segunda terminal.

- Ejecutar el comando ‘nmon’ y se teclea ‘m’ para indica el monitoreo de la RAM.

Figura 17. **Monitor de la memoria RAM con nmon**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y nmon v16i.

- Ejecutar el comando ‘stress-ng’:

  La estructura utilizada para el comando stress-ng fue:

  stress-ng --vm <N> --vm-ops <M>

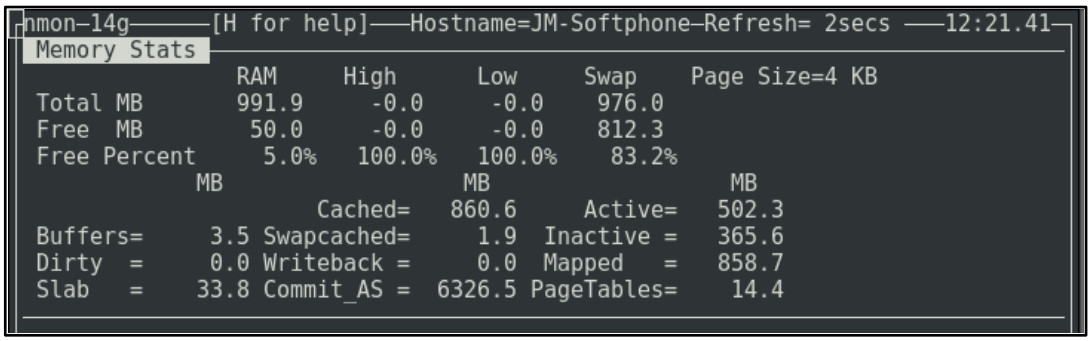
  Donde los parámetros son:

  - --vm <N>: Asigna una cantidad N de estresores que ingresan a la memoria.
  - --vm-ops <M>: Indica que se harán una cantidad M de operaciones de acceso directo a memoria.

  Durante la prueba se usaron arbitrariamente 16 estresores y 40 000 operaciones obteniendo los siguientes resultados:

```
$ stress-ng --vm 16 --vm-ops 40000
stress-ng: info:  [2637] defaulting to a 86400 second run per stressor
stress-ng: info:  [2637] dispatching hogs: 16 vm
stress-ng: info:  [2637] cache allocate: default cache size: 16384K
stress-ng: info:  [2637] successful run completed in 9.88s
```

Figura 18. **Monitoreo de pruebas de estrés de la memoria RAM**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

La información obtenida indica que se utilizó una memoria de acceso dinámico, DIMM, de marca QEMU, que es el emulador de la máquina virtual utilizada, con capacidad de 1 GiB, en la cual se pueden ejecutar 4 048,58 instrucciones directas a memoria por segundo con una capacidad de utilización máxima del 95 %. No se pudo obtener la clase de memoria dinámica debido a que, siendo una máquina virtual, el equipo no brindó dicha información.

4.4.1.3. Estrés de disco

Los pasos a realizar para ejecutar la prueba de estrés sobre el disco son:

- Obtener las características del disco el comando ‘lshw’ con la opción ‘-C disk’:

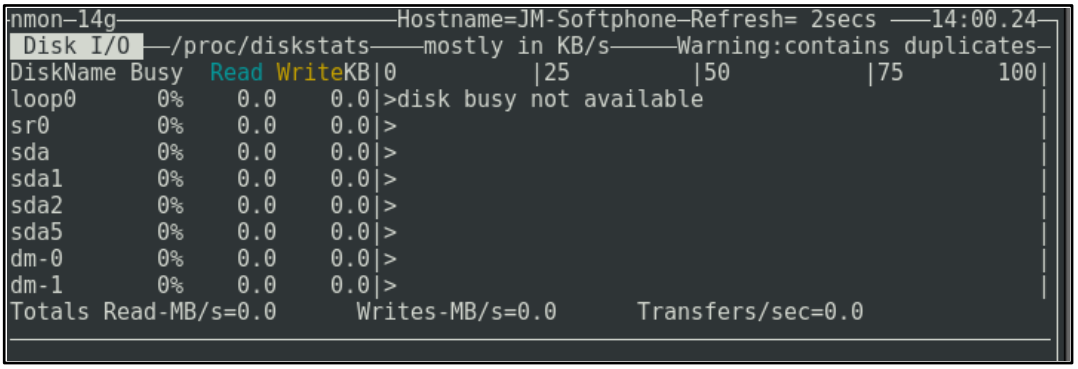
```
$ sudo lshw - C disk
  *-cdrom
       description: DVD reader
       physical id: 0.0.0
       bus info: scsi@1:0.0.0
       logical name: /dev/cdrom
       logical name: /dev/dvd
       logical name: /dev/sr0
       capabilities: audio dvd partitioned partitioned:dos
       configuration: signature=13982071 status=ready
  *-disk
       description: SCSI Disk
       physical id: 0.0.0
       bus info: scsi@2:0.0.0
       logical name: /dev/sda
       size: 40GiB (41GB)
       capabilities: partitioned partitioned:dos
       configuration: logicalsectorsize=512 sectorsize=512
signature=6e8ac7d3
```

- Abrir una segunda terminal.

- Ejecutar el comando ‘nmon’ y se teclea ‘d’ para indica el monitoreo del disco.

```
$ nmon
```

Figura 19. **Monitor del disco con nmon**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y nmon v16i.

- Ejecutar el comando ‘stress-ng’:

La estructura utilizada para el comando stress-ng fue:

stress-ng --hdd <N> --hdd-ops <M>

Donde los parámetros son:

- --hdd <N>: Asigna una cantidad N de estresores que escriben al disco.
- --hdd-ops <M>: Indica que se harán una cantidad M de operaciones de escritura a disco.

Durante la prueba se usaron arbitrariamente 16 estresores y 20 000 operaciones obteniendo los siguientes resultados:

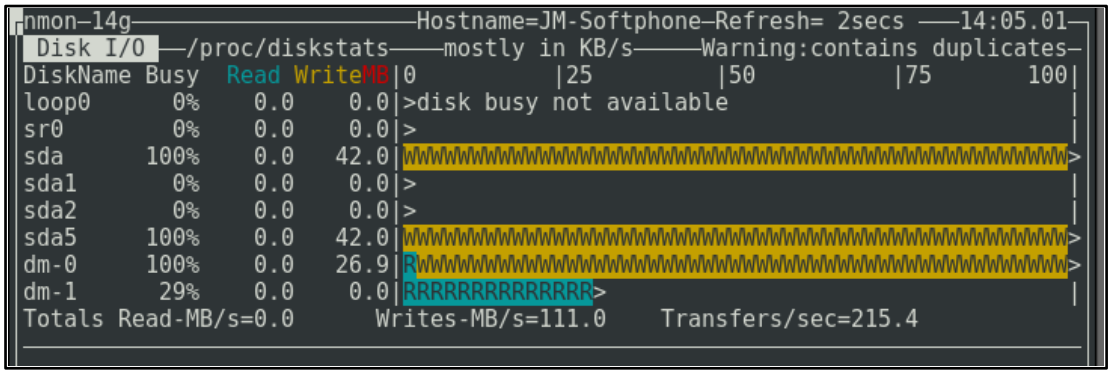
$ stress-ng --hdd 16 --hdd-ops 40000
stress-ng: info: [3171] defaulting to a 86400 second run per stressor

stress-ng: info: [3171] dispatching hogs: 16 hdd

stress-ng: info: [3171] cache allocate: default cache size: 16384K

stress-ng: info: [3171] successful run completed in 58.54s

Figura 20. **Monitoreo de pruebas de estrés del disco**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

La información obtenida indica que se utilizó un disco de tipo SCSI, mapeado como '/dev/sda', con capacidad de 40 GiB, en la cual se pueden ejecutar 683,29 instrucciones de escritura a disco por segundo con una capacidad de estrés máximo del 95 %, una tasa de escritura máxima observada de 204 MB/s.

4.4.1.4. Estrés de tarjeta de red

Los pasos a realizar para ejecutar la prueba de estrés sobre la tarjeta de red fueron:

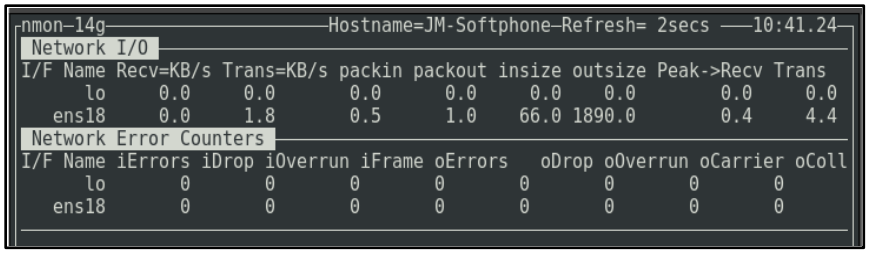
- Mostrar las características del disco el comando 'lshw' con la opción '-C network':

```
$ sudo lshw - C network
  *-network
       description: Ethernet interface
       product: Virtio network device
       vendor: Red Hat, Inc
       physical id: 12
       bus info: pci@0000:00:12.0
       logical name: ens18
       version: 00
       serial: a2:62:26:32:23:6d
       width: 64 bits
       clock: 33MHz
       capabilities: msix bus_master cap_list rom ethernet physical
       configuration: autonegotiation=off broadcast=yes driver=virtio_net
driverversion=1.0.0 ip=10.10.10.8 latency=0 link=yes multicast=yes
       resources: irq:10 ioport:e5e0(size=32) memory:fc454000-fc454fff
memory:fea0c000-fea0ffff memory:fc400000-fc43ffff
```

- Abrir una segunda terminal.

- Ejecutar el comando ‘nmon’ y teclear ‘n’ para monitorear la red.

Figura 21. **Monitor de la tarjeta de red con nmon**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y nmon v16i.

- Ejecutar el comando ‘stress-ng’:

  La estructura utilizada para el comando stress-ng fue:

  stress-ng --udp <N> --udp-ops <M>

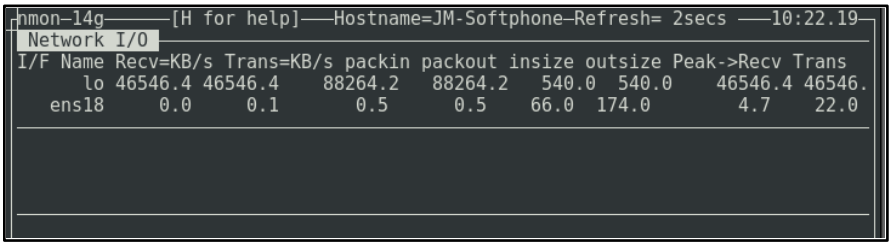
  Donde los parámetros son:

  - --udp <N>: Asigna una cantidad N de estresores que envían y reciben paquetes UDP.
  - --udp-ops <M>: Indica que se harán una cantidad M de operaciones de envío de paquetes de tipo UDP.

  Durante la prueba se usaron arbitrariamente 16 estresores y 500 000 operaciones de obteniendo los siguientes resultados:

  $ stress-ng --udp 16 --udp-ops 500000
  stress-ng: info: [2541] defaulting to a 86400 second run per stressor
  stress-ng: info: [2541] dispatching hogs: 16 udp
  stress-ng: info: [2541] cache allocate: default cache size: 16384K
  stress-ng: info: [2541] successful run completed in 5.24s

Figura 22. **Monitoreo de pruebas de estrés de la tarjeta de red**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

La información obtenida indica que se utilizó una tarjeta de red de tipo VirtIO, mapeado como ‘ens18’, que opera de 33 MHz, la cual se puede ejecutar 95 419,85 instrucciones de escritura / lectura de paquetes UDP por segundo con tasas de transmisión y recepción máximas de 45,46 MB/s.

4.4.2. Pruebas de software

Es necesario probar que el servidor SMTP Appliance funciona de manera adecuada ante cualquier sobrecarga de mensajes de tipo SMTP. Para ello se ejecutará una prueba de estrés de software con la herramienta ‘postal’, descargada de los repositorios con el comando ‘apt’, instalada en una máquina virtual de pruebas Ubuntu Server 18.04, seleccionado arbitrariamente, quien sobrecargará de mensajes de correo aleatorios al servidor SMTP Appliance.

La estructura utilizada de postal fue:

postal <servidor_SMTP> <archivo_de_usuarios>

Donde:

- <servidor SMTP>: es la dirección IP asignada al servidor.
- <archivo de usuarios>: es el conjunto de usuarios que manejará la herramienta postal.

Los parámetros a medir durante la prueba serán: carga en el cpu, carga en memoria, cantidad de mensajes recibidos y tiempo de duración.

Durante la prueba solo se las secciones del receptor y procesador, pues del emisor ya se poseen las medidas de latencia correspondientes.

Los pasos a seguir para esta prueba son:

- Se abre dos terminales del servidor SMTP Appliance. Una para correr el servidor y otra para monitorear el rendimiento del equipo.

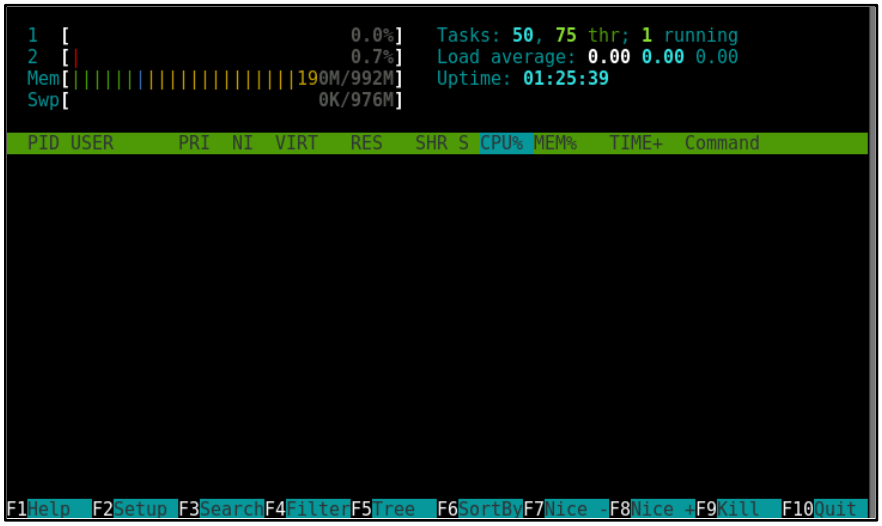
- Abrir una terminal del equipo de prueba.

- Ejecutar el comando ‘htop’ en una terminal del servidor SMTP Appliance.

  $ htop

- Presionar tecla ‘F4’, teclear la palabra ‘python’ e introducir la tecla ‘*Enter*’ para obtener los parámetros de CPU y memoria únicamente del servidor SMTP Appliance.

Figura 23. **Monitoreo del servidor SMTP Appliance – antes del inicio**



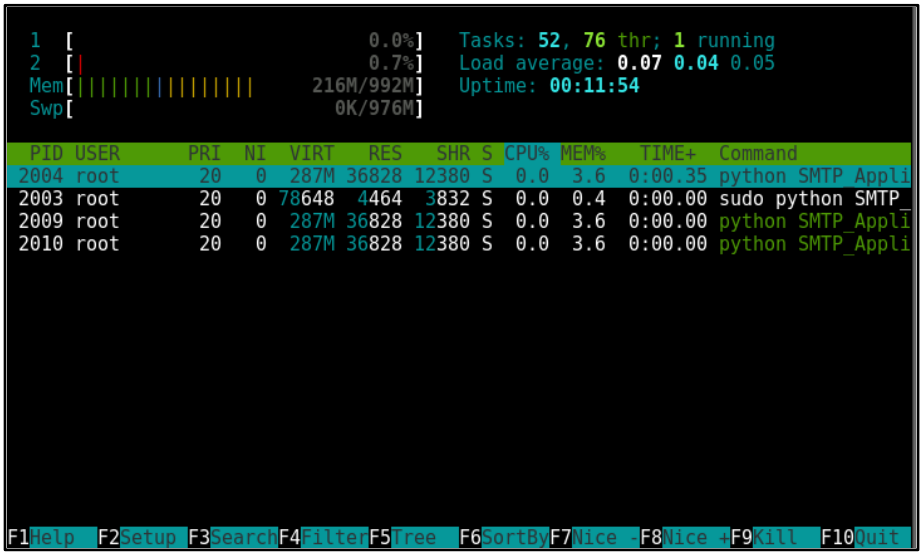
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y htop v2.2.

Se observa que inicialmente se tienen 190 MB cargados a memoria, con un uso del procesador del 0,7 %.

- Desde otra terminal, se corre el servidor SMTP Appliance.

```
$ cd Softphone/code
$ sudo python
>> from SMTP_Appliance import SMTP_Softphone
>> server = SMTP_Softphone()
```

Figura 24. **Monitoreo del servidor SMTP Appliance – estado inicial**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y htop v2.2.

Se puede observar que en total el servidor carga 26 MB a memoria en su estado inicial. El porcentaje de uso del procesador no se ve afectado por el inicio del servidor.

- Crear un archivo de con correos de los usuarios asignados dentro del servidor SMTP Appliance en postal. Se puede utilizar el comando ‘echo’ o ‘nano’.

  $ echo ‘fer@patitos.org’ > usuarios.txt

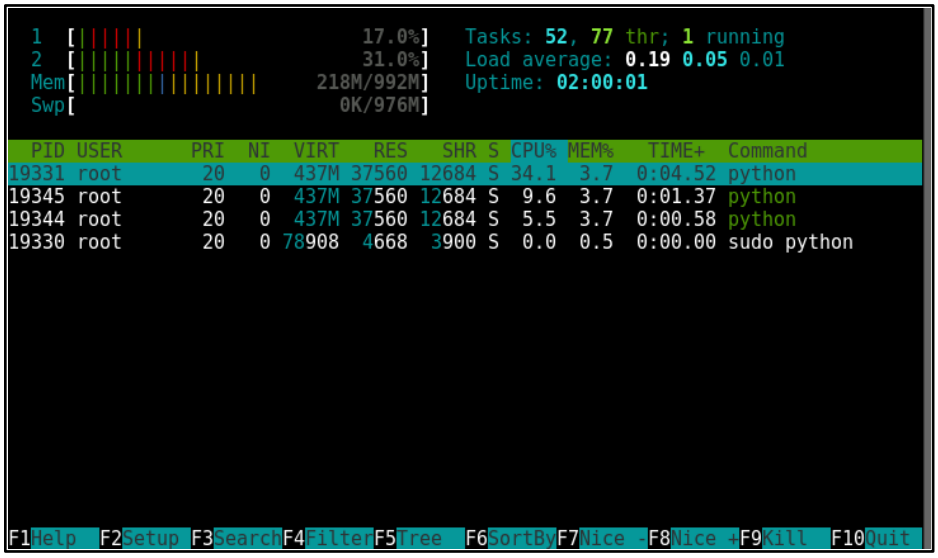
- Ejecutar el comando ‘uptime’ para obtener el tiempo de inicio.

  $ uptime
  14:01:13 up 1:41, 1 user, load average: 0.00, 0.00, 0.00

- Ejecutar el comando postal

  $ postal ip_servidor usuarios.txt

Figura 25. **Monitoreo del servidor SMTP Appliance – durante prueba**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6 y htop v2.2.

Durante la prueba se observa que el servidor aumento hasta llegar a un porcentaje de utilización del CPU del 34,1 % máximo. En RAM solo se incrementó en 2 MB.

- Ejecutar nuevamente el comando uptime.

```
$ uptime
14:01:32 up  1:41,  1 user,  load average: 0.00, 0.00, 0.00
```

La cantidad de mensajes recibidos se obtuvo del servidor SMTP Appliance la cual fue de 915 mensajes aleatorios.

En síntesis, se puede indicar que tanto el receptor como el procesador sel servidor SMTP Appliance son capaces de procesar alrededor de 101,67 mensajes por segundo, aumentando el uso del procesador en un 33,4 % e incrementado el uso de memoria en 28 MB.

4.5. Productos similares

De los sistemas investigados únicamente se encontró el Sensaphone IMS-4000 como una solución similar de este tipo a nivel empresarial. Un análisis comparativo entre las características de este sistema con el SMTP Appliance se detalla a continuación:

Tabla XXVIII. **Sensaphone IMS-4000 vs SMTP Appliance**

Sensaphone IMS-4000	**SMTP Appliance**
Capaz de monitorear hasta 8 sensores de ambiente, como lo son humedad, temperatura, humo, nivel de sonido, intrusión, movimiento, fugas de agua y cortes de energía. Dichos sensores solo son provenientes de la misma empresa.	Integrable con cualquier equipo de monitoreo ya existente.
Puede generar alertas por llamada tradicional, VoIP, SMS y correo.	Se puede utilizar como un servidor físico, virtual o como *container*.
Posee alertas audibles.	Cantidad ilimitada de equipos que puede monitorear.
Puede apagar o encender equipos en base a las alarmas recibidas.	Puede generar alertas por llamada tradicional, VoIP, SMS y correo.
Puede monitorear hasta 64 equipos por medio de IP.	No requiere de interfaz gráfica para su configuración.
Puede conectarse con 31 equipos similares para extender la cantidad de equipos que monitorea.	Código fuente editable.
El control del equipo requiere una interfaz gráfica para su configuración.	Disponible para cualquier entusiasta en tener un sistema similar.
Disponible solo para empresas.	Alternativa *open source*.

Fuente: elaboración propia, empleando Libreoffice v6.1.

No se pudo obtener el costo del Sensaphone IMS-4000 debido a que solo se puede cotizar este equipo a compañías.

Figura 26. **Sensaphone IMS-4000**

Fuente: Sensaphone. *Sensaphone IMS-4000 Enterprise Monitoring Host Unit*. https://www.sensaphone.com/products/sensaphone-ims-4000-enterprise-monitoring-host-unit. Consulta: 3 de noviembre de 2019.

Escoger un sistema u otro se deja a discreción de la empresa que desee contar con un generador de alertas.

5. FUTURAS IMPLEMENTACIONES

5.1. Introducción a futuras implementaciones

Las futuras implementaciones a realizar en el servidor SMTP Appliance serían las siguientes:

- Balanceo de cargas y prevención de fallos con Kubernetes
- Implementación de protocolos SRTP y SRTCP
- Uso de encriptación por TLS en la recepción de mensajes SMTP
- Utilización del *framework* twisted
- Creación de interfaz de administración web con Django
- Módulo para envío de alertas por Whatsapp
- Obtención de estadísticas de las alertas
- Monitoreo de sensores
- Realización de sistema bidireccional

5.2. Kubernetes

Kubernetes es un sistema de código libre para la orquestación de *containers*.

Un *container* es en sí una instancia virtual del sistema operativo que permite segmentar tareas específicas sobre el sistema operativo original, asignándoles la cantidad necesaria de RAM, disco y capacidad de procesamiento requerido para realizar su labor. Los *containers* generalmente son utilizados para aplicaciones que son de vital importancia.

Al hablar de ‘orquestación de *containers*’, se hace referencia a cómo interactúan los *containers* unos con otros. Una de las principales características de la orquestación de *containers* es que se puede tener diversas instancias de un *container* que se muestren al usuario como solo una, es decir que una aplicación puede ser ejecutada en diversos *hosts* simultáneamente, mostrando al usuario como si solo se ejecutará una sola aplicación. Esta ventaja permite tener aplicaciones redundantes, balanceo de cargas, tolerancia a fallos, entre otros.

5.3. SRTP y SRTCP

Dichos protocolos le añaden las capas de encriptación, autenticación, integridad, respuesta contra ataques, entre otras características, a los protocolos RTP y RTCP.

Tanto SRTP como SRTCP fueron elaborados por un pequeño grupo de criptógrafos expertos de Cisco y Sony Ericsson.

Al implementar estos protocolos, permite un gran incremento en la seguridad de VoIP.

Generalmente el uso de estos protocolos es opcional, por lo que se pretende que se pueden habilitar o deshabilitar según el operador del servidor SMTP Appliance lo requiera.

5.4. TLS en la recepción de mensajes SMTP

TLS es un protocolo criptográfico que proporciona una comunicación segura a través de una red. TLS se puede implementar en los protocolos de aplicación

HTTP el cual se convierte HTTPS, SMTP el cual se convierte en SMTPS, IMAP para formar IMAPS, entre otros protocolos de aplicación. Así mismo, se puede implementar en el protocolo de transporte TCP.

Actualmente el receptor del *appliance* no posee ninguna capa de protección al momento de recibir mensajes de SMTP, por lo que al agregarle la capa de encriptación por TLS a la recepción del mensaje, generaría un incremento en seguridad en la recepción de mensajes evadiendo cualquier ataque de la red interna.

5.5. Twisted

Twisted es un *framework* de Python el cual se utiliza para el desarrollo de comunicaciones a través de red. Twisted fue creado por el Instituto Tecnológico de Massachusetts. Este *framework* está diseñado para el montaje de servidores para transmisión de audio y video a nivel productivo.

Se realizó varias experimentaciones con la librería twisted, sobretodo en el envío de paquetes RTP. Al evaluar los parámetros obtenidos en los paquetes RTCP y compararlos con los de la librería 'socket', se concluyó que con twisted se obtiene un *jitter* de cero, mientras que con la librería 'socket' el *jitter* es muy variable obteniendo un máximo de 35, lo que indica que con twisted hay un incremento sustancial en la calidad de las llamadas.

5.6. Django

Django es otro *framework open source* de Python diseñado para el desarrollo de sitios web. Existen diversos sitios web productivos que utilizan Django, entre los cuales se destaca Instagram, The New York Times, Pinterest,

Nasa Science, National Geographic, entre otras (OpenWebinars: 10 Webs famosas que no sabías que usaban Django, 2013.

Se pretende utilizar este *framework* para el desarrollo de una interfaz web capaz de administrar y configurar cada uno de los parámetros del servidor SMTP Appliance.

5.7. WhatsApp

Dado que WhatsApp es uno de los medios de comunicación más utilizados en la actualidad (Next U: Top 10 de las redes sociales más usadas en el mundo, 2020, se pretende crear un módulo de envió de mensajes por WhatsApp con la implementación de la librería de Python 'selenium*'.*

5.8. Estadísticas de alertas

Se almacenarán estadísticas continuamente de las alertas generadas. Dichas estadísticas se pretenden que sean vistas por medio de la interfaz web.

La obtención de estadísticas permitirá a la empresa determinar los puntos de mayor fallo, con lo cual se podrá efectuar un mejor análisis para mitigar los riesgos futuros.

5.9. Sensores

Se piensa integrar un sistema de colección de datos de sensores a conveniencia de la empresa. Para estos sensores se podrán generar alertas las cuales se enviarán a los usuarios correspondientes, así como se generarán gráficas de sus valores en tiempo real.

5.10. Sistema bidireccional

Se pretende realizar un sistema de envío y respuesta, con lo que los individuos interesados en recibir las alertas, puedan realizar acciones inmediatamente sin requerir una visita en sitio.

CONCLUSIONES

1. Un servidor de alertas permite a cualquier organización detectar algún fallo o problema dentro de esta. La función principal de estos es reducir los daños por cualquier falla. Además, son capaces de salvar vidas, como los sistemas de alertas ante catástrofes naturales.

2. El servidor SMTP Appliance debe ubicarse en la etapa siguiente al servidor de monitoreo de la empresa. Para su total funcionalidad, así como seguridad, es requerido que tal servidor cuente con los siguientes elementos: equipos a monitorear, servidor de monitoreo, central telefónica, *firewall*, módulo GSM y equipo de destino.

3. El servidor SMTP Appliance se divide en tres secciones: receptor, procesador y emisor. El receptor recibe el mensaje de alerta, lo pasa por un filtro, luego por una memoria FIFO, después se lo entrega al procesador. El procesador extrae la información de la alerta para adaptarla a cada uno de los módulos del emisor, el cual se compone de los siguientes módulos: SIM, SMS, VoIP y Mail. Finalmente, el emisor hace que la alerta llegue a su destinatario.

4. El análisis financiero indica que la mejor opción de nube es la híbrida. Las pruebas de factores de calidad señalan que el mejor medio de envío de alertas es la llamada telefónica. Las pruebas de estrés muestran que el *appliance* soporta sobrecargas de hardware, procesa 101 mensajes por segundo, consume 28 MB de RAM y usa como máximo el 33,4 % del CPU. El análisis de mercado indica que hay ciertos aspectos del servidor a mejorar para hacerlo más competitivo en el mercado.

Las futuras implementaciones para el servidor SMTP Appliance son: orquestación de *containers* con de Kubernetes, SRTP, SRTCP, SMTP con TLS en el receptor, uso del *framework* de red Twisted, aplicación del *framework* de web Django, envío de alertas por WhatsApp, generación de estadísticas de alertas, monitoreo de sensores y poder efectuar acciones de respuesta mediante un sistema bidireccional.

RECOMENDACIONES

1. El sistema de alerta a usar por una empresa debe ser capaz de informar al destinatario de tal forma que dicho destinatario tenga una rápida respuesta ante el problema, para que se minimice el impacto de la falla en los servicios que esta ofrece.

2. Para sistemas de monitoreo, *firewall* u otros tipos de servidores, es recomendable usar soluciones de tipo *open source* empresarial que, por lo general, son soluciones fiables con costo nulo a nivel de software.

3. Implementar un sistema de VoIP en la nube pública que sustituya el servicio de una CPaaS, con lo cual el costo total en esta nube se reduciría enormemente.

4. Hacer pruebas de estrés de hardware y software en el servidor físico o en la nube, a utilizar en producción. En tales pruebas se recomienda comparar los resultados del servidor de la empresa con los obtenidos en este trabajo de graduación, para tener un punto de comparación en lo que respecta al rendimiento general del servidor.

5. Para aquellos que desean desarrollar un sistema de alerta basado en este trabajo de graduación para una empresa, deberían efectuar las futuras implementaciones al SMTP Appliance para que sea considerado un servidor productivo y así acoplarlo área de servidores.

BIBLIOGRAFÍA

1. 3CX. *¿Qué es voz sobre IP (VoIP)?.* [en línea]. < https://www.3cx.es/voip-sip/voz-sobre-ip/>. [Consulta: 09 de enero de 2019].

2. AGNIHOTRI, Nikhil. *AT Commands, GSM AT command set*. [en línea]. <https://www.engineersgarage.com/tutorials/at-commands-gsm-at-command-set> [Consulta: 22 de mayo de 2019].

3. Altronics. *Módulo GSM/GPRS SIM800L 5V evb*. [en línea]. <https://altronics.cl/modulos-gsm/mod-sim800l-evb>. [Consulta: 03 de septiembre de 2019].

4. Amazon.com, Incorporation. *Fujitsu Primergy TX1320 M3 - Ordenador de Sobremesa*. [en línea]. <https://www.amazon.es/Fujitsu-PRIMERGY-TX1320-E3-1220V6-Torre/dp/B0716XPYVM>. [Consulta: 21 de septiembre de 2019].

5. _____. *Raspberry Pi 3 Model B+*. [en línea]. <https://www.amazon.es/Raspberry-Model-Official-Essentials-Black/dp/B07BFVYMJY>. [Consulta: 12 de septiembre de 2019].

6. _____. *Xia Fly SIM808 Module GSM GPRS*. [en línea]. <https://www.amazon.com/gp/offer-listing/B07MY62M4L>. [Consulta: 02 de septiembre de 2019].

7. Amazon Web Services, Incorporation. *AWS Pricing Calculator*. [en línea]. <https://calculator.aws>. [Consulta: 15 de septiembre de 2019].

8. BAUGHER, Mark. *The Secure Real-time Transport Protocol (SRTP)*. [en línea]. <https://tools.ietf.org/html/rfc3711>. [Consulta: 03 de abril de 2020].

9. Claro Guatemala. *Postpago.* [en línea] <https://www.claro.com.gt/personas/servicios/servicios-moviles/ postpago>. [Consulta: 11 de septiembre de 2019].

10. Citelia. *Redes privadas virtuales-VPN.* [en línea]. <https://citelia.es/blog/redes-privadas-virtuales-vpn/> [Consulta: 16 de enero de 2019].

11. Codetec. *Calcula tu mantenimiento*. [en línea]. <https://codetec.es/servicios/calcula-mantenimiento>. [Consulta: 27 de agosto de 2019].

12. Cleo Solutions. *Simple Mail Transfer Protocols*. [en línea]. <https://www.cleo.com/solutions/integration-connectors/smtp-smtps-simple-mail-transfer-protocol>. [Consulta: 04 de abril de 2020].

13. Definición De. *Definición de SMS* [en línea]. <https://definicion.de/sms/>. [Consulta: 14 de enero de 2019].

14. DELL Technologies. *PowerEdge T30 Mini Tower Server*. [en línea]. <https://www.dell.com/en-us/work/shop/cty/pdp/spd/poweredge-t30> [Consulta: 20 de septiembre de 2019].

15. Django Software Foundation. *Meet Django*. [en línea]. <https://www.djangoproject.com>. [Consulta: 07 de abril de 2020].

16. EBay Incorporation. *Raspberry Pi 3 modelo A+.* [en línea]. <https://www.ebay.com/itm/RASPBERRY-PI-Base-Plate-3-Model-A-Cortex-to-1-4-GHZ-WiFi-5-Ghz/193023563747>. [Consulta: 11 de septiembre de 2019].

