UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UNA RED VIRTUAL BASADA EN DOCKER EN UN AMBIENTE DE REDES DEFINIDAS POR SOFTWARE (SDN). UTILIZANDO ZEROTIER Y RASPBERRY PI’’.

Trabajo de titulación previo a la obtención del
Título de: Ingeniero Electrónico

AUTOR: YILDER ALEXANDER COBOS BRIONES

TUTOR: Ing. JUAN CARLOS GONZÁLEZ GUZMÁN, MSc.

Guayaquil – Ecuador

2021

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Yilder Alexander Cobos Briones con documento de identificación N° 0920187473 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 12 de noviembre de 2021

Atentamente,

.....

Yilder Alexander Cobos Briones

CI: 0920187473

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Yilder Alexander Cobos Briones con documento de identificación N° 0920187473, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del: Proyecto Técnico: “Diseño e Implementación de una Red Virtual Basada en Docker en un Ambiente de Redes Definidas por Software (SDN). Utilizando ZeroTier y Raspberry Pi”, el cual ha sido desarrollado para optar por el título de: Ingeniero electrónico mención en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 12 de noviembre de 2021

Atentamente,

.....

Yilder Alexander Cobos Briones

CI: 0920187473

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

Yo, MSc Juan Carlos González Guzmán, con documento de identificación N° 0908222987, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UNA RED VIRTUAL BASADA EN DOCKER EN UN AMBIENTE DE REDES DEFINIDAS POR SOFTWARE (SDN). UTILIZANDO ZEROTIER Y RASPBERRY PI, realizado por Yilder Alexander Cobos Briones con documento de identificación N° 0920187473 , obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 12 de noviembre de 2021

Atentamente,

....

Ing. Juan Carlos González Guzmán, MSc.

CI: 0908222987

DEDICATORIA

Este proyecto de titulación está dedicado a Dios, a mis padres y a mi esposa, a ellos, les dedico el gran esfuerzo puesto durante todo este tiempo en mi carrera, pues no ha sido fácil, pero gracias a ese acercamiento y palabras de apoyo han sido quienes me han inculcado el sentido de pertenencia en mis estudios, fueron los pilares fundamentales en el arduo camino de mi carrera superior para poder culminarlos y ser un profesional. También dedico a cada una de las personas que siempre me daban palabras de aliento para seguir adelante, en fin, a todos los que creyeron en mí, pues me alentaron a seguir luchando por terminar mi carrera y que pudiera dar un paso importante en mi vida.

Yilder Alexander Cobos Briones

AGRADECIMIENTO

Me siento eternamente agradecido con Dios, porque sin él no hubiera sido posible alcanzar este logro, con mis padres por la ayuda incondicional que siempre me han brindado, con mi esposa por su paciencia y ser ese hombro que me ayudó a sobrellevar parte este esfuerzo, también me gustaría agradecer a la Universidad Politécnica Salesiana sede Guayaquil y a los docentes de la carrera por haberme formado como profesional, y aquellos compañeros que más que clases, compartimos tiempo de vida en un ambiente de buen compañerismo porque todos han aportado con un granito de arena en mi formación como profesional.

Yilder Alexander Cobos Briones

RESUMEN DEL PROYECTO

Este proyecto de titulación: “DISEÑO E IMPLEMENTACIÓN DE UNA RED VIRTUAL BASADA EN DOCKER EN UN AMBIENTE DE REDES DEFINIDAS POR SOFTWARE (SDN). UTILIZANDO ZEROTIER Y RASPBERRY PI”, se basa en construir un sistema de Red Distribuido Virtual (VDN)", utilizando un referente de redes definidas por software llamado ZeroTier, con prestaciones descentralizadas de las grandes empresas de Tecnologías de la Información y la Comunicación (TIC).

Con la creciente adopción de servicios proporcionados por la computación en la nube, el advenimiento del Internet de las cosas (IoT) y la adición desmesurada de dispositivos a las redes, necesitamos explorar nuevas soluciones que sugieran una plataforma tecnológica más flexible y fácilmente extensible que admita los servicios actuales. La investigación sobre estas innovadoras propuestas tecnológicas debe comenzar necesariamente con los fundamentos teóricos. Como tal, este documento hace referencia a toda la tecnología utilizada en el desarrollo de este proyecto. El ayuda a identificar las características de la nueva arquitectura de red que se implementará en redes modernas en un futuro cercano para abordar los desafíos técnicos causados por la creciente demanda de servicios de red.

La solución propuesta para cubrir las demandas citadas se encuentra enmarcada en las redes inteligentes, fundamentalmente en SDN y NFV. Permite establecer la comunicación de dispositivos a través de cualquier red en un enlace p2p, abriéndose campo en las Redes de Área Global, IoT, servidores y más equipos directamente en lugar de un servidor remoto dedicado, donde vamos a ver todos los dispositivos conectados como si estuvieran en el mismo conmutador Ethernet.

Funciona para que los usuarios obtengan servicios y comunicaciones en internet a través de todo tipo de redes del mundo, incluidas las controladas por los ISP con doble NAT, firewalls, y hasta los tormentos de CG-NAT, facilitando las comunicaciones por ser privada, encriptada, autenticada y segura ante atacantes.

Palabras claves.

Redes Definidas por Software (SDN), Virtualización de las Funciones de la Red (NFV), Función de Redes Virtuales (VNF), ZeroTier, Raspberry Pi, Docker, IoT.

ABSTRACTO

This degree project: “DESIGN AND IMPLEMENTATION OF A VIRTUAL NETWORK BASED ON DOCKER IN A SOFTWARE-DEFINED NETWORKING (SDN) ENVIRONMENT. USING ZEROTIER AND RASPBERRY PI", is based on building a Virtual Distributed Network (VDN) system, using a software-defined networking referent called ZeroTier, with decentralized services from large Information and Communication Technology (ICT) companies.

With the increasing adoption of services provided by cloud computing, the advent of the Internet of Things (IoT) and the excessive addition of devices to the networks, we need to explore innovative solutions that suggest a more flexible and easily extensible technology platform that supports current services. Research on these innovative technological proposals must necessarily begin with the theoretical foundations. As such, this document refers to all the technology used in the development of this project. The helps identify the features of the new network architecture that will be implemented in modern networks in the future to address the technical challenges caused by the increasing demand for network services.

The proposed solution to meet the above-mentioned demands is framed in intelligent networks, fundamentally in SDN and NFV. It allows to establish the communication of devices through any network in a p2p link, opening field in the Global Area Networks, IoT, servers and more equipment directly instead of a dedicated remote server, where we will see all the connected devices as if they were on the same Ethernet switch.

It works for users to get services and communications on the Internet through all kinds of networks in the world, including those controlled by ISPs with double NAT, firewalls, and even the torments of CG-NAT, facilitating communications by being private, encrypted, authenticated and secure before attackers.

Keywords.

Software Defined Networking (SDN), Network Functions Virtualization (NFV), Virtual Network Function (VNF), ZeroTier, Raspberry Pi, Docker, IoT.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE CÓDIGOS FUENTES

INTRODUCCIÓN

La idea principal del presente Trabajo de Titulación se basa en el diseño e implementación de una red virtual con tecnología de redes definidas por software para redes de área amplia (SD-WAN).

La red se ha mantenido relativamente estable en los últimos años, ya que aún se basa en la pila TCP / IP. Los principios básicos del reenvío de tráfico (conmutación y enrutamiento en las capas 2 y 3 respectivamente) están bien documentados y no han cambiado en absoluto. Sin embargo, las redes son conocidas desde hace mucho tiempo por la carga de configurar cada dispositivo por separado, ya sea un conmutador o un enrutador, representa una instancia independiente con su propia lógica.

En la actualidad, las TIC (Tecnologías de la Información y las Comunicaciones), se han centrado principalmente en mejorar su eficiencia. En términos de su infraestructura, se ha producido una transición de mantener una sala de servidores con varios hosts físicos a ejecutar servidores virtualizados alojados por un proveedor de nube con características propias de una nueva arquitectura de redes, mejorando capacidades y agilidad en el despliegue de funciones, logrando un menor tiempo de comercialización de nuevos servicios para clientes finales.

En ese contexto, las SDN se presentan como un nuevo frente para cubrir las nuevas necesidades y promete transformar las arquitecturas y la gestión de las redes que se conocen en la actualidad al estar la red centralizada en la capa de control, las SDN ofrecen a los operadores de red flexibilidad para configurar, administrar, proteger y optimizar los recursos de la red a través de programas dinámicos y automatizados. El problema de escalabilidad y la transición de nuevos protocolos de red definidos anteriormente pretenden ser resueltos por un paradigma emergente actual de Redes Definidas por Software (SDN) llamado ZeroTier.

ZeroTier ha surgido como un enfoque para fomentar la innovación en la red a través de una mayor flexibilidad, capacidad de programación, gestión y rentabilidad. La inteligencia de su red se encuentra lógicamente centralizada en su controlador basado en software, el mismo que mantiene una visión global de la red. Como resultado, la red ZeroTier aparece frente a las aplicaciones y a la gestión de redes como un conmutador lógico y único capaz de comunicar equipos, comunicar redes y demás centros de datos elevando su comunicación a redes de área global, de una manera sencilla de una manera segura por mucho sobre otros similares.

Así, el propósito de esta tesis es mejorar la disponibilidad del servicio de datos basados en SDN de centros pequeños o medianos en compañía de un sistema de orquestadores (Docker) donde los requisitos de flexibilidad y escalabilidad de las instancias de virtualización de una nueva red serían prácticamente instantáneas. De esta forma, los administradores no tienen que esperar a que los fabricantes lancen sus programas, lo cual es muestra de la relativa independencia de esta tecnología, con el desarrollo del software por parte de las empresas proveedoras.

CAPÍTULO I

1. EL PROBLEMA

1.1. Descripción del Problema

Si bien la crisis provocada por la emergencia sanitaria de la pandemia (COVID-19) ha tenido impactos severos en las condiciones de vida y en el ámbito laboral de todas las personas, así surgen preocupaciones tras los retos adicionales que la pandemia ha provocado a las personas y por ende a sus familias. Las compañías a través de sus equipos cuyos miembros trabajan juntos desde ubicaciones geográficamente apartadas han reconocido el desafío de incluir nuevos conceptos tecnológicos de comunicaciones en las estaciones de trabajo al desear vincular sus computadoras y servicios a través de Internet y tratarlas como si estuviesen en la misma ubicación geográfica.

Aunque este flujo de trabajo ofrece muchas ventajas claras, existen casos en los cuales los equipos tienen servicios informáticos externos en una red que normalmente los operadores te impiden acceder directamente a tus archivos, tus servidores, tu propio servicio multimedia o tu nube personal, en fin, tienen servicios en una red controlada, pues cada vez que quisiéramos conectarnos a ellos externamente tendríamos que administrar puertos en routers, firewall o switch y muchas veces peor que eso cuando nos encontramos detrás de una red que tiene configuración CG-NAT de operador, directamente no tendríamos como acceder a esa información que puede ser tan importante y necesaria para nosotros.

Otro problema es la baja escalabilidad que dispone la infraestructura de red convencional que no puede responder de manera efectiva a cambios repentinos en el tráfico de la red y que es difícil de segmentar. Además de ordenadores, tablet, y teléfonos smartphone, los dispositivos inteligentes como cámaras, electrodomésticos, periféricos de comunicación, etc. Se están convirtiendo en sistemas de seguridad. Se trata de cómo integrar todos estos dispositivos de varios fabricantes en la red de forma segura y en un túnel bien estructurado. A menudo, estos elementos están conectados a la red como dispositivos de red comunes, lo que representa un gran riesgo para los usuarios comunes. Los usuarios externos pueden acceder a toda la red a través de dispositivos infectados. Por ejemplo, piratas informáticos que utilizan la conexión a Internet de dispositivos inteligentes o proveedores que utilizan la llamada "puerta trasera" de su producto. En ambos casos, no hay ninguna razón para que tengan acceso a la red y comprometer los datos privados.

Oportunamente, cada vez se crean más proyectos de software como Kubernetes o compilando una aplicación multiservicio compleja. A veces, contribuye a la productividad la posibilidad de tratar las máquinas como si estuviesen juntas, ya que no se debe correr el riesgo de exponer servicios inacabados a Internet. Este paradigma se puede lograr mediante las redes definidas por software (SDN), una tecnología relativamente nueva que proporciona una trama de red dinámica completamente integrada por software.

Los alumnos de la carrera de Ingeniería en Electrónica mención Telecomunicaciones e Ingeniería en Telecomunicaciones, deben estar alineados a los nuevos desafíos en el campo de las Redes de Nueva Generación. Para ello es importante que la Universidad enfoque los laboratorios de Redes y Telecomunicaciones en la investigación continua de retos de ingeniería de red y la introducción de nuevos conceptos prometedores que juegan un papel importante para impulsar la evolución y dar forma al futuro de las redes de comunicaciones.

1.2. Antecedentes

Debido a que el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil actualmente no cuenta con infraestructura e hipervisores SDN que faciliten la tarea de realizar la apertura técnica como equipos, hardware y software para trabajar con redes virtuales, que es soporte primordial para el alumno que cursa el último semestre de la carrera de Electrónica o Telecomunicaciones.

El desarrollo de este proyecto será de gran beneficio, principalmente para los estudiantes que cursan las materias de redes de computadores y redes de comunicaciones que han descubierto el sentido y están entusiasmados por cambiar el estatus de las redes actuales permitiendo el desarrollo e investigación de nuevas tecnologías donde se podrá cambiar el protocolo de red actual TCP/IP.

1.3. Importancia y Alcance

Los estudiantes pueden fortalecer sus conocimientos técnicos tanto teóricos como prácticos, los cuales garantizarán la competencia académica requerida por el alumno el cual estará listo para enfrentar los nuevos desafíos de ingeniería de red que es el reto principal de las organizaciones en nuestro tiempo en un mundo cada vez más resiliente, para lo cual se requiere de profesionales especializados en Gestión de la Tecnología de Información y específicamente, en la asimilación, explotación y apropiación de las TIC, ya que éstas se convierten en un factor estratégico para poder lograr ventajas sostenibles en el mercado.

El alcance de este proyecto es de probar y desplegar escenarios de comunicaciones en redes de área extensa haciendo uso de las NFV y SDN, para evidenciar el despliegue de los recursos de virtualización, como resultado de sólidos conocimientos para enfrentarse a los cambios que se presentarán a futuro en un mundo cada vez más conectado y exigente.

El trabajo está dirigido al ámbito de la Tecnología de la Información y Comunicación (TIC) con función de escalabilidad de conocimientos que se adquieran en la práctica del software.

1.4. Delimitación

1.4.1. Geográfica

Por motivo de seguridad y salud ocupacional, dada la situación mundial de emergencia sanitaria por la pandemia (COVID-19) el diseño y la implementación del proyecto se realizó en el domicilio del autor, para luego del desarrollo establecerlo en el Laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil.

1.4.2. Temporal

El periodo para la ejecución del proyecto fue desde noviembre 2020 hasta septiembre 2021. Tiempo requerido para el diseño, implementación y correcto funcionamiento del proyecto.

1.4.3. Académica

Se aplicaron conocimientos técnicos, prácticos adquiridos durante el transcurso de la carrera de Ingeniería Electrónica, con base en las materias de Programación, Redes y Redes de Comunicaciones.

1.5. Objetivos

1.5.1. Objetivo General.

Diseñar e implementar una Red Virtual para las redes de área local y global, utilizando un software de código abierto con tecnología de redes definidas por software, en un sistema de contenedores de Docker soportados en una tarjeta electrónica raspberry pi para el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana - Sede Guayaquil.

1.5.2. Objetivos Específicos.

- Diseñar e implementar un módulo de Red Virtual.
- Generar un escenario de comunicaciones entre dispositivos portátiles y PC.
- Analizar el funcionamiento de un Servidor NAS con Tecnología SDN – Escenario SD-WAN.
- Documentar los resultados de la conectividad y tráfico de datos entre equipos internos y remotos de la red.

1.5.3. Marco Metodológico

Se utilizará la metodología analítica-experimental, ya que al dejar un respaldo en base a procedimientos y análisis de resultados los interesados en este proyecto podrán

manipular equipos, realizar cambios en las configuraciones, rediseñar, en fin, gestionarlo para el conocimiento en redes definidas por software esperando aportar así a los futuros profesionales especialistas en redes de comunicaciones.

Aplicando esta metodología, el proyecto garantizará el alcance académico y técnico de los estudiantes permitiendo con firmeza su participación en las redes de la próxima generación resiliente.

Se pretende emplear un interruptor de código abierto llamado ZeroTier que utiliza algunos de los últimos desarrollos en materia de SDN, para permitir a los usuarios tener a disposición una comunicación P2P - LAN virtual (red de área local virtual) segura y encriptada a través de Internet, lo cual significa que al contrario de lo que sucede con los servidores de VPN tradicionales, las comunicaciones no tienen que pasar a través de su central, los mensajes se envían directamente de host a host, ganando así eficiencia con la garantía de una latencia mínima.

Siguiendo este proyecto, conectará clientes, servidores y servicios juntos en una red punto a punto simple y controlada, debido a que la red definida por software agiliza la implementación de las herramientas de conmutación y control con la adición de nodos complementarios y un poco de enrutamiento a través de iptables en la Raspberry Pi, podremos dirigir todo el tráfico a través de ZeroTier y hacia Internet a través de su red Local en una LAN virtual.