17. _____. *1PCS SIMCOM SIM900*. [en línea]. <https://www.ebay.com/itm/1PCS-SIMCOM-SIM900-Quad-band-GSM-GPRS-Shield-Development-Board-Antenna-K9/222130277143>. [Consulta: 02 de septiembre de 2019].

18. Gesbyte Soluciones Tecnológicas. *PC MINIATX RYZEN 3 3200*. [en línea]. <https://gesbyte.com/producto/pc-miniatx-ryzen-3-3200-ddr4-8gb-240gb-ssd-windows10prof>. [Consulta: 20 de septiembre de 2019].

19. Google, LLC. *Google Cloud Pricing Calculator.* [en línea]. <https://cloud.google.com/products/calculator>. [Consulta: 16 de septiembre de 2019].

20. HANDLEY, Mark. *SDP: Session Description Protocol*. [en línea]. <https://tools.ietf.org/html/rfc2327>. [Consulta: 11 de enero de 2019].

21. HERNANDEZ, Juan Estuardo. *Ubuntu Server*. [en línea]. <http://911-ubuntu.weebly.com/ubuntu-server.html>. [Consulta: 13 de abril de 2019].

22. _____. *Nethserver Small Business - Controlador de Dominio híbrido*. [en línea]. <http://911-ubuntu.weebly.com/nethserver>. [Consulta: 6 de abril de 2019].

23. _____. *Zabbix Monitoreo de "Redes, Servicios y aplicaciones"*. [en línea]. <http://911-ubuntu.weebly.com/zabbix/zabbix-monitoreo-de-redes-servicios-y-aplicaciones>. [Consulta: 16 de abril de 2019].

24. _____. *Proxmox Plataforma de Virtualización.* [en línea]. <http://911-ubuntu.weebly.com/Proxmox>. [Consulta: 5 de abril de 2019].

25. Informatica Mayes. *Mantenimiento informático*. [en línea]. <https:///www.mayes.es/mantenimiento-informatico.htm>. [Consulta: 30 de agosto de 2019].

26. Instituto Federal de Telecomunicaciones. *Sabías qué la Telefonía Móvil*. [en línea]. < http://www.ift.org.mx/usuarios-telefonia-movil/sabias-que-la-telefonia-movil>. [Consulta: 08 de enero de 2019].

27. Mantenimiento Digital. *Tarifa de soporte infrormatico y mantenimiento*. [en línea]. <https://www.mantenimientodigital.com/descargas/ Tarifas.pdf>. [Consulta: 29 de agosto de 2019].

28. Microsoft Corporation. *¿Qué es la nube pública, privada e híbrida?* [en línea]. <https://azure.microsoft.com/es-es/overview/what-are-private-public-hybrid-clouds> [Consulta: 13 de septiembre de 2019].

29. _____. *Pricing calculator*. [en línea]. <https://azure.microsoft.com/en-us/pricing/calculator>. [Consulta: 17 de septiembre de 2019].

30. MUTHUKADAN, Baiju. *Selenium with Python*. [en línea]. <https://selenium-python.readthedocs.io/>. [Consulta: 10 de abril de 2020]

31. NOJ LÓPEZ, Jacob Isreal. *Pruebas de estrés en servidores web* [en línea]. <https://es.slideshare.net/corey486/pruebas-de-estres-windows-linux>. [Consulta: 20 de diciembre de 2019].

32. Plivo Incorporation. *Plivo Pricing*. [en línea]. <https://www.plivo.com/pricing>. [Consulta: 19 de septiembre de 2019].

33. ROSENBERG, Jonathan. *SIP: Session Initiation Protocol.* [en línea]. <https://tools.ietf.org/html/rfc3261>. [Consulta: 10 de enero de 2019].

34. SCHULZRINNE, Henning. *RTP: A Transport Protocol for Real-Time Applications* [en línea]. <https://tools.ietf.org/html/rfc3550>. [Consulta: 12 de enero de 2019].

35. _____. *RTP Profile for Audio and Video Conferences with Minimal Control*. [en línea]. <https://tools.ietf.org/html/rfc3551>. [Consulta: 13 de enero de 2019].

36. Sensaphone. *IMS-4000 Enterprise Monitoring Node Expansion Unit*. [en línea]. <https://www.sensaphone.com/products/ims-4000-

enterprise-monitoring-node-expºansion-unit>. [Consulta: 03 de noviembre de 2019].

37. SILGADO, Alicia. *¿Qué es el SMTP? Ventajas e inconvenientes de un servidor SMTP.* [en línea]. <https://blog.mailrelay.com/es/2017/04/25/que-es-el-smtp>. [Consulta: 14 de enero de 2019].

38. Tecnología+Informática. *Que es un Firewall o Cortafuegos. Tipos*. [en línea]. <https://tecnologia-informatica.com/que-es-firewall-como-funciona-tipos-firewall>. [Consulta: 23 de enero de 2019].

39. The Linux Foundation. *Orquestación de contenedores para producción*. [en línea]. <https://kubernetes.io/es/>. [Consulta: 05 de abril de 2020].

40. Tigo Guatemala. *Adquiere un Plan Sin Contrato en línea*. [en línea]. <https://www.tigo.com.gt/movil/postpago/plan-sin-contrato> [Consulta: 10 de septiembre de 2019].

41. Twilio Incorporation. *Twilio pricing*. [en línea]. <https://www.twilio.com/pricing>. [Consulta: 19 de septiembre de 2019].

42. Twisted Matrix Labs. *What is Twisted?* [en línea]. <https://twistedmatrix.com/trac>. [Consulta: 08 de abril de 2020].

43. Vonage Business Communications. *Pricing*. [en línea]. <https://www.nexmo.com/pricing>. [Consulta: 20 de septiembre de 2019].

44. Zabbix SIA. *Zabbix Manual*. [en línea]. <https://www.zabbix.com/documentation/current/manual>. [Consulta: 03 de marzo de 2019].

APÉNDICES

Apéndice 1. **Manual del usuario del SMTP Appliance**

En la sección 3.2.1. se detalló el sistema de archivos de la aplicación utilizado en la elaboración del servidor. Algunos de estos archivos tienen variables de configuración las cuales definen el comportamiento del *appliance* y pueden ser modificables por el desarrollador. De estos archivos se destacan:

- SMTP_Appliance.py: Contiene variables para conectarse a la base de datos del servidor de monitoreo Zabbix, en caso de poseer alguno. El objetivo de conectarse esta base de datos es modificar la tabla de correos electrónicos de forma automática con los métodos del *appliance*. Estas variables son:

  ZABBIX_HOST = '<dirección IP o FQDN del servidor Zabbix>'
  ZABBIX_USER = '<usuario de la base de datos>'
  ZABBIX_PASSWD = '<contraseña de la base de datos>'
  ZABBIX_DB = '<nombre de la base de datos>'

  Es de notar que para poder conectarse a la base de datos de Zabbix, se requiere la creación de un usuario dentro de ella y cuyos parámetros deben coincidir con los ingresados en este archivo.

  Antes de crear el usuario de base de datos, se debe abrir una terminal de Zabbix y ejecutar el siguiente comando:

Continuación del apéndice 1.

$ sudo mysql_secure_installation

Al usar este comando se solicitará asignar una contraseña al usuario 'root', que por defecto no tiene ninguna, habiltar el acceso remoto desde otros equipos, borrar bases de datos de prueba, entre otros. Después se debe ingresar a la base de datos con el usuario 'root' y crear el usuario el cual va a ser usado con los parámetros previamente descritos.

$ sudo mysql -u root -p
Password: #Contraseña asignada anteriormente
mysql> GRANT ALL PRIVILEGES ON **<zabbix_db>**.* TO
"**<zabbix_user>**"@"**<IP o FQDN del SMTP_Appliance>**" ->
IDENTIFIED BY "**<zabbix_passwd>**";
mysql> FLUSH PRIVILEGES;

- serverlib.py: Este es el archivo del receptor. Las variables que definen el comportamiento del receptor son:

DEFAULT_TIMEOUT = <tiempo de espera del *socket*>
SMTP_PORT = <puerto del servidor>
MAX_LINE_SIZE = <cantidad de caracteres a recibir por línea>
SMTP_MAXWAIT_TIME = <tiempo de espera de conexión>
MAX_ITEMS_SIZE = <cantidad de mensajes del filtro>
DEL_ALLITEMS_GAP = <tiempo de espera para eliminar elementos>
DEL_ITEM_GAP = <tiempo de espera para eliminar mensajes>
DONT_PROCESS_TUPLE = ('mensaje 1', 'mensaje 2',...)

Continuación del apéndice 1.

SMTP_USERS = OrderedDict([('usuario 1', 'nombre de usuario 1', 'grupo de usuario 1'), ('usuario 2', 'nombre de usuario 2', 'grupo de usuario 2'), …])

Las primeras cuatro variables definen el comportamiento del *socket*.

Generalmente, los *sockets* en Python tienen por defecto un mecanismo de bloqueo, que no permite el cierre de este hasta que se reciba algún mensaje. Tal bloqueo no permite el cierre de la aplicación, por ello se asignó un tiempo de espera para recibir mensajes, indicado en la variable DEFAULT_TIMEOUT.

La variable SMTP_PORT indica el puerto por el cual se estarán recibiendo los mensajes de tipo SMTP destinados a este *appliance*.

La variable MAX_LINE_SIZE define que la cantidad de caracteres que puede recibir por mensaje de tipo SMTP.

Este servidor evita que los equipos de monitoreo se queden conectados indefinidamente, por lo que este servidor espera la cantidad de tiempo definida en la variable SMTP_MAXWAIT_TIME para el equipo conectado envíe un mensaje de tipo SMTP, de lo contrario la conexión se dará por terminada.

Las variables de la cinco a la ocho definen el comportamiento del filtro temporal de mensajes de correo.

Continuación del apéndice 1.

Ya que no se puede contar con memoria RAM infinita, la variable MAX_ITEMS_SIZE define la cantidad de mensajes de correo que puede albergar el filtro. En caso de excederse, el filtro efectúa un análisis del primer mensaje dentro de su lista y en caso de ya haya sido procesado, es eliminado e ingresa el mensaje nuevo tanto al filtro como a la memoria FIFO, de lo contrario el mensaje nuevo es el que se elimina.

El filtro temporal será vaciado dentro del tiempo definido en la variable DEL_ALLITEMS_GAP, lo que eliminará la carga sobre la memoria RAM.

En caso de que un mensaje de correo recibido sea repetido, si el tiempo entre el mensaje dentro del filtro y el mensaje repetido no supera al tiempo definido en la variable DEL_ITEM_GAP, o en caso de que el mensaje dentro del filtro aún no se ha procesado, el mensaje repetido es eliminado, de lo contrario se elimina el mensaje dentro del filtro, luego se agrega el repetido al filtro, así como a la memoria FIFO.

Al filtro se le puede indicar cierta cantidad de mensajes los cuales no hay que procesar. Estos mensajes se definen dentro de la variable DONT_PROCESS_TUPLE.

Dentro de la variable SMTP_USERS, dentro de la clase SMTPServer, se define los usuarios que son capaces de recibir mensajes de correo.

- Phone/__init__.py: Dentro de este archivo existen varias variables de las cuales las más importantes son:

Continuación del apéndice 1.

LANG = ‘<lenguaje>’
VOIP_USERNAME, VOIP_DOMAIN, VOIP_PASSWORD, VOIP_DISPLAY_NAME = ‘<extensión asignada>’, ‘<dirección IP o FQDN del servidor FreePBX>’, ‘<contraseña asignada>’, ‘<nombre de la aplicación>’
SOFTPHONE_MAIL, SOFTPHONE_PASSWORD = '<correo del servidor>', '<contraseña de la aplicación>'
MAIL_NUMBERS = OrderedDict([(‘usuario 1’, ‘extensión de usuario 1’, ‘número telefónico de usuario 1’), (‘usuario 2’, ‘extensión de usuario 2’, ‘número telefónico de usuario 2’),…])

La variable LANG define el lenguaje en el cual será transmitido el mensaje vía voz. El formato a seguir para esta variable es el de etiquetas IETF. Por defecto se le ha asignado un el valor de ‘en’ que representa al lenguaje inglés.

La segunda línea de variables son los parámetros para efectuar de llamadas por VoIP, véase Apéndice 4 en la sección de Extensiones.

La tercera línea son las variables del correo electrónico del servidor. Para que el servicio de correo funcione, se debe crear una cuenta en el servidor de Gmail, activar la verificación de dos pasos de la cuenta y crear una contraseña de aplicación para este *appliance,* parámetro el cual se debe ingresar en la variable SOFTPHONE_PASSWORD.

La variable MAIL_NUMBERS, ubicada en la clase SMTP_Softphone, contiene la información de los usuarios para el envío del mensaje de alerta. Al igual que la variable SMTP_USERS, los usuarios están definidos por su correo electrónico. Cada usuario debe estar registrado tanto en MAIL_NUMBERS como en SMTP_USERS.

Es de recalcar que cada variable que se encuentra entre comillas representa una cadena de caracteres, denominada *string*, y las otras son números enteros.

El *appliance* aparte de tener variables, también cuenta con métodos para la administración del mismo. Los métodos se describen en la siguiente tabla:

Métodos del SMTP Appliance

Método	**Descripción**
__init__(timeout = DEFAULT_TIMEOUT, lang = LANG)	Método de inicialización del servidor. Se ejecuta cuando se inicializa la clase SMTP_Softphone, por ejemplo server = SMTP_Softphone). Los parámetros son: • timeout; entero: Tiempo de espera del *socket* receptor de mensajes. Por defecto tiene la variable DEFAULT_TIMEOUT, sin embargo se puede modificar si así se desea. • lang; *string*: Representa el lenguaje del audio a enviar por llamada VoIP y telefónica. Por defecto tiene la variable LANG.

Continuación del apéndice 1

addUser(new_user, phone_number = '', extension = '', username = '', group = 'My_Gruop')	Agrega un nuevo usuario a las variables SMTP_USER y MAIL_NUMBERS. Los valores a ingresar son: • new_user; *string*: Nuevo usuario a ingresar. Debe seguir el formato de correo electrónico, por ejemplo: fer@ejemplo.com. • phone_number; *string*: Número de teléfono del nuevo usuario. • username; *string*: Nombre del nuevo usuario • group; *string*: Grupo al que pertenece el nuevo usuario.
updateUser(new_user, phone_number = '', extension = '', username = '', group = 'My_Gruop', position = 0)	Permite sustituir un usuario en la posición indicada por uno nuevo en las variables SMTP_USERS y MAIL_NUMBERS. Además sustituye los correos en la base de datos del servidor de monitoreo Zabbix. Al usar este método se ingresan los siguientes argumentos: • new_user; *string*: Nuevo usuario. Debe seguir el formato de correo electrónico, por ejemplo fer@ejemplo.com. • phone_number; *string*: Número de teléfono del nuevo usuario. • extension; *string*: Extensión VoIP a asignar al nuevo usuario. Este valor debe estar registrado en la central telefónica. • username; *string*: Nombre del nuevo usuario. • group; *string*: Grupo al que pertenece el nuevo usuario. • position; entero: Número de posición del usuario a ser sustituido dentro de las variables de registro de usuarios.
enableMail()	Habilita el módulo Mail. No posee argumentos.
disableMail()	Inhabilita el módulo Mail.

Continuación del apéndice 1.

enableSIMCall()	Habilita el módulo SIM.
disableSIMCall()	Inhabilita el módulo SIM.
enableSMS()	Habilita el módulo SMS.
disableSMS()	Inhabilita el módulo SMS.
enableSMS_fulltext()	Habilita el modo texto completo. Este modo permite enviar varios mensajes de texto hasta completar el envío de la alerta, de lo contrario solo se enviará una parte del mensaje de alerta en un solo mensaje de texto.
disableSMS_fulltext()	Inhabilita el modo texto completo.
enableVoIP()	Habilita el módulo VoIP.
disableVoIP()	Inhabilita el módulo VoIP.
setLang(lang = LANG)	Modifica el lenguaje en el cual se estarían enviado las alertas por voz.
setOrder(order = (0, 1, 2, 3))	Cada módulo del emisor es representado por un valor numérico, en donde el módulo Mail representa 0, el módulo VoIP representa 1, el SIM 2 y el SMS 3. Este método permite ingresar el orden en el que se ejecutará la transmisión del mensaje de alerta en base a la representación numérica de cada módulo ingresada en el argumento *order;* tupla. Es decir que, si el argumento *order* vale '(1,2,3)', se enviará la alerta iniciando por el módulo VoIP, luego por el módulo SIM y culminará con el envío de SMS.
stop()	Detiene y cierra a todos los módulos del emisor.
close()	Cierra el receptor.

Continuación del apéndice 1.

Para iniciar el servidor, como se indica en la sección 3.2.1., se efectúa los siguientes comandos en una terminal de servidor SMTP Appliance:

```
$ cd Softphone/code #Ubicarse en el directorio code del appliance
$ sudo python #Ingresar al intérprete de Python
>> from SMTP_Appliance import SMTP_Softphone #Importar la clase STMP_Softphone de la librería SMTP_Appliance
>> server = SMTP_Softphone() #Inicializar el servidor, con lo que se llama al método __init__.
```

Luego ya se puede empezar enviar mensajes de alerta y configurar el servidor con los métodos previamente descritos.

Si se desea ejecutar cualquiera de los métodos, con excepción de __init__, una vez iniciado el servidor, se debe ejecutar lo siguiente en el intérprete de Python:

```
>> server.<método> <argumentos si los hay>
```

Si se desea finalizar el servidor se deben ejecutar tanto los métodos *stop()* como *close()*, sin importar el orden en el que se ejecuten.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 2. **Proxmox VE**

Un virtualizador es una plataforma que es capaz de administrar los recursos en hardware del equipo físico en el que se encuentra para crear máquinas virtuales que compartan estos mismos recursos.

Proxmox VE, *Virtual Environment*, es un potente virtualizador utilizado a nivel empresarial, sin costo alguno, con muchos beneficios similares a los que ofrecen otros tipos de virtualizadores como lo son OracleVM, VMWare vSphere, Citrix XenServer y Windows Hyper-V.

Previo a instalación

Para la instalación de Proxmox VE, se requiere de un servidor físico o servidor virtual, el cual requiere la activación de la 'virtualización añidada' en el equipo físico, lo que no es muy recomendable a nivel empresarial. La imagen ISO de Proxmox se puede encontrar en el siguiente link:

https://www.Proxmox.com/en/downloads

Del cual se escogerá el archivo con el nombre de 'Proxmox VE 5.2 ISO Installer'. Se puede escoger otro ISO más reciente.

Previo a la instalación es necesario seguir los siguientes pasos:

Continuación del apéndice 2.

1. Validar si el equipo es apto para virtualizar, por lo que se debe validar si la opción de ‘Hardware Virtualization Extensions’ está habilitado en la BIOS.

2. Arrancar el equipo desde una memoria USB con un SO Linux con el cual se pueda acceder en forma de versión de prueba.

3. Abrir una terminal de Linux.

4. En caso de tener un procesador Intel debe ejecutar el siguiente comando:

   ```
   $ grep --color vmx /proc/cpuinfo
   ```

   En caso de tener un procesador AMD debe ejecutar el siguiente comando:

   ```
   $ grep --color svm /proc/cpuinfo
   ```

   Como resultado obtendrá la palabra vmx/svm resaltada, dependiendo de la cantidad de procesadores, será las veces que aparecerá la palabra vmx/svm. De no obtener ningún resultado significa que el procesador no es capaz de virtualizar.

5. Generar una memoria USB con el instalador del sistema operativo, SO, de Proxmox. Para este paso se recomienda el uso de la herramienta ‘Rufus’, que ha mostrado resultados satisfactorios para generar memorias USB con instaladores de sistemas operativos.

Continuación del apéndice 2.

Instalación

La instalación de Proxmox 5.2 implanta en el servidor físico lo siguiente:

- Debian x64 como sistema operativo.
- Particiona los discos como LVM y Thinly-Provisioned Logical Volumes.
- Instalación y configuración de "Proxmox Ve *Kernel*", con soporte tanto para KVM así como LXC.
- Herramientas de copia de respaldo y restauración
- Instalación y configuración del interfaz Web HTML5, para la administración del entorno virtualizado.

Siguiendo la guía de instalación, los pasos de instalación son los siguientes:

1. Iniciar el equipo desde la USB con Proxmox.

Una vez iniciado aparecerá la siguiente pantalla:

Continuación del apéndice 2.

Instalación de Proxmox – Pantalla de Inicio

Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

Se debe seleccionar la opción ‘Install Proxmox VE’.

2. A continuación aparecerá la opción de términos de licencia:

Continuación del apéndice 2.

Instalación de Proxmox – Términos de licencia

Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

Este acuerdo de licencia indica que este producto es *open source*. Seleccione la opción 'I agree'.

3. La siguiente pantalla es para seleccionar la forma de distribución de las particiones del disco:

Continuación del apéndice 2.

Instalación de Proxmox – Particionamiento del disco

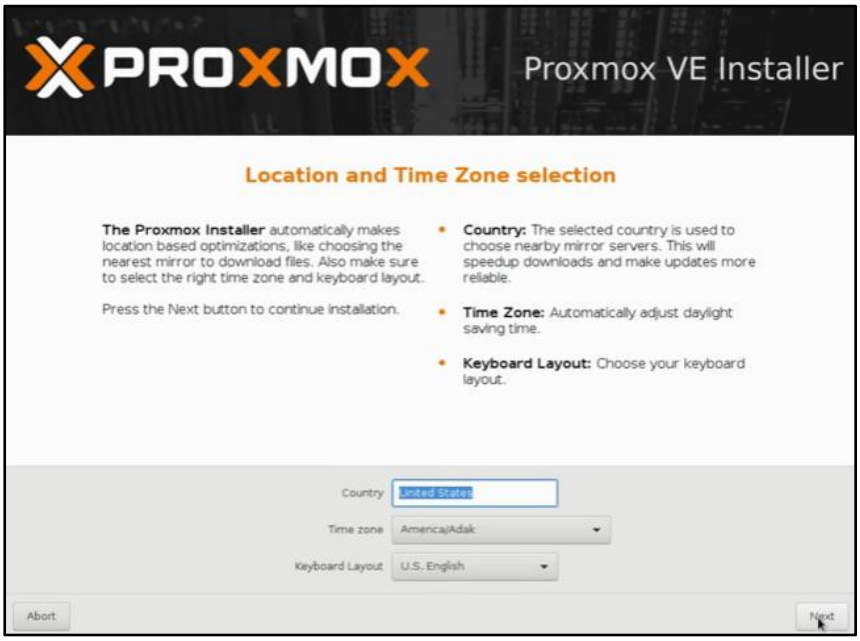
Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

Si selecciona el botón ‘Options’, podrá ver todos los parámetros de las particiones del disco: tipo de sistema de archivos, tamaño, cantidad de particiones, entre otros. En caso de solo seleccionar ‘Next’ tomará la configuración de las particiones por defecto. Para caso de esta guía se seleccionará únicamente ‘Next’

4. Siguiendo con el proceso aparecerá lo siguiente:

Continuación del apéndice 2.

Instalación de Proxmox – País, zona horaria y teclado



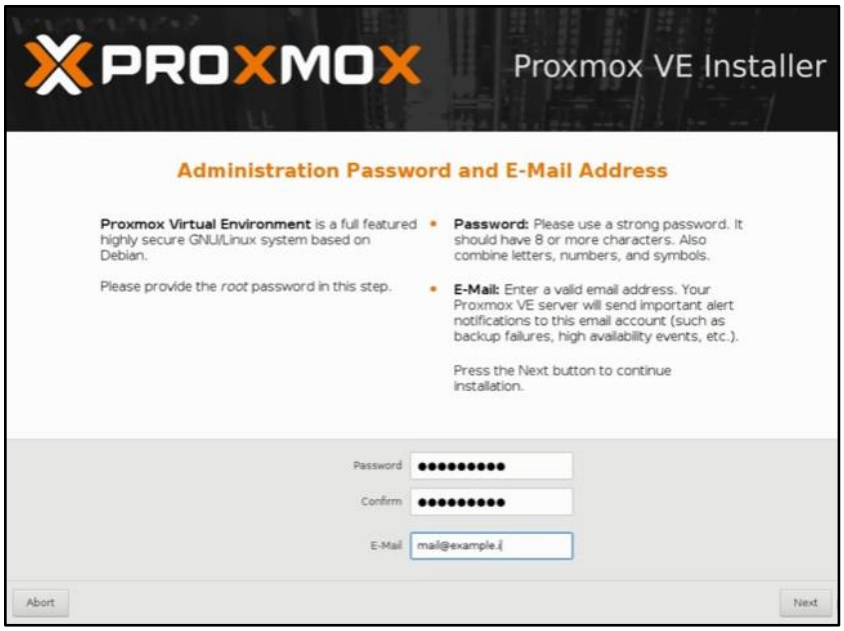
Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

En esta sección debe indicar el país, zona horaria y teclado. Si se posee un servidor de DHCP, esta información se auto completa.

Continuación del apéndice 2.

5. Ingresar contraseña para usuario ‘root’ y correo:

Instalación de Proxmox – Contraseña y correo



Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

Esta contraseña se utilizará para los accesos vía SSH e interfaz web. El correo aún necesita ser configurado para que se puedan recibir alertas sobre fallas, se recomienda utilizar la guía de asignación de correo de Proxmox.

Continuación del apéndice 2.

6. Se deben colocar los parámetros de la interfaz de red a utilizar:

Instalación de Proxmox – Parámetros de la interfaz de red

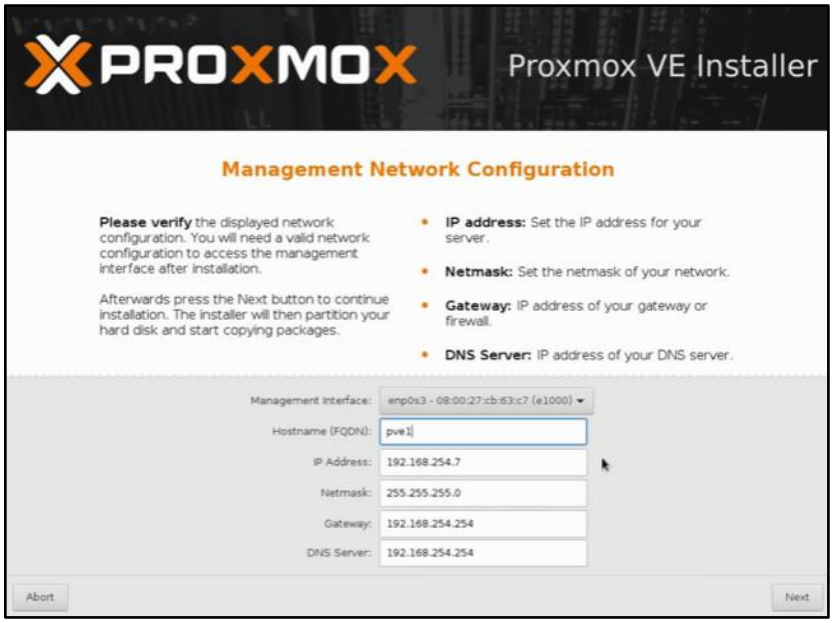
Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

Si se tiene un servidor de DHCP, solamente se recomienda modificar el *hostname*. Al darle click al botón ‘Next’, se empezará a realizar la instalación.

Continuación del apéndice 2.

7. Se deben colocar los parámetros de la interfaz de red a utilizar:

Instalación de Proxmox – Parámetros de la interfaz de red



Fuente: Tecno Dad Life. *Install Proxmox to Virtualize All Your Servers*. https://www.youtube.com/watch?v=ZNpTP_En_bo. Consulta: 5 de abril de 2019.

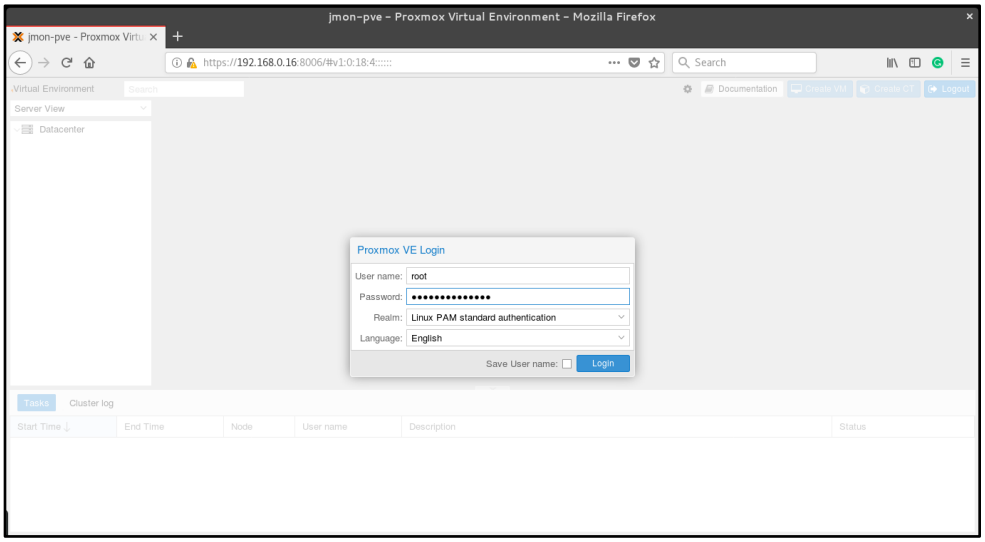
Si se tiene un servidor de DHCP, solamente se recomienda modificar el *hostname*. Al darle click al botón ‘Next’, se empezará a realizar la instalación.

8. Para ingresar a la administración web servidor Proxmox se debe realizar desde el navegador de otro equipo conectado a la red, ingresando la siguiente URL:

Continuación del apéndice 2.

https://<la dirección IP del servidor>:8006

Instalación de Proxmox – Ingreso a interfaz web imagen 1

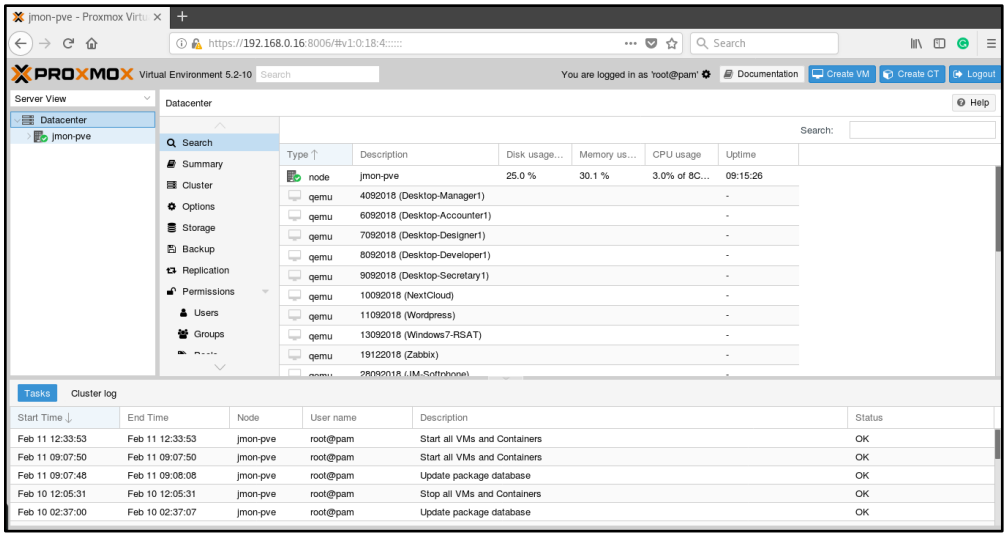


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Se ingresa con las credenciales de usuario, el cual por defecto es ‘root’, y contraseña con lo que ya se puede empezar a administrar:

Continuación del apéndice 2.

Instalación de Proxmox – Ingreso a interfaz web imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Configuración inicial

Una vez instalado es recomendable realizar los siguientes pasos para obtener un servidor actualizado con todas las herramientas:

1. Ingrese a la línea de comandos del servidor vía SSH, con una pantalla VGA o desde la interfaz web.

2. Los siguientes comandos garantizan que las variables de lenguaje estén correctamente configuradas:

```
$ export LANGUAGE=en_US.UTF-8
$ export LANG=en_US.UTF-8
```

Continuación del apéndice 2.

```
$ export LC_ALL=en_US.UTF-8
$ locale-gen en_US.UTF-8
$ dpkg-reconfigure locales
```

Al aplicar el último comando aparecerá una segunda pantalla en la que debe seleccionarse ‘en_US.UTF-8’.

3. Se debe modificar el archivo ‘/etc/apt/sources.list’ dejándolo de la siguiente forma:

```
deb http://ftp.debian.org/debian stretch main contrib non-free

# PVE pve-no-subscription repository provided by Proxmox.com,
# NOT recommended for production use
deb http://download.Proxmox.com/debian/pve stretch pve-no-subscription
# security updates
deb http://security.debian.org stretch/updates main contrib non-free
```

4. Después de ello se debe inhabilitar paquetes para versión empresarial dejando al archivo ‘/etc/apt/sources.list.d/pve-enterprise.list’ de la siguiente forma:

```
#deb https://enterprise.Proxmox.com/debian/pve stretch pve-enterprise
```

Continuación del apéndice 2.