Una vez establecida la conectividad, tendrá la oportunidad de utilizar la capacidad de tunelización de ZeroTier empleando algunas funciones inteligentes de Linux para permitir que el tráfico salga de su red ZeroTier desde su servidor e indicar a un cliente que envíe su tráfico en esa dirección. Esto significa que puedo conectarme a mi red ZeroTier a través de las aplicaciones iOS / Android / Windows y estar efectivamente en una misma red sin importar la ubicación física del host, por lo tanto, la comunicación está disponible para todos los dispositivos siempre que estén conectados en la misma red virtual.

El equipo constará de una tablet para gestión del administrador donde podrá visualizar remotamente toda la información del sistema que compone la red virtual.

En cuanto al análisis de laboratorio para la gestión de información en un sistema NAS, buscando cumplir con uno de los objetivos planteados, se realizará con el software FreeNAS – OpenMediaVault, Sistema Operativo basado en licencias BSD (Berkeley Software Distribution – Distribución de Software Berkeley) que es un tipo de sistema operativo derivado de Unix, que nos permitirá conectarnos a internet y tener las prestaciones para nuestra red virtual en una estructura de cliente - servidor.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Redes de Datos Modernas

Durante las décadas en que se creó Internet hasta como la conocemos ahora, hubo la necesidad de crear nuevas aplicaciones a la vanguardia de las escalables mega transmisiones de datos. Con las aplicaciones recientemente utilizadas, la naturaleza del tráfico de Internet ha cambiado drásticamente. La infraestructura de red está evolucionando gradualmente de la distribución metálica a la óptica. En un lugar donde no es posible cubrir toda el área global con una conexión de red fija, se establecen conexiones inalámbricas en ubicaciones remotas a través de señales de satélites que se encuentran en órbitas geoestacionarias.

Las nuevas aplicaciones y servicios, el grado y la variabilidad de los dispositivos finales requieren soluciones flexibles a corto plazo. Con la llegada de nuevas tendencias en el campo de las tecnologías de la información y comunicación (TIC), paulatinamente comienzan a aparecer nuevas tecnologías en las áreas de automatización, digitalización u orquestación. Una de las ideas básicas para los diseños emergentes de las redes de datos modernas es la estandarización abierta, esto puede o no ser un proceso políticamente de poder, y la abstracción de las funcionalidades de la red en varios niveles de gestión de red.

No solo en las redes corporativas se espera una mayor combinación de tecnología, con el tiempo, SDN estará trabajando de la mano con inteligencia artificial, dentro de las empresas TIC, se crearán departamentos especializados en la transformación de los modelos de comunicaciones estandarizados, trabajando con la mentoría de la Unión Internacional de Telecomunicaciones o ITU (International Telecommunications Union), estarán enfocados en el procesamiento del tráfico de información en tiempo real con una depreciable tasa de error y latencia absoluta.

Los dos conceptos básicos para las redes de datos modernas son SDN (Redes Definidas por Software) y NFV (Virtualización de Funciones de Red).

2.1.1. Redes Definidas por Software

Las Redes Definidas por Software o SDN (Software-Defined Networking en inglés) constituye una tecnología de red basada en el concepto de separación de la gestión y el control de una red de su hardware físico, es decir la separación física del plano de control del plano de reenvío de datos en la infraestructura de su red, esta tecnología emplea controladores, que son los responsables de gestionar el reenvío de información de los conmutadores o switch en redes cableadas. Hoy en día se han desarrollado muchos controladores SDN, tanto de código abierto como comerciales, por lo que es uno de los principales aspectos para tener en cuenta en el entorno actual de soluciones con aplicaciones de Redes Definidas por Software.

Las Redes Definidas por Software (SDN), son el componente básico de una nueva forma de administrar las Redes Basadas en Intención o IBN (Intent - Based Network en inglés). A medida que crecía el número de usuarios, dispositivos y aplicaciones distribuidas en la red, también lo hacía el entorno de red. El nuevo método de administración de red transforma las redes orientadas al hardware y el trabajo de configuración manual de la red en redes controladas por controladores que capturan las intenciones comerciales y las traducen en políticas que se pueden automatizar y aplicar de manera uniforme en toda la topología administrada. Enfatizamos en referir que el punto central de actividad en la red SDN es el controlador. Los principales dominios en los que los controladores cooperan entre sí son: parte de acceso, WAN (Wide Área Network), centro de datos y nube.

Cada vez surgen más aplicaciones que necesitan interferir con la infraestructura de la red y cambiar su configuración por motivos de seguridad, enrutamiento del flujo de red o equilibrio de carga. Estos pasos a menudo son imposibles debido a la necesidad de configurar cada dispositivo por separado. El nuevo enfoque de gestión de red, que es una red definida por software, responde a las tendencias actuales especialmente en:

- Adición y eliminación de dispositivos de la topología
- Reserva dinámica de la capacidad de la red.
- Desarrollo de servicios en la nube.
- Implementar o eliminar políticas en las redes.
- Independencia del fabricante de equipos de red.

En SDN, el HW (hardware) de la red no contiene el SW (software) de control, sino que la toma de decisiones lógicas y es procesada por el controlador. Todos los dispositivos de la red están conectados a este controlador, de esta forma, pasamos del concepto de control distribuido al control centralizado.

El concepto centralizado parece ventajoso desde el punto de vista de la planificación y la gestión, que ahorra sobre todo el coste de los recursos humanos para la gestión de la red. Por razones de redundancia y respaldo, es deseable que no solo esté presente una única instancia del controlador en la red, sino que haya más de una, para asegurar una alta disponibilidad de las funcionalidades de control. Por lo general, no es posible trasladar la infraestructura de red de la tradicional a la definida por software de la noche a la mañana, ya que no solo necesitamos suficientes recursos financieros, sino también conocimientos para cambiar a SDN. (Mikéska, 2019, p. 18)

Los principales beneficios del concepto SDN son la configurabilidad directa basada en la separación de las capas de control y datos, la abstracción de dispositivos de red, la gestión centralizada, proveer mayor control y flexibilidad, la apertura del estándar independientemente del fabricante y la presencia de API (Application Programming Interface), que permite al usuario comunicarse a través de una interfaz estandarizada con el controlador. (Flachs, 2020, p. 4)

2.1.1.1. Arquitectura SDN. La arquitectura de las redes definidas por software permite una respuesta rápida y eficiente a los requisitos del usuario y los cambios en la red a través de una unidad de control centralizada, que se muestra en la Figura No. 1 como un controlador. Representa una forma centralizada de diseñar y administrar la infraestructura de red. Es un enfoque que separa la parte de datos y control, que le permite percibir toda la red con suficiente abstracción de las funciones de las capas inferiores. (Bill Kleyman, 2015)

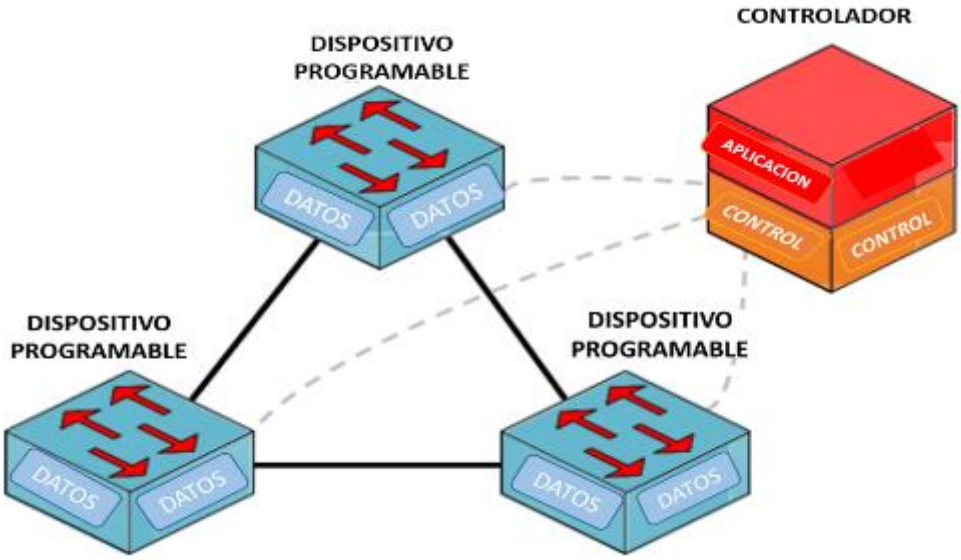


Figura 1. Arquitectura de Red SDN.
Fuente: Adaptado de Arquitectura de red SDN, (p.17), por Martín Mikéska, [Mikéska, 2019].

Algunas fuentes disponibles mal interpretan el concepto SDN con todo lo que incluye el software. Sin embargo, las cuatro definiciones más comunes son:

- Plano de control y datos. Los dispositivos de red se convierten así en elementos simples que envían unidades de datos. La parte de control transmite las instrucciones de la parte de datos utilizando el protocolo de comunicación (más en el subcapítulo 2.1.1.1.1).

- El plano de control (inteligencia de red) se traslada a un dispositivo centralizado, el llamado controlador SDN o Sistema operativo de red "NOS" (Network Operating System). Suele ser una plataforma de software que se ejecuta en un servidor, lo que facilita el control de elementos de la parte de datos en una vista abstracta de toda la red.

- Red programable que permite el control y la administración a través de muchas aplicaciones que se comunican con el controlador mediante la Interfaz de Programación de Aplicaciones o API (Application Programming Interface por sus siglas en inglés). El controlador luego hace cumplir los requisitos para los elementos de la parte de datos.

- El cambio y el enrutamiento se basan en el flujo de datos. Esto se define mediante un conjunto de valores e instrucciones entre los dispositivos de origen y de destino. La abstracción del flujo de datos le permite unificar el comportamiento de varios elementos de red, incluidos conmutadores, enrutadores, cortafuegos, servidores, entre otros.

La idea SDN es la creación de una infraestructura de red dinámica y altamente programable que puede administrar los componentes básicos de la red mientras está separada de las aplicaciones y los servicios de red. Esto debería permitir una mejor escalabilidad y confiabilidad, una segmentación más sencilla, una mayor eficiencia y flexibilidad, capacidad de programación, administración central y una arquitectura dinámica. (Mikéska, 2019)

2.1.1.1.1. ***Capa de Infraestructura o de Datos.*** Incluye los dispositivos físicos, los dispositivos que forman la capa de red virtualizada y los que están interconectados mediante un medio de transmisión metálico, cable óptico o de forma inalámbrica, juntos forman una parte de datos con las mismas características que en la arquitectura tradicional.

Existen reglas y tablas de traspaso para trabajar con unidades de datos entrantes. La parte de control transmite instrucciones a la parte de datos de acuerdo con ciertos datos (MAC Dirección, IP Dirección, VLAN IDENTIFICACIÓN, etc.) A continuación, el dispositivo puede reenviar, descartar, replicar o procesar los paquetes. Si las instrucciones necesarias no están disponibles, los paquetes se pasan a la sección de control para su posterior procesamiento. (Mikéska, 2019, p. 18)

2.1.1.1.2. ***Capa de Control.*** La llamada “El cerebro" de la red, ubicado entre la aplicación y la parte de datos, representa un intermediario entre el usuario y la infraestructura de la red. Es un punto estratégico que mantiene una visión global de toda la red, incluye protocolos de señalización, control y enrutamiento, gestión de sesiones, autenticación, autorización y contabilidad o AAA (Authentication, Authorization and Accounting en inglés) y muchas otras características.

El componente principal es uno o más controladores SDN que se ejecutan en un servidor físico o virtual. El controlador completa las tablas de traspaso para compilar la ruta más adecuada, procesa los eventos de la aplicación y la información del tráfico de datos, garantiza la seguridad, configura parámetros y atributos de elementos, etc. Con una unidad de control centralizada, es mucho más fácil obtener información utilizable sobre el flujo de datos en la red en tiempo real y tomar decisiones rápidas y eficientes basadas en ellos. La tarea principal del plano de control es, por lo tanto, la creación de instrucciones, que el controlador transmite a los elementos de la parte de datos a través de la interfaz sur (controlador / recursos), tal y como se observa en la Figura 2. El procesamiento de estas instrucciones tiene lugar en el hardware de los dispositivos, que tradicionalmente es un método eficaz, ya que el proceso de toma de decisiones de hardware es muy rápido y, por lo tanto, reduce el retraso general. (Mikéska, 2019, p. 18)

2.1.1.1.3. ***Capa de Aplicaciones***. Es un grupo de aplicaciones que utilizan Northbound API, (enlace entre las aplicaciones y el controlador SDN). Es la interfaz para implementar herramientas para la gestión de la red, la gestión de la configuración, la notificación de fallos, la monitorización de eventos, la monitorización del tráfico de datos, etc. Gracias a estas aplicaciones, la red está completamente programable. Por tanto, el entorno se adapta fácilmente a las necesidades y los requisitos de los usuarios que cambian rápidamente. La parte de la aplicación define además los principios de seguridad que el controlador aplica al dispositivo de la parte de datos.

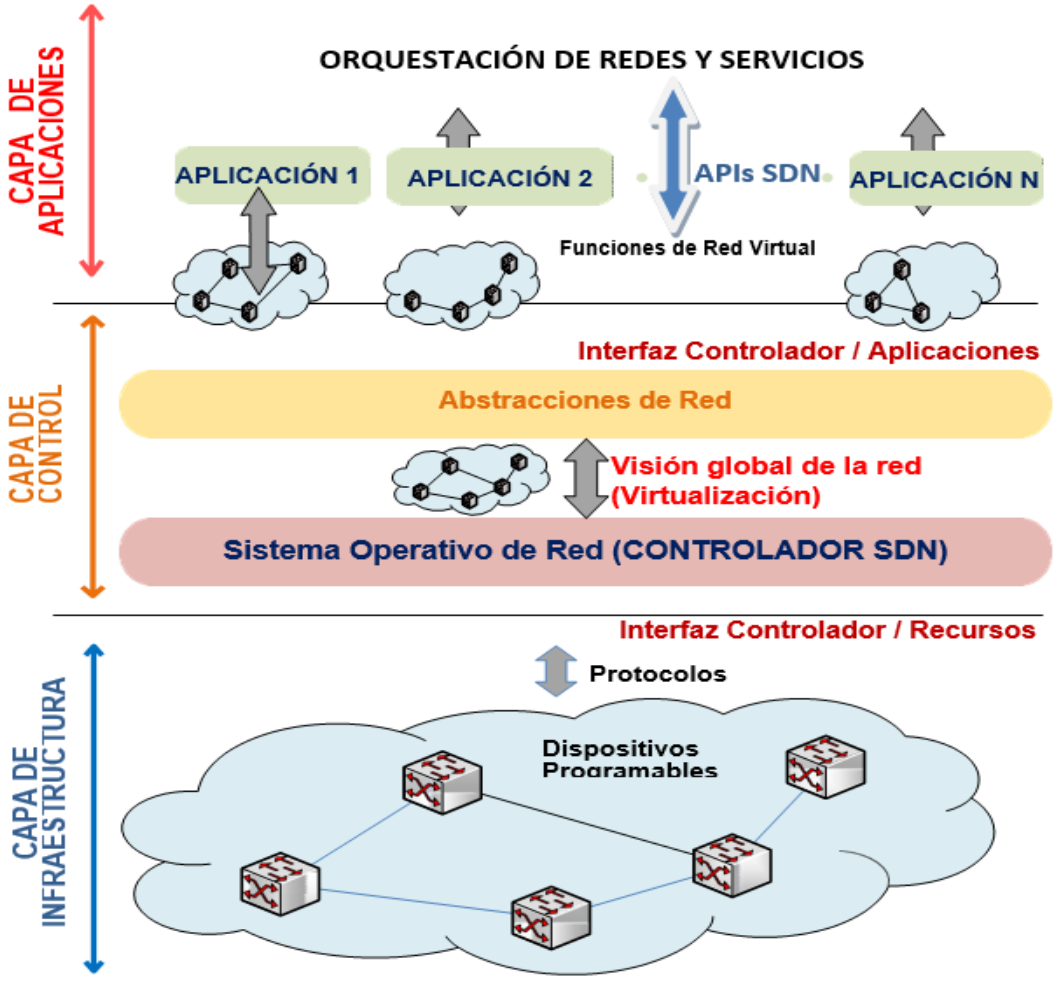


Figura 2. División Básica de la Red SDN.

Fuente: El autor- Adaptado de División Básica de Red SDN, (p. 19), por Martín Mikéska, [Mikéska, 2019].

2.1.1.2. Interfaces de Comunicación. Debe haber un protocolo de comunicación adecuado para intercambiar información desde las aplicaciones a través del controlador a los dispositivos de la red. En una arquitectura SDN , el controlador proporciona dos interfaces de comunicación abiertas principales:

2.1.1.2.1. ***Interfaz Controlador / Recursos (control de hardware).*** Se utiliza para transferir información desde el elemento central de la parte de control, es decir, del controlador hacia los dispositivos en la capa de datos (ilustrado en Figura 2), permite realizar cambios en tiempo real de acuerdo con los requisitos del usuario.

El primer estándar abierto de esta interfaz fue el protocolo OpenFlow (lo describiremos con más detalle en la sección 2.1.1.5.1), que sigue siendo uno de los más comunes. Sin embargo, este no es el único estándar SDN, a pesar de algunos términos de uso SDN y OpenFlow indistintamente. En algunos casos, el controlador le permite combinar OpenFlow con otros estándares o protocolos propietarios.