5. Una vez realizado los dos pasos anteriores, ya se puede actualizar el sistema operativo con el siguiente comando:

```
$ apt-get update && apt-get -y dist-upgrade && apt-get remove --purge && apt-get -y autoremove --purge && apt-get clean && apt-get autoclean
```

Después de aplicar este comando se recomienda reiniciar.

6. Instalar herramientas de monitoreo:

```
$ apt install htop iotop bwm-ng bmon nmon nethogs pydf ncdu tree sshfs nmap mtr-tiny traceroute mc ntfs-3g ldap-utils arj vbetool sysfsutils lshw $(check-language-support)
```

La descripción de cada una de las herramientas se encuentra aquí.

7. Limpiar *kernel* antiguo:

```
$ touch '/please-remove-proxmox-ve'
$ apt-get purge $( dpkg --list | grep -P -o "pve-kernel-\d\S+" | grep -v $(uname -r | grep -P -o ".+\d") )
$ update-grub
```

Se recomienda reiniciar.

8. Instalar herramientas para compresión y descompresión de archivos:

Continuación del apéndice 2.

$ apt install unace zip unzip p7zip-full

Una vez haya completado estos pasos, el servidor Proxmox ya está listo para producción.

Creación de máquina virtual

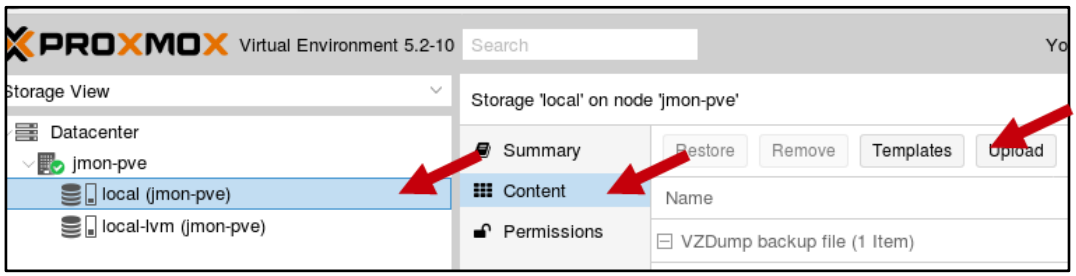
A continuación se describen los pasos para crear una máquina virtual en Proxmox VE 5.2. Los pasos son:

1. Ingresar a la interfaz de administración web.

2. Cargar al servidor la imagen ISO de SO a instalar.

   Para ello se debe seleccionar el nodo, el almacenamiento ‘local’, ‘content’, ‘upload’ y cargar la imagen ISO.

Creación de VM – Carga de ISO imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6

Continuación del apéndice 2.

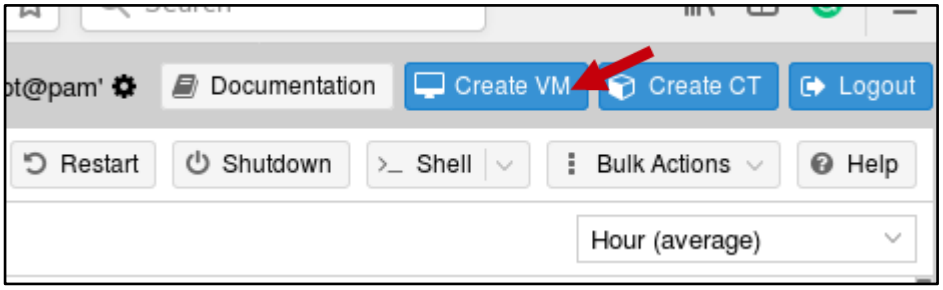
Creación de VM – Carga de ISO imagen 2

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Es de notar que post la instalación, Proxmox genera dos almacenamientos: local y local-lvm, para la gestión de máquinas virtuales, de acrónimo VMs. En el almacenamiento 'local' se almacenan las copias de respaldos de VMs, *templates* de *containers* e imágenes ISO. Además el almacenamiento 'local' es el directorio '/var/lib/vz'. El almacenamiento 'local-lvm' sirve para guardar los discos virtuales de las máquinas virtuales, así como los discos de *containers*.

3. Seleccionar el botón de crear máquina virtual que se encuentra en la esquina superior derecha:

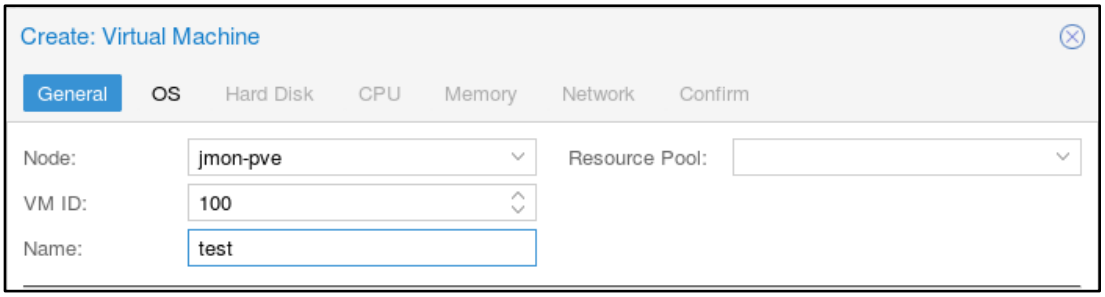
Creación de VM – Botón de crear



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

4. A continuación aparecerá la una ventana general:

Creación de VM – Ventana General



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

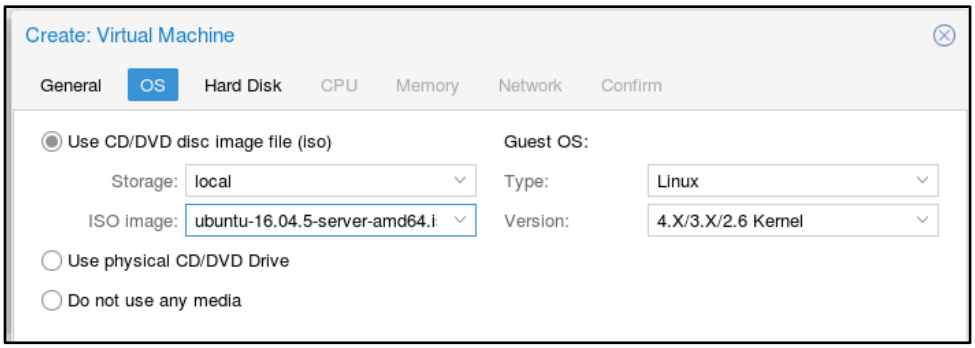
Aquí se debe ingresar los siguientes datos:

- ***Node***: Se debe seleccionar en que equipo, o nodo, será creado la máquina virtual. Proxmox VE permite que varios equipos Proxmox se comuniquen entre sí formando un *cluster*, lo que le permite manipular máquinas virtuales de cualquier equipo enlazado.

- ***VM ID***: Este es el identificador principal del equipo virtual. Proxmox creará un archivo de configuración en '/etc/pve/qemu-server' en el cual albergará toda la configuración principal de la VM. Además los discos virtuales a crear se identificarán con este ID.
- ***Name***: Es un alias genérico que recibirá la VM.
- ***Resource Pool***: Esto es para un conjunto de VMs con ciertos permisos especiales. Este parámetro es opcional.

5. Al presionar 'Next' aparecerán las opciones del sistema operativo, en las que se debe seleccionar la imagen ISO cargada del almacenamiento 'local', el tipo de sistema operativo a usar y su versión:

Creación de VM – Ventana OS



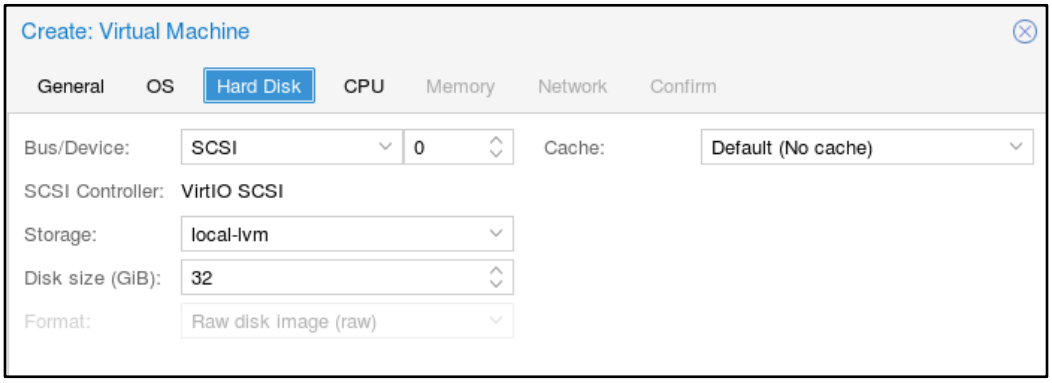
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

6. Después le aparecerán las opciones del disco virtual. Estas son:

Continuación del apéndice 2.

- ***Bus/Device***: Tipo de disco. Entre las opciones están: IDE, SATA, VirtIO Block y SCSI. Tanto VirtIO, como SCSI son para mejor rendimiento. En SO Linux se recomienda usar uno de estos tipos de disco, mientras que en Windows se requiere de cargar un *driver* en la instalación.
- ***Storage***: La ubicación en donde será almacenado el disco virtual. Por defecto se utiliza 'local-lvm'.
- ***Disk size***: Tamaño del disco.
- ***Format***: Indica en que formato se almacenará el disco. Las opciones soportadas son: raw, qcow2 y vmdk. El almacenamiento 'local-lvm' tiene por defecto solo almacenar discos de formato raw.
- ***Cache***: Esta es una opción para mejorar el rendimiento de un disco. En este caso se utilizará la opción por defecto 'No cache'.

Creación de VM – Ventana Disco Virtual

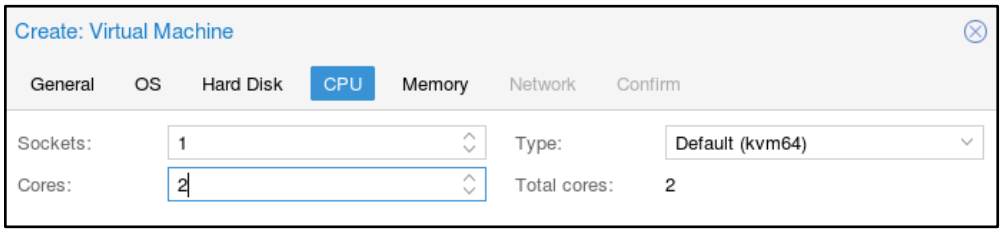


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 2.

7. A continuación vienen las opciones del procesador. En esta sección se selecciona la cantidad de núcleos y el tipo de procesador a emular:

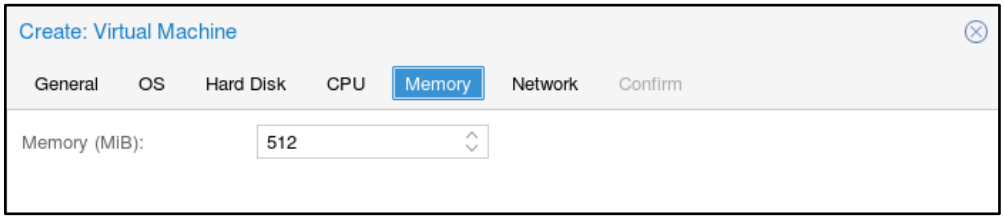
Creación de VM – Ventana CPU



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

8. En el siguiente paso se debe indicar la cantidad de memoria a utilizar.

Creación de VM – Ventana Memoria

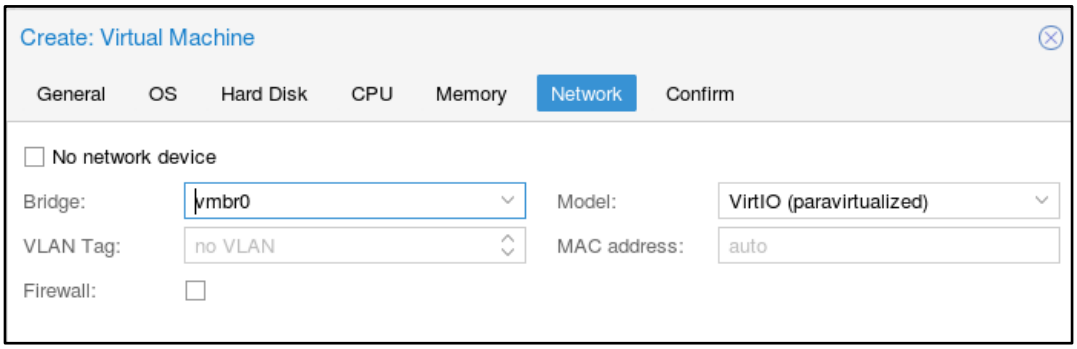


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

9. Lo siguiente a configurar son las opciones de red:

Continuación del apéndice 2.

Creación de VM – Ventana Red



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Los datos a ingresar son:

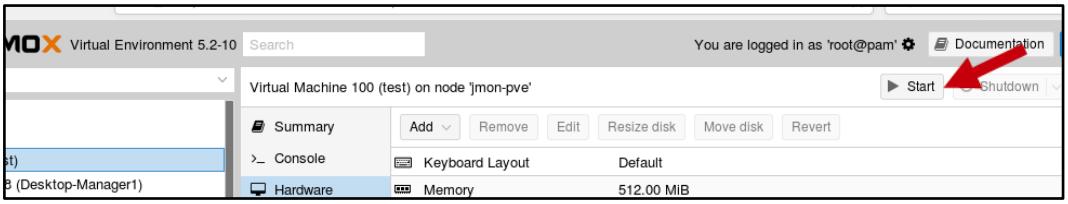
- ***Bridge***: Es la interfaz puente. Esta interfaz es la encargada de administrar los recursos de una interfaz física de red. Por defecto se crea la interfaz puente 'vmbr0'.
- ***VLAN Tag***: Esta es el número de VLAN a utilizar. La interfaz bridge permite la separación de las tarjetas de red para crear VLANs internas. En este proyecto se usarán dos VLANs: Nativa y 10.
- ***Firewall***: Esta opción habilita el *firewall* de Proxmox.
- ***Model***: Los modelos son parámetros que indican el rendimiento de la tarjeta de red. La VirtIO es el mejor modelo, sin embargo, en SO Windows, se requiere de la instalación de un *driver* para su uso.
- **MAC**: se puede asignar cualquier dirección MAC.

10. Confirmar la creación de la VM.

Continuación del apéndice 2.

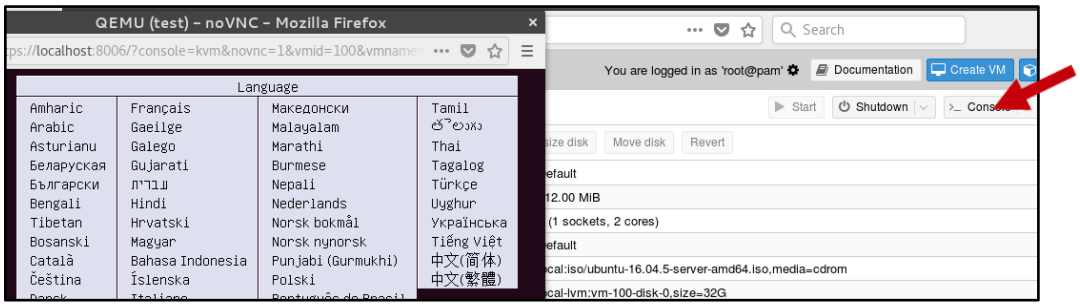
11. Una vez creada ya se puede empezar la instalación.

Creación de VM – Iniciar VM imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Creación de VM – Iniciar VM imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Mapeo de unidad USB

El mapeo de la unidad USB permite a una máquina virtual adueñarse de un puerto USB del servidor Proxmox. Esto permitió al equipo SMTP Appliance utilizar un módulo USB a UART con el cual envía los comandos AT.

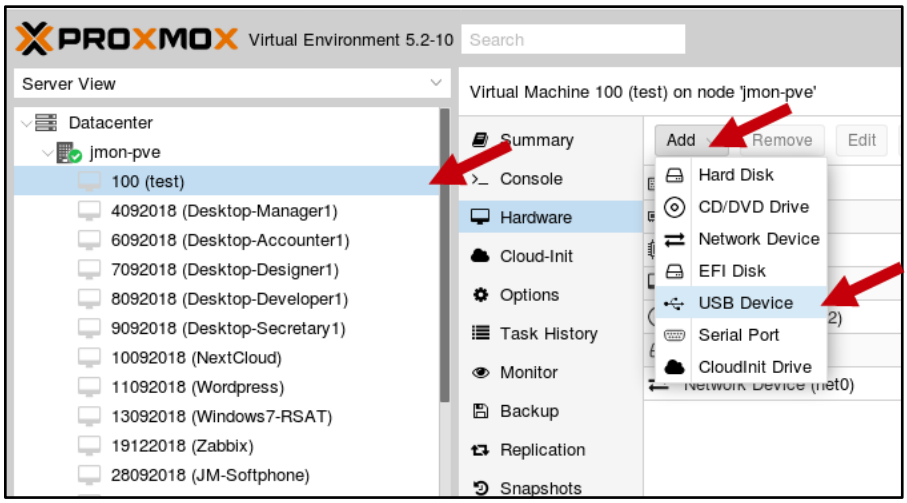
Para mapear una unidad USB se realiza lo siguiente:

Continuación del apéndice 2.

1. Insertar la unidad USB en el servidor Proxmox.

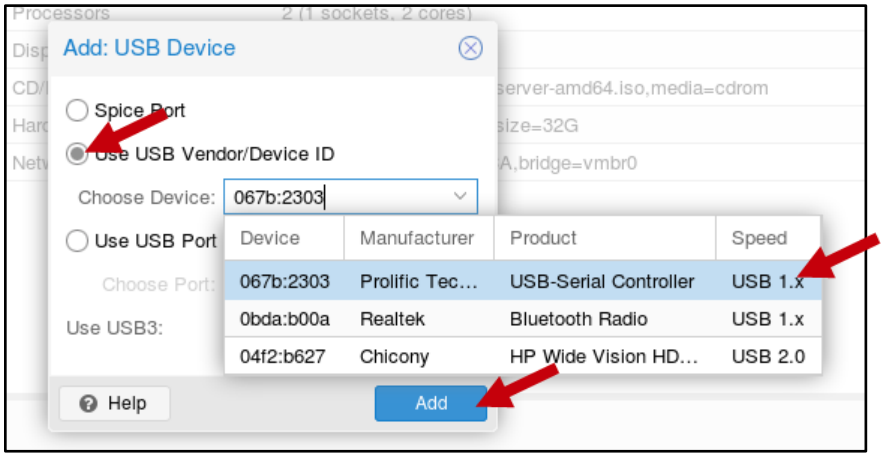
2. Agregar una unidad USB a la máquina virtual.

Mapeo de Puerto USB imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Mapeo de Puerto USB imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 2.

Si es un dispositivo USB 3 se debe habilitar la opción 'Use USB3'.

Mapeo de dispositivo de audio

Para poder utilizar el dispositivo de audio del servidor Proxmox en un equipo virtual, es requerido que el equipo virtual cuente con los *drivers* para SPICE.

SPICE es una solución para ambientes virtuales que permite emular un equipo virtual como si fuese un equipo de escritorio. A diferencia de un equipo virtual convencional, SPICE permite interactuar con los dispositivos de audio, video y USB de la máquina local la cual solicite manipular la máquina virtual. Además SPICE ofrece una experiencia del usuario similar al uso de una máquina virtual como un equipo local. Linux cuenta por defecto soporte para SPICE, mientras que en Windows, se requiere de la instalación de unos *drivers*.

En el equipo SMTP Appliance es requerido habilitar SPICE para enviar datos de audio desde el servidor hasta el módulo GSM. Dado que este equipo es Linux, no requiere de ninguna instalación de *drivers* especiales.

Los pasos para el mapeo del audio son los siguientes:

1. Instalar la herramienta virt-viewer en un equipo local.

   En Linux se puede instalar esta herramienta con el siguiente comando:

   $ sudo apt install virt-viewer

Continuación del apéndice 2.

En Windows se debe descarga esta herrienta de la página oficial o haciendo click aquí.

2. Ingresar a la línea de comandos del servidor Proxmox.

3. Modificar el archivo de configuración de la VM.

   Para ello es necesario ingresar al archivo '/etc/pve/qemu-server/<VM ID>.conf', siendo <VM ID> el ID del equipo virtual el cual se desea configurar, y se debe agregar la siguiente línea:

   args: -device AC97,addr=0x18

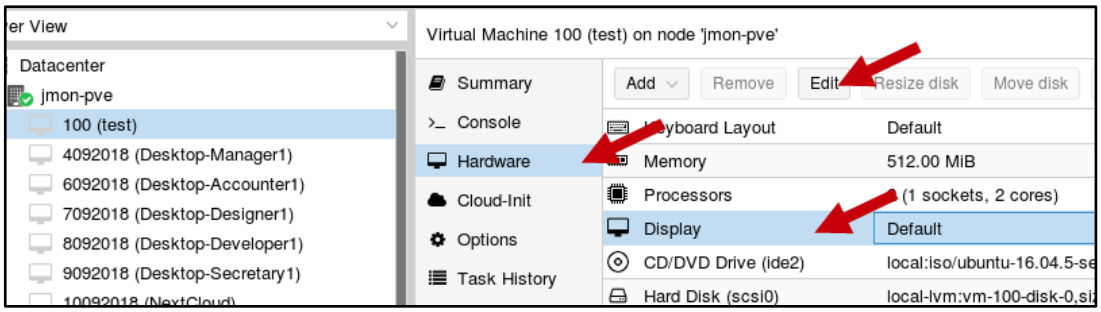
   o

   args: -device intel-hda,id=sound5,bus=pci.0,addr=0x18 -device hda-micro,id=sound5-codec0,bus=sound5.0,cad=0 -device hda-duplex,id=sound5-codec1,bus=sound5.0,cad=1

4. Modificar la pantalla de la VM a SPICE, vía interfaz web.

Continuación del apéndice 2.

Modificación de pantalla imagen 1



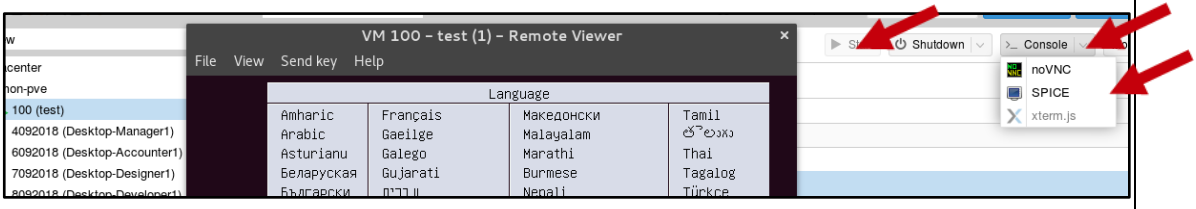
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Modificación de pantalla imagen

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

5. Iniciar la VM con consola tipo SPICE:

Iniciación de VM con consola tipo SPICE



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

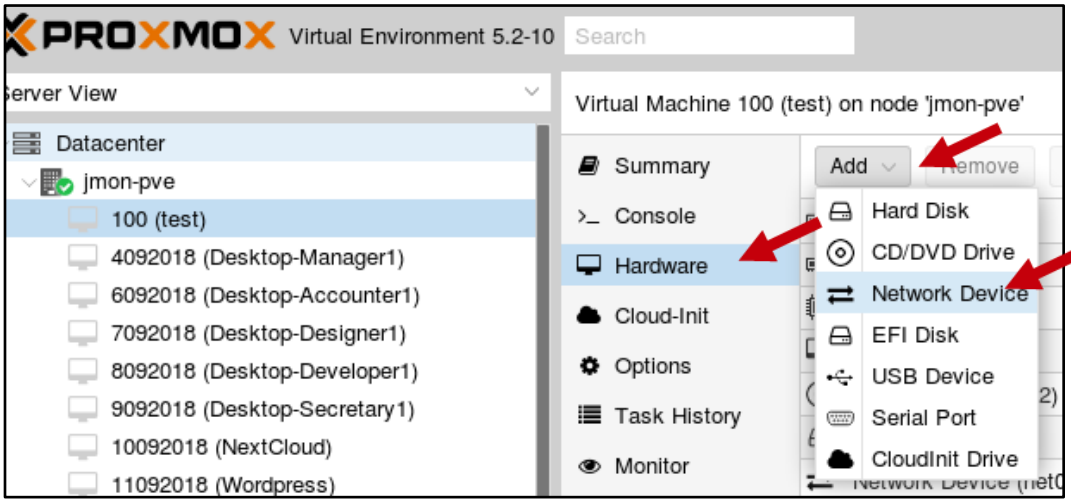
Una vez iniciado el equipo, se cargan todos los *drivers* de SPICE.

Adición de tarjeta de red

El equipo Nethserver necesita trabajar con dos tarjetas de red. Una es para manejar el tráfico interno de las demás máquinas virtuales proveyendo los servicios correspondientes, y la otra es la conexión a Internet.

Las siguientes figuras muestran como agregar una tarjeta de red:

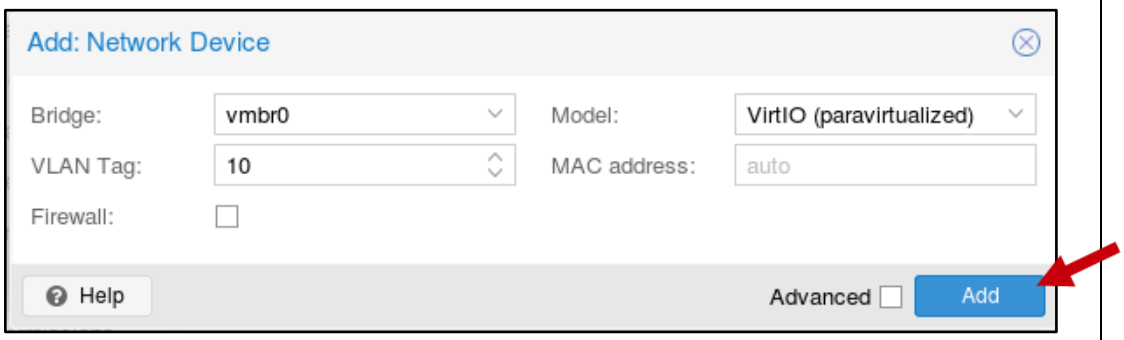
Agregar una tarjeta red imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 2.

Agregar una tarjeta red imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

En el paso anterior se puede seleccionar cualesquiera características de la interfaz.

Esquema de máquinas virtuales

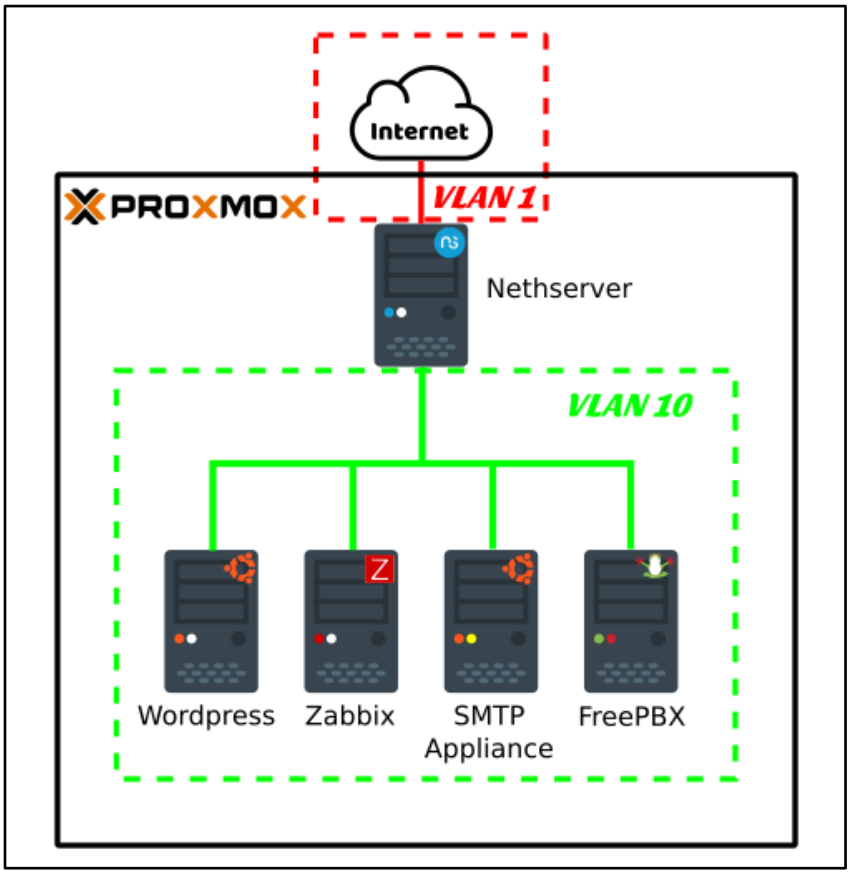
El esquema de máquinas virtuales realizado representa la topología de red de las máquinas virtuales creada dentro del servidor Proxmox. Como se mencionó en la sección 3.2.4, se utilizarán dos VLANs, la nativa o uno, la cual viene por defecto al crear una tarjeta de red, y la diez.

El equipo Nethserver tiene dos interfaces de red, cada una conectada a cada VLAN, mientras que los equipos de Wordpress, Zabbix, SMTP Appliance y FreePBX tendrán una sola interfaz conectada a la VLAN 10.

El esquema es el siguiente:

Continuación del apéndice 2.

Esquema de máquinas virtuales



Fuente: elaboración propia, empleando Inkscape v0.92.4.

Para cada máquina virtual se seleccionaron las siguientes características:

Continuación del apéndice 2.

Características de las máquinas virtuales

Equipo	RAM	CPUs	Disco	Características especiales
Wordpress	1 GB	1	30 GB	-
Zabbix	1 GB	1	30 GB	• Display: VMWare
SMTP Appliance	2 GB	1	40 GB	• Display: SPICE • Puertos USB mapeados: 1
FreePBX	1 GB	1	30 GB	-
Nethserver	1 GB	1	50 GB	• Tarjetas de red: 2; VLAN 1 y VLAN 10
TOTAL	6 GB	5	180 GB	

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 3. **Nethserver**

Nethserver es una plataforma *open source* diseñada para cumplir con todas las soluciones de red requeridas para pequeñas y medianas empresas. Nethserver cumple con las características de un servidor de tipo *Small Business*, capaz de brindar todos los servicios requeridos para la administrar una red como lo son: DHCP, DNS, *firewall*, IPS, *active directory*, entre otros.

Como alternativa a Nethserver, existe Microsoft Small Business Server. Otras soluciones *Small Business* de tipo *open source* están Zentyal, ClearOS, Uninvention, entre otras, aunque estas, en comparación con Nethserver, requieren la compra de *plugins* para utilizar ciertos servicios que en Nethserver son libres de uso.

Instalación

Al instalar Nethserver 7 el equipo poseerá las siguientes características:

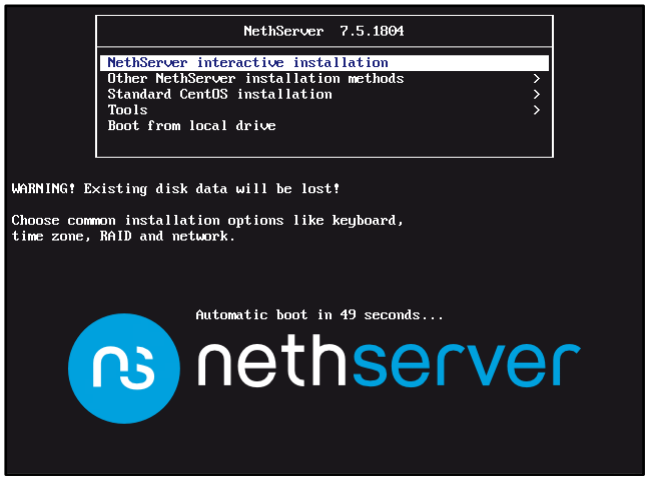
- CentOS 7 x64 como sistema operativo.
- Herramientas para administración de servicios de *firewall*, DHCP, DNS, *Port Forwarding*, entre otros.
- Instalación y configuración del interfaz Web HTML5, para administrar el servidor.

Para instalar Nethserver 7 se usarán los siguientes pasos:

1. Descargar la imagen ISO del sitio oficial y cargarlo al servidor Proxmox.
2. Crear VM con 2 tarjetas de red, VLANs 10 y 1, e iniciarla con la imagen ISO.