Otros estándares de interfaz de recursos incluyen una base de datos abierta de vSwitch “OVSDB” (Open vSwitch Database en inglés), el reenvío y separación de elementos de control o ForCES (Forwarding and Control Element Separation), Protocolo de reenvío ajeno o POF (Protocol Oblivious Forwarding en inglés), Biblioteca OpenFlow revisada o ROFL (Revised OpenFlow Library en inglés), HAL (Lenguaje de abstracción de hardware), abstracción programable de ruta de datos o PAD (Programmable Abstraction of Data Path en inglés), OpenState, Cisco OpFlex, etc.

2.1.1.2.2. ***Interfaz Controlador / Aplicaciones (aplicaciones de SDN).*** Permite la comunicación entre el controlador en la parte de control y las aplicaciones o servicios en la parte lógica de la aplicación de SDN. Es un conjunto de interfaces programables abiertas que facilitan la innovación, permiten la automatización, ayudan a administrar el tráfico de datos e implementan nuevos servicios.

La interfaz norte proporciona una abstracción de los dispositivos de red y la topología general. Esto significa que ofrece una interfaz general que permite que las aplicaciones de software funcionen sin conocer las características de los dispositivos individuales. Gracias a esto, es posible desarrollar aplicaciones o servicios que funcionen en dispositivos de diferentes fabricantes, aunque pueden diferir significativamente en la implementación. Gracias a esta abstracción, podemos virtualizar la red, es decir, separar el servicio de red de la red física.

El más común es RESTful API basado en la tecnología de Transferencia de Estado Representacional REST (Representational State Transfer por sus siglas en inglés) usando el protocolo HTTP (Hyper Text Transfer Protocol) - (Protocolo de transferencia de Hipertexto), comúnmente utilizado para transferir tráfico web. RESTful API es simple y extensible, por lo que se ha convertido en el método dominante de llamar API a través de redes. Otra interfaz utilizada es Java API o Python API.

Los fabricantes de controladores diseñan sus propias interfaces del norte (Northbound API) con una definición específica. Hablamos, por ejemplo, de los lenguajes de programación Frenetic, Nettle, NetCore, Procera, Pyre- tic, NetKAT, PANE y muchos otros, que proporcionan potentes mecanismos que facilitan el desarrollo de módulos y aplicaciones de software. (Mikéska, 2019, p. 22)

2.1.1.3. Controladores SDN. La SDN está dirigida por un controlador. El conocimiento del controlador cubre toda la topología de la red. La gestión de la red por parte del controlador consiste en la comunicación con los elementos individuales de la red y en la implementación de la programación de las funciones de la red. Una red de datos con un elemento de control central se denomina solución centralizada. Elegir un controlador SDN dependerá de una solución comercial o de código abierto, esto juega un papel importante.

Una solución de código abierto reducirá la probabilidad de bloqueo del proveedor. Los protocolos estandarizados como OpenFlow u OVSDB (Open vSwitch Database) se utilizan para la comunicación con la interfaz norte (Northbound API). Al elegir un controlador, también se debe prestar gran atención a la funcionalidad, idoneidad y soporte de la solución en sí, que en gran parte es proporcionada por controladores comerciales.

El controlador se comunica con los elementos de la red mediante un protocolo abierto, que suele ser OpenFlow (2.1.1.5.1). Para mantener una alta disponibilidad de control de red, el controlador se ejecuta en un clúster o en una máquina virtual. Estos dispositivos se operan más en la red debido a la redundancia.

Para implementar la red de acuerdo con el concepto SDN, es necesario seleccionar un controlador adecuado para definir las métricas correctas para la red recién creada. El número de controladores que está disponible crece constantemente. (Flachs, 2020, p. 6)

2.1.1.3.1. *Controladores Comerciales SDN*. Al referirnos a controladores comerciales nos enfocamos por soluciones comercial con soporte, cuando se reporta cualquier problema al proveedor de un producto específico, el cual debe garantizar un soporte acorde a lo establecido SLA (Acuerdo de nivel de Servicio). Otro parámetro importante que cumple con las soluciones comerciales es la vida útil. Al elegir un controlador adecuado, es necesario comparar el número de implementaciones en ejecución, las áreas de cobertura y posiblemente la información sobre cuánto tiempo son funcionales estas implementaciones.

Actualmente el desarrollo de controladores SDN para uso comercial son manejados por casi todos los principales fabricantes de equipos de red, como Cisco Systems, Huawei Technologies, Juniper Networks, Ericsson, Contrail, NEC, Aruba, NXP, Inocybe y otros.

Tres tecnologías se comparan a continuación, incluyen controladores SDN de Cisco, HPE-Aruba y Huawei. Estos controladores se comparan en las áreas de automatización, segmentación, monitorización y gestión de redes. Otros parámetros discutidos en comparación incluyen las operaciones simplificadas y automatizadas, seguridad y virtualización de funciones y herramientas de diagnóstico intuitivas. (Flachs, 2020, p. 7)

Arquitectura de red digital de Cisco DNA-C

Cisco DNA-C. El centro de arquitectura de red digital (Digital Network Architecture – Center por sus siglas en inglés) es una plataforma de hardware centralizada que proporciona automatización empresarial LAN, WLAN y WAN.

El controlador DNA-C en su arquitectura interna incluye, además de paquetes de software funcionales el APIC-EM como componente que media la comunicación entre la interfaz gráfica de usuario DNA-C y dispositivos de red. Juntos, brinda una herramienta de control para el despliegue automático de elementos de red, segmentación de red, monitoreo activo, resolución de problemas y administración de toda la infraestructura de red. Todas las funciones del controlador están disponibles a través de la interfaz REST API, La integración de otras herramientas y aplicaciones permite una implementación más rápida de nuevos servicios, aumentando así la flexibilidad de la red. La interfaz sur (Southbound API) se utiliza para comunicarse con los dispositivos de la parte de datos, ejecutan protocolos SNMP y NETCONF.

Como hemos analizado el concepto DNA-C también representa la arquitectura de red global que supone el término Acceso Definido por Software o SDA (Software Defined Access por sus siglas en inglés). Esto cambia significativamente la visión de las redes de datos tradicionales y puede resumirse en los siguiente.

- Utilizando la tecnología de red virtual VXLAN, se crea una red lógica (superposición estructurada) sobre la topología física de la red (capa subyacente).
- Basado en la verificación de la identidad mediante el protocolo IEEE 802.1X, los dispositivos finales se asignan a las redes virtuales adecuadas. se enruta todo el tráfico de datos de estos dispositivos.
- Se usa la intención IBN (Intent-Based Networking),

Solución HPE - Aruba Mobile First Campus

HP Enterprise (HPE) se ha expandido para incluir tecnología inalámbrica a través de la adquisición de Aruba Networks. Las tecnologías inalámbricas son principalmente puntos de acceso, controladores y software de control. La empresa cuenta con una solución para SDN en forma de varios elementos, que son la gestión Air'Wave, Clear Pass y Mobility Master para la configuración de la red inalámbrica y la plataforma Clarity, Network Analytics Engine, NetInsights y Cape Networks para la gestión y parametrización del software. La segmentación dinámica de Aruba Mobile permite que las organizaciones conecten a usuarios y a dispositivos a puertos alámbricos y, a través de un túnel, conectarlos a un controlador o a una VLAN o subred adecuada y descargar una política dinámica (con parámetros de seguridad y de QoS) al puerto al cual estén conectados. Esto extiende la funcionalidad de redes tradicionales alámbricas 802.1x a flujos de trabajos que se desplegaban sólo en redes inalámbricas. Aruba Mobile ofrece un paquete de administración de red llamado Management Center, que se enfoca en la red central y los centros de datos, desafortunadamente no en las redes del campus y no tiene soporte inalámbrico. (Flachs, 2020, p. 7)

Huawei Agile Campus Network Solution

La Agile Network Solution es la solución de red de próxima generación diseñada por Huawei para el mercado empresarial. Esta solución aprovecha el concepto SDN y los últimos logros de investigación en la industria. Basado en los más de 20 años de estudios, Huawei brinda una experiencia de implementación de red con esta solución y permite que las redes sean más ágiles para los servicios.

La solución de red Agile para campus de Huawei implementa la movilidad libre mediante Agile Controller y conmutadores ágiles, una solución Wi-Fi distribuida ágil y puntos de acceso WLAN de alta densidad y construye un campus red que se centra en las experiencias del usuario y el servicio, garantizando experiencias consistentes para los usuarios de oficinas móviles.

Los conmutadores ágiles con un controlador de acceso (AC) nativo integrado funcionan con los puntos de acceso WLAN para converger las redes cableadas e inalámbricas. Además de ofrecer una capacidad de reenvío de alto rendimiento, la solución de Huawei también implementa la administración unificada de redes cableadas e inalámbricas. A través de la implementación de la red Wi-Fi en todos los escenarios, la solución de Huawei proporciona una señal completa cobertura y acceso de usuarios de alta densidad en varios escenarios. (HUAWEI TECHNOLOGIES CO., 2015)

Los principales controladores comerciales se enumeran en la Tabla 1. Incluye a las compañías proveedores y el protocolo controlador/datos utilizados en cada uno.

Tabla 1. Controladores SDN Comerciales.

Nombre	Compañía	Interfaz Controlador/Datos
Agile Controller-Campus	Huawei Technologies	NETCONF, SSH, HTTPS
Altiplano	Nokia	SNMP, NETCONF
APIC-EM	Cisco Systems	CLI, SNMP, NETCONF
Controlador SDN PF6800-BX9000	NEC	OpenFlow
DNA-C	Cisco Systems	CLI, SNMP, NETCONF
DR2000 ADCampus Director	H3C	OpenFlow
NorthStar Controller	Juniper Networks, Inc.	BGP, PCEP, NETCONF, SNMP
OneController	Extreme Networks	OpenFlow, OVSDB, SNMP, XML
SD-Branch	Aruba Networks	OpenFlow
VAN SDN Controller	HP Enterprise	OpenFlow
Vista Manager EX	Allied Telesis	SNMP
VortiQa	NXP	OpenFlow

Fuente: El autor - Adaptado de Controladores SDN Comerciales, (p.55), Martín Mikéska, [Mikéska, 2019].

2.1.1.3.2. ***Controladores SDN de Código Abierto***. Hay varios controladores de código abierto. Entre las grandes cantidades que ofrece el mercado actualmente, es muy difícil determinar un elemento de control adecuado para una red de datos. Para tomar la decisión correcta de elegir un controlador SDN, varias organizaciones de tecnología procesan comparaciones y evaluaciones de su desempeño. (Flachs, 2020, p. 8)

Para seleccionar un controlador de código abierto adecuado, es necesario realizar una encuesta. Durante la encuesta, es necesario encontrar toda la información disponible sobre los controladores SDN en la literatura, sitios web, blogs y otras fuentes disponibles. Una encuesta realizada por expertos en TI en el Instituto Fraunhofer de Tecnología de la Información Segura encontró que los principales actores entre los controladores SDN son: POX, Ryu, Trema, FloodLight y Open-Daylight. (Flachs, 2020, p. 8)

POX es un controlador que se deriva de un controlador de NOX. Es un elemento de red que está escrito en el lenguaje de programación Python. Se utiliza para examinar y depurar la red SDN, virtualizar funciones de red, diseñar el controlador y definir modelos de programación.

Ryu cuenta con el apoyo de NTT (Nippon Telegraph and Telephone Public Corp.) Se basa en un sistema de componentes que se comunican entre sí. Muchos de ellos están predefinidos en la instalación básica. Estos componentes se pueden modificar, expandir y ensamblar en unidades más grandes para adaptar el controlador a la aplicación. Se puede utilizar cualquier lenguaje de programación para desarrollar componentes. El lenguaje básico en el que está escrito el núcleo del controlador es Python.

Trema cuenta con el respaldo de NEC Labs (Nippon Electric Corporation) y sus características más importantes son código fácil de escribir y alto rendimiento. El lenguaje de secuencias de comandos es Ruby, que se utiliza para aumentar la productividad y el rendimiento cuando se trabaja con el controlador.

El compilador escrito en C. Trema también es un marco, gracias al cual es posible construir su propio controlador SDN en Ruby o C. Este marco incluye bibliotecas básicas para crear conmutadores compatibles con OpenFlow. (Flachs, 2020, p. 8)

Floodlight es un controlador SDN que se originó a partir del controlador Beacon original desarrollado en la Universidad de Stanford. Consta de varios módulos, cada uno de los cuales proporciona servicios a los demás módulos. Los módulos proporcionan la lógica de control mediante la API de Java y la Interfaz de programación de aplicaciones (REST API). El controlador es compatible con arquitecturas de sistemas operativos Linux, Windows y Mac. (Flachs, 2020, p. 9)

OpenDaylight es un controlador que tiene como objetivo cubrir la mayoría de los componentes principales de la arquitectura SDN que utiliza código robusto. En torno a

este controlador se ha creado una comunidad de rápido crecimiento que contribuyen al desarrollo del código. Las ramas individuales del código se utilizan para productos comerciales.

Las tecnologías de todos los controladores SDN que se utilizan para la interfaz norte son (REST - Representional State Transfer - Representación de Transferencia de Estado), (JSON - JavaScript Object Notation - Notación de Objetos JavaScript), Java/RPC (Remote Procedure Calling - Llamada a Procedimiento Remoto), OSGi (Open Services Gateway Initiative - Iniciativa de puerta de enlace de Servicios Abiertos). (Khondoker, Zaalouk, Marx, & Bayarou, 2014), (Flachs, 2020, p. 9)

Las tecnologías para la interfaz sur se pueden dividir en dos grupos básicos, que son gestión y control.

Las tecnologías de gestión son OVSDB, OF-Config (OpenFlow-Config), SNMP (Simple Network Management Protocol - Protocolo de Simple Administración de Red) y XMPP (Extensible Messaging and Presence Protocol - Protocolo Extensible de Mensajería y Comunicación de Presencia).

Las tecnologías de control incluyen OpenFlow, XMPP, PCE/PCEP (Path Computation Element**/**Path Computation Element Communication Protocol), ForCES (Forwarding and Control Element Separation - Separación de Elementos de Reenvío y Control), I2RS (Interface to Routing System), BGP (Border Gateway Protocolo - Protocolo de Puerta de Enlace) y BGP-LS (Border Gateway Protocol Link-State).

Seleccionar un controlador usando un criterio es trivial. La toma de decisiones de varios criterios ya es un problema de toma de decisiones de soluciones. Existen varios métodos para resolver este problema, en este caso se utilizó una solución utilizando el método de Proceso Analítico Jerárquico o AHP (Analytic Hierarchical Process por sus siglas en inglés).

El método AHP utiliza la priorización por pares y tiene un mecanismo de verificación de coherencia integrado. Para utilizar este método correctamente, es necesario asignar los valores de propiedad a una escala de prioridad de pares. La adaptación se crea mediante mecanismos de interpolación y extrapolación monótonos. (Flachs, 2020)

Según expertos del Instituto Fraunhofer, el método Ryu se basa en el uso del método MCDM, específicamente AHP aplicado mediante un mecanismo de mapeo, preferiblemente el controlador Ryu en los cinco controladores SDN seleccionados anteriormente mencionados. (Khondoker, Zaalouk, Marx, & Bayarou, 2014)

Los principales controladores de código abierto se enumeran en la Tabla 2. La tabla incluye el lenguaje de programación y el protocolo controlador/datos utilizados en cada uno.

Tabla 2. Controladores SDN de Código Abierto.

Nombre	Lenguaje	Interfaz Controlador/Datos
Beacon	Java	OpenFlow
Floodlight	Java	OpenFlow
Kandoo	Go	OpenFlow
Maestro	Java	OpenFlow
NOX	C++	OpenFlow
ODL	Java	OpenFlow
ONOS	Java	OpenFlow, OVSDB, NETCONF, BGP
OpenContrail	Java/Python	BGP, XMPP, NETCONF
OpenDaylight	Java/Python	OpenFlow, BGP, PCEP, NETCONF
OpenMUL	C/Python	OpenFlow, OVSDB, NETCONF
OPNFV	Java/Python	OpenFlow
POX	Python	OpenFlow, OVSDB
Ryu	Python	OpenFlow, OVSDB, NETCONF
Ryuretic	Python	OpenFlow
Trema	C/Ruby	OpenFlow

Fuente: El autor - Adaptado de Controladores SDN de código abierto, (p. 58), por Martín Mikéska, [Mikéska, 2019].

2.1.1.4. Conmutador SDN. Los conmutadores o llamados interruptores en Redes Definidas por Software pueden ser de software o de hardware. Un conmutador SDN basado en software suele implementarse en una plataforma de servidor estándar y permite las conexiones entre elementos de red y aplicaciones. Un conmutador SDN de hardware suele ser un dispositivo híbrido que gestiona la conmutación de tramas tradicionales.

El conmutador SDN tiene un plano de control y datos lógicamente separado. Si el conmutador SDN recibe datos por primera vez para los que no tiene instrucciones almacenadas para su procesamiento, se pone en contacto con el controlador SDN. Obtiene información sobre cómo procesar el flujo de datos. El interruptor almacena esta información en su memoria y luego no es necesario contactar al controlador cuando recibe datos con el mismo encabezado. El conmutador SDN y el controlador SDN utilizan con mayor frecuencia el protocolo OpenFlow para la comunicación. (Flachs, 2020, p. 10)

En las dos subsecciones siguientes, se enumeran algunos de los productos disponibles.

2.1.1.4.1. ***Conmutadores SDN de Software.*** Un conmutador de software es un conmutador virtual implementado en forma de software en un dispositivo. Se utiliza para simplificar la comunicación entre dispositivos. Este tipo de conmutador puede diseñarse para la comunicación entre entornos virtuales o como firmware para un conmutador de hardware.

En la tabla no. 3 se enumeran los principales conmutadores SDN de software, sus proveedores y la disponibilidad de fuente abierta o comercial de cada uno.

Tabla 3. Conmutadores SDN de Software.

Nombre	Fabricante	Disponibilidad
Abrir vSwitch	Linux Foundation	Fuente abierta
BESS	Berkeley NetSys Lab	Fuente abierta
Bmv2	P4	Fuente abierta
CPqD	CPqD	Fuente abierta
OpenContrail vSwitch	Tungsten Org	Fuente abierta
Pantou/OpenWRT	Stanford	Fuente abierta
Referencia de OpenFlow	Stanford	Fuente abierta
VPP	FD.io	Fuente abierta
Índigo	Project Floodlight	Fuente abierta
Interruptor Lagopus	Lagopus Org	Fuente abierta
Interruptor LINC	FlowForwarding	Fuente abierta
Interruptor Light	Big Switch Networks	Fuente abierta
Interruptor Snabb	Snabb	Fuente abierta
Interruptor VortiQa	NXP Semiconductors	Comercial
Interruptor ZeroTier	ZeroTier	Comercial
B4N SwitchOS	Brain4Net	Comercial
OcNOS	IP Invision	Comercial
PF1000	NEC	Comercial
PicOS	Pica8	Comercial

Fuente: El autor - Adaptado de Conmutadores SDN Software, (p. 10), por František Flachs, [Flachs, 2020].

2.1.1.4.2. ***Conmutadores SDN de Hardware.*** La conmutación de datos por medio de un dispositivo diseñado por hardware se realiza mediante la función de chips individuales, que se sueldan a la placa base del dispositivo de red o se conectan a ella mediante una ranura de bus de datos. Algunos componentes se pueden actualizar en el conmutador, otros son fijos. (Flachs, 2020, p. 11)

Un conmutador de hardware tiene varios tipos de memoria que son de diferente naturaleza. Los conmutadores habilitados para SDN extraen su funcionalidad de control al controlador SDN, estos conmutadores son denominados conmutadores de SDN puros

ya que en ellos todas las funciones de control de un conmutador tradicional (como los protocolos de enrutamiento que se usan para crear bases de información de redireccionamiento) se ejecutan en el controlador central. La funcionalidad en el conmutador se restringe exclusivamente al nivel de los datos.

En un conmutador híbrido, las tecnologías SDN y los protocolos de conmutación tradicionales funcionan simultáneamente. Un responsable de red puede configurar el controlador SDN para descubrir y controlar ciertas corrientes de tráfico mientras que los protocolos de redes tradicionales y distribuidas continúan dirigiendo el resto del tráfico en la red.

En la tabla no. 4 se enumeran los principales fabricantes de interruptores SDN de hardware y una lista de sus productos individuales por compañías.

Tabla 4. Conmutadores SDN de Hardware.

Fabricante	Productos Compatibles con SDN
Alcatel-Lucent	OmniSwitch: 9900, 6900, 6865, 6860, 6560, 6465, 6450, 6350
Allied Telesis	SwitchBlade - x8100, x908; Serie - x950, x930, x610, x550, x530, x510, x310
Arista	7500, 7300, 7280R, 7250X, 7160, 7050X
Aruba Networks	5400R, 3810, 2930M, 2930F, 2920, 2540, 2530
Centec	V580, V350, V330, V150
Cisco Systems	Catalizador - 9600, 9500, 9400, 9300, 9200, 6800, 6500, 4500E, Serie C3850, 3650, 3560-CX, 2960-L Nexus 3000 Series, Nexus 31128PQ, Nexus 3232C, Nexus 3264Q
Dell EMC	N4000, N3000, N2000, N1500, N1100
H3C	S7500X, S6520X, S5560S, S5130S, S3100V3
HP Enterprise	Altoline - 6960, 6940, 6920, 6900; Serie - 10500, 8200, 5500, 5400, 5130, 3800, 2920
Huawei Technologies	S12700, S9700, S7700, S6720-HI, S5730-HI, S5720-HI
Juniper Networks	QFX - 10016, 10008, 10002, 5200, 5100; EX9200
NEC	PF5248, PF5240
Nuage Networks	Serie 7850 NSG-E200 / 300, 7210 SAS
Pica8	P5401, P5101, P3930, P3922, P3297
Quanta	T5032-LY6, T1048-LB9, T3040-LY3, T5016-LB8D BMS T3048-LY9, T3048-LY8, T3048-LY2R, T3048-LY2
Redes Edgecore	AS - 7700, 5800; ECS - 4620, 4510, 4210, 4120, 4100, 3500
Redes extremas	BlackDiamond - X8, 8000, 7100; Serie - X870, X770, X690, X590
Schweitzer Engineering Lab	SEL-2740S

Fuente: El autor - Adaptado de Conmutadores SDN Hardware, (p.11), por František Flachs, [Flachs, 2020].

2.1.1.5. Protocolos Utilizados en SDN. La comunicación entre las capas de datos y control suele ser un asunto exclusivo y lo define el fabricante. Muy a menudo, la compatibilidad entre dispositivos de diferentes fabricantes, como la interfaz de línea de comandos o CLI (Command Line Interface en inglés), no está garantizada.

La definición de protocolos estandarizados para la comunicación entre los planos de control y datos es esencial para las SDN. La definición de estos protocolos que funcionan con API de capas individuales permite la interoperabilidad entre dispositivos de diferentes fabricantes.

Los protocolos se pueden dividir en dos categorías básicas según su funcionamiento. Los protocolos de gestión de bases de datos Open vSwitch “OVSDB” (Open vSwitch Data base Management Protocol por sus siglas en inglés) y OF-Config el protocolo de Configuración de OpenFlow (OpenFlow Config por sus siglas en inglés), se utilizan para activar y desactivar un puerto o para crear un túnel lógico. Por el contrario, el protocolo OpenFlow se utiliza para programar el flujo de paquetes en un dispositivo de red. (Voruganti & Subramanian, 2016)

2.1.1.5.1. *OpenFlow.* Es un protocolo de Capa 2 del modelo ISO / OSI (Organización Internacional para la Estandarización / Modelo de Interconexión de Sistemas Abiertos). OpenFlow desacopla el plano de control del plano de datos y media el enlace de comunicación entre los dispositivos de red y el controlador.

En una arquitectura OpenFlow, siempre debe haber al menos un controlador que se comunique con uno o más interruptores. Los siguientes cuatro puntos describen el principio básico de la solución OpenFlow:

- El controlador crea y modifica la tabla de flujo dentro del interruptor agregando o eliminando entradas para cada flujo.
- El interruptor evalúa los paquetes entrantes y busca una coincidencia en la tabla de flujo, luego ejecuta las instrucciones apropiadas.
- Si el interruptor no encuentra ninguna coincidencia en la tabla de flujo, el paquete se encapsula y se pasa al controlador o se descarta
- El Controlador actualiza la tabla de flujo en el conmutador con la llegada de paquetes nuevos, aún sin asignar.

OpenFlow es una de las muchas partes de la arquitectura general SDN. Es el estándar abierto más utilizado y extendido de la interfaz sur del controlador. La arquitectura de OpenFlow comprende tres componentes principales, como se muestra en la figura 3 y se mencionan a continuación:

(1) Los interruptores compatibles con OpenFlow constituyen el plano de datos.
(2) El plano de control tiene uno o más controladores OpenFlow.
(3) Interfaz OpenFlow. SSL (Secure Sockets Layer).

Los dispositivos compatibles con OpenFlow se dividen en dos grupos:

- OpenFlow-Only: Los dispositivos con esta designación admiten el procesamiento de paquetes solo mediante canalizaciones OpenFlow y no permiten ninguna otra forma de procesamiento de datos.

- OpenFlow-hybrid: En dispositivos híbridos se permite procesar datos tanto mediante operaciones OpenFlow como mediante operaciones de conmutación L2 estándar, tradicionalmente configurados en el dispositivo (Flachs, 2020, p. 12)

Esquemáticamente tenemos el flujo de paquetes. Para todo esto el Switch OpenFlow se ve precisado de unos componentes los cuales se ilustran a continuación.

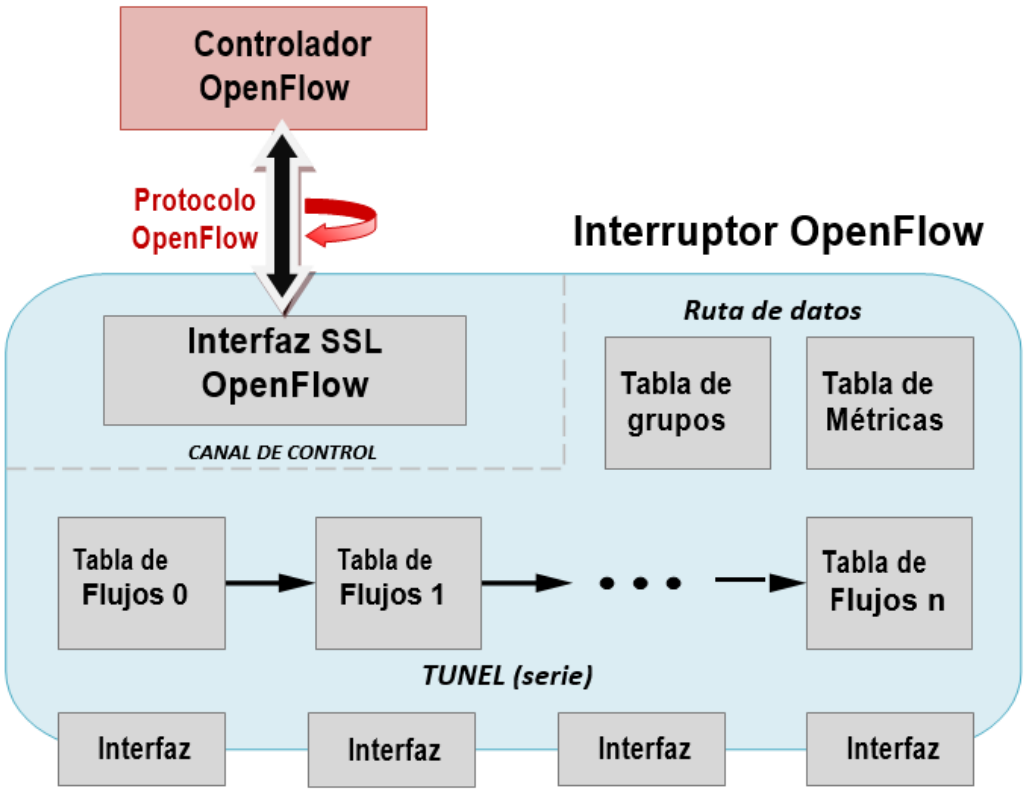


Figura 3. Componentes Principales de un Conmutador OpenFlow.
Fuente: El autor - Adaptado de Estructura OpenFlow, por Martín Mikéska, [Mikéska, 2019].

El conmutador se comunica con el controlador y el controlador gestiona el conmutador a través del protocolo de conmutación OpenFlow. Utilizando el protocolo de conmutación OpenFlow, el controlador puede añadir, actualizar y eliminar entradas de flujo en las tablas de flujo, tanto de forma reactiva (en respuesta a las necesidades de los usuarios) como de forma automática. (Mikéska, 2019, p. 26)

Interruptor OpenFlow

Consta de una o más tablas de flujo y una tabla de grupos, que realizan la búsqueda y el reenvío de paquetes y un protocolo de canal de control seguro (puede haber uno o varios) que se comunican hacia el controlador externo.

Su arquitectura también consta de tres tipos de tablas, que suelen implementarse en hardware o firmware, se realizará un breve análisis a continuación, interpretando los puntos relevantes de cada una de ellas.

Tabla de flujos. Contiene información según la cual se procesan los paquetes entrantes. Puede haber más de una de estas tablas dentro de un solo conmutador. Contiene asociada una regla para identificar el flujo, una acción que aplicar al flujo y un contador para estadísticas.

Ahora es apropiado definir el término "flujo". En general, un flujo es una secuencia de paquetes que atraviesan una red que comparten un conjunto particular de valores en el encabezado correspondiente a una entrada particular en la tabla de flujo. Un ejemplo es un flujo que consta de paquetes con la misma dirección IP de origen y destino, identificador de VLAN, interfaz, etc. (Mikéska, 2019, p. 27)

Un protocolo OpenFlow permite que el controlador SDN controle la comunicación que tiene lugar en el interruptor OpenFlow. Las instrucciones de control básicas son definir, editar y eliminar flujos de datos. Un flujo de datos es un conjunto de reglas que se vincula a todos los paquetes que pertenecen a ese flujo. El interruptor procesa paquetes de un flujo determinado y decide si reenviarlos a uno o más puertos, descartarlos o reenviarlos al controlador para un mayor manejo de excepciones.

Cada flujo de datos contiene campos de coincidencia de paquetes, prioridad de paquetes, varios contadores, instrucciones de procesamiento de paquetes, tiempos de espera y más. Estas reglas se almacenan en tablas (ver figura 4). Cada paquete antes de que salga del conmutador a través de la interfaz de salida se puede procesar mediante un conjunto de reglas en las tablas de OpenFlow. (Flachs, 2020, p. 13)

Tabla de grupos. Los flujos individuales se pueden unificar en una tabla de grupo, que por lo tanto son una manera eficiente de indicar que el mismo set de instrucciones deben aplicarse a múltiples flujos. Entonces, las instrucciones afectan a todo el grupo de flujos, cada tabla de flujo contiene:

- Identificador de grupo: Entero sin signo.
- Tipo de grupo: Para determinar su semántica.
- Contador: Para llevar el número de paquetes procesados por el grupo.
- Set de acciones: A aplicar.

Tablas de medición (Estadísticas). Con estas tablas se mide la tasa de paquetes asignadas a ellas y se facilita el control de este. Consiste en entradas de medidas por flujo esto proporciona a OpenFlow implementar operaciones de QoS simples, como limitar el tráfico.

- Tipo de Banda : Entero sin signo de 32 bytes que identifica unívocamente.
- Tasa de bit : Contiene el ancho de banda mínimo asignado al flujo.
- Contador : Se actualiza cuando un paquete es procesado por la banda de contadores. Se utilizan para hacer estadísticas de todo tipo, asociadas a las tablas, entradas de tabla, puertos, colas, grupos, etc.
- Tipo de Argumentos específicos : Argumentos opcionales.

Un interruptor OpenFlow no debe contener todos los contadores, sólo aquellos marcados como requeridos, si un contador especifico no está disponible en un switch su valor debe ser el máximo de este.

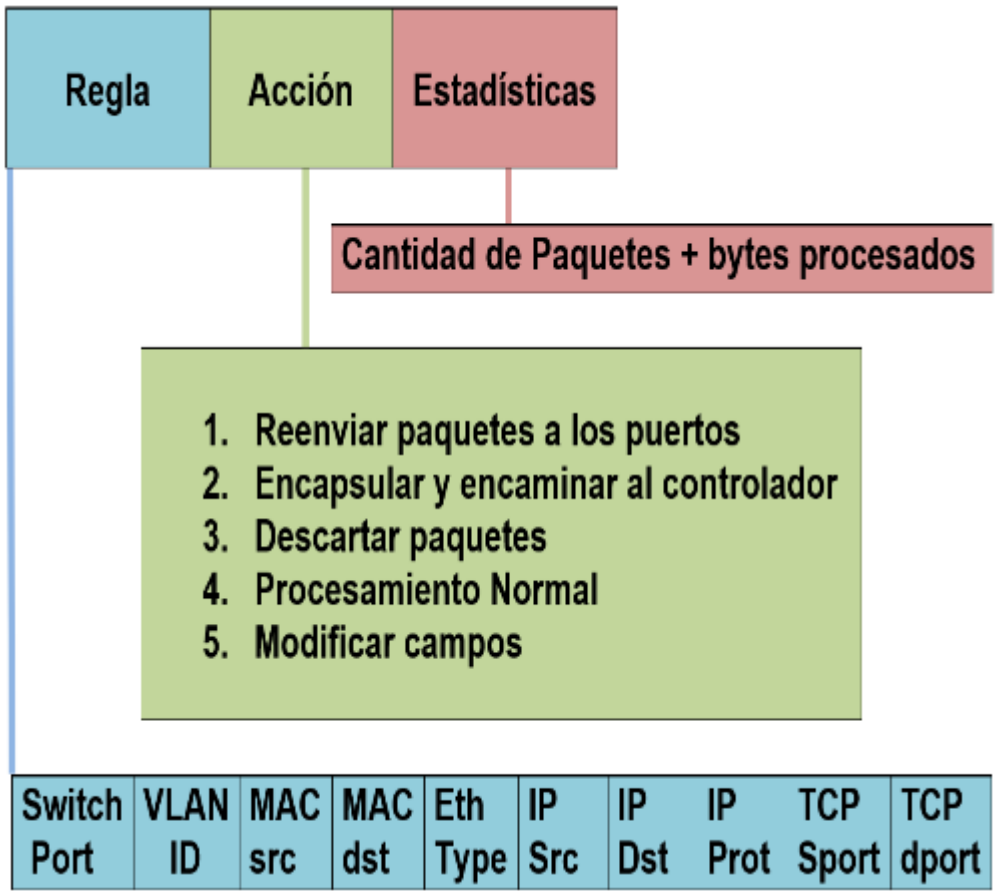


Figura 4. OpenFlow: Entradas de la Tabla de Flujo

Fuente: Propia del autor

Interfaz OpenFlow

Es la interfaz que conecta el switch y el controlador para actualizar la información de la ruta de datos, se realiza a través de un canal seguro, utiliza un protocolo criptográfico SSL (Transport Layer Security) con el fin de hacer la red más segura y que sea realmente el controlador autorizado el que está cambiando la configuración del interruptor OpenFlow.