3. Seleccionar instalación interactiva.

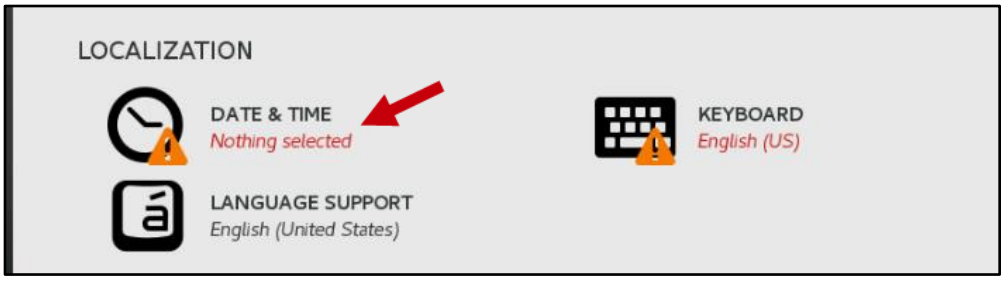
Instalación de Nethserver – Pantalla de inicio



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

4. Seleccionar la zona horaria.

Instalación de Nethserver – Zona Horaria imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Instalación de Nethserver – Zona Horaria imagen 2

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

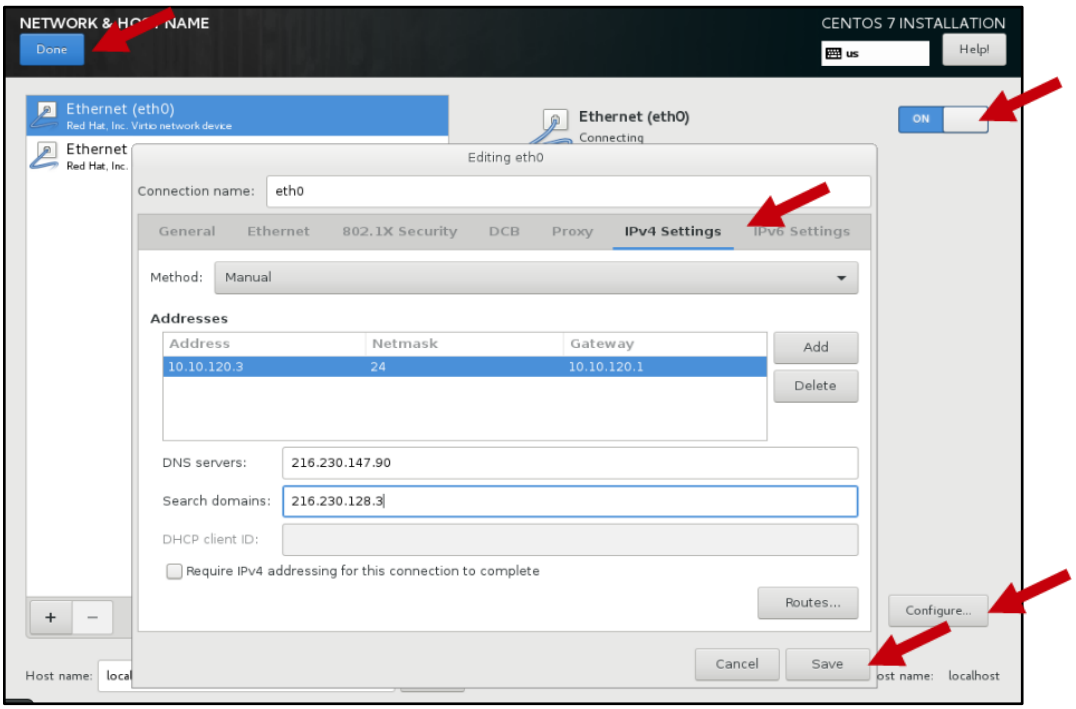
5. Configurar tarjeta de red de VLAN 1 la cual se recomienda que tenga una dirección IP estática.

Continuación del apéndice 3.

Instalación de Nethserver – Red imagen 1

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación de Nethserver – Red imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

También es posible utilizar la opción de DHCP para equipos conectados directamente a un *router* o cualquier otro dispositivo que provea dicho servicio.

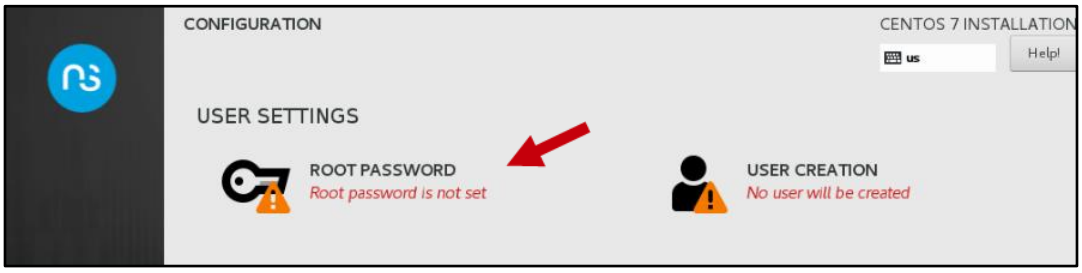
6. Comenzar instalación

Instalación de Nethserver – Comenzar instalación

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

7. Crear contraseña para usuario root

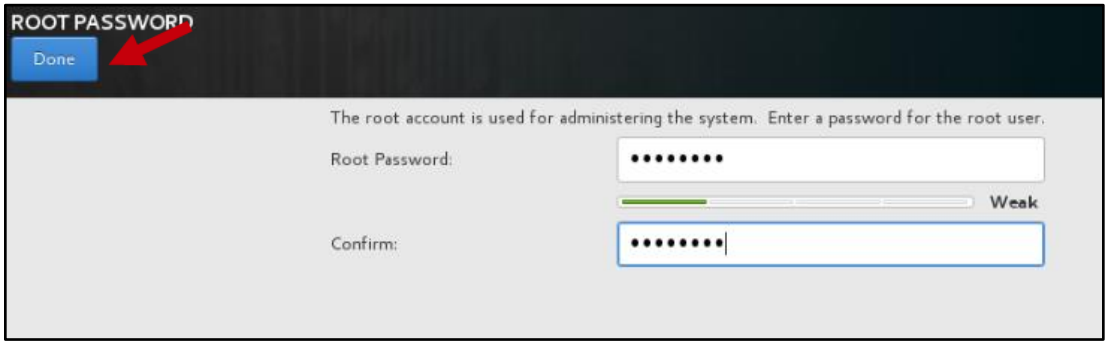
Instalación de Nethserver – Contraseña de root imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Instalación de Nethserver – Contraseña de root imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

8. Ingresar a la terminal del servidor vía consola.

9. Validar que la IP asignada se encuentre en el archivo de configuración de la interfaz de la VLAN 1 */etc/sysconfig/network-scripts/ifcfg-eth0*.

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

En caso que los parámetros IPADDR, GATEWAY y NETMASK no estén bien configurados, se les debe asignar los valores correctos, luego se reinicia el equipo.

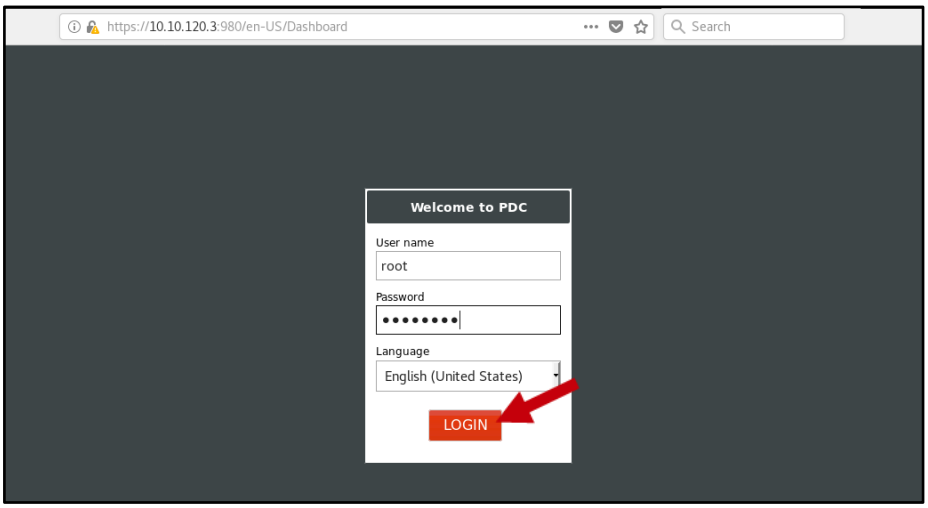
10. Ingresar vía web

Para ingresar a la administración web servidor Nethserver se debe realizar desde el navegador de otro equipo conectado a la red, ingresando la siguiente URL:

Continuación del apéndice 3.

https://<la dirección IP del servidor>:980

Instalación de Nethserver – Ingreso vía web



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Debe utilizar las credenciales ingresadas en la instalación.

Configuración inicial

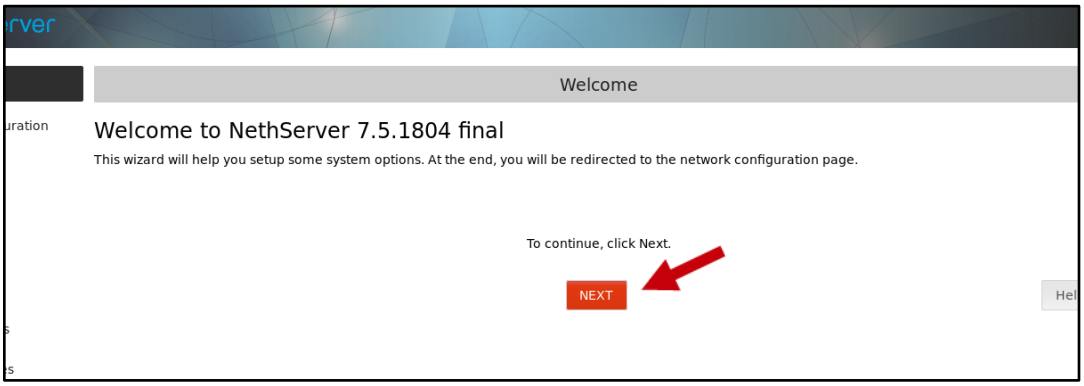
Una vez finalizado el proceso de instalación, al primer ingreso vía web se solicitará las primeras configuraciones del servidor. Los pasos a seguir son los siguientes:

1. Ingresar vía web

2. Pasar la ventana de inicio presionando ‘Next’:

Continuación del apéndice 3.

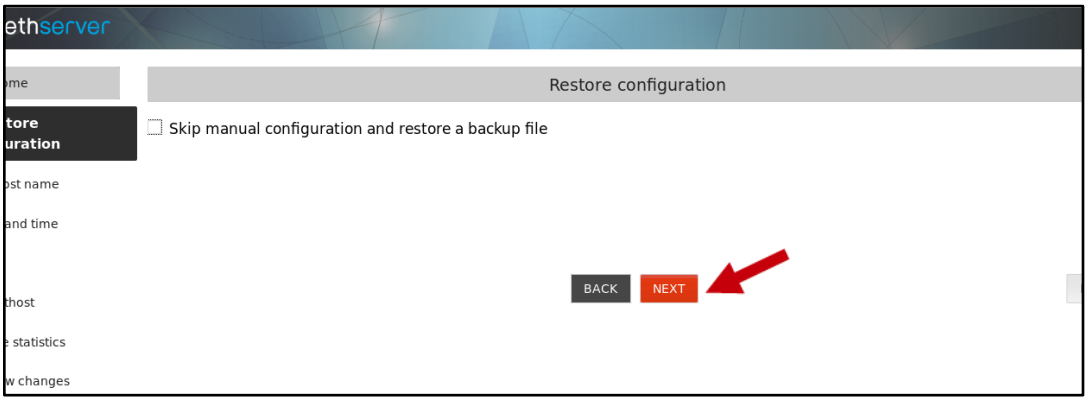
Configuración inicial de Nethserver – Ventana de inicio



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

3. Restaurar la configuración previa:

Configuración inicial de Nethserver – Ventana de restauración

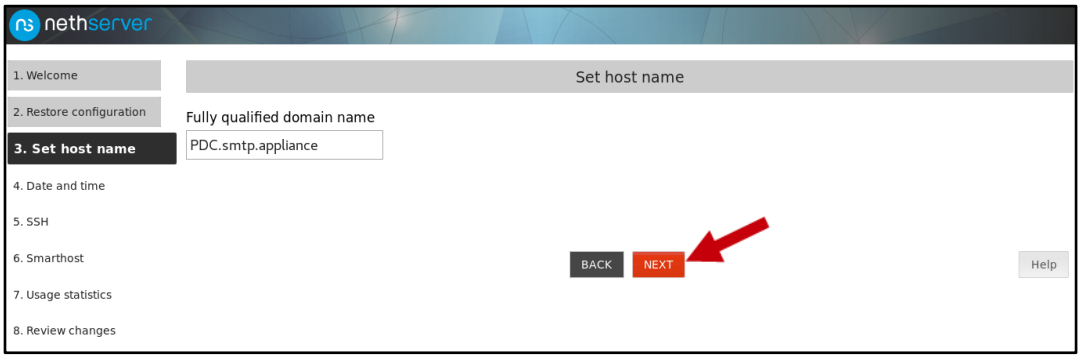


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Nethserver permite realizar una copia de respaldo a su configuración. En este paso se puede restaurar. En este caso solo se saltará presionando ‘Next’.

4. Ingresar un nombre de la PC en conjunto con un nombre de dominio completo, ‘FQDN’:

Configuración inicial de Nethserver – Ventana de FQDN

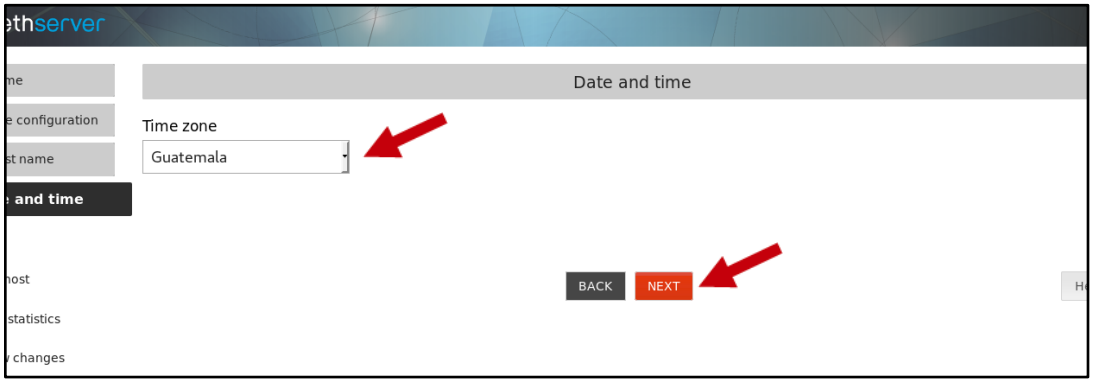


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

5. Seleccionar zona horaria:

Continuación del apéndice 3.

Configuración inicial de Nethserver – Ventana de zona horaria

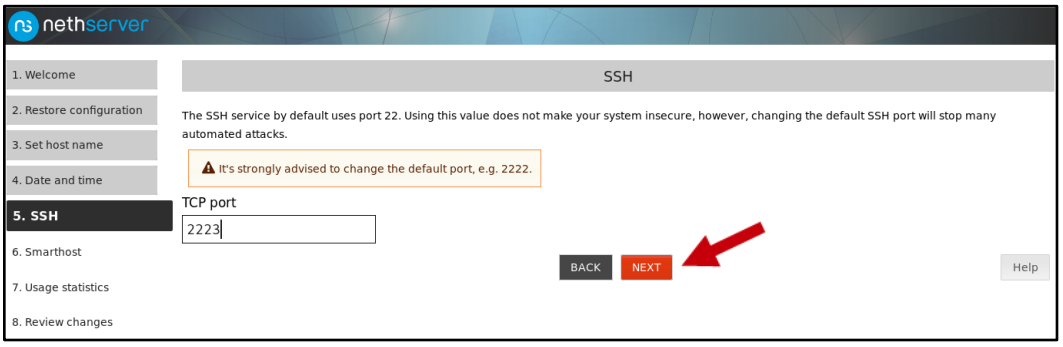


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Por defecto viene la zona horaria indicada en el proceso de instalación

6. Modificar el puerto para el ingreso vía SSH:

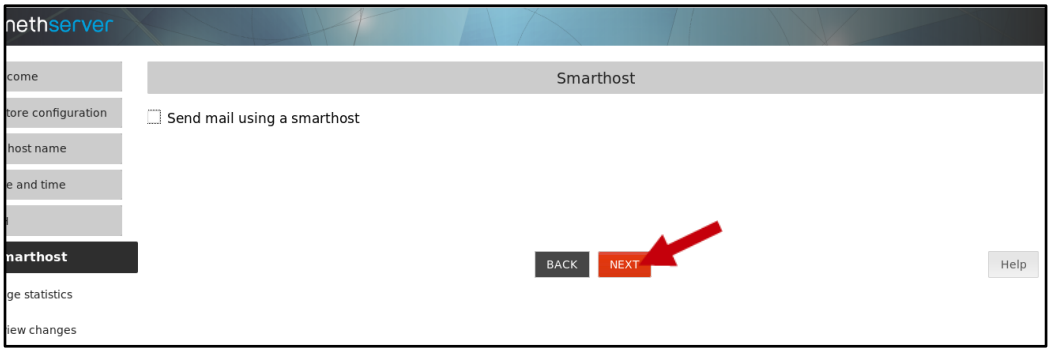
Configuración inicial de Nethserver – Ventana de SSH



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

7. Configurar de correo:

Configuración inicial de Nethserver – Ventana de correo

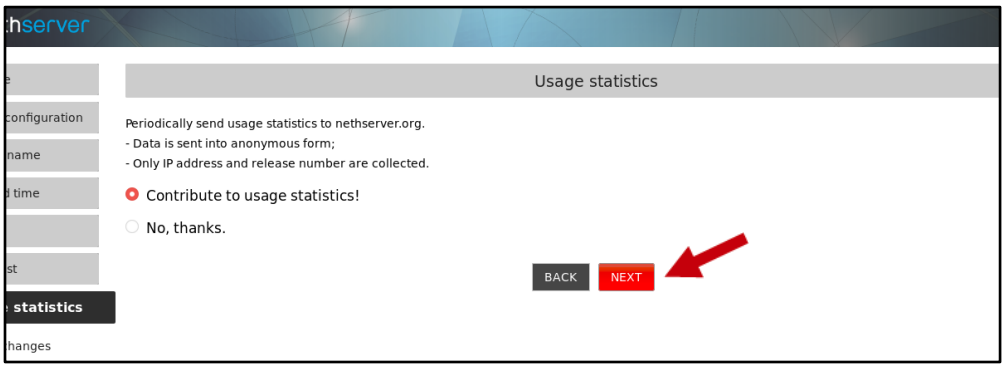


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6:

En esta sección se establece que servidor de correo se va a utilizar para enviar notificaciones. Para cuestiones de este documento se saltará este paso presionando 'Next'.

8. Contribuir al envío de estadísticas:

Configuración inicial de Nethserver – Ventana de estadísticas

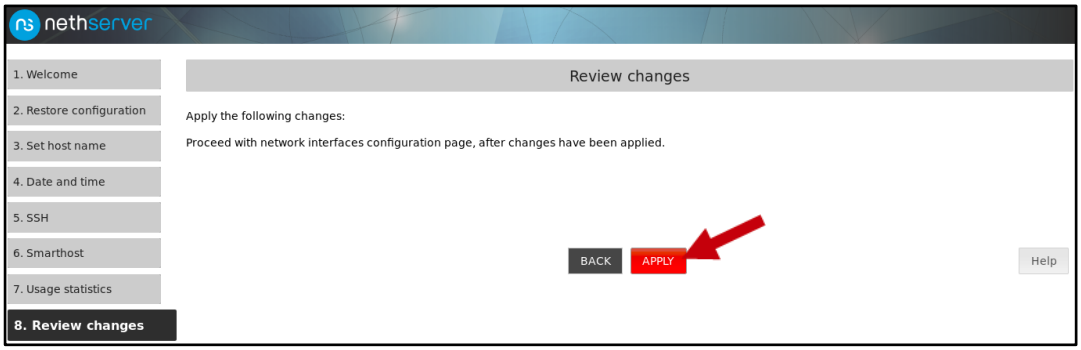


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Al contribuir al envío permite a las organizaciones *open source* realizar mejoras en sus productos.

9. Aplicar cambios:

Configuración inicial de Nethserver – Ventana de revisión



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Configuración de red

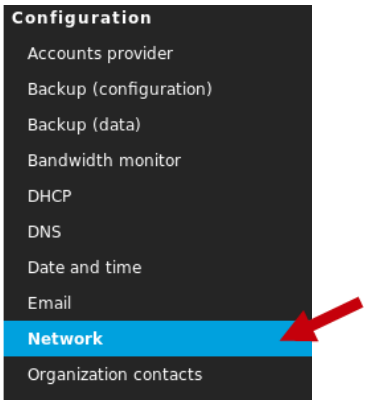
Nethserver define lar tarjetas de red por roles los cuales son: Internet, LAN, DMZ y Guests. Cada rol define su comportamiento. Si la tarjeta se define como Internet, la interfaz se utilizará para conectar a Internet, si se define como LAN, la interfaz se utilizará para proveer servicios dentro de una red local, si se define como DMZ, la interfaz solo permite conectar equipos que se encuentren dentro de esta red, si se define como Guests, es para pruebas.

En este caso se definirá la interfaz como LAN la que se encuentra conectada a la VLAN 10. La otra interfaz conectada a la VLAN 1 se definirá como Internet.

Continuación del apéndice 3.

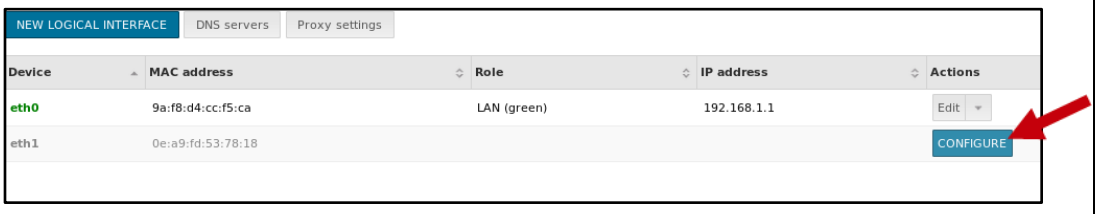
Las direcciones IP estáticas asignadas serán las 10.10.10.1 y 10.10.120.3 para las interfaces de LAN e Internet respectivamente. La máscara de subred será la 255.255.255.0.

Configuración de interfaz VLAN 10 Imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

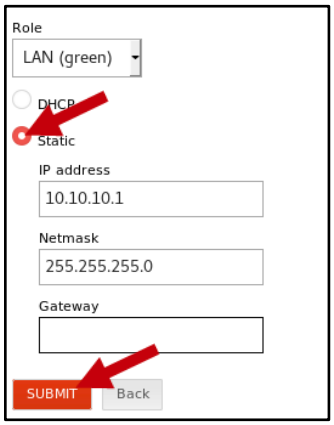
Configuración de interfaz VLAN 10 Imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

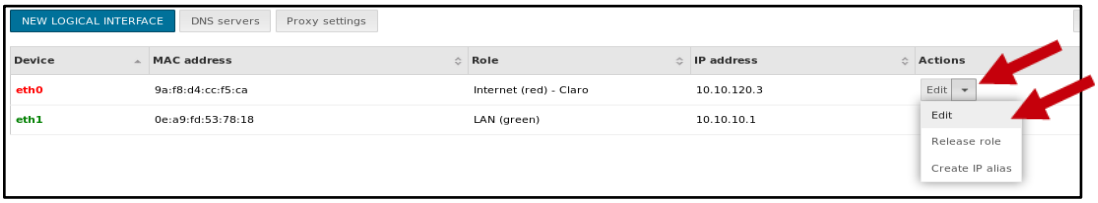
Configuración de interfaz VLAN 10 Imagen 3



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

La interfaz anterior será utilizada como *gateway* de la VLAN 10 por lo que no es requerido asignarle un *gateway*.

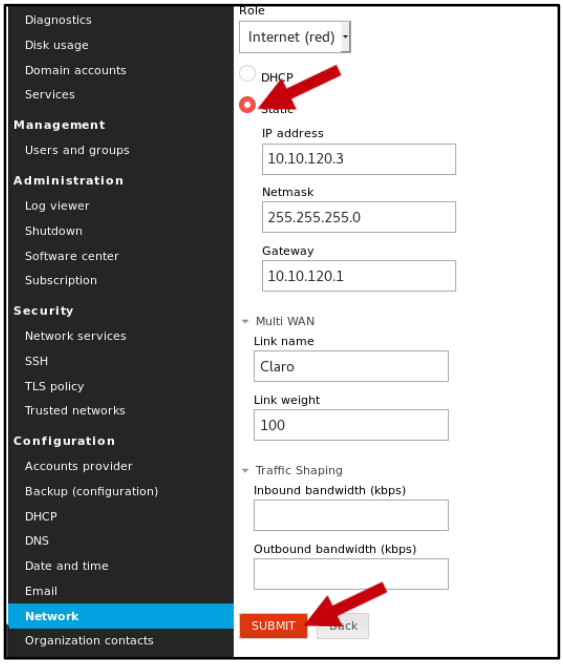
Configuración de interfaz VLAN 1 Imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Configuración de interfaz VLAN 1 Imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Configuración de DHCP

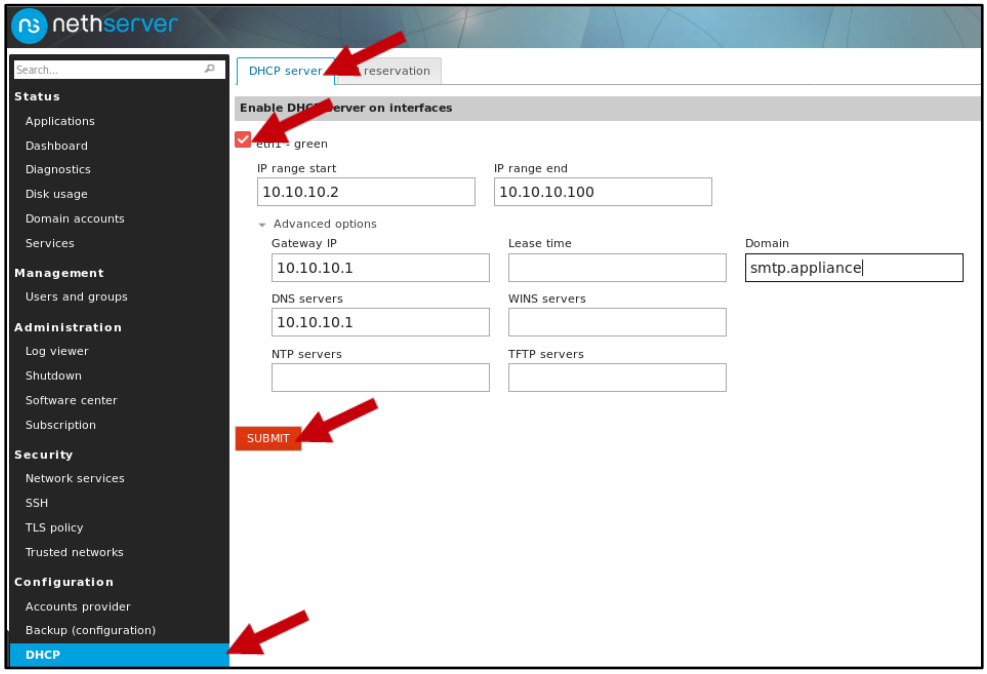
Uno de los servicios principales de Nethserver es el de DHCP en una LAN. En la configuración de DHCP se debe definir lo siguiente:

- **IP range start**: Dirección IP de inicio del rango de DHCP. Para este caso se comenzará en la 10.10.10.2.
- **IP range end**: Dirección IP final del rango de DHCP. Para este caso se finalizará en la 10.10.10.100.

- **Gateway IP**: *Gateway* a asignar. Para este caso será la dirección IP 10.10.10.1.
- **DNS**: Dirección IP del servidor DNS. Nethserver también actuará como DNS en la LAN.
- **Domain**: Se asignará como nombre de dominio al asignado en el paso 4 de la configuración inicial.

Los demás parámetros son opcionales.

Configuración de DHCP



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

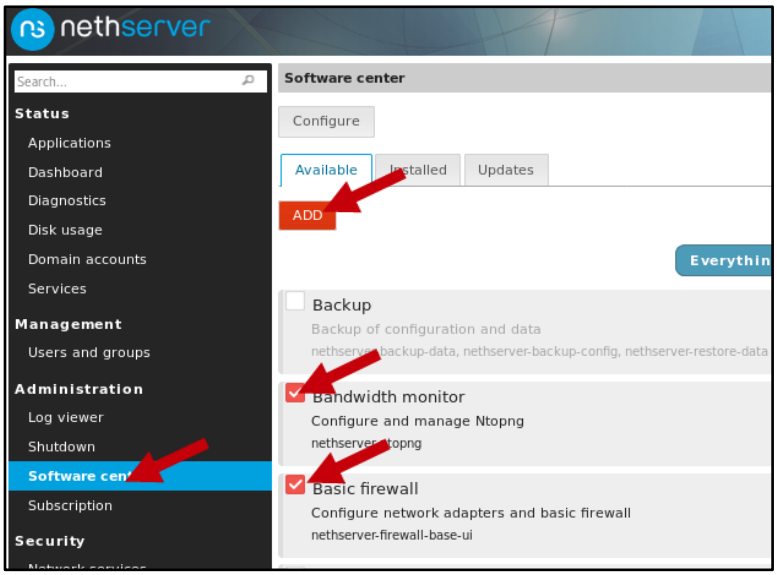
Firewall con reservación de dirección IP

Una de los servicios básicos de Nethserver es el de *firewall*. El *firewall* se instalará y configurará para que requiera de una reservación de dirección IP por dirección MAC, la reservación por IP es parte del servicio DHCP, para que el equipo que reciba una dirección IP pueda comunicarse con el internet u otros equipos dentro de la LAN. El motivo de agregar esta característica es de tener un mayor control de los equipos dentro de la LAN.

Para la instalación y configuración del *firewall* se hace lo siguiente:

i. Instalar los módulos ‘Bandwidth Monitor’ y ‘Basic Firewall’:

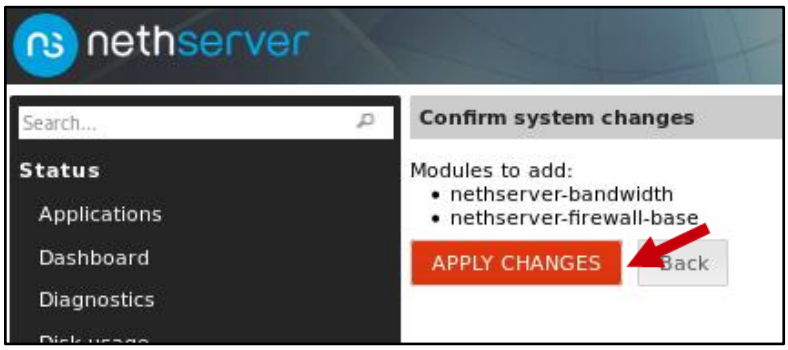
***Firewall* – Instalación imagen 1**



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

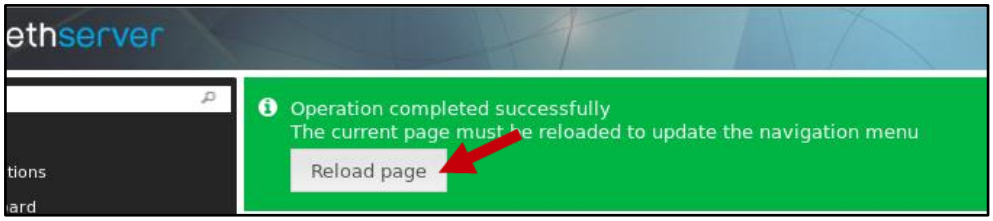
Continuación del apéndice 3.

Firewall – Instalación imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Firewall – Instalación imagen 3



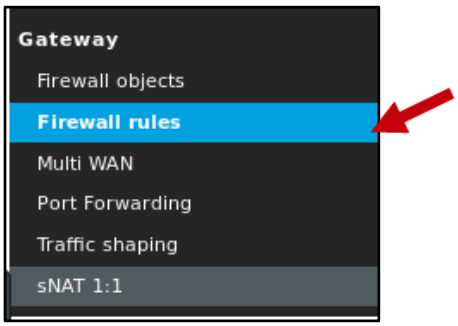
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Las dos figuras anteriores se repiten a lo largo de cualquier instalación de un paquete, por lo que serán omitidas.

1. Configurar las reglas de *firewall*:

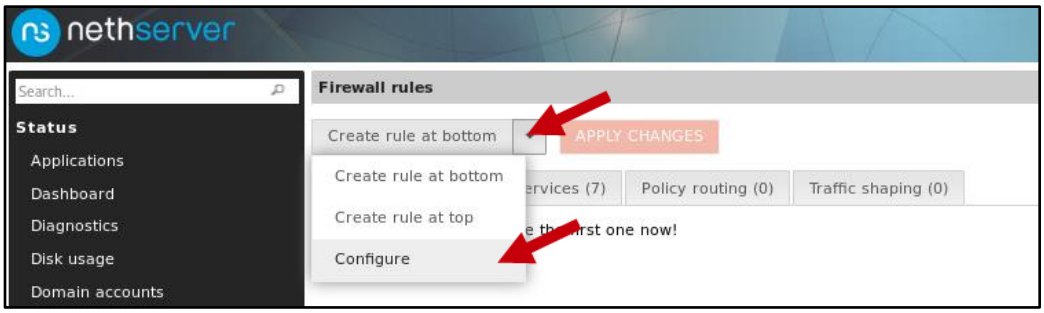
Continuación del apéndice 3.

Firewall – Configuración imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

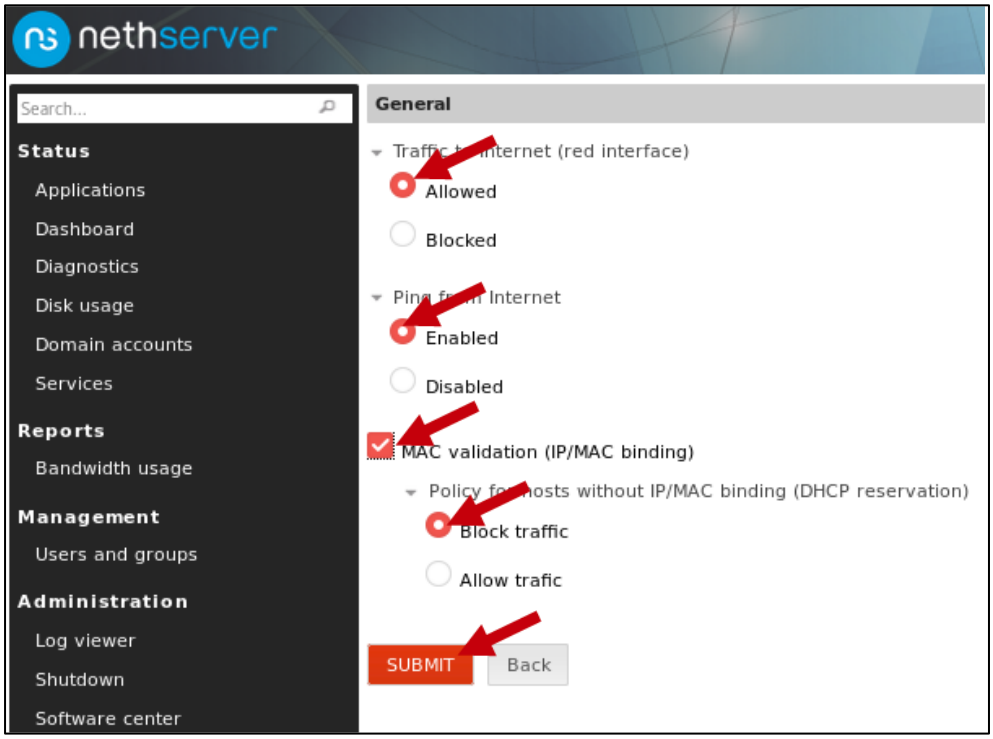
Firewall – Configuración imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Firewall – Configuración imagen 3



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

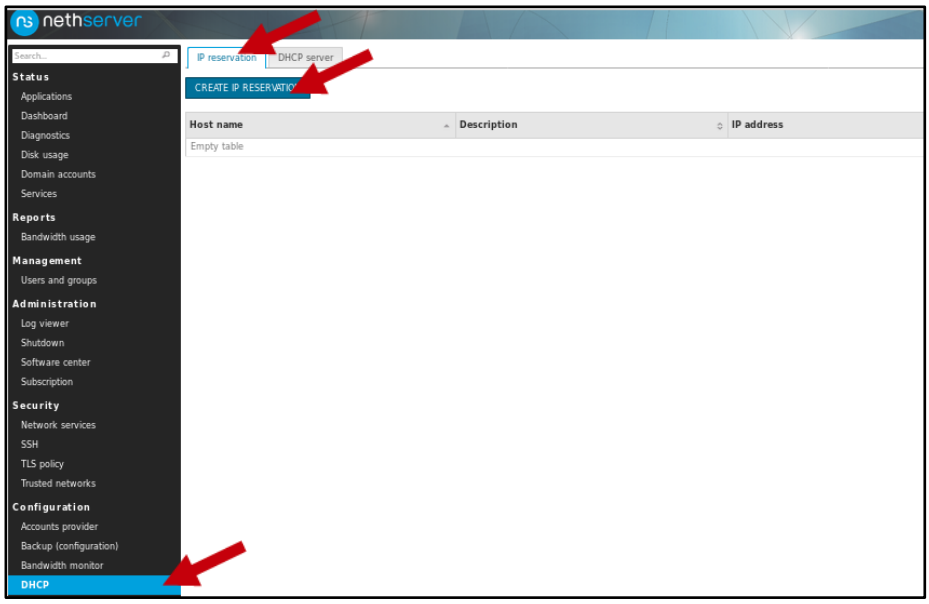
Las opciones mostradas en la figura anterior son:

- ***Traffic to Internet***: Esta opción indica si está permitido el tráfico desde la LAN hacia Internet. Se selecciona ‘allowed’.
- ***Ping from Internet***: Indica si se habilita el reconocimiento de equipos dentro de la LAN, por parte de equipos en Internet que estén utilizando la herramienta ‘ping’.

- ***MAC validation***: Esta opción indica permitir o denegar el tráfico de equipos que se encuentren con una dirección IP reservada por dirección MAC. Esta opción se habilita y se selecciona la opción 'Block Traffic' para bloquear el tráfico de equipos con IP no reservada.

2. Reservar direcciones IP por direcciones MAC:

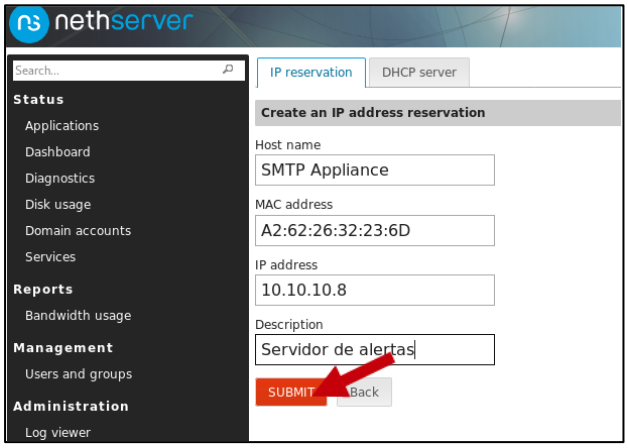
Firewall – Reservación de IP por MAC imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Firewall – Reservación de IP por MAC imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Al realizar la reservación por DHCP el equipo ya tiene acceso a Internet y conectarse a los equipos dentro de la LAN. Además en la reservación de dirección IP, los equipos toman de nombre el asignado en el parámetro ‘Hostname’ con lo que pueden ser identificados dentro de la LAN.

Configuración de filtrado web con antivirus

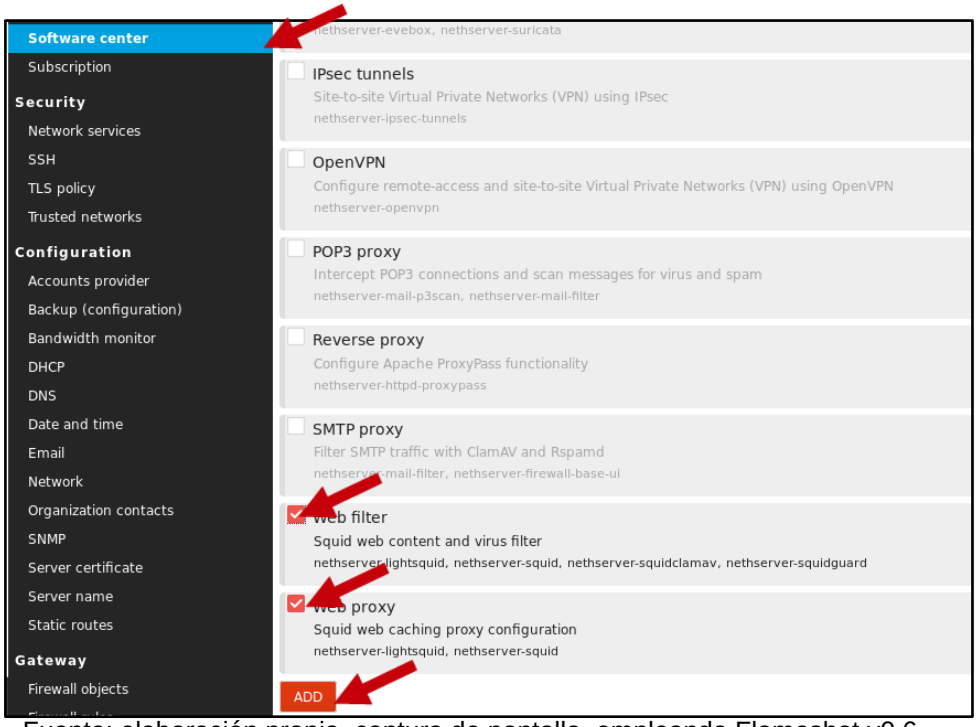
Nethserver también es capaz de brindar los servicios de filtrado de contenido, con chequeo de virus, para conexiones de tipo http y https utilizados en la mayoría de transacciones hacia internet.

Para su configuración se seguirán los siguientes pasos:

Continuación del apéndice 3.

1. Instalación de los módulos ‘Web Filter’ y ‘Web Proxy’

Proxy y filtro web – Instalación



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

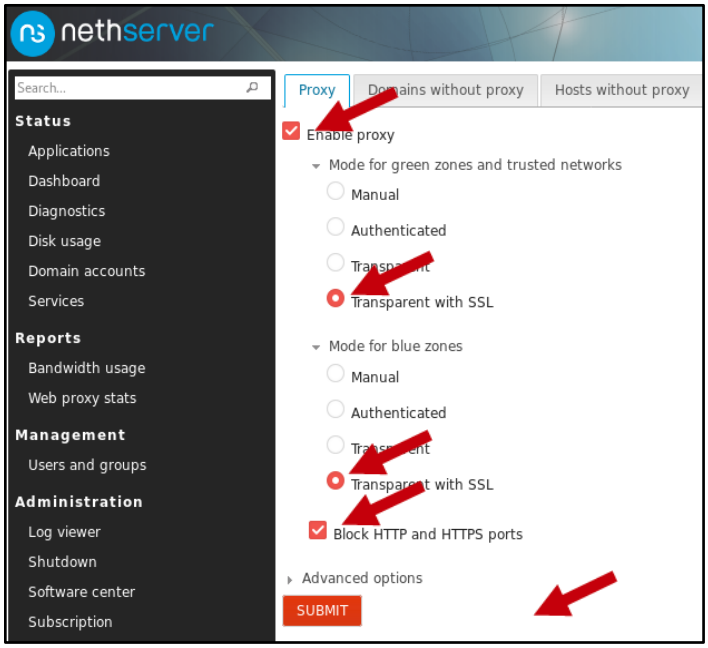
2. Activación de servicio de Proxy

Continuación del apéndice 3.

Proxy y filtro web – Activación de proxy Imagen 1

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Proxy y filtro web – Activación de proxy Imagen 2

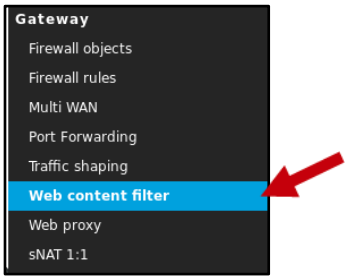


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Estas opciones permiten controlar todo el tráfico http y https para obligarlo a pasar por este equipo. Además solo permitirá tráfico con certificado SSL válido.

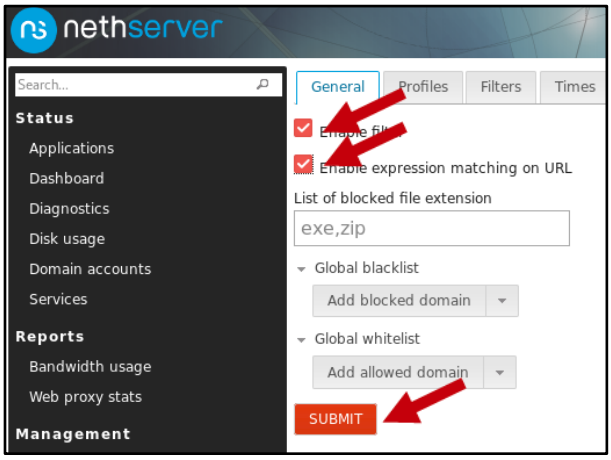
3. Activar filtrado de contenido web.

Proxy y filtro web – Activación de filtro imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Proxy y filtro web – Activación de filtro Imagen 2

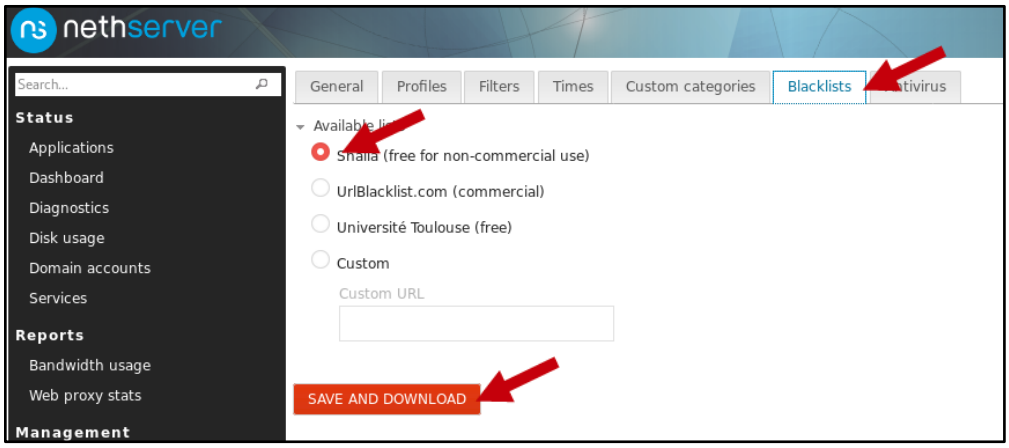


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

4. Descargar lista de sitios web a filtrar.

Proxy y filtro web – Descarga de *blacklist*



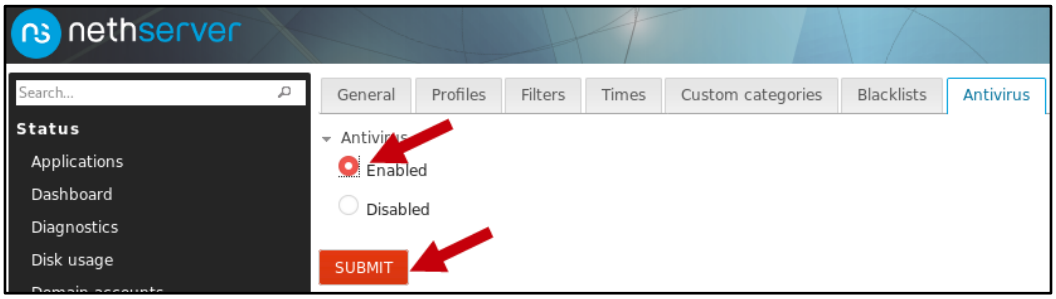
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Nethserver separa los sitios web por categorías y cada *blacklist* tiene ciertos dominios restringidos asignados a las categorías, por lo que el comportamiento del filtro de sitios web depende de que *blacklist* se utilice.

5. Activar antivirus.

Continuación del apéndice 3.

Proxy y filtro web – Antivirus

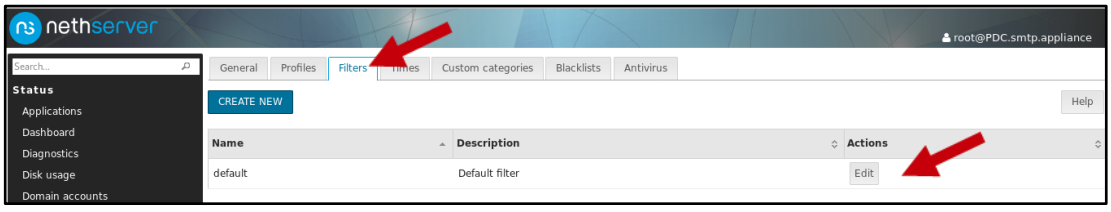


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Nethserver es capaz de buscar cualquier tipo de virus o contenido malicioso de los sitios web.

6. Configurar filtro.

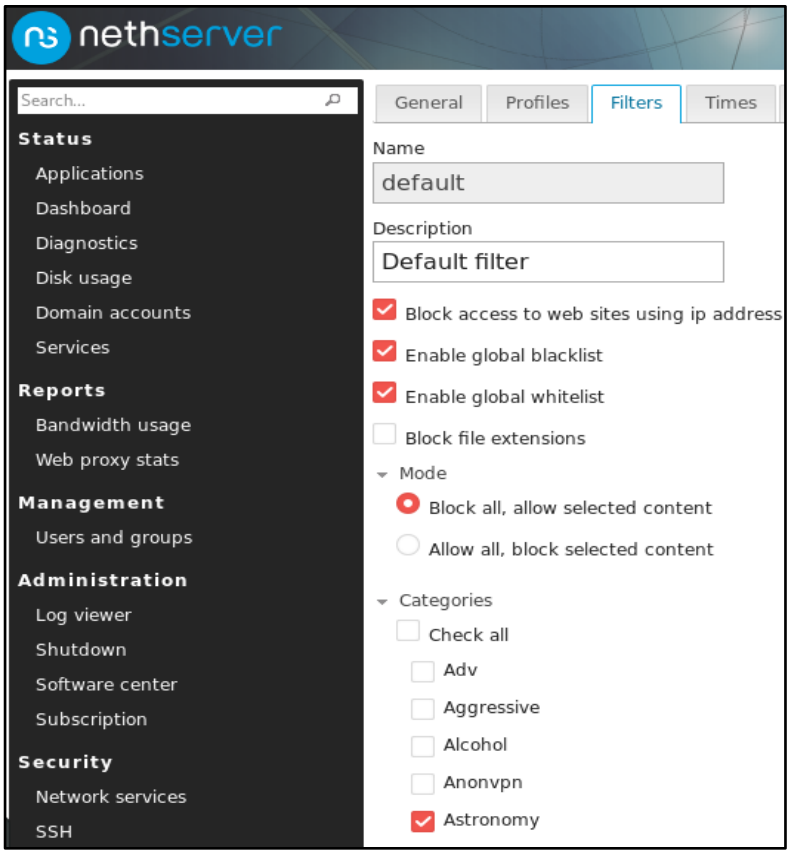
Proxy y filtro web – Configuración de filtro imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Proxy y filtro web – Configuración de filtro imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

En esta sección se seleccionan las categorías de páginas web a filtrar. Por buenas prácticas de seguridad, se recomienda utilizar el filtro en modo ‘Block all, allow selected content’, el cual solo permite el tráfico según la categoría seleccionada.

Continuación del apéndice 3.

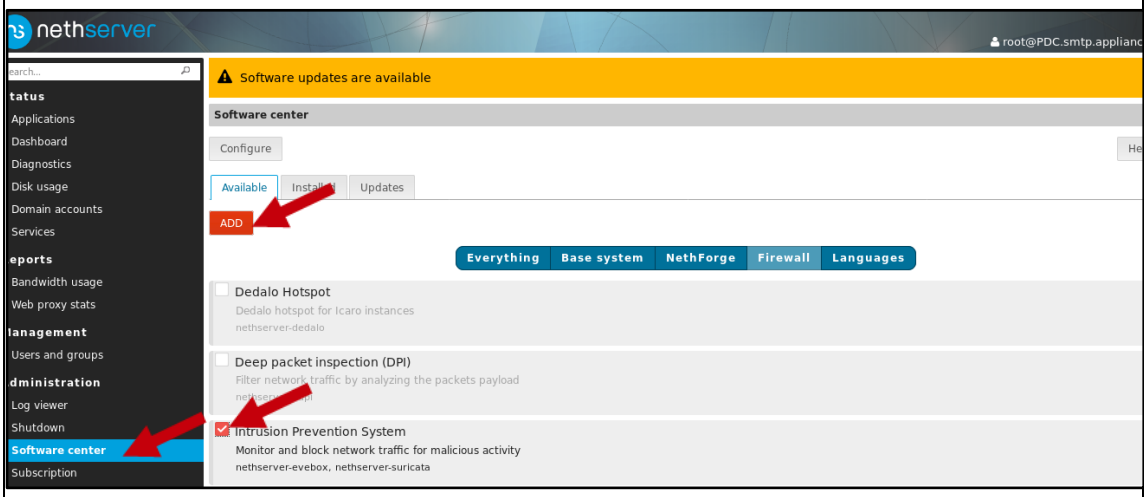
Configuración de IPS

IPS, *Intruision Prevention System*, es uno de los servicios que permite añadir una capa de seguridad a los equipos dentro de la LAN. IPS permite detectar actividades maliciosas que puedan afectar a los equipos.

Los pasos de configuración e instalación del IPS son:

1. Instalación del módulo IPS

IPS – Instalación

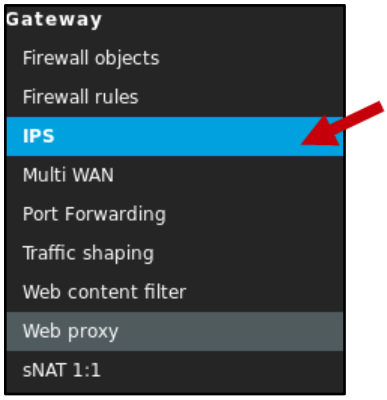


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

2. Configuración de IPS

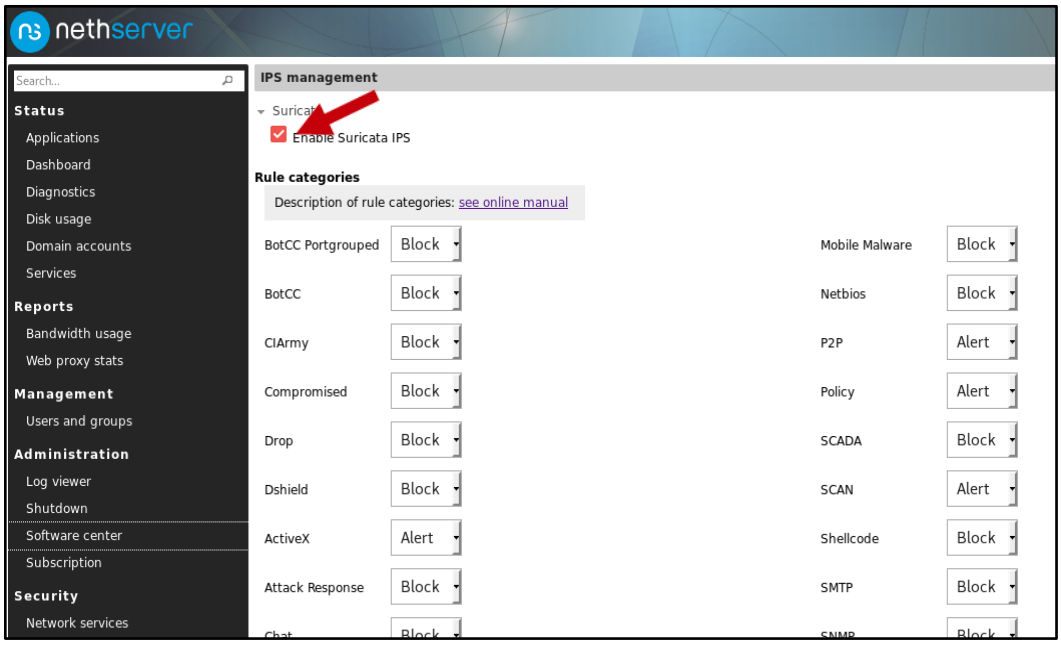
Continuación del apéndice 3.

IPS – Configuración imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

IPS – Configuración imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 3.

Para este módulo es recomendable revisar el manual para establecer que tráfico es conveniente bloquear. Además se pueden revisar las alertas generadas en la sección de *Applications* > *Evebox*.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 4. **FreePBX**

FreePBX es una solución de tipo empresarial para centrales telefónicas. La mayoría de funciones de FreePBX son de tipo *open source*, sin embargo, existen funciones extra, las que en esta ocasión no son necesarias, que requieren efectuar un pago para su uso.

Otras soluciones similares están: FusionPBX, MIRTA PBX, Yate, 3CX, etc.

Instalación

Al instalar FreePBX 14 el equipo poseerá las siguientes características:

- CentOS 7 x64 como sistema operativo.
- Asterisk 13 o 15 como base para la central telefónica.
- Instalación y configuración del interfaz web HTML5, para la administración del servidor.

Para instalar FreePBX se usarán los siguientes pasos:

1. Descargar la imagen ISO del sitio oficial y cargarlo al servidor Proxmox.

2. Crear VM con una tarjeta red que pertenezca a la VLAN 10.

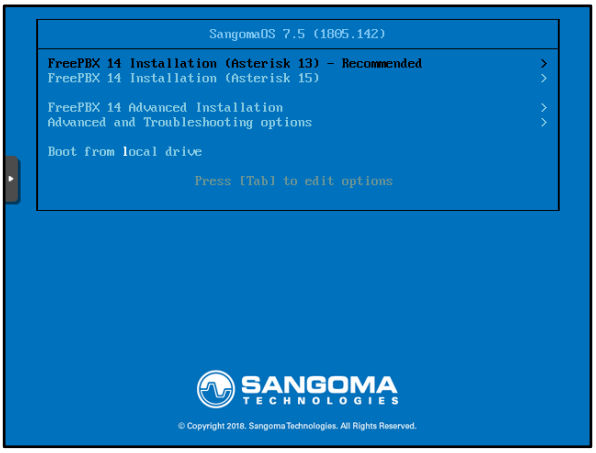
3. Realizar reservación de dirección IP por dirección MAC en Nethserver.

4. Iniciar VM con la imagen ISO.

Continuación del apéndice 4.

5. Seleccionar instalación de FreePBX 14 con Asterisk 13.

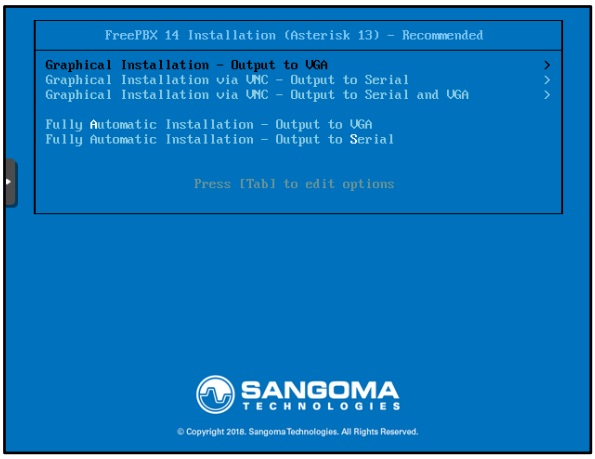
Instalación de FreePBX – Pantalla de inicio



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

6. Seleccionar instalación gráfica a VGA.

Instalación de FreePBX – Modo de instalación

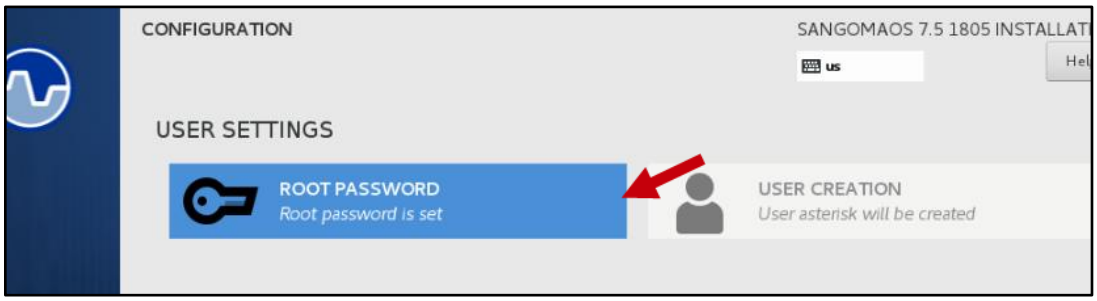


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 4.

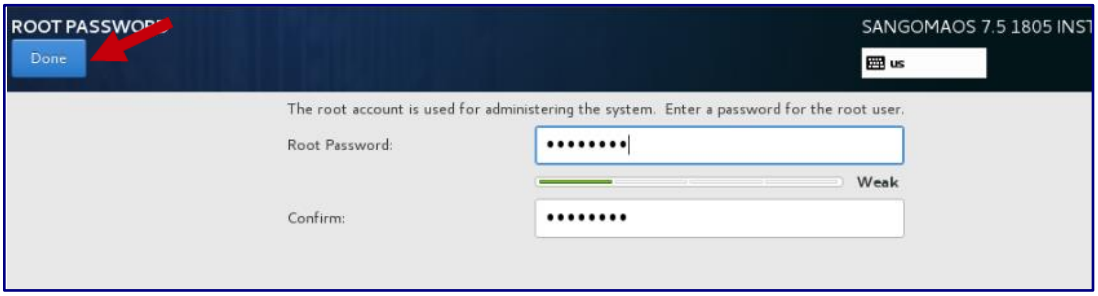
7. Asignar contraseña a usuario ‘root’.

Instalación de FreePBX – Contraseña de root imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación de FreePBX – Contraseña de root imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Después de la instalación aparecerá la opción para reiniciar el equipo.

8. Ingresar vía web

Para ingresar a la administración web servidor FreePBX se debe entrar desde el navegador de otro equipo conectado a la red ingresando la siguiente URL:

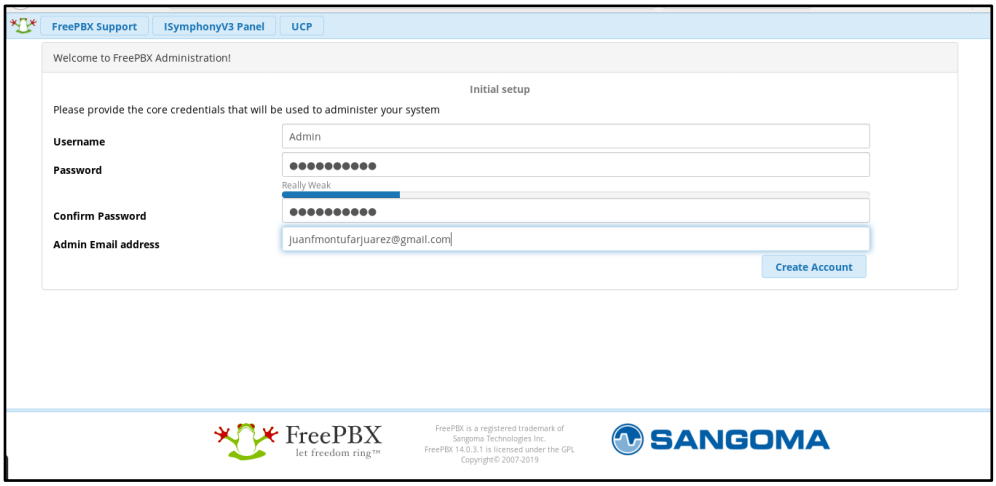
http://<la dirección IP del servidor>

Se recomienda instalar una VM de escritorio conectado a la VLAN 10 para proceder con esta configuración o bien utilizar la herramienta de *Port Forwarding* de Nethserver.

Configuración inicial

Al ingresar por primera vez se le solicitará ingresar un usuario para administrar la interfaz web de FreePBX.

Configuración inicial de FreePBX – Usuario administrador



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

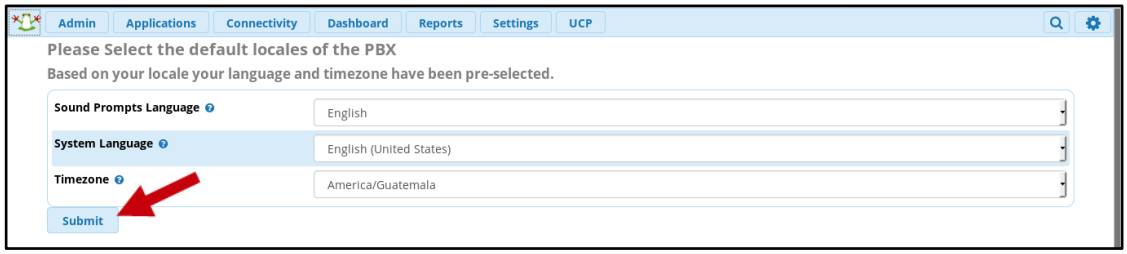
Después de ello se observará la ventana de inicio e ingresar con las credenciales usadas en el paso anterior.

Configuración inicial de FreePBX – Ingreso a administración

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

A continuación se mostrará la ventana para selección del lenguaje y zona horaria.