En el canal se proporciona un cifrado asimétrico basado en la Seguridad de la Capa de Transporte (TLS), aunque también se permiten conexiones sin cifrar del Protocolo de Control de Transmisión (TCP). Esto de acuerdo con su configuración.

Estos canales se utilizan principalmente en los centros de datos en los que el controlador y el interruptor se encuentran en el mismo entorno, lo que provoca que para minimizar los mensajes de control. OpenFlow define interfaces físicas, lógicas y reservadas (en la práctica usamos comúnmente la palabra "puertos"). El interruptor puede conectarse a otros interruptores, dispositivos de red o dispositivos de usuario final a través de estas interfaces.

Alcance OFP (Protocolo OpenFlow)

Estándar abierto de comunicación entre el controlador y los dispositivos.

Como ya hemos comentado anteriormente el controlador centraliza todas las configuraciones de los elementos de la red, se podría decir que funciona como un sistema operativo de red, ya que tiene una visión global de los flujos de datos que se registran y que se comunican en la red. En fin, sintetizando, el controlador se comunica mediante el protocolo OpenFlow con todos los elementos de la red.

Como conclusión, OpenFlow es un estándar para control avanzado de redes de código abierto, fue desarrollado por investigadores de las Universidades de Stanford y California con el fin de estandarizar la comunicación entre interruptores y controladores basados en software en arquitecturas SDN. OpenFlow se agrega como una función para los conmutadores comerciales permitiendo a sus usuarios llevar a cabo laboratorios, sin necesidad de estar ligados directamente al proveedor de los dispositivos de red. OpenFlow está siendo implementado por los principales fabricantes de dispositivos de red del mercado.

El protocolo OpenFlow fue diseñado como un proveedor múltiple, lo que significa que se puede ejecutar en dispositivos de varios fabricantes. El estándar de protocolo OpenFlow está definido por Open Networking Foundation (ONF) para la implementación de SDN en dispositivos de red. OpenFlow podría llegar a las redes presentes en un futuro puesto que existe un gran compromiso con fabricantes de dispositivos de red como Cisco, HP, T- Mobile. En los routers podrían integrarse en su firmware mediante actualizaciones.

2.1.1.5.2. ***NETCONF (Network Configuration Protocol).*** Protocolo de Configuración de Red en RFC 6241. Es un protocolo basado en XML (Extensible Markup Language) para llamadas a procedimiento remoto (RPC). (Remote Procedure Call). Este mecanismo se utiliza para la administración remota de dispositivos, como instalar, manipular y eliminar configuraciones. Admite comunicación basada en transacciones y comunicación de red a través de múltiples dispositivos.

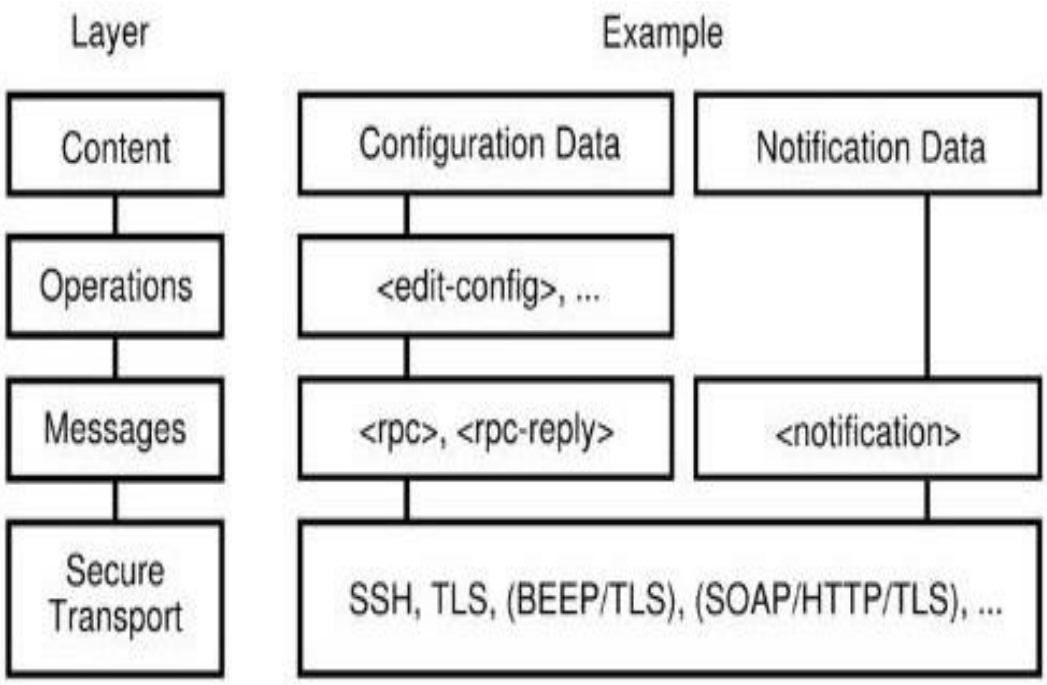


Figura 5. Estructura del Protocolo NETCONF.
Fuente: Protocolo NETCONF, (p. 14), por František Flachs, [Flachs, 2020].

NETCONF define la estructura mediante las siguientes capas:

- Capa de contenido: Permite la configuración del dispositivo y el almacenamiento de datos de notificación.
- Capa de operaciones: Define un conjunto de operaciones para manipular datos en el almacén de datos.
- Capa de mensajes RPC: Soporte para notificaciones y llamadas a procedimientos remotos.
- Capa de transporte seguro: La capa se basa en la comunicación del Protocolo de control de transmisión o TCP (Transmission Control Protocol por sus siglas en ingles) para una transferencia de información confiable. SSH (Secure Shell) o TLS se utiliza para el cifrado. (Flachs, 2020, p. 14)

2.1.1.5.3. ***REST (Representational State Transfer) API.*** Transferencia de Estado de Representación. Este protocolo se ha convertido en el estándar para comunicarse con servicios web porque es flexible, escalable y multiplataforma. REST se basa en el modelo de consulta-respuesta HTTP/S (HyperText Transfer Protocol/Secure) (Protocolo de transferencia de hipertexto / Seguro) en la comunicación cliente-servidor.

REST utiliza los siguientes métodos para la comunicación cliente-servidor:

- GET: obtenga datos de la ruta especificada en el Localizador uniforme de recursos (URL). (Uniform Resource Locator).
- POST: escribe datos en la ubicación especificada por la ruta en la URL. El método se usa solo para escribir datos nuevos.
- PUT: reemplaza los datos definidos por una ruta existente en la URL. No se puede utilizar para crear datos hasta ahora inexistentes.
- DELETE: elimina los datos existentes de la ruta definida en la URL.

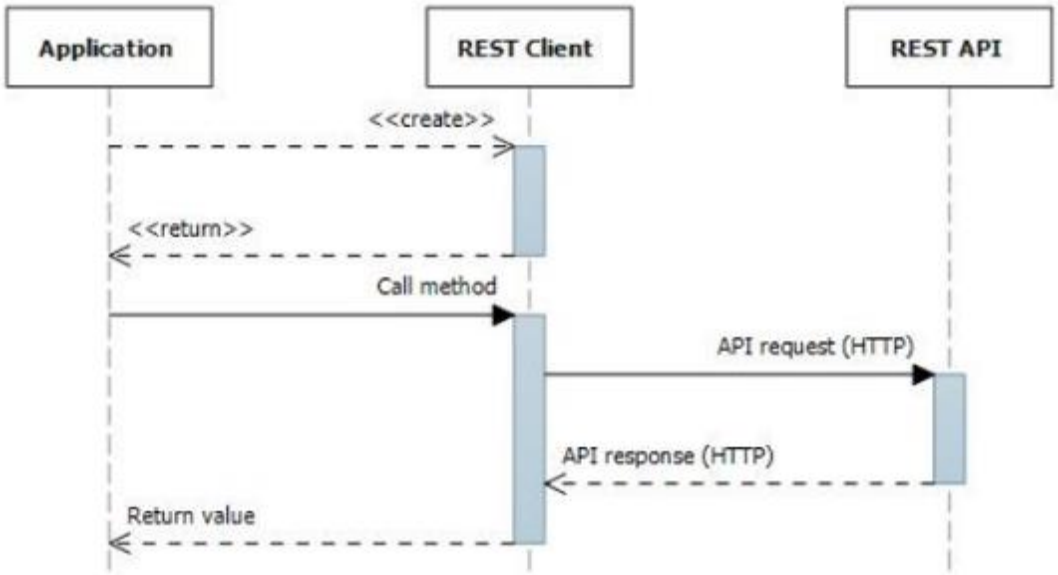


Figura 6. Comunicación de la Aplicación con el Servidor mediante REST API.
Fuente: Aplicación REST API, (p. 15), por František Flachs, [Flachs, 2020].

La comunicación con la API se realiza con mayor frecuencia mediante objetos JSON. Es un formato de archivo estándar abierto para intercambiar datos que utiliza un formato legible por humanos. Los objetos de datos son pares atributo-valor. Este es un formato de uso común en varias aplicaciones. Hoy reemplaza el formato XML. En la imagen 6 Se muestra la comunicación entre la aplicación y el servidor mediante la REST API. (Flachs, 2020, p. 15)

2.1.1.5.4. ***SNMP (Simple Network Management Protocol).*** Protocolo Simple de Administración de Red. Se utiliza para trabajar con la capa de administración y se define en RFC 1157. El protocolo utiliza un agente en el lado del dispositivo administrado, que es un software que recupera información de un dispositivo de red. La información se almacena la Estación de Administración de Red NMS (Network Management Station), donde se procesa posteriormente. En caso de que la evaluación del mensaje de estado supere el límite establecido por el usuario, se notifica al administrador de una anomalía en la red.

Este protocolo se puede utilizar para monitorear el estado de los dispositivos de red. Hasta cierto punto, se puede utilizar para configurar fácilmente la funcionalidad del dispositivo. Los datos del dispositivo recopilados se organizan en una estructura MIB (Management Information Base). (Flachs, 2020, p. 16)

2.1.1.5.5. ***SSH (Secure Socket Shell).*** Es un protocolo que se utiliza para conexiones encriptadas remotas entre dispositivos de red. Proporciona varios niveles de seguridad en las comunicaciones al cifrar los datos transmitidos. SSH es el sucesor de protocolos inseguros como telnet, rlogin o FTP. Los usos más comunes de este protocolo son para proporcionar acceso de usuario seguro o para procesos automatizados, transferencia de datos interactiva y automatizada, entrada de comandos remota y administración de infraestructura de red. (Flachs, 2020, p. 16)

El protocolo funciona en un modelo cliente-servidor, lo que significa que la comunicación se inicia en el lado del cliente. El cliente inicia la comunicación enviando una solicitud para contactar al servidor. Posteriormente, se utiliza la clave pública del servidor para verificar su identidad. Después de la fase de configuración inicial, se utilizan una clave de cifrado simétrica y un algoritmo hash para garantizar la seguridad y la integridad de los datos transferidos entre el cliente y el servidor.

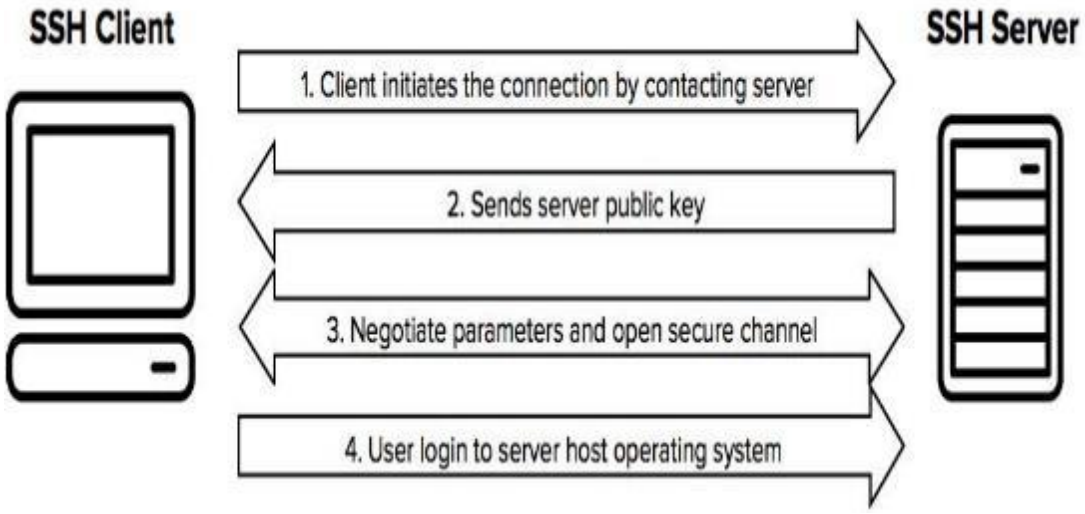


Figura 7. Funcionalidad del Protocolo SSH.
Fuente: Protocolo SSH, (p. 16), por František Flachs, [Flachs, 2020].

2.1.1.5.6. ***PCEP (Path Computation Element Communication Protocol).*** Protocolo de plano de control SDN. Se define en RFC 5440-4655. Este es uno de los protocolos de control más comunes para la comunicación entre el controlador central y los dispositivos de red. Funciona con la interfaz norte del dispositivo. Permite al PCE, el controlador maestro, mapear todos los dispositivos en una red que se administran en un dominio SDN determinado. En una extensión adicional de la funcionalidad PCE, es posible determinar las rutas entre el controlador y el dispositivo por medio de LSP (Labeled Switched Paths - Rutas Conmutadas Etiquetadas) estáticas, enrutamiento segmentario o combinando funciones de servicio. En la imagen 8 se muestra la topología en la que aparece este protocolo de control.

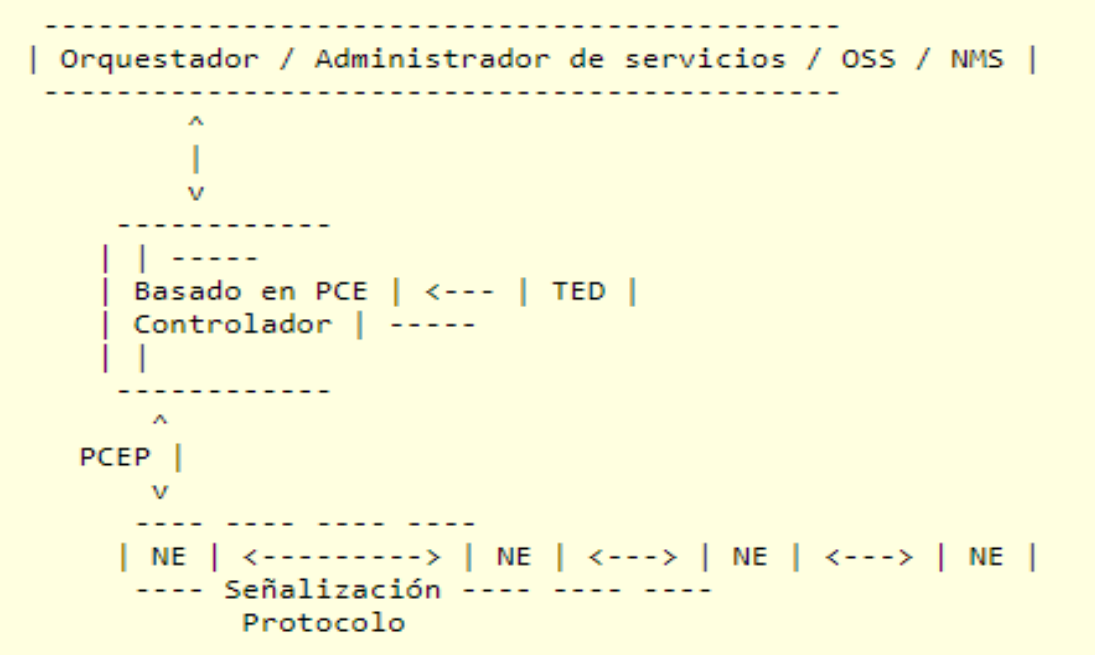


Figura 8. Arquitectura del Controlador Central con un Plano de Control.
Fuente: Arquitectura del Controlador Central, por Alfonso Farrel, [Farrel,2017].

El Componente de Red OSS (Operational Support System). Solicita al controlador central que establezca la conectividad. El controlador PCE calcula las rutas en la red utilizando la topología de la red, los recursos disponibles y otra información admitida por las redes. La información se almacena en la base de datos de ingeniería de tráfico o TED (Traffic Engineering Database en inglés). Posteriormente, el PCE envía solicitudes a los Elementos de Red o NE (Network Elements) utilizando el protocolo PCEP en el siguiente paso, se utiliza una capa de control, mediante la cual se establece la conexión requerida y se reservan los recursos de la red.

PCEP requiere soporte solo en la red de borde a diferencia de OPEN Flow, PCEP es fácil de administrar, implementar y solucionar problemas en cualquier red grande de ISP. (Flachs, 2020, p. 17)

2.1.2. Virtualización de las Funciones de la Red (NFV)

La virtualización de funciones de red mueve las funciones de red del hardware propietario al software cuyo objetivo es transformar la manera en que los operadores de red diseñan sus redes, mediante la evolución de la tecnología de virtualización de servidores, con el fin de consolidar los diferentes tipos de funciones de red, en equipos estándar de propósito general (servidores, conmutadores y dispositivos de almacenamiento), los cuales pueden estar ubicados en centros de datos (DC: Data Centers). (Dorantes Alonso, 2017, p. 4)

Las Funciones de red virtual (VNF: Virtual Networks Functions) se ubica según sea necesario en varios lugares de la red. Esto elimina la necesidad de instalar nuevos equipos patentados. También podemos configurar escenarios híbridos donde las funciones que se ejecutan en recursos virtualizados coexisten con funciones de recursos físicos.

2.1.2.1. Tecnología NFV. Tiene como objetivo un modelo de nube y se creó como una iniciativa de la industria (proveedores de red, operadores móviles) para aumentar la flexibilidad para implementar nuevas funciones y servicios de red. El resultado es una inversión y unos costes operativos más bajos, una puesta en marcha más rápida VNF, mayor retorno de la inversión, apertura de la red virtualizada y más oportunidades para probar nuevas tecnologías.

La meta NFV es cambiar la forma en que se diseña, administra e implementa la infraestructura de red mediante la tecnología de virtualización. La idea principal es unificar muchos dispositivos de hardware en plataformas de servidor, conmutadores y almacenamiento estándar. Tanto en redes fijas como móviles, podemos utilizar tecnología NFV para virtualizar la gran mayoría de dispositivos, servicios y aplicaciones. Luego, implementamos instancias virtuales en centros de datos, nodos de red, puntos finales, etc. Los componentes de red comúnmente virtualizados incluyen: (Mikéska, 2019)

- Elementos de Conmutación: Enrutadores e interruptores de baja potencia, BNG, CG-NAT, DNS.
- Nodos de red móvil: Registros HLR / HSS, puertas GGSN / PDN, nodos NB/ eNB, unidades RNC.
- Tunelización: IPSec/SSL - VPN Gateway.
- Analizadores de tráfico: DPI, medición QoE, vigilancia SLA.
- Señalización NGN: SBC, SOY S.
- Dispositivo de control: Servidores AAA, plataformas de control.
- Optimización a nivel de aplicación: CDN, servidores de caché, equilibradores de carga, aceleradores de aplicaciones.
- Dispositivo de seguridad: Cortafuegos, servidores proxy, sistemas de detección de intrusos, programas antivirus y antispam. (Standards, 2012)

2.1.2.2. Servicios NFV. Las funciones de la red virtual son gestionadas por el sistema de gestión de elementos (EMS: Element Management System). También es responsable de su creación, configuración, monitoreo, desempeño y seguridades EMS.

Además, proporciona información sobre VNF para el sistema de apoyo al tráfico (OSS: Operations Support System), que junto con el sistema de soporte empresarial (BSS: Business Support System) ayuda a los proveedores a implementar y administrar terminales de servicios como pedidos, facturación, renovaciones periódicas de equipos, resolución de problemas, etc. También incluye bases de datos para almacenar información y modelos de datos, que describen el proceso de implementación, el ciclo de vida de funciones, servicios y recursos.

2.1.2.3. Arquitectura NFV. ETSI define la arquitectura NFV en bloques individuales y puntos de referencia entre ellos, lo que permite que diferentes sistemas trabajen juntos. Se puede encontrar una descripción y explicación detallada de todas las partes y puntos de referencias de esta arquitectura en la documentación oficial de ETSI. (ETSI Group (ISG), 2014) O más claramente en el libro (Stallings, 2017)

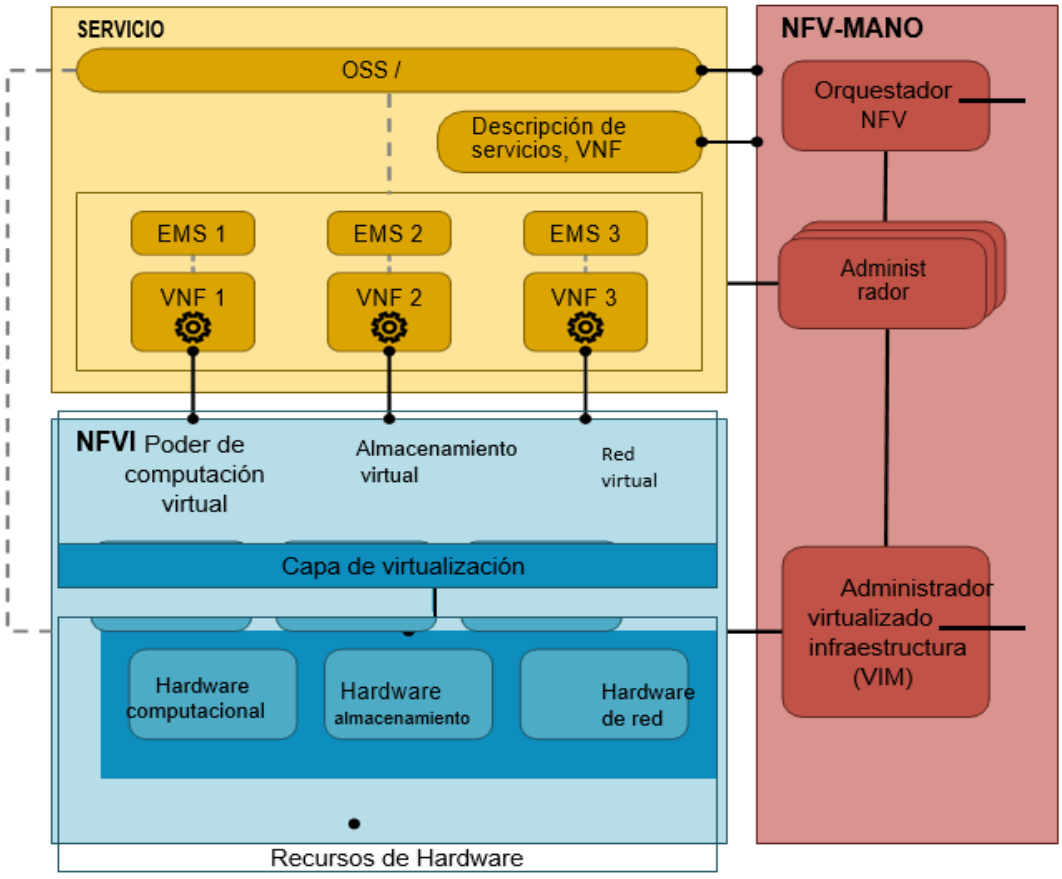


Figura 9. Arquitectura de la Tecnología NFV.
Fuente: El autor - Adaptado de Arquitectura NFV, (p. 46), por Martín Mikéska, [Mikéska, 2019].

Esta parte presenta la arquitectura NFV y su principio, Imagen # 9. muestra los componentes individuales, su interconexión y, sobre todo, la división básica de la arquitectura NFV en tres bloques fundamentales:

- Incluyen un conjunto de funciones de redes virtuales (VNF: Virtualized Networks Functions) que se implementan en una o más máquinas virtuales (VM: Virtual Machine en inglés).
- Se ejecuta en recursos de infraestructura tanto físicos como virtualizados en base a Infraestructuras de virtualización de Funciones de red (NFVI: Network Functions Virtualization Infrastructure).
- La orquestación y gestión de NFV (NFV-MANO: NFV Management & Orchestration).

2.1.2.3.1. ***NFVI.*** (Network Functions Virtualization Infrastructure), Incluye todos los recursos de hardware y software, conexión a centros de datos y nubes públicas y privadas. Los recursos físicos incluyen informática, almacenamiento y hardware de redes que proporciona procesamiento, almacenamiento y conectividad para funciones de redes virtuales. VNF a través de la capa de virtualización. La creación de una capa de virtualización no se especifica estrictamente. Podemos utilizar, por ejemplo, el hipervisor, que simplemente abstrae y divide los recursos físicos en partes lógicas, que luego asigna a las VM individuales que ejecutan VNF. Otra solución es mediar en la capa de virtualización utilizando un sistema operativo que agregue el software apropiado al servidor de hardware en que implementaría VNF como una aplicación.

2.1.2.3.2. ***VNF***. Función de red virtualizada o VNF (Virtualized Network Functions) es la implementación de software de una función de red, capaz de funcionar en la NFVI. Puede ir acompañada de un Sistema de Gestión de Elementos EMS (Element Management System) siempre que sea aplicable a la función concreta, que entiende y gestiona una VNF individual y sus peculiaridades. La VNF es la entidad correspondiente a los nodos de red actuales, que ahora se espera que se entreguen como puro software libre de dependencia del hardware. (ETSI Group (ISG), 2014)

2.1.2.3.3. ***NFV-MANO.*** Administración y orquestación de virtualización de funciones de red (NFV-MANO: NFV Management and Orchestration). Se centra en todas las tareas específicas de virtualización necesarias en un entorno NFV.

NFV MANO tiene tres bloques funcionales principales:

- *Los orquestadores de NFV*. Constan de dos capas, orquestación de servicios y orquestación de recursos, que controlan la integración de nuevos servicios de red y VNF en un marco virtual. Los orquestadores de NFV también validan y autorizan las solicitudes de recursos de infraestructura de NFV (NFVI).

- *Los administradores de VNF.* Supervisan el ciclo de vida de las instancias de VNF.

- *El administrador de VIM.* El gestor de infraestructura virtualizada (VIM: Virtualised Infrastructure Manager) es responsable del control y la gestión de los recursos informáticos, de almacenamiento y red pertenecientes a la NFVI. Proporcionan visibilidad dentro de la gestión de la infraestructura y manejan la administración de recursos.

Juntos, estos bloques son responsables de implementar y conectar funciones y servicios cuando se necesitan en toda la red.

El Grupo de Especificación de la Industria del Instituto Europeo de Estándares de Telecomunicaciones (ETSI) (ISG NFV) definió la arquitectura MANO para facilitar el despliegue y la conexión de servicios a medida que se desacoplan de los dispositivos físicos dedicados y se trasladan a máquinas virtuales (VM).

En la Figura 9, relacionado con la arquitectura de la tecnología de la virtualización de las funciones de la red también se observan las interfaces que conectan los componentes de los bloques funcionales del marco arquitectural NFV. Las VNFs serán controladas tanto por el sistema de gestión de elementos (EMS) como por la NFV-MANO, y la capa de virtualización ofrecerá los recursos físicos de las ubicaciones elegidas por NFV-MANO a las VNFs. (Mikéska, 2019, p. 47)

2.1.2.4. Relación entre NFV y SDN. Ambas tecnologías están estrechamente relacionadas, se complementan, pero no dependen una de la otra. Sobre todo, aporta una administración más sencilla, una mayor flexibilidad y una innovación más rápida. El concepto NFV se puede implementar sin el uso de tecnología SDN. Sin embargo, al combinarlos, podemos lograr mejores resultados y aumentar el potencial general de ambas soluciones.

Las diferencias generalmente se pueden describir mediante estos puntos:

- NFV se trata de virtualización de dispositivos de red, mientras SDN se trata de virtualización de redes.
- NFV transfiere funciones de servicio de red desde dispositivos físicos propietarios a servidores virtualizados, mientras SDN separa la parte de control y de datos del dispositivo de red.
- NFV opera las funciones de red de forma virtual, pero SDN se mueve El control funciona en un controlador centralizado y proporciona una arquitectura programable.
- NFV reduce CapEx, OpEx, consumo de energía y espacio. SDN proporciona abstracciones de red de funciones de capa inferior, lo que permite una gestión de red flexible y una innovación más rápida.

2.1.3. Aplicaciones de SDN-NFV

Nos vamos a enfocar en dos aplicaciones específicas que se ha planteado en este proyecto como casos de uso de la virtualización de funciones de red que demuestran cómo se utiliza actualmente el VNF para hacer frente a una serie de retos, así como para proporcionar soluciones mejoradas a estos y otros obstáculos de la red con el fin de mejorar los servicios y reducir los gastos.

2.1.3.1. Virtualización de Redes. Es el uso principal con el que las tecnologías NFV y SDN están siendo utilizadas por las empresas de telecomunicaciones en el mundo para lograr mejorar la capacidad y servicios de los sistemas de comunicaciones.

La tecnología de virtualización de redes eleva la capacidad de las redes definidas por software o SDN (Software-Defined Networking), gracias a que desvincula por completo los recursos de redes del hardware subyacente. Permite a los proveedores expandir y acelerar el desarrollo y la innovación de los servicios.

También ayuda a mejorar los requisitos críticos de la red, como el aprovisionamiento. Cada red virtual es un conjunto de nodos y enlaces virtuales denominados Entorno de Virtualización de Red (NVE). Una de las características más importantes de una red virtual es el soporte de protocolos y servicios que se ejecutan simultáneamente para una aplicación específica. (Lanner, 2019)

2.1.3.1.1. *Red de Área Local Virtual (VLAN).* Una VLAN (Red de área local virtual o LAN Virtual) es un método para crear redes lógicas independientes dentro de una misma red física. Gracias a las redes virtuales (VLAN), es posible liberarse de las limitaciones de la arquitectura física (geográficas, de direccionamiento, etc. Ya que se define una segmentación lógica basada en el agrupamiento de equipos según determinados criterios (direcciones MAC, número de puertos, protocolos, subred, el acceso, condiciones de seguridad del equipo, etc.).

Las VLANs están definidas por los estándares IEEE 802.1D, 802.1p, 802.1Q y 802.10.

2.1.3.1.2. *Redes Virtuales Privadas (VPN*). Una red privada virtual de las siglas en inglés de VPN (Virtual Private Network), es una tecnología de red que permite superponer una extensión segura de la red local (LAN) sobre una red pública o no controlada como internet, sin necesidad de que sus integrantes estén físicamente conectados entre sí, sino a través de Internet. Las conexiones VPN vienen acompañadas de un cifrado de los paquetes que se transmiten con ellas. Permite que los dispositivos en la red envíen y reciban datos sobre redes compartidas o públicas como si fuera una red privada con toda la funcionalidad, seguridad y políticas de gestión de una red exclusiva confidencial. (Fernádez, 2015, p. 6)

Básicamente existen cuatro arquitecturas de conexión VPN:

- VPN de Acceso Remoto (Remote Access VPN). Esta conexión es segura y se realiza a través de internet mediante un Remote Access Server. Permite a los usuarios conectarse a una red privada para acceder a servicios y recursos de forma remota. Una vez autenticado el cliente de VPN tendrá un nivel de acceso muy similar al que tienen en la red local y podrá conectarse sin importar el lugar que se encuentre físicamente geolocalizado.

- VPN de Sitio a Sitio (Site-to-Site VPN). El servidor VPN posee un vínculo permanente a Internet, acepta las conexiones vía Internet provenientes de los sitios y establece un túnel VPN entre las redes, en oficinas geográficamente distantes, y las conecta a través de internet para mantener una comunicación segura y privada entre las redes. La VPN de sitio a sitio se basa en la comunicación de enrutador a enrutador; uno actúa como un cliente VPN y otro enrutador como un servidor VPN. La comunicación entre los dos enrutadores comienza después de validar una autenticación entre ellos.

- Tunneling. La técnica de tunneling consiste en encapsular un protocolo de red sobre otro (protocolo de red encapsulador) creando un túnel dentro de una red de computadoras. El establecimiento de dicho túnel se implementa incluyendo una PDU (unidades de datos de protocolo) determinada dentro de otra PDU con el objetivo de transmitirla desde un extremo al otro del túnel sin que sea necesaria una interpretación intermedia de la PDU encapsulada.

- VPN Over LAN. Una aplicación realmente desconocida pero muy útil y potente consiste en establecer redes privadas virtuales dentro de una misma red LAN. Una aplicación muy típica de este modelo se utiliza para aumentar la seguridad en redes de acceso inalámbrico, separándolas así de la red física para evitar posibles fugas de información o accesos no autorizados. Sirve para aislar zonas y servicios de la red interna. (Fernádez, 2015)

Protocolos utilizados en VPN:

Para hacer referencia a los aspectos técnicos de los Protocolos utilizados en las VPN, citaremos los más idóneos para efectuar una solución de comunicación privada virtual. De esta forma veremos un breve análisis de sus principales características según el escenario de aplicación.

- IPSec o Internet Protocol Security: Es un Protocolo de Seguridad en Internet, es una extensión del tradicional protocolo IP para redes VPN. Es lo suficientemente seguro para ser usado por empresa para conectar sucursales o a sus usuarios de forma remota. Cifrará cualquier conexión, verificando cada sesión y codificando individualmente los paquetes por lo que asegura privacidad e integridad de los datos.

- L2TP o Layer 2 Tunneling Protocol: Es un protocolo de túnel utilizado para soportar la red privada virtual (VPN) para encapsular los datos que a su vez utilizarán IPSec para encriptarlos y enrutarlos por la red. A este método se le llama línea virtual ya que es de uso sencillo y el encabezado del paquete cuenta con la información suficiente como IP para que el servidor identifique al usuario.

- PPTP o Point-to-Point Tunneling Protocol: Protocolo de túnel punto a punto, su función es encriptar y encapsular los paquetes con el protocolo IP de una forma simple. Es uno de los protocolos más rápidos, aunque con una protección menos contundente que los anteriores por tener una encriptación más frágil.