Configuración inicial de FreePBX – Lenguaje y zona horaria

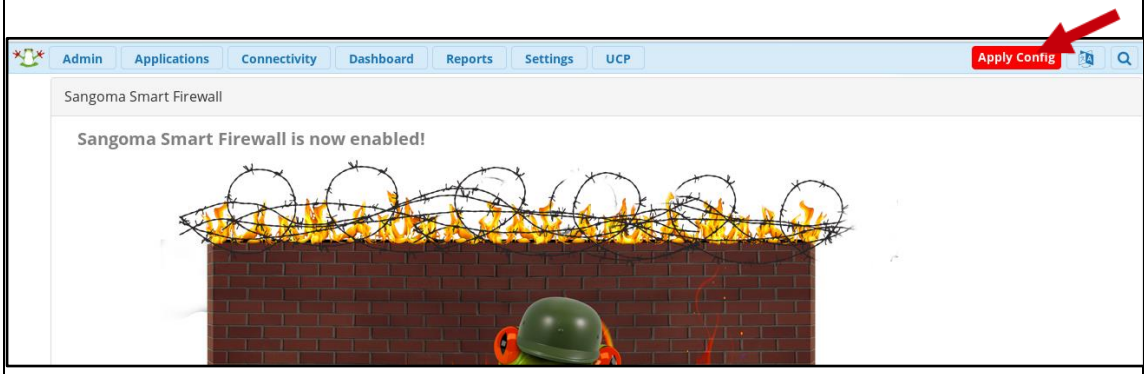


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 4.

Por defecto FreePBX posee un propio *firewall* el cual debe ser habilitado.

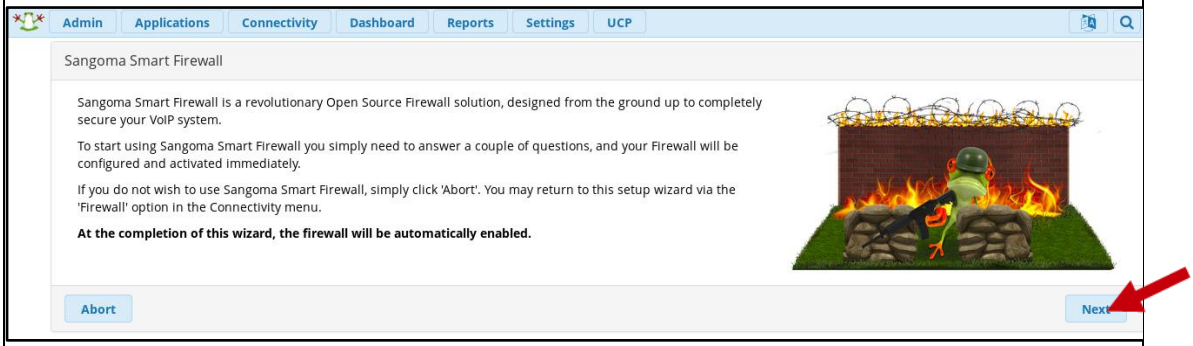
Configuración inicial de FreePBX – *Firewall* imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Las ventanas que aparecen después realizan preguntas simples de Si o No acerca de la configuración del *firewall* de FreePBX. Para esta ocasión se aceptarán todas las preguntas.

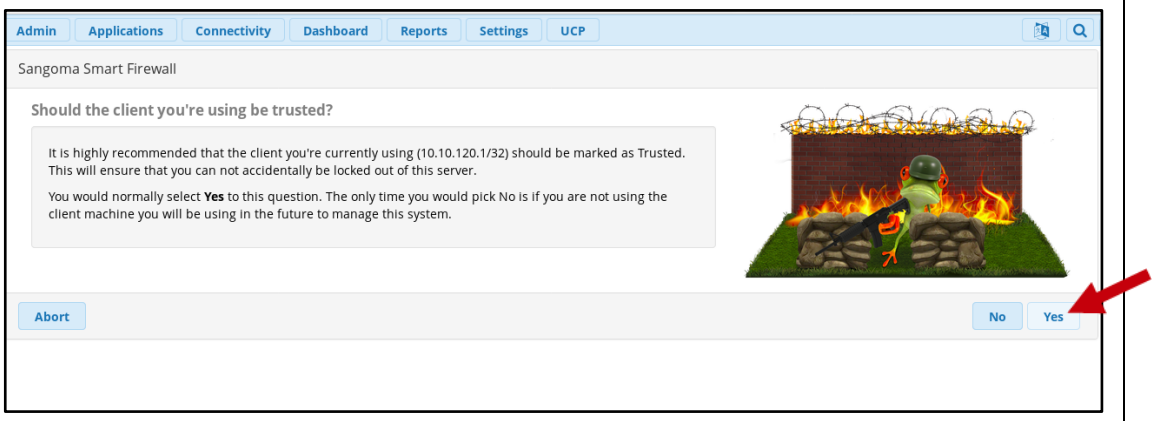
Configuración inicial de FreePBX – *Firewall* imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

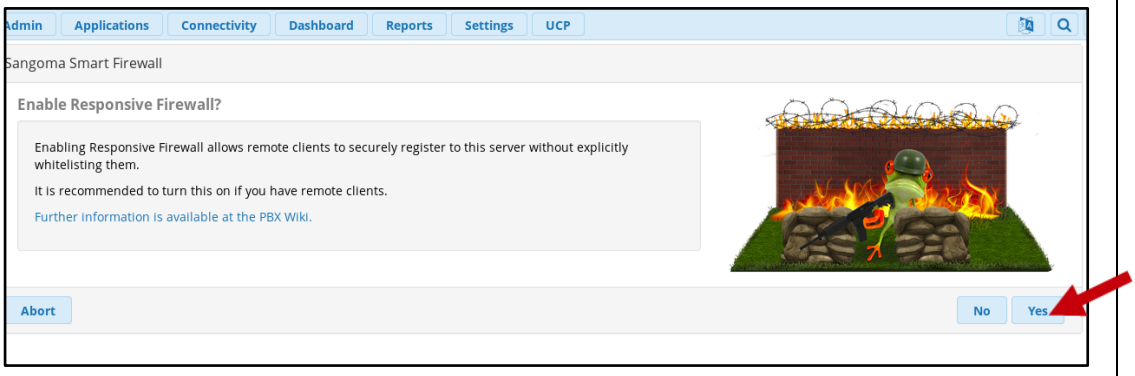
Continuación del apéndice 4.

Configuración inicial de FreePBX – *Firewall* imagen 3



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

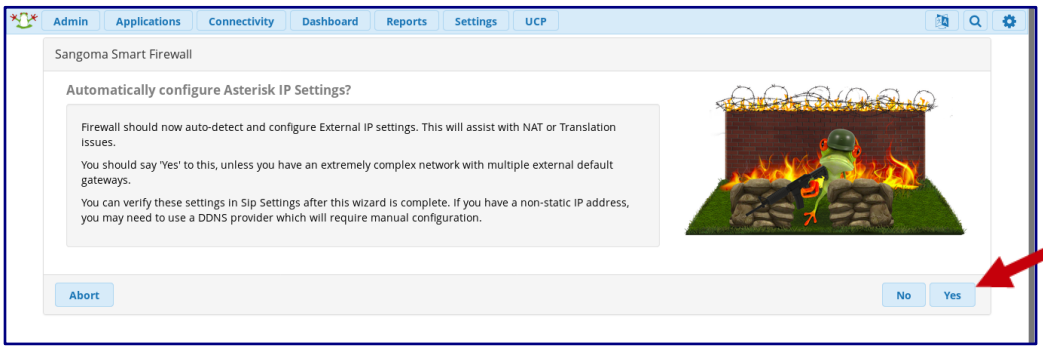
Configuración inicial de FreePBX – *Firewall* imagen 4



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 4.

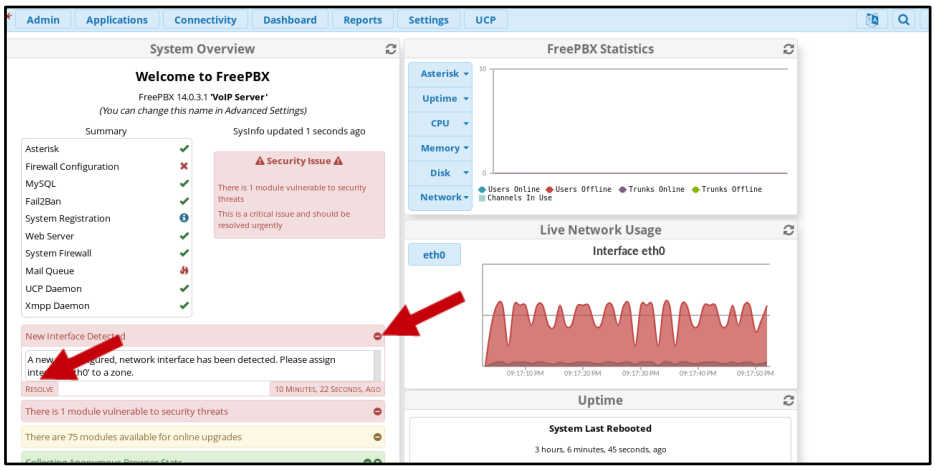
Configuración inicial de FreePBX – *Firewall* imagen 5



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Una vez finalizada se podrá acceder a la ventana de inicio. Después de ello, para usar el *firewall* de FreePBX, la interfaz del servidor debe ser asignada como de tipo ‘Internet’.

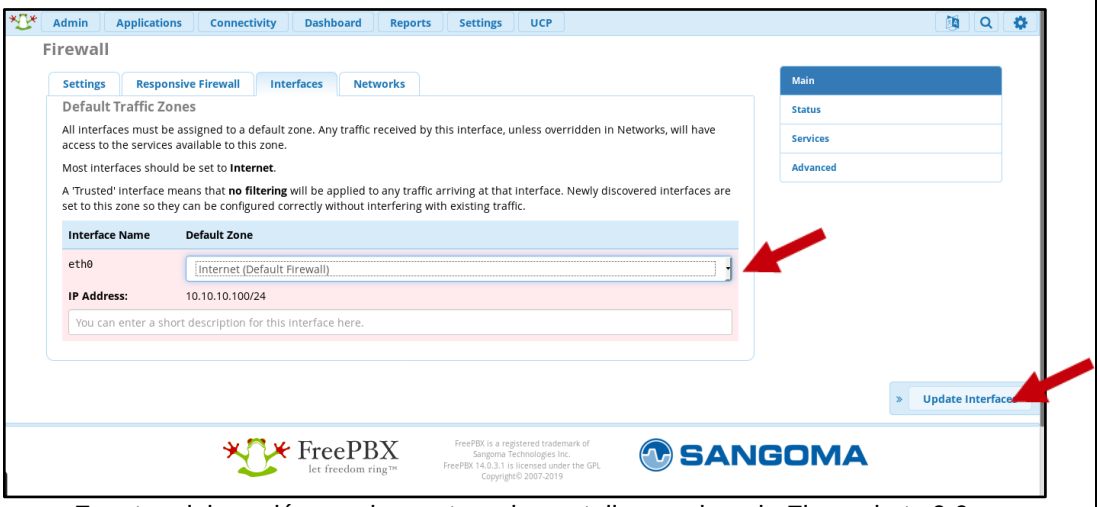
Configuración inicial de FreePBX – Interfaz imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 4.

Configuración inicial de FreePBX – Interfaz imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Extensiones

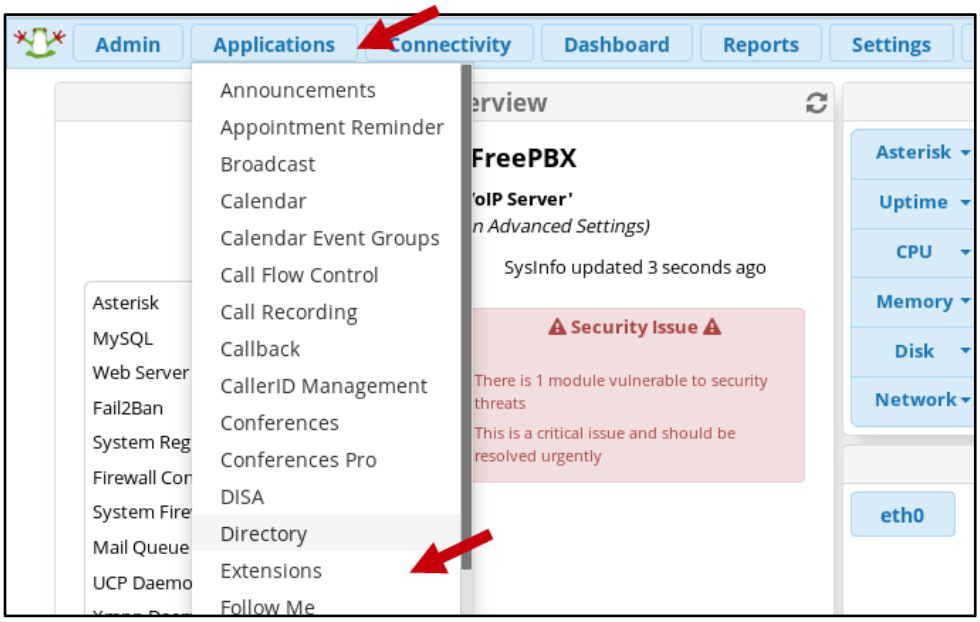
Para que se puedan comunicar los equipos de VoIP, es requerido identificarlos mediante un número de extensión. Si se desea efectuar una llamada desde un teléfono VoIP a otro, en las aplicaciones se requiere marcar el número de extensión del teléfono VoIP deseado.

Para asignar números de extensión a los equipos en FreePBX se realiza lo siguiente:

1. Ingresar a interfaz web.

2. Ingresar a *Applications* > *Extensions*.

Extensiones – Paso inicial



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

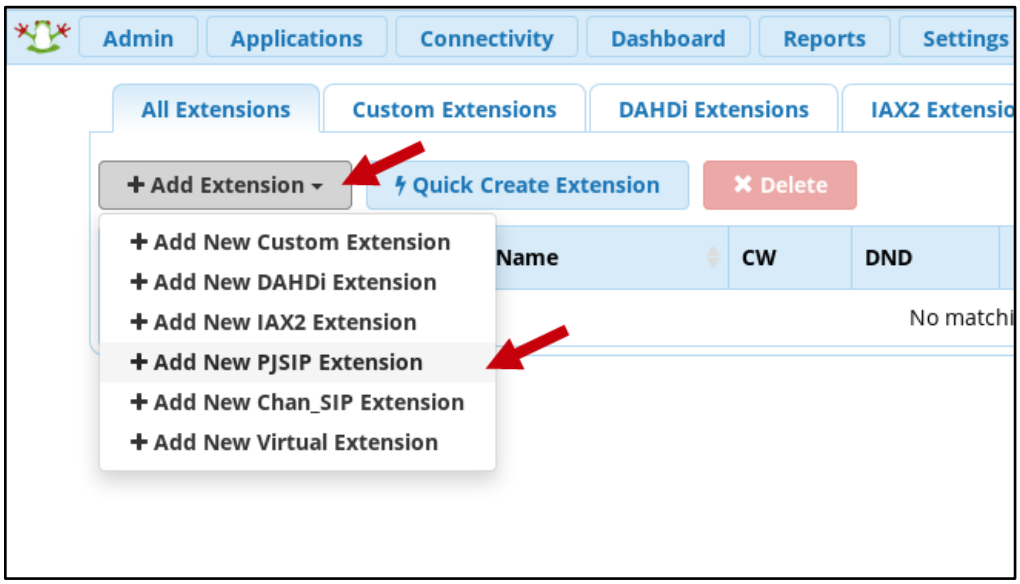
3. Crear una extensión de tipo PJSIP.

FreePBX maneja los protocolos DAHDi, IAX2, PJSIP que es la nueva versión de SIP y Chan_SIP que es la versión antigua de SIP, para que equipos VoIP se puedan comunicarse. En este documento se utilizará el protocolo PJSIP el cual utiliza por defecto el puerto 5060.

Para que un equipo pueda realizar llamadas, es necesario que este se registre en la central telefónica mediante un proceso de autenticación que requiere de un usuario, el cual será el número de extensión asignado, y una contraseña.

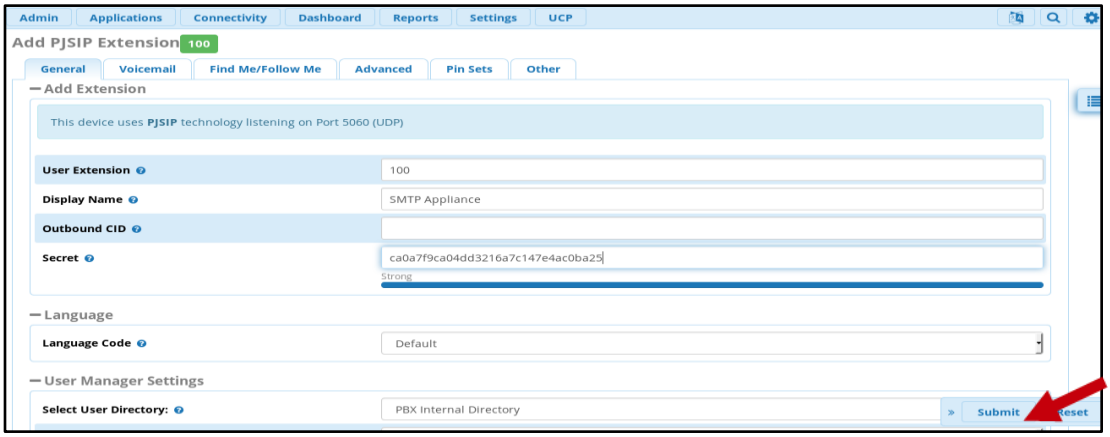
Continuación del apéndice 4.

Extensiones – Asignación imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Extensiones – Asignación imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

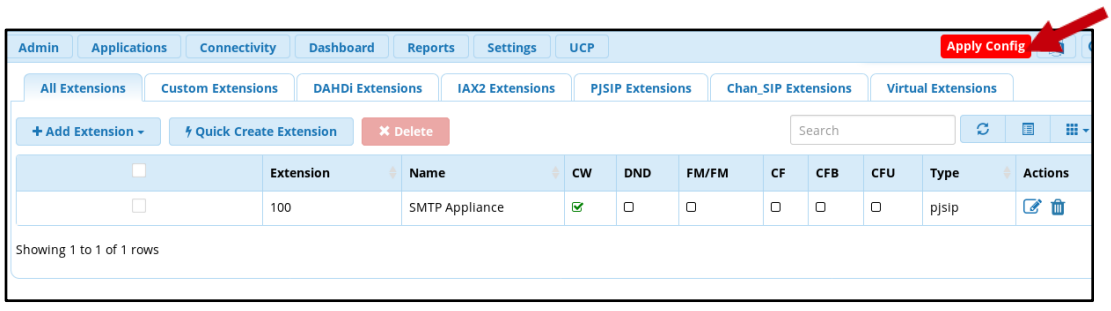
Los elementos relevantes de la imagen anterior son:

- ***User Extension*:** Es la extensión asignada.
- ***Display Name*:** Es el nombre con el cual otro equipo lo puede identificar. Este parámetro no es tan relevante como los otros.
- ***Secret*:** es la contraseña del usuario.

En la configuración de los otros equipos son requeridos los datos de *User Extension* que se usa como parámetro *username*, *Secret* que se usa como parámetro *password*, dirección IP de la central telefónica utilizada como *domain* o *outbound proxy* y puerto de SIP, aunque este último viene por defecto en varios equipos VoIP el uso del puerto 5060 para SIP por lo que solo es necesario ingresarlo a la configuración de no utilizar el puerto por defecto.

4. Aplicar cambios.

Extensiones – Aplicar cambios



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 4.

Conexión a VPN

Para conectar este servidor a una VPN utilizando el protocolo OpenVPN, se realizan los siguientes pasos:

1. Abrir una terminal del servidor.

2. Descargar y transferir al servidor un archivo de tipo ‘ovpn’. Los archivos gratuitos pueden ser descargados de la página de VPNBook.

3. Correr el siguiente comando:

   $ openvpn <nombre del archivo tipo ‘ovpn’>

Dentro del servidor se creará una interfaz de prefijo ‘tun’ la cual se utiliza para conectarse a la red VPN gratuita.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 5. **SMTP Appliance**

SMTP Appliance es el servidor de generación de alertas. Dicho servidor, como se mencionó en los capítulos, cuenta con una serie de archivos creados en el lenguaje de programación Python 2.7 que permiten su funcionamiento haciéndolo un equipo único en el mercado.

El sistema operativo utilizado es igual que el que posee el servidor Wordpress el cual es Ubuntu Server 18.04 x64.

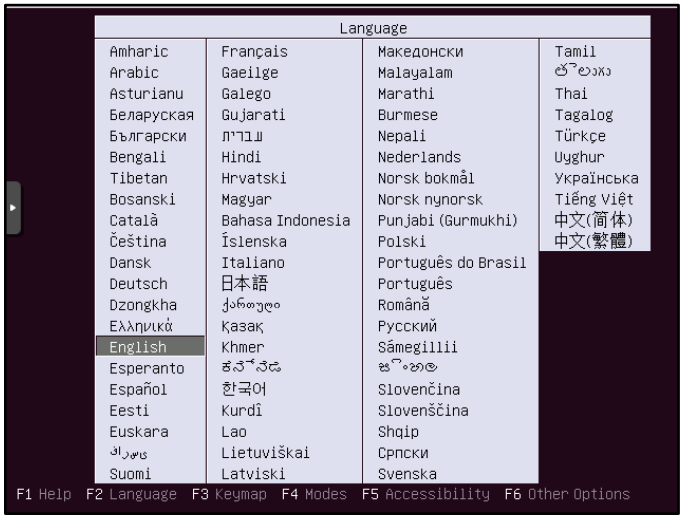
Instalación

Para instalar el SMTP Appliance se usarán los siguientes pasos:

1. Descargar la imagen ISO del sitio oficial y cargarlo al servidor Proxmox.
2. Crear VM con una tarjeta red que pertenezca a la VLAN 10, unidad USB y mapeo de audio.
3. Realizar reservación de dirección IP por dirección MAC en Nethserver.
4. Crear una extensión en FreePBX para este equipo.
5. Iniciar VM con la imagen ISO.
6. Seleccionar lenguaje de inicio.

Continuación del apéndice 5.

Instalación SMTP Appliance – Lenguaje de inicio



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

7. Seleccionar ‘Instalar Ubuntu Server’.

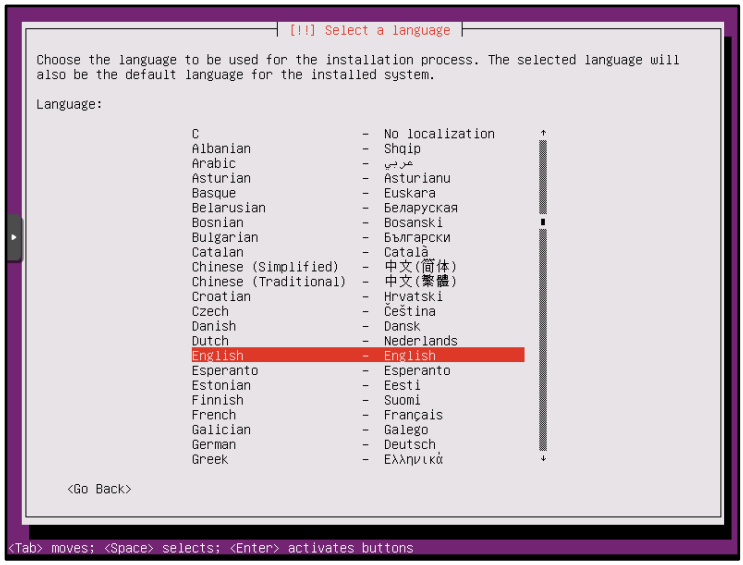
Instalación SMTP Appliance – Modo de instalación

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 5.

8. Seleccionar lenguaje de instalación.

Instalación SMTP Appliance – Lenguaje de instalación

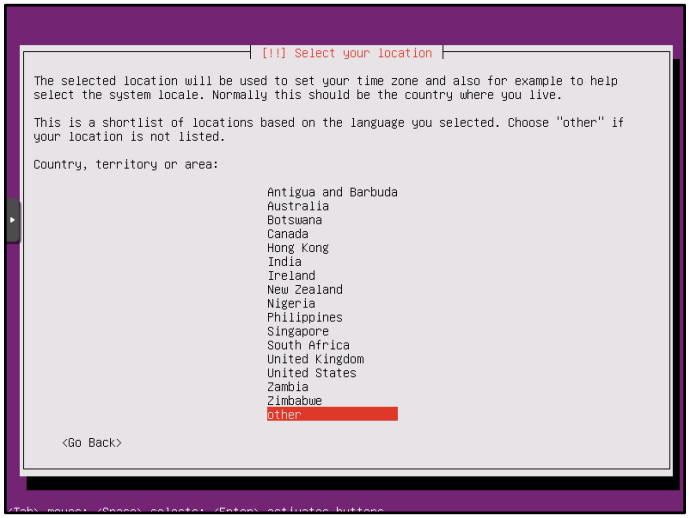


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

9. Seleccionar ubicación.

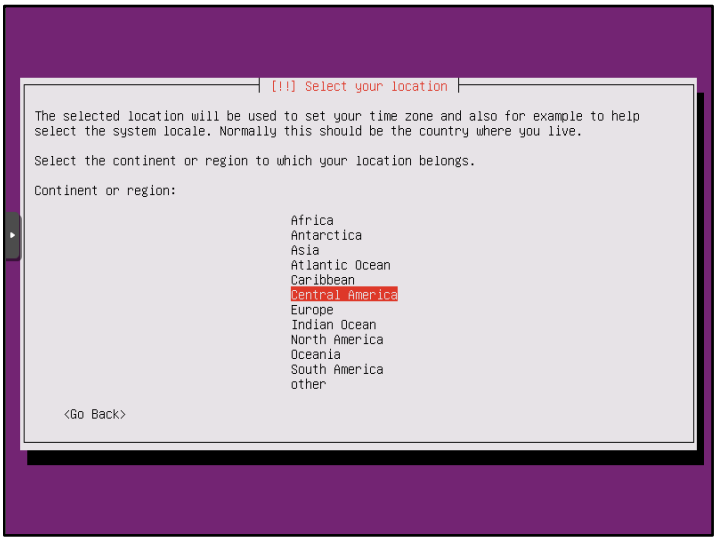
Continuación del apéndice 5.

Instalación SMTP Appliance – Ubicación imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

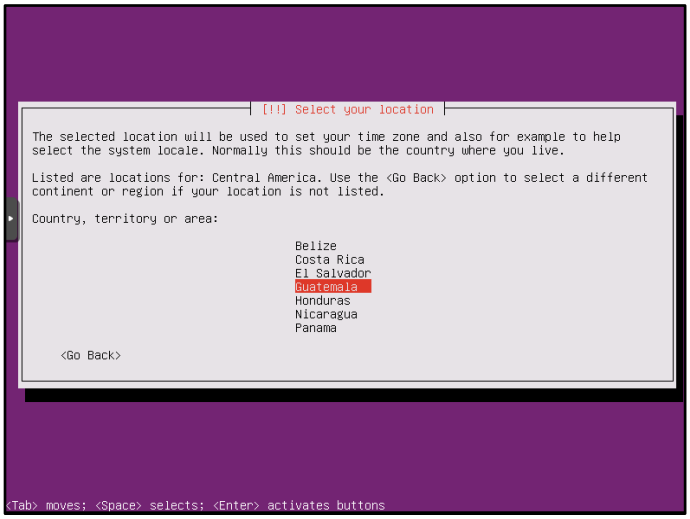
Instalación SMTP Appliance – Ubicación imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

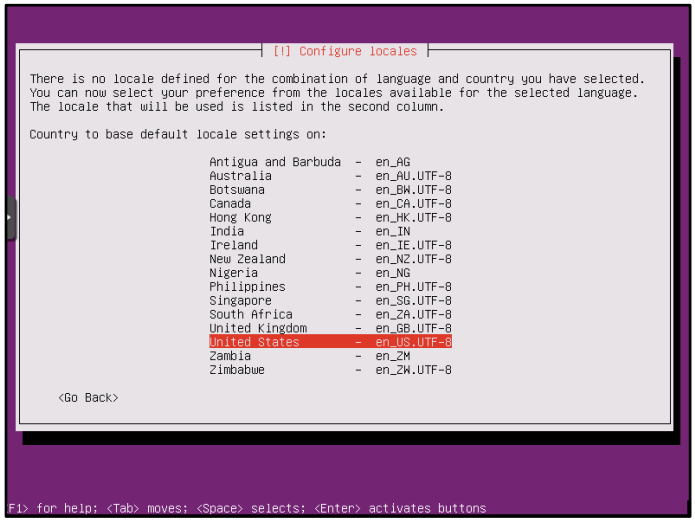
Continuación del apéndice 5.

Instalación SMTP Appliance – Ubicación imagen 3



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación SMTP Appliance – Ubicación imagen 4

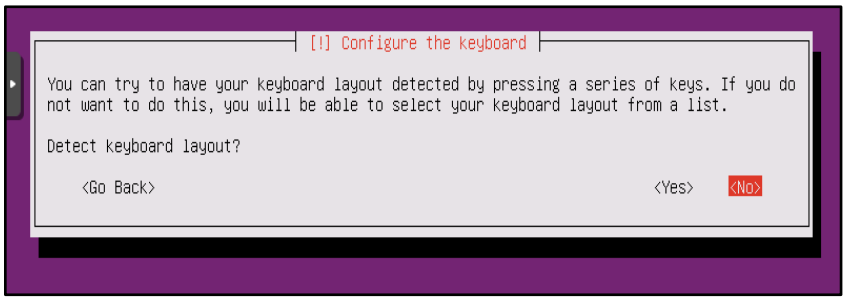


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 5.

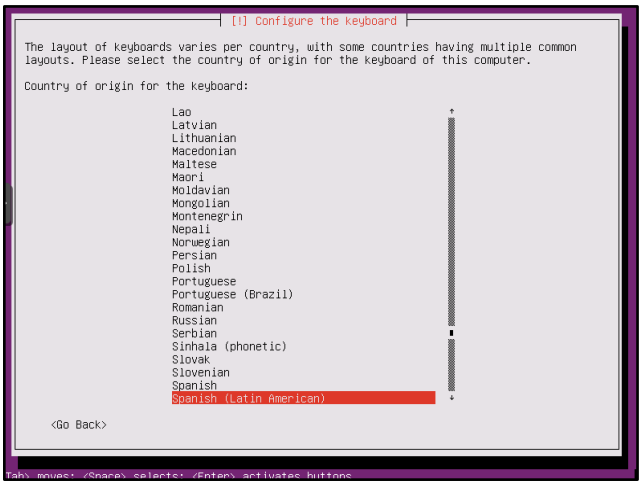
10. Configurar teclado.

Instalación SMTP Appliance – Teclado imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

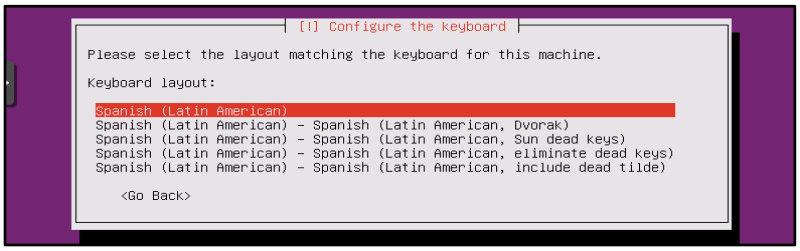
Instalación SMTP Appliance – Teclado imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 5.

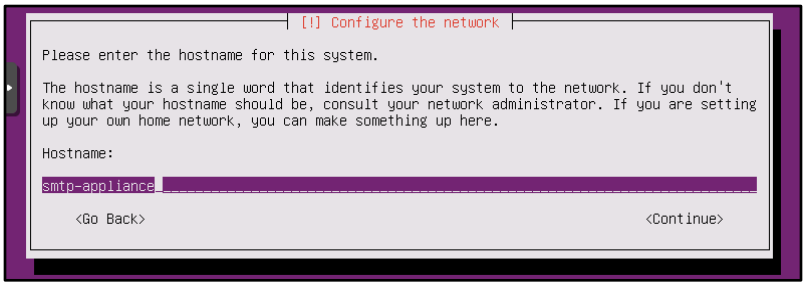
Instalación SMTP Appliance – Teclado imagen 3



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

11. Ingresar *hostname* del *appliance*.

Instalación SMTP Appliance – *Hostname*

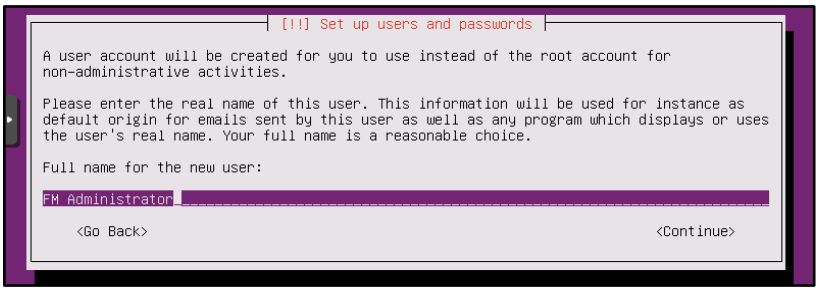


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

12. Ingresar usuario administrador y contraseña.

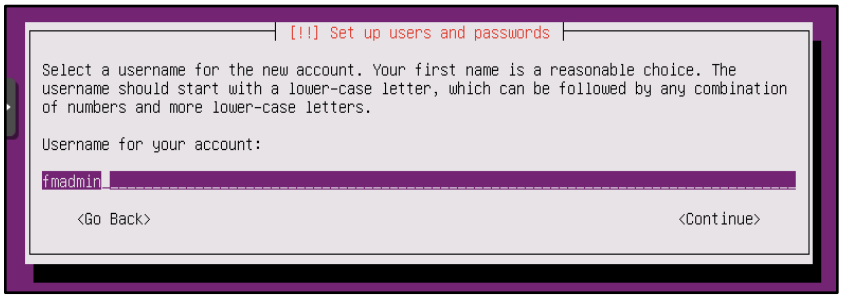
Continuación del apéndice 5.