- OpenVPN: Es una tecnología de código abierto que utiliza la biblioteca de OpenSSL y el protocolo de Seguridad de la capa de transporte (TLS: Transport Layer Security). Además de ser un software cliente para conectarnos a una VPN, también es un protocolo de red punto a punto que permite establecer el túnel entre el cliente servidor utilizando OpenSSL para la encriptación. Además, es capaz de usar los protocolos de transporte TCP o UDP para la transmisión de los datos. (Profesionalreview, 2021)

2.1.3.1.3. ***Redes Activas y Programables.*** Las redes activas (Active Networking) orientadas hacia el control de la red, conceptualizando una interfaz de programación (API) que expone los recursos (procesamiento, almacenamiento, colas de paquetes, etc.) en nodos de red individuales y soporta la construcción de funcionalidades personalizadas para aplicar a un subconjunto de paquetes que pasan a través del nodo.

2.1.3.1.4. ***Redes Overlay***. Una red Overlay también llamada superpuesta, es una red virtual de nodos encapsulados y enlazados lógicamente, que está construida sobre una o más redes subyacentes partiendo de una infraestructura de red LAN/WAN que funcionará como base, admitiendo que es una red donde todos los componentes activos son de capa 3, es decir, son enrutadores que implementan algún protocolo dinámico como OSPF, EIGRP o IS-IS. Un ejemplo de este tipo de red virtual es la LAN Extensible Virtual ó VXLAN (Virtual Extensible LAN).

VXLAN (LAN Extensible Virtual)

Es el protocolo más utilizado para crear redes superpuestas que se encuentran por encima de la red física y que permiten el uso de redes virtuales. Para la red Overlay, VXLAN describe el protocolo de encapsulación y los procesos de creación de túneles necesarios para su funcionamiento.

VXLAN es un método de superposición que extiende el tráfico en un esquema de expansión de redes de capa 2 sobre una infraestructura de capa 3 encapsulando tramas de capa 2 en paquetes UDP. A través de su encapsulación es posible aumentar considerablemente el número de redes que se desea transportar, sin importar el alcance geográfico, admite la virtualización de la red del centro de datos así mismo aborda sus necesidades en múltiples inquilinos proporcionando la segmentación necesaria y una red superpuesta a gran escala.

Más allá de simplemente dividir una red en subgrupos, las VXLAN pueden virtualizar una red entera valiéndose de su identificador de segmento de 24 bits, conocida como VNI (VXLAN Network Identifier), esta extensión de la funcionalidad nos permite alcanzar un máximo de 16 millones de redes VXLANs posibles que pueden converger en una misma implementación, permitiéndonos aumentar en gran medida la capacidad y la escalabilidad de las redes virtuales VLAN (Virtual LAN) que trae consigo refiriéndonos al concepto tradicional de VLANs bajo el estándar 802.1Q que tiene una limitación en la cantidad de VLANs que se pueden crear. Esta capacidad está restringida por el tamaño del identificador de su segmento el cual es de 12 bits, por lo que el límite máximo de VLANs se ubica en 4096 VLANs disponibles. Lo cual es especialmente importante para los entornos virtuales sin contar las complejas arquitecturas de nubes actuales.

Por otra parte, VXLAN, tal como ya hemos dicho, propone la creación de una red virtual capa 2 (L2) sobre una capa inferior de capa 3 (L3), lo que otorga un nuevo nivel de abstracción, en este nivel se mantiene la independencia entre las redes virtuales creadas o VNs (Virtual Network). Esto es, que no existirá comunicación entre los elementos de diferentes VNs salvo que una regla de seguridad lo permita de forma explícita y un enrutador la implemente.

Además, se considera la posibilidad de definir una VN entre los dispositivos que se encuentran ubicados en diferentes centros de datos. Así podríamos tener dos máquinas virtuales ubicadas en dos centros de datos diferentes pero que están integradas a la misma red virtual o a la misma VXLAN.

Una de las ventajas de VXLAN es que a través de su VNI se puede aislar el tráfico de cada segmento de red, sin importar si una dirección MAC se encuentra repetida o duplicada al interior de cada segmento. Esto se debe porque su VNI encapsula la dirección física de cada VM en una trama que finalmente procesa el módulo VTEP, de esta forma los switches no entrarán en conflicto cuando se ejecuten los procesos sobre su tabla MAC. (MORENO & GARCIA, 2019, pág. 13)

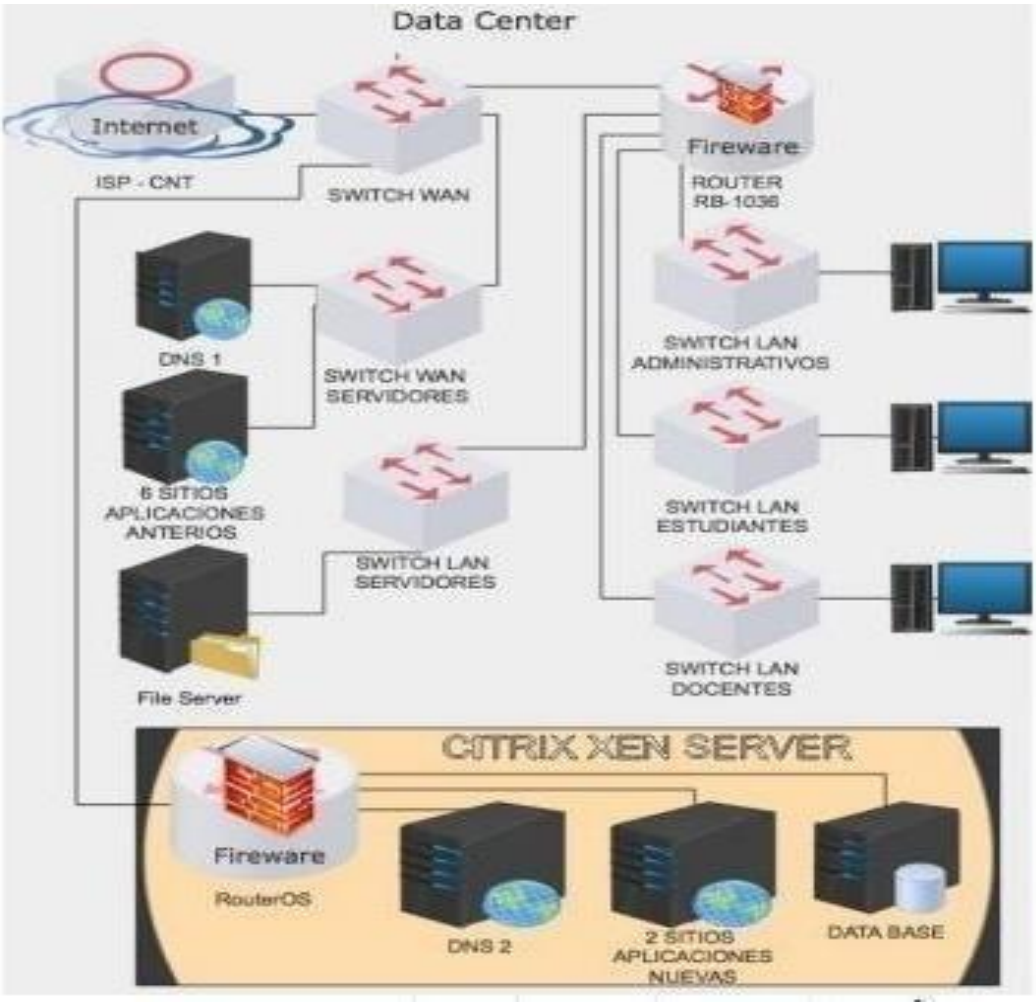


Figura 10. Escenario de Redes Virtuales, (Print Screen).
Fuente: Virtualización de Redes y Servidores, J. Casierra, 2018, Revista Científica.

2.1.3.2. Virtualización de Almacenamiento. La virtualización del almacenamiento es el proceso de consolidar varios dispositivos físicos en distintos volúmenes de almacenamiento virtualizado a partir de distintos sistemas de soluciones definidos por software de diversos fabricantes, reorganizándolos en agrupamientos virtuales que implica archivar, organizar y distribuir los datos según se requiera, además de lógicos, o en unidades de almacenamiento. La virtualización de almacenamiento, de acuerdo con el lugar en que se realice en sí, se puede clasificar en dos grupos, pudiendo ser: virtualización basada en dispositivo y en la red.

2.1.3.2.1. *Virtualización basada en Dispositivo*. En este tipo, la virtualización se realiza en arreglos de dispositivos de almacenamiento. Cada host dispone de un dispositivo virtual que se encuentra relacionado con una ubicación física dentro del arreglo de dispositivos, como discos duro y las unidades de estado sólidos SSD (Solid State Drive).

2.1.3.2.2. ***Virtualización basada en Red.*** En este modelo, la virtualización se realiza en la misma red, en la cual se emplean switches inteligentes u otros equipos de virtualización; en redes como SAN (Red de Área de Almacenamiento), NAS (Almacenamiento Conectado a Red), y DAS (Almacenamiento de Conexión Directa). (Lugo C, 2014)

Los ambientes virtualizados serán el futuro de las redes de comunicaciones, por lo cual, podemos afirmar que existen varios protocolos que cumplen con los estándares de calidad que los clientes exigen y necesitan.

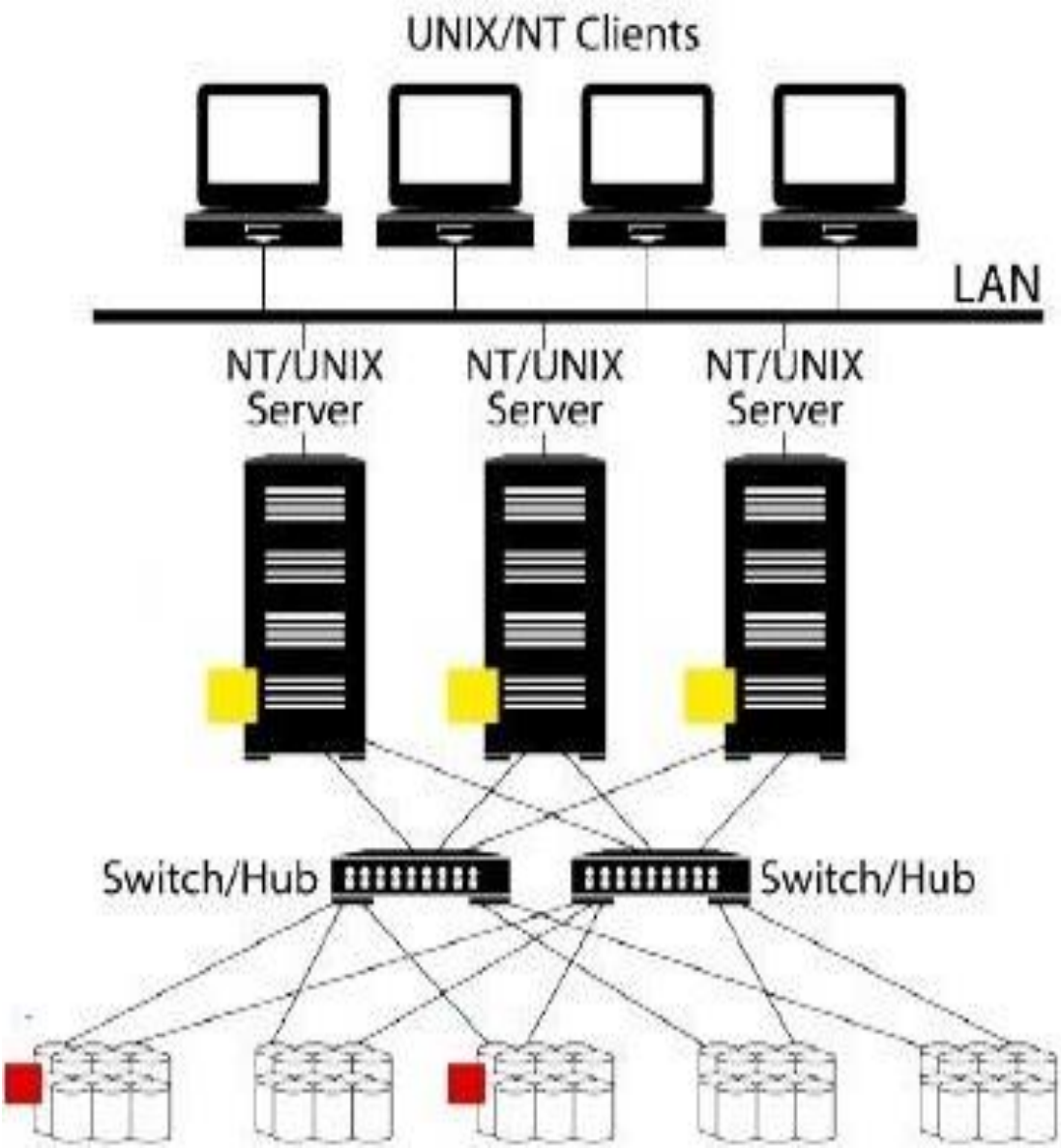


Figura 11. Virtualización de Almacenamiento, (Print Screen).
Fuente: https://sites.google.com/site/wwwvirtualizacioneducom/

2.1.4. Integraciones SDN

2.1.4.1. SD-WAN (Software-Defined Wide Área Network). Son las Redes de Área Global Definidas por Software, es un enfoque definido por software para la administración de redes WAN, aprovechando el poder de la digitalización para simplificar la gestión de la red, aumentar la velocidad en la ejecución de las aplicaciones, la optimización de la experiencia y la eficiencia del usuario, la mejora del rendimiento y la agilidad de las aplicaciones y, lo que es más importante, la reducción de los costes relacionados con la independencia de los flujos de datos.

El aumento del tráfico en Internet obliga a las empresas a optimizar el rendimiento de las redes WAN y las aplicaciones que se ejecutan en estas. Ante esta necesidad, se están imponiendo las redes definidas por software SD-WAN.

Una de las razones de la necesidad de fortalecer la infraestructura WAN es, por ejemplo, la transición a SaaS (Software as a Services), IaaS (Infrastructure as a Services) y PaaS (Platform as a Services). Es un nuevo enfoque para utilizar servicios, infraestructura y plataformas a través de una red de datos, Internet. Con la apertura de Internet a nuevas tecnologías, se están creando al mismo tiempo posibles amenazas a la seguridad y requisitos de cumplimiento.

Con la llegada de SD-WAN, la administración de la red se puede simplificar con una arquitectura WAN centralizada que facilita el escalado de los puntos finales donde quiera que estén conectados a través del bloque de red administrado. Otra ventaja es la optimización del rendimiento de las aplicaciones de usuario que se ejecutan en la nube. En el caso de una falla en la conexión, el tráfico de datos entre las conexiones dedicadas se ajusta dinámicamente para garantizar un funcionamiento ininterrumpido, que muy a menudo se marca como crítico. Parte de este concepto es también el enfoque en la seguridad, que se proporciona a nivel de sucursal. La comunicación con las sucursales individuales se realiza mediante túneles encriptados del tipo VPN.

Las plataformas de gestión de flujo existen en varios diseños, desde cajas físicas, pasando por servicios de red virtual, hasta instancias de dispositivos de red en la nube.

2.1.4.1.1. *Ventajas de SD-WAN.* Las principales ventajas de usar SD-WAN, aparte de mejorar el rendimiento de las aplicaciones, es que permite administrar de mejor manera el ancho de banda, adaptándolo a las necesidades y prioridades de las empresas de forma totalmente personalizada. Todo este control y gestión se realiza desde una única plataforma centralizada, lo que permite, gracias a esto, que cualquier cambio, se aplique casi inmediatamente.

Otra ventaja relevante que aportan las redes WAN definidas por software es la seguridad en las comunicaciones, ya que los datos están cifrados de extremo a extremo, y están supervisadas todas las comunicaciones por un firewall totalmente gestionable. La reducción de los costes tanto para el proveedor de telecomunicaciones como para la

empresa, ya que por el lado del ISP necesita menos hardware específico para poder ofrecer al cliente sus necesidades contractuales, y para el cliente se traduce en una reducción del coste a pagar mensualmente.

La rapidez de cualquier cambio que se necesite en cuanto a comunicaciones de las compañías es casi instantánea, ya que, al estar todo gestionado por software, centralizado todo desde un solo lugar, las organizaciones se benefician de una solución que aporta agilidad a los procesos, reduce costes, mejora la seguridad, ofrece confiabilidad y proporciona un alto rendimiento. Todo esto para conseguir adaptarse a los nuevos desafíos que presenta el panorama digital actual. (Oscar, 2021)

2.1.4.1.2. ***Seguridad en SDWAN.*** Con el paso de los años y la evolución en las telecomunicaciones, y el traspaso de todos los sistemas físicos hacia la nube ha propiciado que las empresas se den cuenta que los sistemas WAN actuales tienen falencias en la generación de redes actuales, y en las que se aproximan con el tráfico generado por derivar todo a la nube.

El aumento de tráfico en la última década ha dado lugar a la complejidad de la gestión de los datos y la falta de rendimiento de aplicaciones, junto a la vulnerabilidad de los datos. Esto se ha agravado más debido a que cada vez hay más empresas que dependen de Internet y la nube para todas sus operaciones diarias.