Instalación SMTP Appliance – Nombre completo de usuario



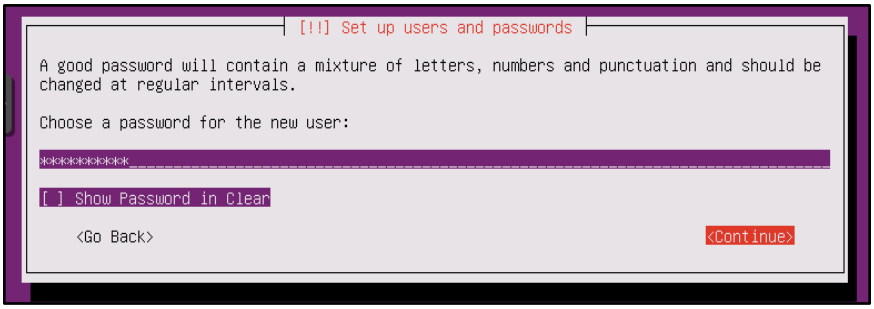
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación SMTP Appliance – Usuario



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

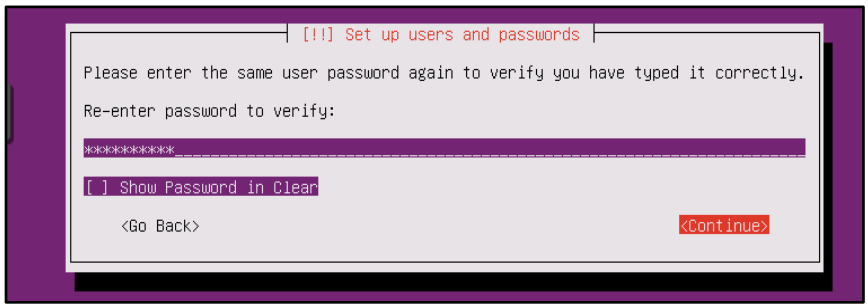
Instalación SMTP Appliance – Contraseña imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

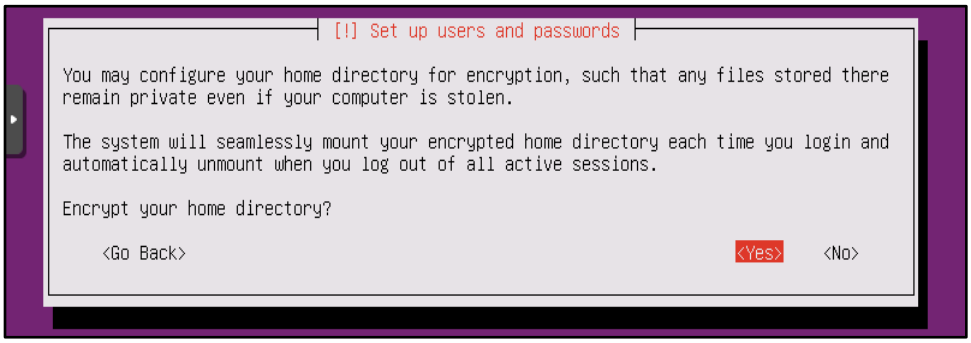
Continuación del apéndice 5.

Instalación SMTP Appliance – Contraseña imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación SMTP Appliance – Encriptar directorio home

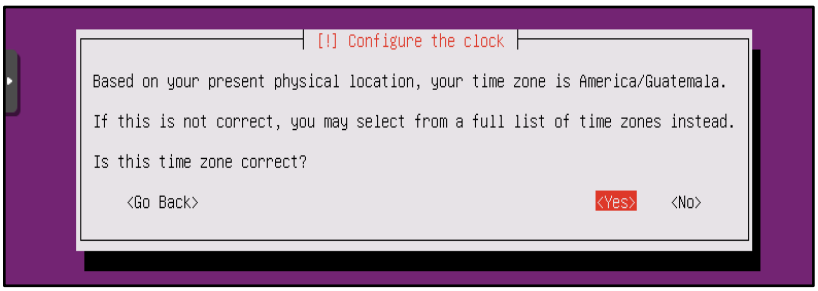


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

13. Configurar zona horaria.

Continuación del apéndice 5.

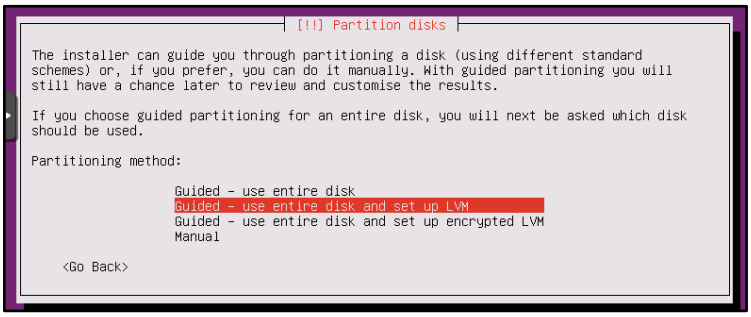
Instalación SMTP Appliance – Asignación de zona horaria



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

14. Configurar particiones de disco con LVM.

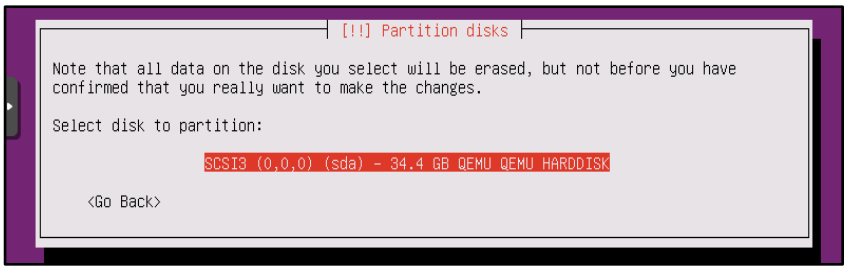
Instalación SMTP Appliance – Disco imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

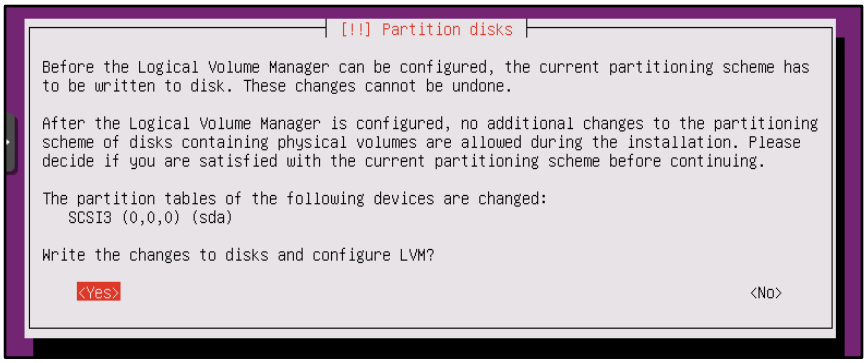
Continuación del apéndice 5.

Instalación SMTP Appliance – Disco imagen 2



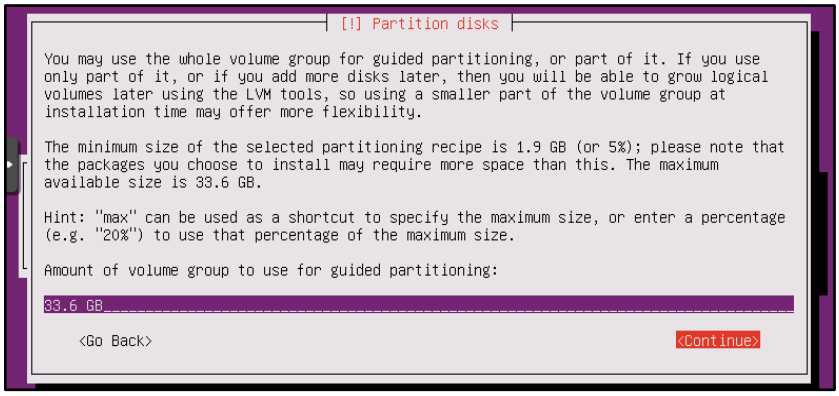
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación SMTP Appliance – Disco imagen 3



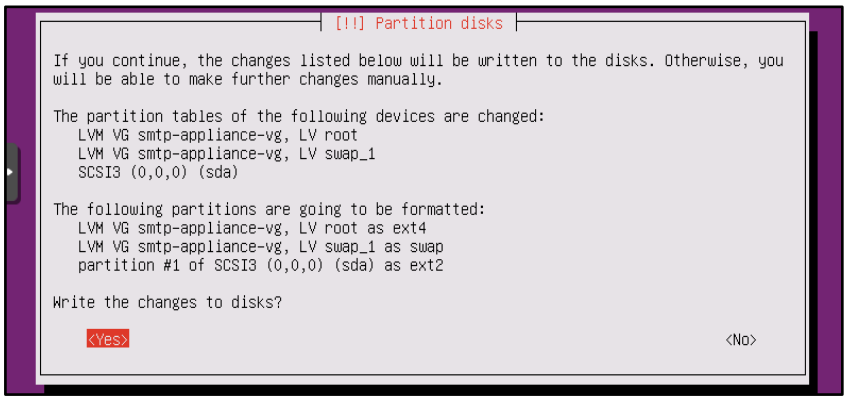
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación SMTP Appliance – Disco imagen 4



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación SMTP Appliance – Disco imagen 5

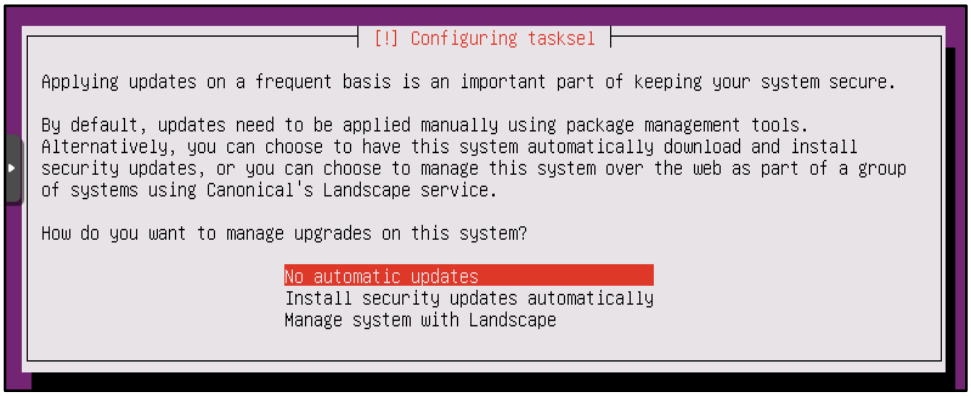


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

15. Inhabilitar actualizaciones automáticas.

Continuación del apéndice 5.

Instalación SMTP Appliance – Actualizaciones



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

16. Instalar paquetes normales y OpenSSH *server*.

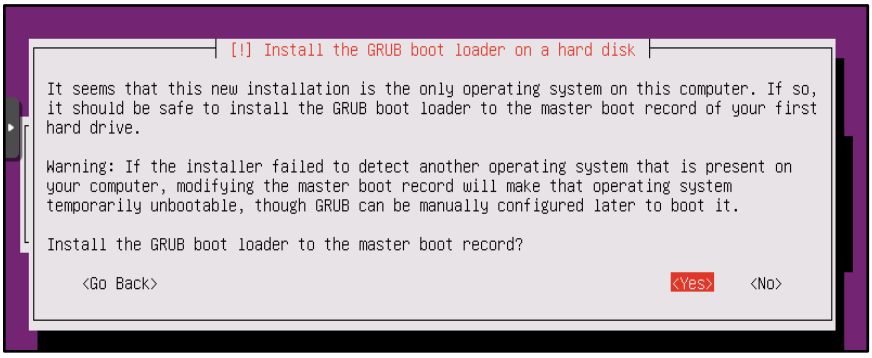
Instalación SMTP Appliance – Selección de software

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

17. Instalar GRUB.

Continuación del apéndice 5.

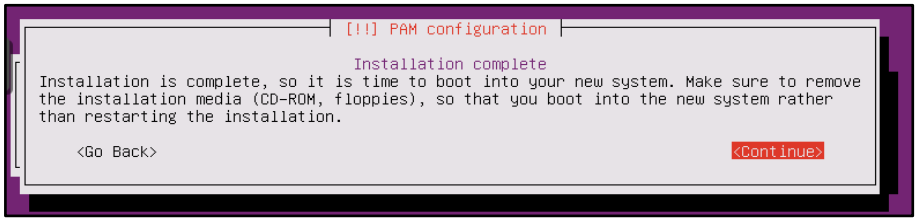
Instalación SMTP Appliance – GRUB



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

18. Reiniciar.

Instalación SMTP Appliance – Reinicio



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Instalación finalizada.

Configuración inicial

Para la configuración inicial de este equipo se siguen los siguientes pasos:

Continuación del apéndice 5.

1. Ingresar al servidor vía SSH o con la consola de la VM.

2. Modificar el archivo ‘/etc/apt/source.list’, eliminando los prefijos ‘gt.’ o cualquier otro de los repositorios y habilitando el repositorio ‘canonical’, lo que lo dejaría de la siguiente forma:

```
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade
# to newer versions of the distribution.
deb http://archive.ubuntu.com/ubuntu/ xenial main restricted
# deb-src http://archive.ubuntu.com/ubuntu/ xenial main restricted
deb http://archive.ubuntu.com/ubuntu/ xenial-updates main restricted
# deb-src http://archive.ubuntu.com/ubuntu/ xenial-updates main restricted
deb http://archive.ubuntu.com/ubuntu/ xenial universe
# deb-src http://archive.ubuntu.com/ubuntu/ xenial universe
deb http://archive.ubuntu.com/ubuntu/ xenial-updates universe
# deb-src http://archive.ubuntu.com/ubuntu/ xenial-updates universe
deb http://archive.ubuntu.com/ubuntu/ xenial multiverse
# deb-src http://archive.ubuntu.com/ubuntu/ xenial multiverse
deb http://archive.ubuntu.com/ubuntu/ xenial-updates multiverse
# deb-src http://archive.ubuntu.com/ubuntu/ xenial-updates multiverse
deb http://archive.ubuntu.com/ubuntu/ xenial-backports main restricted universe multiverse
# deb-src http://archive.ubuntu.com/ubuntu/ xenial-backports main restricted universe multiverse
deb http://archive.canonical.com/ubuntu xenial partner
# deb-src http://archive.canonical.com/ubuntu xenial partner
```

Continuación del apéndice 5.

```
deb http://security.ubuntu.com/ubuntu xenial-security main restricted
# deb-src http://security.ubuntu.com/ubuntu xenial-security main restricted
deb http://security.ubuntu.com/ubuntu xenial-security universe
# deb-src http://security.ubuntu.com/ubuntu xenial-security universe
deb http://security.ubuntu.com/ubuntu xenial-security multiverse
# deb-src http://security.ubuntu.com/ubuntu xenial-security multiverse
```

3. Actualizar el sistema operativo.

```
$ sudo apt update && sudo apt -y dist-upgrade && sudo apt remove –purge && sudo apt -y autoremove --purge && sudo apt clean && sudo apt autoclean
$ sudo rm -R /tmp/*
```

Después de este paso se debe reiniciar.

4. Remover *kernels* antiguos.

```
$ sudo apt purge $( dpkg --list | grep -P -o "linux-image-\d\S+" | grep -v $(uname -r | grep -P -o ".+\d") )
$ sudo update-grub
```

Después de este paso se debe reiniciar.

5. Instalar herramientas para monitoreo, pruebas de estrés, administración remota y descompresión de archivos.

Continuación del apéndice 5.

$ sudo apt install stress-ng nmon htop bwm-ng bmon nethogs pydf ncdu tree mc nmap mtr-tiny traceroute ntfs-3g sshfs ldap-utils unace rar unrar zip unzip p7zip-full p7zip-rar arj $(check-language-support)

6. Instalar paquetes necesarios de Python.

$ sudo apt install python python-pip python-serial
$ sudo pip install gTTS

7. Instalar ‘git’ para descargar repositorios, ‘sox’ y ‘lame’ para manejo de archivos de audio.

$ sudo apt install git sox lame

8. Instalar interfaz gráfica de Xubuntu con el siguiente comando:

$ apt install xubuntu-desktop

9. Instalar audio y habilitar sus módulos

$ sudo apt install libasound2 alsa-utils alsa-oss
$ sudo modprobe snd-ens1371
$ sudo modprobe snd-mixer-oss

Después de este paso se debe reiniciar.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 6. **Zabbix**

Zabbix es un *appliance open source* destinado al monitoreo de equipos. Para ello Zabbix utiliza agentes los cuales se encargan de recolectar y entregar información del estatus de los equipos en los que están instalados. Zabbix también puede obtener información de los equipos mediante el uso de los protocolos SNMP, IPMI o JMX.

Instalación

Al instalar Zabbix 4.4 el equipo poseerá las siguientes características:

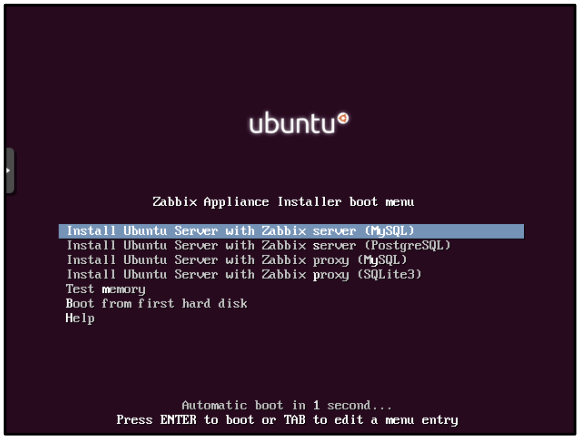
- Ubuntu 18.04 x64 como sistema operativo.
- Herramientas de monitoreo.
- Instalación y configuración del interfaz web HTML5, para administrar el servidor.

Para instalar Zabbix 4.4 se usarán los siguientes pasos:

1. Descargar la imagen ISO del sitio oficial y cargarlo al servidor Proxmox.
2. Crear VM con una tarjeta red que pertenezca a la VLAN 10.
3. Realizar reservación de dirección IP por dirección MAC en Nethserver.
4. Iniciar VM con la imagen ISO.
5. Seleccionar instalar Zabbix server con MySQL.

Continuación del apéndice 6.

Instalación Zabbix – Ventana de inicio



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

La instalación del servidor Zabbix permite el uso de bases de datos MySQL o PostgreSQL. También se puede instalar en modo proxy, enviando la información recolectada de los equipos al servidor Zabbix principal. El modo proxy permite disminuir la carga sobre un servidor Zabbix en caso de tener bastantes equipos que monitorear.

Después de este paso, no es requerido intervenir en el proceso de instalación.

Para el ingreso a terminal, ya sea vía SSH o consola, se debe utilizar las siguientes credenciales:

usuario: appliance
contraseña: zabbix

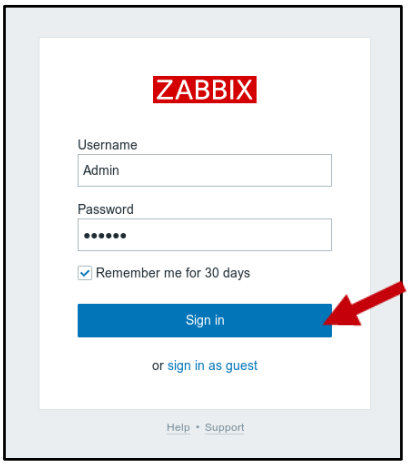
Continuación del apéndice 6.

6. Ingresar a administración vía web.

Para el ingreso vía web, la instalación utiliza por defecto la siguiente url y credenciales:

URL: http://<la dirección IP o FQDN del servidor>/zabbix
usuario: Admin
contraseña: zabbix

Instalación Zabbix – Ingreso a interfaz web



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

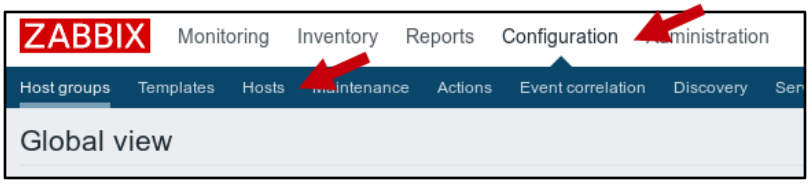
Configuración de *Hosts*

Los *hosts* en Zabbix son los equipos a monitorear. El *host* ya debe contar con el agente de zabbix instalado y configurado, para este proceso ver sección la 'Configuración inicial' de 'Wordpress' en la siguiente sección. Para su configuración en Zabbix, se realiza lo siguiente:

Continuación del apéndice 6.

1. Ingresar a interfaz web.
2. Irse a la ventana de *hosts*.

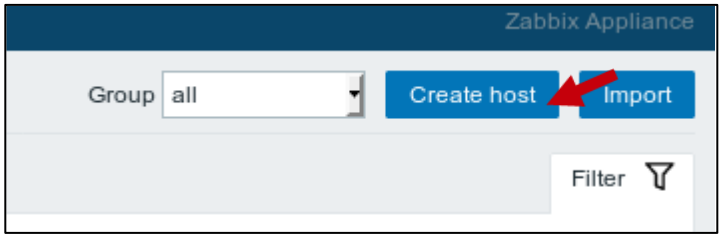
Configuración de *hosts* – Ingreso a ventana de *hosts*



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

3. Seleccionar *create host*.

Configuración de *hosts* – Selección *create host*



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

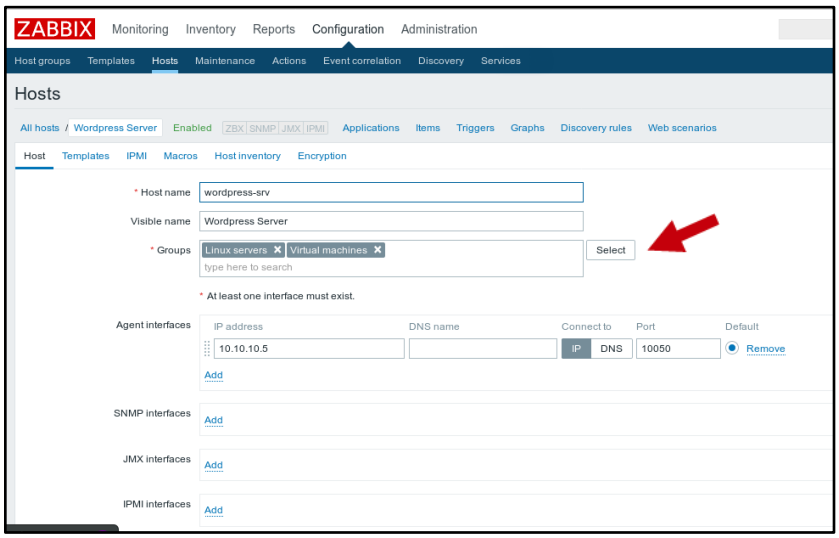
4. Ingresar parámetros generales del *host* a crear.

Los parámetros a asignar son:

- ***Hostname***: Nombre con el cual será identificado el equipo dentro de Zabbix. En este caso se ingresó wordpress-srv.

- ***Visible name***: Si es ingresado, es el nombre que se podrá ver en la interfaz web y alertas enviadas. En este caso se utilizó 'Wordpress Server'.
- ***Groups***: Grupos a los que pertenece. En este caso se seleccionaron las opciones de 'Linux servers' y 'Virtual Machines'.
- ***Agent interfaces***: Aquí se ingresa la dirección IP a asignar al servidor Wordpress en caso de usar un agente de Zabbix. En este caso se utilizó la 10.10.10.5.

Configuración de *hosts* – Parámetros generales

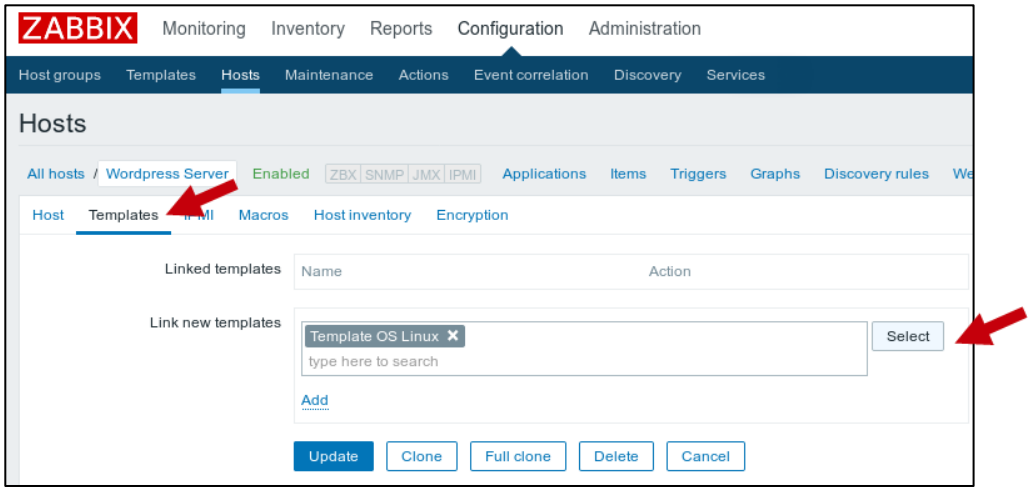


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

5. Agregar *templates*.

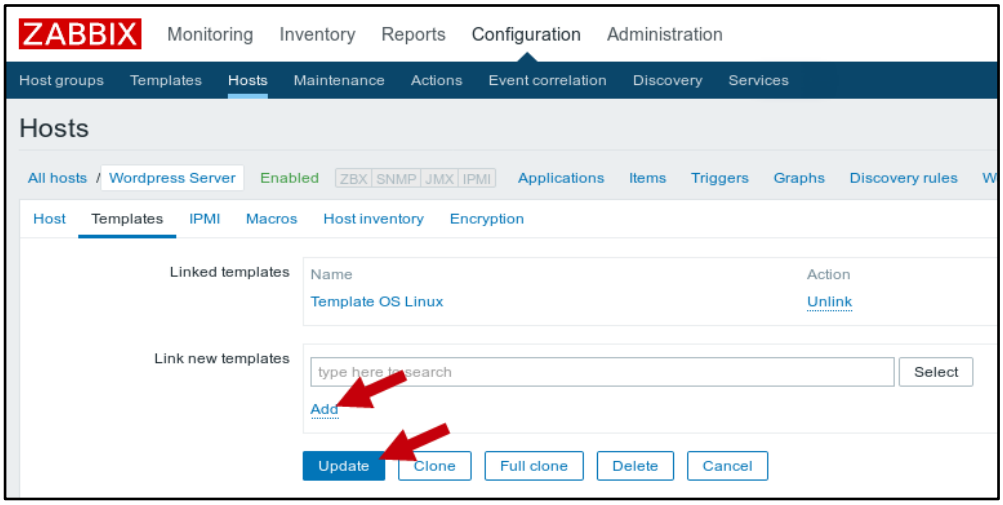
Continuación del apéndice 6.

Configuración de *hosts* – *Templates* imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Configuración de *hosts* – *Templates* imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 6.

Configuración de servicio de correo

Zabbix compara la información recibida de un *host*, con valores predeterminados, conocidos como '*Triggers*', en los *templates* para poder levantar una alerta y enviarla a través de correo, SMS o Jabber.

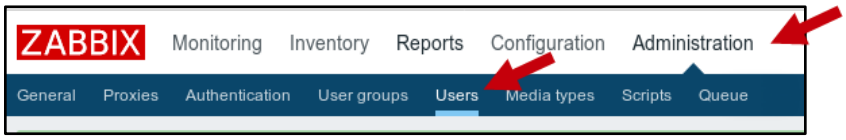
Para esta ocasión solo se configurará el envío de alertas por medio de correo hacia el servidor SMTP Appliance.

Los pasos son:

1. Ingresar a interfaz web

2. Ubicarse a la ventana de tipos de medio

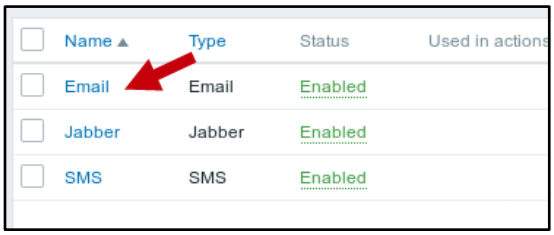
Configuración de correo – Ingreso a ventana de medios



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

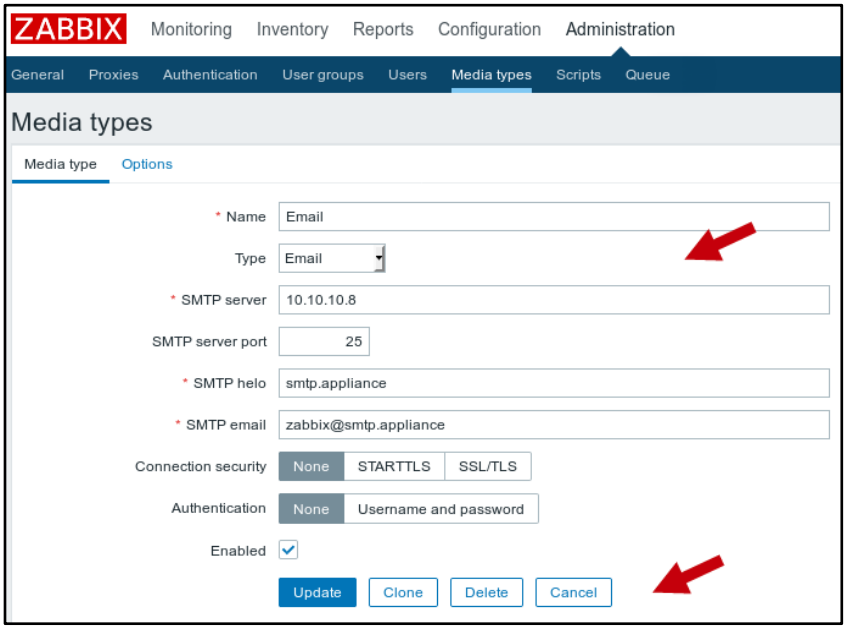
3. Configurar el medio de correo.

Configuración de correo – medio correo imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Configuración de correo – medio correo imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

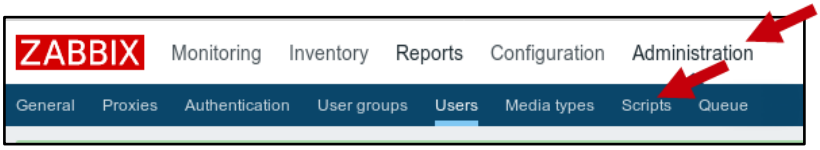
Los valores ingresados en la figura anterior son:

- ***Name***: El nombre del medio.

- ***SMTP server***: Dirección IP o FQDN del servidor SMTP.
- ***SMTP server port***: Puerto de comunicación con el servidor SMTP.
- ***SMTP email***: Correo de este equipo. Dicho correo fue escogido de forma arbitraria.
- ***Connection security***: Seguridad de conexión. En este caso no se tiene habilitada la seguridad en la recepción de mensajes SMTP por lo que se seleccionó *None*.
- ***Authentication***: Método de autenticación. En este caso no se implementó un usuario y contraseña para la autenticación.

4. Irse a la ventana de usuarios.

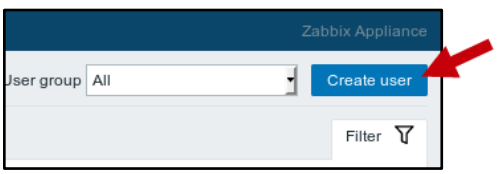
Configuración de correo – Ingreso a ventana de usuarios



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

5. Seleccionar ‘crear usuario’.

Configuración de correo – Selección crear usuario



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

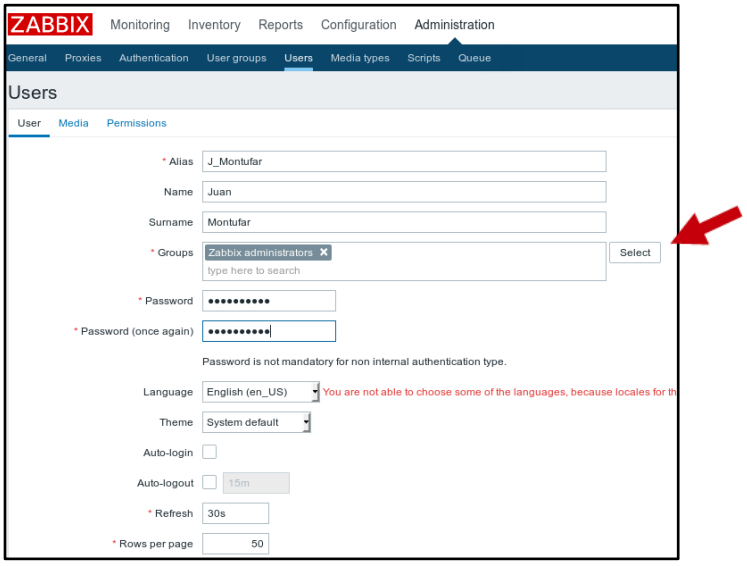
Continuación del apéndice 6.

6. Ingresar parámetros generales del usuario.

Los parámetros relevantes son:

- **Alias**: Nombre con el cual será identificado el usuario dentro de Zabbix.
- ***Name***: Nombre del usuario.
- ***Surname***: Apellido del usuario.
- ***Groups***: Grupos a los que pertenece el usuario. De esto depende los permisos de este usuario.
- ***Password***: Contraseña de ingreso.
- ***Language***: Idioma a utilizar.

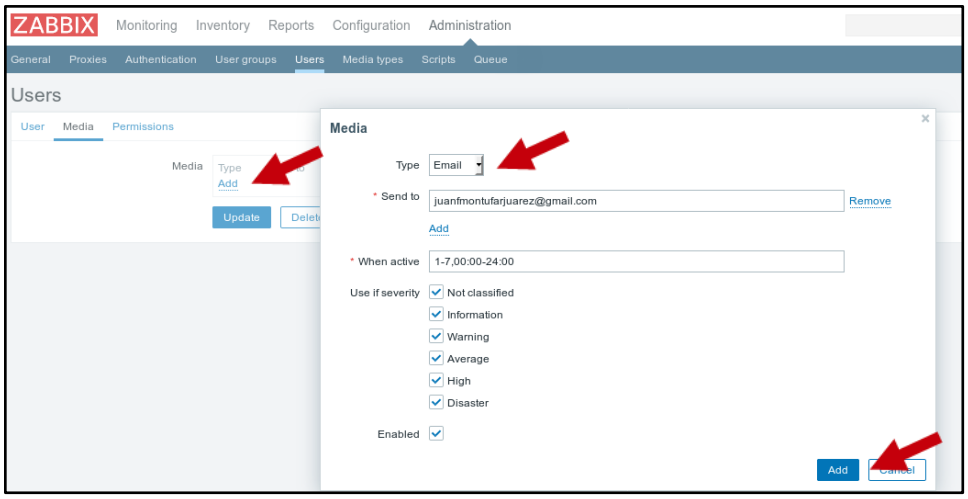
Configuración de correo – Parámetros generales de usuario



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

7. Agregar un medio para el usuario

Configuración de correo – Correo de usuario imagen 1



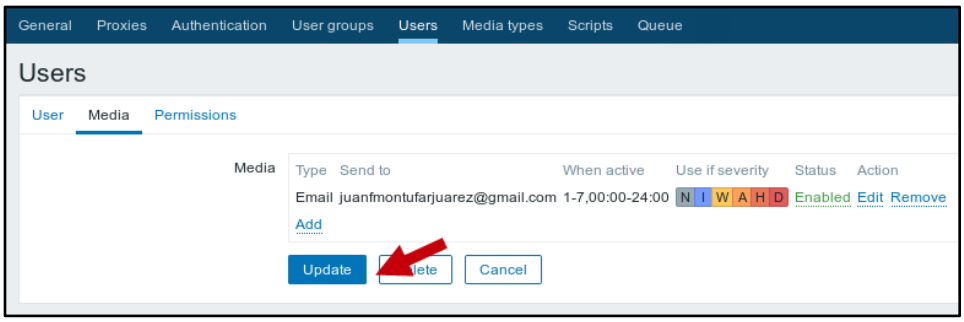
Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Los valores a ingresar en la imagen anterior son:

- ***Type***: tipo de medio a utilizar.
- ***Send to***: Correo del usuario.
- ***When active***: días y horas en el cual el servicio este activo
- ***Use if severity***: las alertas son agrupadas por severidad. Aquí se seleccionan los tipos de alertas a recibir.
- ***Enabled***: Es para habilitar el envío de alertas

Continuación del apéndice 6.

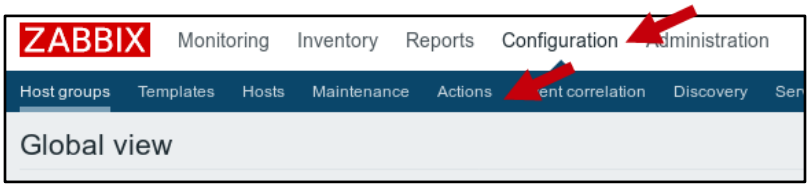
Configuración de correo – Correo de usuario imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

8. Irse a la ventana de acciones.

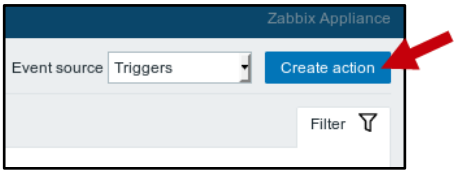
Configuración de correo – Ingreso a ventana de acciones



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

9. Seleccionar ‘crear acción’.

Configuración de correo – Selección crear acción

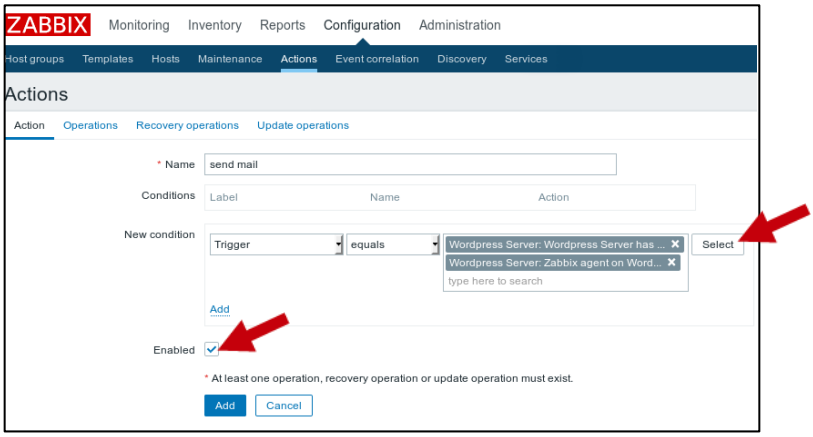


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 6.

10. Asignar las condiciones para el envío de correo.

Configuración de correo – Condiciones de envío

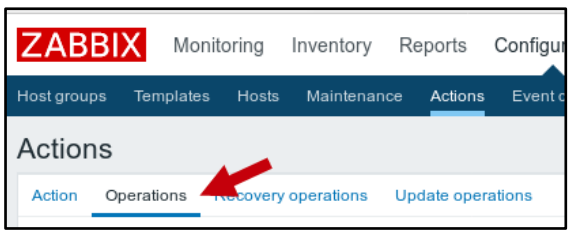


Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

El servidor Zabbix se configuró para enviar mensajes de alertas cuando el agente de Zabbix instalado en el servidor Wordpress se encuentre inactivo durante 5 minutos y cuando el agente regrese a estar activo.

11. Configurar envío de alerta.

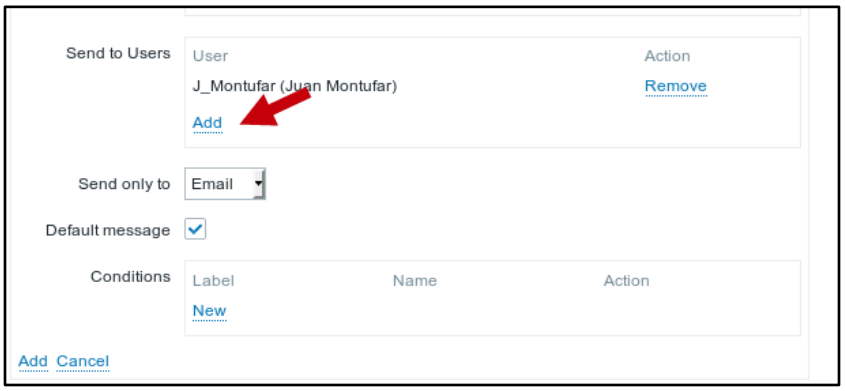
Configuración de correo – Envío de alerta imagen 1



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 6.

Configuración de correo – Envío de alerta imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

12. Configurar envío de correo de cuando el problema ha sido resuelto.

Configuración de correo – Envío de asunto resuelto imagen 1

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Configuración de correo – Envío de asunto resuelto imagen 2

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

13. Agregar acción.

Configuración de correo – Agregar acción

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 7. **Wordpress**

Wordpress es un gestor de contenidos enfocado a la creación de cualquier tipo de página web. Dicho gestor de contenidos será instalado en un Ubuntu Server 18.04 con servidor web Apache y base de datos MariaDB.

Dentro de una organización, generalmente se requiere que contenga una página web propia, por lo que un servidor con Wordpress es bastante útil al momento de realizar dicha página web.

Otras alternativas a Wordpress están Wix, HostGator, BigCommerce, Yumla, entre otros.

Instalación

El proceso de instalación del SO es igual al proceso de instalación del servidor SMTP Appliance.

Configuración inicial

Los pasos a seguir previo a la instalación del SO son:

1. Seguir los pasos desde el 1 al 5 de la sección de ‘Configuración inicial’ del SMTP Appliance.

2. Agregar repositorios de MariaDB con los siguientes comandos:

```
$ sudo apt install software-properties-common
```

Continuación del apéndice 7.

```
$ sudo apt-key adv --recv-keys --keyserver
hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8
$ sudo add-apt-repository 'deb [arch=amd64,i386,ppc64el]
http://mariadb.mirror.anstey.ca/repo/10.2/ubuntu xenial main'
$ sudo apt update
```

3. Instalar base de datos MariaDB, servicio de http Apache2 y lenguaje de programación php7.

```
$ sudo apt install apache2 php7.0 libapache2-mod-php7.0 mariadb-
server mariadb-client
```

4. Instalar librerías adicionales de php.

```
$ sudo apt install php7.0-common php7.0-cli php7.0-gd php7.0-json
php7.0-mysql php7.0-curl php7.0-mbstring php7.0-imap php7.0-ldap
php7.0-curl php7.0-intl php7.0-mcrypt php-imagick php7.0-xml php7.0-
zip php-bz2 php-curl php-gd php-imagick php-intl php-mbstring php-
xml php-zip
```

5. Recrear tablas de administración de MariaDB.

```
$ sudo systemctl stop mysql
$ sudo /usr/bin/mysql_install_db --no-defaults --basedir=/usr --
datadir=/var/lib/mysql
$ sudo systemctl restart mysql
```

6. Mejorar seguridad de MariaDB.

Continuación del apéndice 7.

$ sudo mysql_secure_installation

Previo a este comando se solicitará modificar la contraseña del usuario por defecto ‘root’, desactivar cuentas anónimas así como las cuentas de acceso remoto. Se recomienda ingresar ‘Yes’ a todas las opciones siempre y cuando MariaDB esté recientemente instalado.

7. Instalar agente de zabbix para la versión 4.4.

$ wget https://repo.zabbix.com/zabbix/4.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_4.4-1+bionic_all.deb
$ dpkg -i zabbix-release_4.4-1+bionic_all.deb
$ apt update
$ apt install zabbix-agent

8. Modificar los siguientes parámetros en el archivo de configuración del agente ‘/etc/zabbix/zabbix_agentd.conf’:

Server=<IP del Servidor Zabbix>
ServerActive=<IP del Servidor Zabbix>
Hostname=<*hostname* de este equipo>
ListenIP=<IP de este equipo>

9. Habilitar el servicio del agente al inicio del equipo:

$ sudo update-rc.d zabbix-agent enable
$ sudo update-rc.d zabbix-agent defaults

Continuación del apéndice 7.

10. Iniciar agente:

```
$ sudo systemctl start zabbix-agent
```

Wordpress con SSL

Los pasos para instalar Wordpress son:

1. Descargar wordpress en directorio '/var/www'.

```
$ cd /var/www
$ sudo wget https://wordpress.org/latest.tar.gz
```

2. Descomprimir el archivo 'latest.tar.gz'.

```
$ sudo tar -xzvf latest.tar.gz
```

Una vez descomprimido se creará un directorio denominado 'wordpress'.

3. Modificar permisos de la carpeta 'wordpress'.

```
$ sudo chown -R www-data:www-data /var/www/wordpress
```

4. Crear base de datos con usuario administrador.

```
$ mysql -u root –p
```

Continuación del apéndice 7.

Se le solicitará la contraseña ingresada en el paso 6 de la sección anterior.

mysql> CREATE DATABASE **miBD**;

mysql> GRANT ALL PRIVILEGES ON **miBD**.* TO "**mi_usuario**"@"localhost" -> IDENTIFIED BY "**mi_contraseña**";

mysql> FLUSH PRIVILEGES;

mysql> EXIT

5. Desactivar configuración preestablecida del servidor web Apache.

$ sudo a2dissite 000-default.conf

$ sudo systemctl restart apache2.service

4. Activar módulos SSL en servidor web Apache.

$ sudo a2enmod ssl rewrite headers env dir mime setenvif

$ sudo systemctl restart apache2.service

5. Modificar el archivo */etc/hosts* modificando la siguiente línea:

127.0.1.1 **mi_*hostname*.mi_dominio** mi_*hostname*

Después de este paso se debe reiniciar.

6. Crear certificado propio de seguridad, en caso de no tener uno propio.

Continuación del apéndice 7.

$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/**apache-selfsigned-wordpress.key** -out /etc/ssl/certs/**apache-selfsigned-wordpress.crt**

Después de ejecutar este comando se le solicitará algunos parámetros del certificado en el cual se debe ingresar el FQDN de este equipo.

7. Crear un archivo para que contenga un *virtual host* de la página de Wordpress.

$ sudo nano /etc/apache2/sites-available/**wordpress.conf**

8. Ingresar lo siguiente al archivo creado:

```
# |----- HTTP Puerto 80  ------|
<VirtualHost *:80>
        # |----- Directorio de la Aplicación Web  ------|
        DocumentRoot "/var/www/wordpress"

        # |----- Nombre del Servicio Aplicación Web  ------|
        ServerName mi_hostname.mi_dominio
        # |----- Alias del Servicio Aplicación Web  ------|
        ServerAlias wordpress
        # |----- Email Administrador  ------|
        ServerAdmin mi_correo
        # |----- Traslado automático HTTP a HTTPS  ------|
        Redirect permanent / https://mi_hostname.mi_dominio
```

Continuación del apéndice 7.

```
        # |----- Configurar registro de errores  ------|
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
# |----- HTTPS Puerto 443  ------|
<VirtualHost *:443>
        # |----- Directorio de la Aplicación Web  ------|
        DocumentRoot "/var/www/wordpress"
        # |----- Nombre del Servicio Aplicación Web  ------|
        ServerName mi_hostname.mi_dominio
        # |----- Alias del Servicio Aplicación Web  ------|
        ServerAlias wordpress
        # |----- Email Administrador  ------|
        ServerAdmin mi_correo
        # |----- Directivas Básicas Aplicación Web  ------|
        <Directory "/var/www/wordpress">
                allow from all
                Options None
                Require all granted
        </Directory>
        # |----- IMPORTANTE información Cifrado, Certificado y Llave
Pública  ------|

        SSLEngine on
        SSLCertificateFile              /etc/ssl/certs/apache-selfsigned-
        wordpress.crt
        SSLCertificateKeyFile           /etc/ssl/private/apache-selfsigned-
        wordpress.key
```

Continuación del apéndice 7.

```
        # |----- Configurar registro de errores  ------|
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
```

9. Habilitar sitio web.

```
$ cd /etc/apache2/sites-available/
$ sudo a2ensite wordpress.conf
$ sudo apache2ctl configtest
$ sudo systemctl restart apache2.service
$ sudo apache2ctl -S
```

10. Ingresar a interfaz web.

Para ingresar a interfaz web con https se debe colocar la siguiente URL:

https://<la dirección IP o FQDN del servidor>

11. Escoger lenguaje de wordpress.

Continuación del apéndice 7.

Instalación Wordpress – Lenguaje

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

12. Ingresar parámetros de la base de datos.

Instalación Wordpress – Base de datos imagen 1

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 7.

Instalación Wordpress – Base de datos imagen 2

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

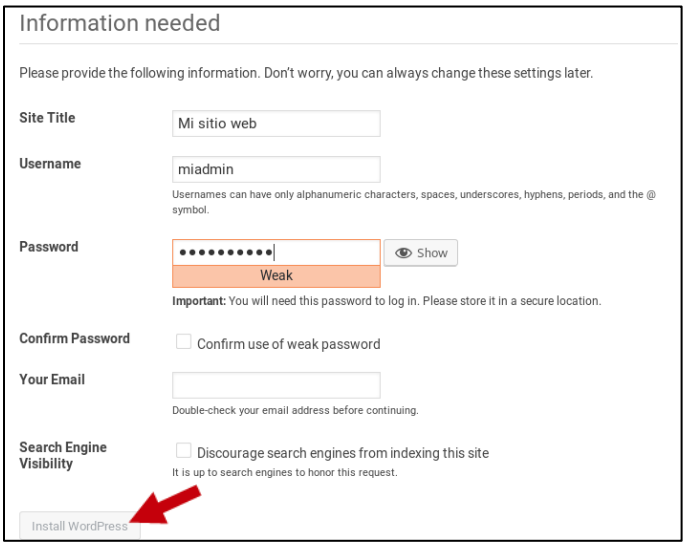
13. Instalar sitio web.

Instalación Wordpress – Instalación de sitio web imagen 1

Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Continuación del apéndice 7.

Instalación Wordpress – Instalación de sitio web imagen 2



Fuente: elaboración propia, captura de pantalla, empleando Flameshot v0.6.

Una vez ingresado ya se puede empezar a crear sitios web.

Si se desea ingresar al sitio web directamente, solo debe ingresar la URL siguiente:

https://<la dirección IP o FQDN del servidor>

Si desea ingresar a la consola de administración de páginas web ingrese lo siguiente:

https://<la dirección IP o FQDN del servidor>/wp-admin

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 8. **Módulo GSM**

El módulo SIM900 está diseñado para una amplia variedad de aplicaciones M2M e IoT, por ejemplo enviar SMS, realizar llamadas telefónicas, conectarse a un servidor web mediante HTTP, conectarse a un servidor FTP, enviar correos electrónicos, entre otros.

El módulo SIM900 pertenece a la clase de tipo GSM/GPRS. GSM, *Global System for Mobile communications*, es una red que se utiliza para la transmisión móvil de voz y datos, la cual es la más utilizada alrededor del mundo (*Internet Archive*: *Global System for Mobile Communications*, 2012). GPRS, *General Packet Radio Services,* es una extensión de la red GSM la cual permite el envío de más mensajes cortos, también denominados SMS, mensajes multimedia o correo electrónico. Por lo tanto el módulo SIM900 utiliza para comunicarse la red GSM con extensiones GPRS.

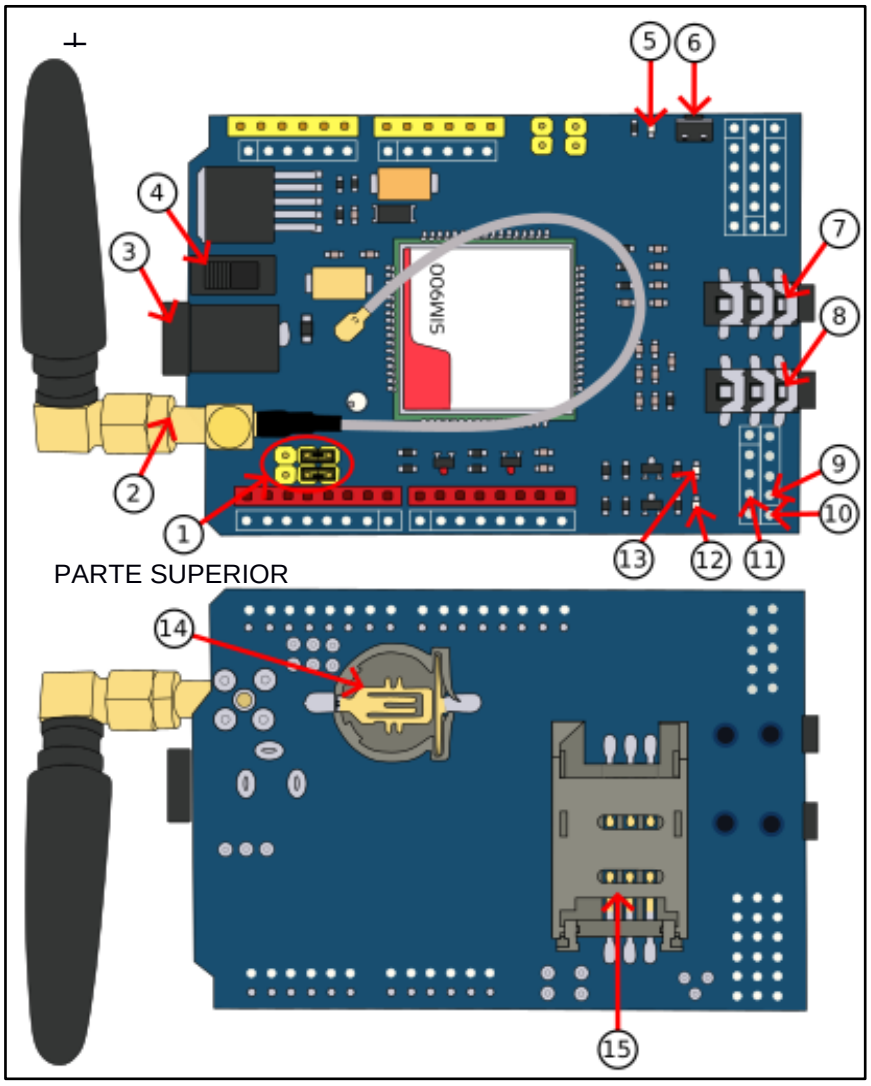
Otras alternativas al SIM900 están: SIM900A, SIM800, SIM800L, SIM808, entre otros.

Partes del módulo GSM

Las partes importantes del módulo utilizado se presentan en la siguiente figura y tabla:

Continuación del apéndice 8.

Partes de módulo SIM900



Fuente: elaboración propia, empleando Inkscape v0.92.4.

Continuación del apéndice 8.

Datos de partes del módulo SIM900

No.	Elemento	Datos importantes
1	Selector del puerto serial	Indica que puerto serial, de los dos que hay, se usará, según la ubicación de los conectores que unen a los pines. El *baud rate* por defecto del módulo es de 19200 baudios.
2	Antena	Permite conectarse a redes GSM/GPRS.
3	Entrada de alimentación	Los tipos de alimentación que pueden conectarse a esta entrada son: 5V 2A, 9V 1A y 12V 1A.
4	*Switch* de encendido	Enciende el módulo.
5	Led de encendido	Indica si el módulo se encuentra encendido.
6	*Switch* de red	Permite iniciar el proceso de comunicación con redes GSM.
7	Entrada de audífonos	Recibe datos de audio durante una llamada.
8	Entrada de micrófono	Envía datos de audio durante una llamada.
9	Transmisor, Tx	Uno de los dos transmisores seriales.
10	Tierra, GND	PIN de tierra. Debe conectarse con otro PIN de tierra de otro dispositivo.
11	Receptor, Rx	Uno de los dos receptores seriales.
12	Led de estado	Indica que se puede comunicar con redes GSM media vez este encendido.

Continuación del apéndice 8.

13	Led de red	Indica cuando está conectado a una red GSM. Si este led se mantiene encendido, aún se encuentra buscando una red, mientras que si se encuentra titilando, ya ha encontrado una red GSM.
14	Entrada de batería para reloj RTC	Al colocar una batería de 3 V, permite activar un reloj de tiempo real, en caso de requerir que se trabaje sobre la hora actual.
15	Entrada de tarjeta SIM	El módulo GSM SIM900 funciona con tarjetas SIM que puedan conectarse a redes 3G y posean una operación de 3 V o 1,8 V. Es requerido validar el buen funcionamiento de la SIM vía interfaz serial.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Puesta en marcha módulo GSM

Para la puesta en marcha del módulo GSM se debe efectuar lo siguiente:

1. Conectar módulo USB a UART al servidor SMTP Appliance.

   Para este paso hay que asegurarse que el módulo USB a UART sea colocado en el puerto correcto, según lo especificado en las características de la máquina virtual, y garantizar que sea reconocido por el *appliance* con el siguiente comando:

Continuación del apéndice 8.

```
$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 002: ID 067b:2303 Prolific Technology, Inc. PL2303
Serial Port
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

2. Insertar tarjeta SIM en la parte posterior del módulo GSM.

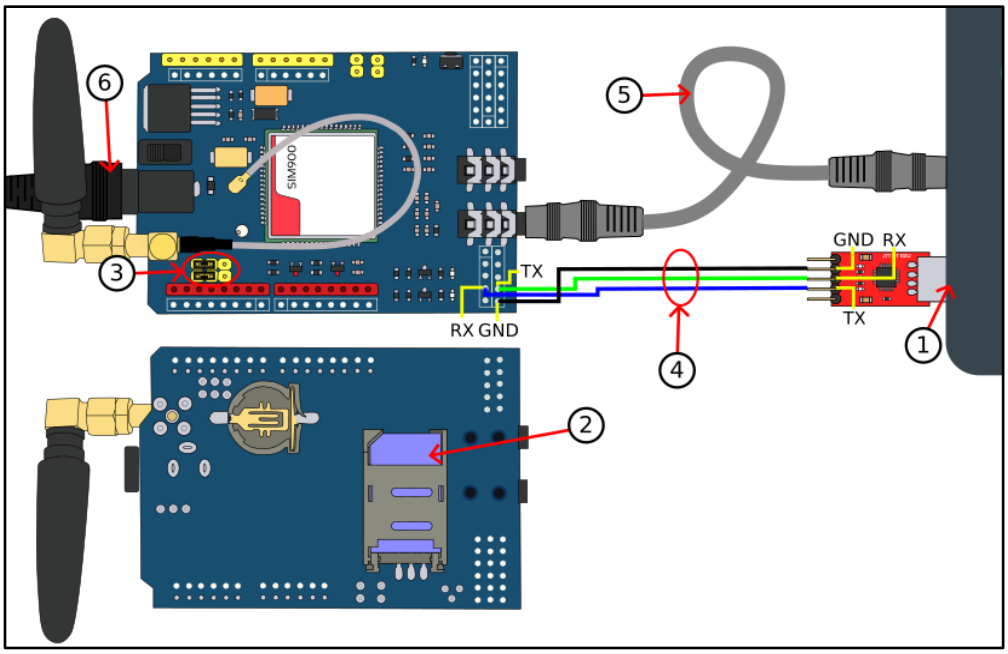
3. Cambiar de posición los conectores del selector del puerto serial del módulo GSM una posición a la izquierda para habilitar comunicación serial.

4. Conectar los pines de GND con GND, Tx con Rx y Rx con Tx del módulo SIM900 con el módulo USB a UART.

5. Conectar cable auxiliar en la entrada de micrófono del módulo GSM y salida del servidor SMTP Appliance.

6. Conectar fuente de alimentación al módulo GSM.

Puesta en marcha módulo SIM900 – Pasos 1 al 6



Fuente: elaboración propia, empleando Inkscape v0.92.4.

7. Iniciar lector de interfaz serial en el servidor SMTP Appliance para garantizar correcto funcionamiento de la tarjeta SIM utilizada.

   Para ello se utiliza el comando ‘screen’ de la siguiente forma:

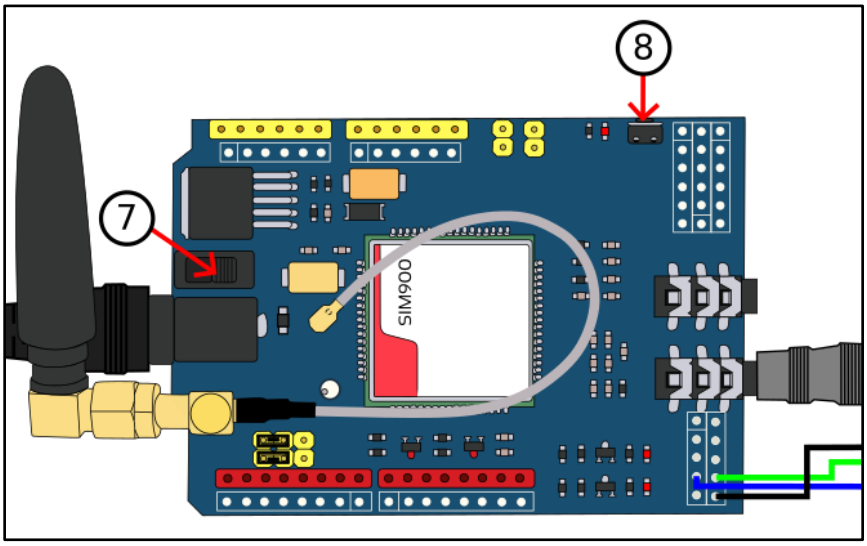
   $ screen **<dispositivo>** ***<baud rate>***

   En Linux los dispositivos USB se leen como ‘/dev/ttyUSBx’. Generalmente el primer dispositivo USB se reconoce como ‘/dev/ttyUSB0’, que es el que se va a utiliza. Además el *baud rate* de módulo utilizado está configurado de forma arbitraria a 9600, no a 19200 como viene de fábrica.

8. Encender módulo GSM con el *switch* de encendido.

9. Presionar el *switch* de red durante 2 segundos. Se encenderán los leds de estado y red

Puesta en marcha módulo SIM900 – Pasos 7 y 8



Fuente: elaboración propia, empleando Inkscape v0.92.4.

Para validar el correcto funcionamiento del módulo se deben mostrar los siguientes mensajes en 'screen':

RDY
+CFUN: 1
+CPIN: READY
+PACSP: 1
Call Ready

Continuación del apéndice 8.

El mensaje ‘Call Ready’ indica que el dispositivo está listo para realizar llamadas o enviar SMS. De no salir los mensajes anteriores, se debe validar la compatibilidad de la SIM utilizada, así como las conexiones realizadas. Para salir de ‘screen’ se debe utilizar la serie de teclas ‘Ctrl + a → k → y’, ó ‘Ctrl + a → :quit’. Es posible utilizar screen para pruebas con interfaz serial.

Fuente: elaboración propia, empleando Libreoffice v6.1.

Apéndice 9. **Equipo de destino**

El equipo de destino es un *smartphone* con sistema operativo Android. En esta sección se detallará como se configuraron las aplicaciones siguientes: Gmail para correo, Mizudroid para VoIP y OpenVPN para conexión a VPN.

Gmail

Gmail es una aplicación del grupo de Google que viene por defecto en los dispositivos Android, por lo que su configuración proviene desde la configuración inicial del *smartphone* utilizado. Como alternativa a Gmail están: Outlook, Blue Mail, Yahoo, myMail, ThunderBird entre otros.

OpenVPN – Connect

OpenVPN es un protocolo libre para establecer conexiones a VPN. Para configurar el dispositivo de destino se debe realizar lo siguiente:

1. Descargar la app de ‘OpenVPN – Connect’ de ‘Google PlayStore’.

2. Transferir un archivo de tipo OVPN al *smartphone*.

   Para que tanto el servidor FreePBX como el equipo de destino se puedan comunicar, ambos deben de estar conectados a la misma VPN.

3. Abrir el archivo OVPN.

Continuación del apéndice 9.

4. Ingresar credenciales de usuario y contraseña para conectarse a la VPN.

5. Activar la conexión a la VPN.

   Si el paso anterior no funciona, se debe intentar conectar a otra VPN tanto en FreePBX como el equipo de destino.

Generalmente se suele tener una baja tasa transmisión de datos a través de VPN, sin embargo brinda lo suficiente para transmitir datos de VoIP.

Mizudroid

Mizudroid es una aplicación gratuita desarrollada por Mizutech. Dicha aplicación fue seleccionada debido a que su configuración es sencilla, provee una alta calidad en el envío y recepción de paquetes de voz, es totalmente compatible con centrales telefónicas Asterisk 13, así como no se queda inhibido en el proceso de una llamada como otras aplicaciones probadas.

Los pasos para la configuración de Mizudroid son:

1. Crear una extensión en FreePBX para este equipo.

2. Descarga la app en ’Google Playstore’.

3. Abrir app y seleccionar tema.

Continuación del apéndice 9.

4. Seleccionar la opción de ‘MÁS’ o ‘ ⋮ ’. 5. Seleccionar la opción de ‘Configuración’. 6. Ingresar dirección IP de interfaz de prefijo *tun* del servidor FreePBX en el parámetro ‘Servidor’, la extensión asignada a este equipo en el parámetro ‘Usuario’ y el valor de *secret* como ‘Contraseña’. 7. Seleccionar ‘OK’. En el teléfono se desplegará el mensaje de conectado.

Fuente: elaboración propia, empleando Libreoffice v6.1.