Se ha vuelto extremadamente difícil proteger los diferentes recursos de una empresa debido a la cantidad de datos y accesos distintos que son necesarios a cualquier nivel. Gracias a la posibilidad de poder configurar y distribuir el ancho de banda en la red WAN, y filtrar todos los datos por las diferentes VPN que usa SD-WAN, se está facilitando el control y seguridad para las empresas.

En los sistemas usados por las empresas, normalmente un firewall o antivirus gestionado desde un servidor central se encargaba de gestionar todo el tráfico, controlándolo desde un solo lugar, en cambio con el sistema SD-WAN, los datos no tienen que regresar al servidor central de seguridad donde esté ubicado, ya que cada lugar es seguro sin tener que depender de un servidor principal. (Oscar, 2021)

2.1.4.1.3. ***Rendimiento de SDWAN.*** La arquitectura de WAN que hasta ahora usábamos en diferentes sedes o empresas del mismo grupo es limitada en cuanto a funcionalidades o configuraciones. A raíz de más implementaciones de aplicaciones en la nube, la arquitectura WAN ha experimentado un aumento del tráfico que repercute negativamente en el rendimiento de aplicaciones o traspaso de información. Además, debido al bajo rendimiento o colapso que puede producirse por el aumento de tráfico, los gastos derivados a los servicios WAN han aumentado exponencialmente debido a la falta de una gestión correcta del ancho de banda utilizado.

La única tecnología que actualmente permite a las empresas gestionar todos esos problemas de rendimiento en la red es SD-WAN, ya que permite la gestión del

ancho de banda de manera eficiente, y, además, ofrece un mejor enrutamiento, protección contra amenazas y simplificación de la administración de redes WAN. Pero no acaba aquí las ventajas, además de lo expuesto anteriormente, añade las siguientes preeminencias de rendimiento que se resumen en:

- Disponer de alta disponibilidad, con SLA predecible, en todas las aplicaciones que use la empresa, para así poder saber en qué momento es necesario más ancho de banda y para qué aplicación es.
- Disponer de enrutamiento dinámico del tráfico de aplicaciones para adaptar el ancho de banda necesario a cada aplicación.
- Disponer de seguridad integrada, sin tener que depender de un solo servidor que gestione todo el tráfico y seguridad, optimizando así, los costes y rendimiento de la red.
- Disponer de políticas de reconocimiento de aplicaciones en tiempo real, para detectar posibles problemas en la red y poder actuar antes de que suceda cualquier problema
- Todo el tráfico es seguro a través de Internet debido al uso de SD-WAN.
- Las aplicaciones que usen la nube están optimizadas para un mayor rendimiento.
- Puede realizar cualquier ampliación de la WAN sin contratiempos, y que los cambios de apliquen en tiempo real.
- Un único panel de administración centralizado para realizar toda la configuración y administración, para poder gestionar todo de manera sencilla, rápida y eficaz. (Oscar, 2021)

2.1.4.2. SD-Access (Acceso Definido por Software). Es una de las primeras soluciones de red basadas en las intenciones de empresas e instituciones. Esta solución proporciona servicios automatizados de extremo a extremo como segmentación, calidad de servicio y análisis de tráfico para usuarios, dispositivos y para operación de la aplicación.

Al mismo tiempo, automatiza las políticas de usuario, por lo que las organizaciones pueden proporcionar un control de acceso adecuado en los puntos finales administrados. Los derechos y las políticas se establecen para usuarios, dispositivos y aplicaciones específicas en la red.

La configuración de la funcionalidad de la red dentro de la organización está controlada por un elemento central. Los beneficios de SD-Access incluyen automatización plug-and-play, segmentación de red automatizada y asignación de políticas de grupo e interfaces de control abiertas y programables para la integración con aplicaciones de terceros. El enfoque SD-Access brinda la capacidad de virtualizar el tráfico de red en una infraestructura física.

2.1.4.3. SD-Branch (Rama SD). Es una nueva forma de extender los principios definidos por software a las sucursales. Cuenta con hardware simplificado, administración centralizada remota y automatización a través de la capacidad de programación. Todos los sitios que integran este concepto se pueden gestionar de forma remota desde una ubicación central utilizando un único sistema de control. Otra ventaja es la capacidad de ejecutar diagnósticos y luego realizar mantenimiento en toda la sección administrada de la red.

Desde el punto de vista de la seguridad, se utilizan tecnologías más avanzadas para la protección perimetral. Se utilizan NGFW (firewall de próxima generación), sistemas de prevención de intrusiones (IPS), herramientas de detección de malware o herramientas de repulsión de denegación de servicio distribuido (DDoS).

Se perfila como un sucesor natural en la evolución de las redes de área amplia definidas por software (SD-WAN), SD-branch simplifica el proceso de ramificación de redes al colapsar múltiples funcionalidades de red en una sola plataforma. Las empresas suelen implementar sucursales SD en ubicaciones totalmente nuevas o como una actualización/reemplazo de las cajas de las sucursales (Bobs) en sucursales más pequeñas, como oficinas regionales, tiendas, bancos, etc.

Se espera que muchas organizaciones eliminen gradualmente sus enrutadores de sucursales existentes en favor de paquetes SD-WAN durante los próximos años y los clientes tendrán la opción de comprar servicios de sucursales SD con soporte para funciones de red más amplias, incluido SD-WAN, enrutamiento, seguridad y wifi. Para las organizaciones que eligen SD-WAN administrada de un proveedor de servicios, el proveedor de servicios definirá las opciones. (Flachs, 2020, pág. 5)

Entre los principales proveedores involucrados en la creación de productos y servicios de SD encontramos a Cisco, VMware, Dell, Nokia, Juniper Networks, HPE Aruba, Cradlepoint, Citrix, Silver Peak, VMware, Talari y Versa Networks entre otros.

2.1.5. Perímetro Definido por Software (SDP).

Una nueva tecnología segura de acceso remoto llamada SDP o (Software Defined Perimeter en inglés) está atrayendo la atención como sucesora de la VPN. El (SDP) es un nuevo enfoque de seguridad cibernética que mitiga los ataques basados en red. Protege tanto los activos de TI heredados como los servicios en la nube de todos los niveles de clasificación. Funciona ocultando los activos críticos de TI dentro de una nube invisible e indetectable. (ebizLatam, 2020)

En términos simples, un perímetro definido por software es una tecnología basada en una arquitectura Zero Trust que puede limitar el acceso de un dispositivo a aplicaciones y servicios basándose en docenas de criterios configurables. Para dispositivos fuera de la red corporativa, una solución SDP tiene la capacidad de crear conexiones 1-1, o “micro” túneles, entre los usuarios y los recursos que necesitan.

2.2. Raspberry Pi

La Raspberry Pi es una serie de ordenadores de placa reducida, única y de bajo coste, desarrollado en Reino Unido por la Raspberry Pi Foundation, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital, abre las puertas de la experimentación en proyectos de electrónica. (Raspberry Pi, Foundation, n.d.)

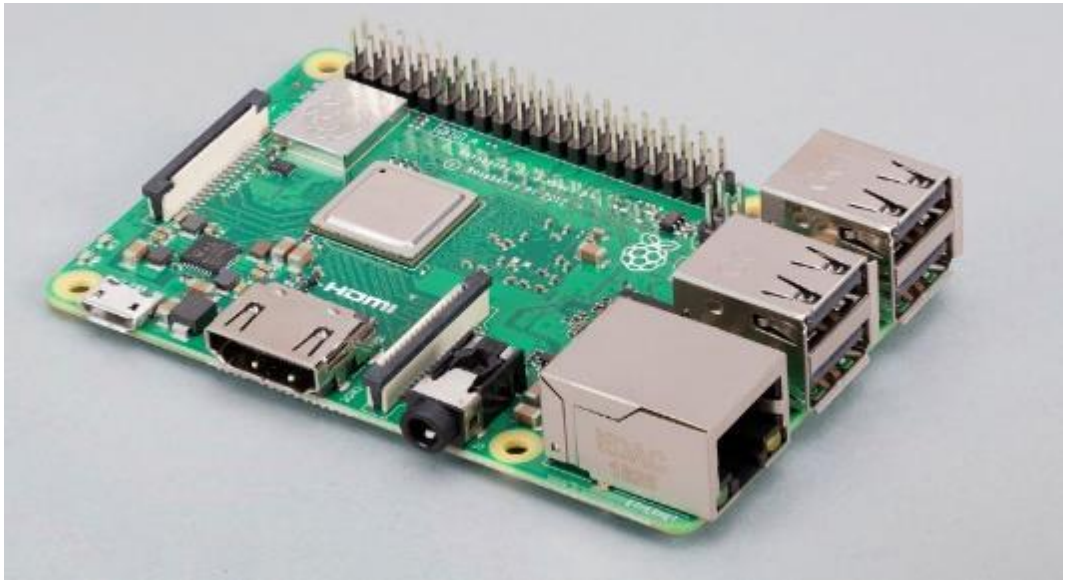


Figura 12. Raspberry Pi 3B+, (Print Screen).
Fuente: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

El diseño electrónico incluye un chip Broadcom de la serie BCM, que contiene una unidad de procesamiento central (CPU) denominado ARM (Advanced RISC Machine) o máquina RISC avanzada, la combinación de la arquitectura RISC (Simple Reduced Instruction Set) y su bajo consumo energético lo convierten en la opción perfecta frente a los chips de computadoras de escritorio que demandan altos consumos en arquitecturas CISC (Complex Instruction Set).

Todo esto, sin embargo, significa que la Raspberry Pi no es compatible con el software de los PC tradicionales. La mayoría del software para ordenadores de escritorio y portátiles se construyen teniendo en cuenta la arquitectura de conjunto de instrucciones x64, presente en los procesadores como AMD, Intel y VIA. Por consiguiente, este software (x64) no funciona en la Raspberry Pi por ser de arquitectura ARM.

La Raspberry pi Foundation da soporte para las descargas de las distribuciones para la arquitectura ARM, Raspbian (derivada de Debían), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora). Promueve principalmente el aprendizaje del lenguaje de programación Python, Java, TI Basic y C.

2.2.1. Característica y Especificaciones Técnicas de la Raspberry Pi

El BCM serie 2837 basado en ARM es el secreto que explica cómo la Raspberry Pi es capaz de funcionar con tan sólo una fuente de alimentación de 5V 1A suministrada por el puerto micro-USB a bordo. Es también la razón por la cual no hay ningún disipador térmico sobre el dispositivo: el bajo consumo de energía del chip se traduce directamente en muy poco calor residual, incluso durante las tareas de procesamiento más complejas, un procesador grafico (GPU) Video Core, la memoria RAM LPDDR (abreviatura de Low-Power Double Data Rate), también conocida como Low-Power DDR SDRAM o LPDDR SDRAM, es un tipo de memoria de acceso aleatorio dinámico síncrona de doble velocidad de datos que está destinada a dispositivos de bajo requerimientos.

El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta microSD para el almacenamiento permanente y el arranque del sistema operativo. La alimentación eléctrica está dada por un transformador adaptador de 110VAC a 5VDC.

Raspberry pi Modelo B+, cuenta con conectores suficientes para permitir que el usuario pueda tener diferentes opciones de desarrollo en laboratorios de electrónica con un diseño absoluto de placa de microordenador.

Aquí se presentan sus especificaciones técnicas:

- Procesador Broadcom BCM2837B0, Cortex-A53 (ARMv8) SoC de 64 bits a 1,4 GHz.
- RAM SDRAM LPDDR2 de 1 GB.
- LAN inalámbrica IEEE 802.11.b / g / n / ac de 2,4 GHz y 5 GHz(Wi-Fi).
- Bluetooth 4.2, BLE.
- Un conector de RJ45 conectado a un integrado lan 9512 -jzx de SMSC que nos proporciona conectividad en el orden de los Mbps sobre cables ethernet.
- 4 puertos USB 2.0 (rendimiento máximo 300 Mbps).
- Cabecera GPIO extendida de 40 pines.
- HDMI de tamaño completo.
- Puerto de cámara CSI para conectar una cámara Raspberry Pi.
- Puerto de pantalla DSI para conectar una pantalla táctil Raspberry Pi.
- Salida estéreo de 4 polos y puerto de video compuesto.
- Puerto microSD para cargar su sistema operativo y almacenar datos.
- Entrada de alimentación de 5 V / 2,5 A CC.
- Soporte Power-over-Ethernet (PoE) (Requiere PoE HAT separado).
- Conector de audio Jack 3,5 mm.
- Conector hembra para buses serie y GPIO.
- Puerto para conector JTAG.
- Conector para videocámara HD Raspberry Pi (775-7731).
- Dimensiones: 86 x 56 x 20 mm (modelo B+).

En la imagen 13 que se muestra a continuación, se observa cada una de las partes, puertos y ranuras de la tarjeta Raspberry Pi.

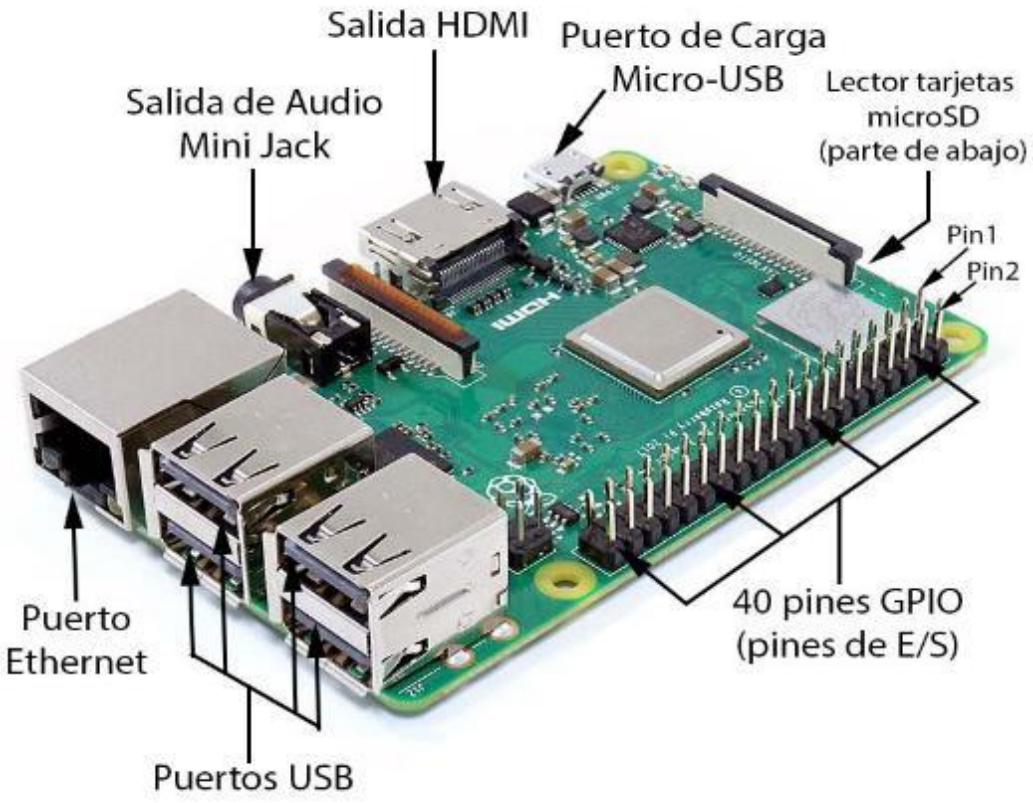


Figura 13. Puertos de la Raspberry Pi 3B+, (Print Screen).
Fuente: https://www.arcadexpress.com/blog/wp-content/uploads/2020/01/raspb_detalle3.jpg

2.2.2. Puesta en Marcha de la Raspberry Pi

2.2.2.1. La Tarjeta SD. Debido a una limitación de hardware en Raspberry Pi Zero, 1 y 2, la partición de inicio en la tarjeta microSD debe ser de 256 GB o menos; de lo contrario, el dispositivo no se iniciará. Los modelos posteriores de Raspberry Pi 2 con SoC BCM2837, Raspberry Pi 3, 4 y 400 no tienen esta limitación. Esto no afecta al sistema operativo Raspberry Pi, que siempre usa una pequeña partición de arranque.

2.2.2.2. Raspberry Pi Imager. Es la herramienta que permite la instalación del Sistema Operativo de Raspberry Pi, permitirá crear dispositivos de arranque de diversa índole a partir de unidades de disco externas como microSD, descargue e instale la aplicación para el software recomendado de la computadora que deberá contar con un lector de tarjetas SD.

La imagen 14 muestra la GUI de la Herramienta Raspberry pi Imager.

.

Figura 14. Herramienta Raspberry Pi Imager, (Print Screen).
Fuente: Fuente: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

2.2.2.3. Imagen de Raspberry Pi. Las imágenes oficiales de los SO están disponibles para descargar desde la página de descargas del sitio web de raspberrypi.org. El sistema operativo Rasbian con imagen de escritorio y software recomendado es el escogido para nuestro proyecto contenida en el archivo ZIP tiene un tamaño superior a 4 GB, descomprimida para obtener la imagen iso.

Figura 15. Sistema Operativo Raspberry Pi, (Print Screen).
Fuente: Fuente: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

2.3. ZeroTier

El servicio de virtualización de red ZeroTier, es un software de red de código abierto, construido sobre una red global peer to peer criptográficamente segura de extremo a extremo.

Utiliza algunos de los últimos desarrollos en materia de SDN para permitir a los usuarios crear redes seguras y manejables, y tratar los dispositivos conectados como si estuviesen en la misma ubicación física. Proporcionando capacidades de gestión y virtualización de red avanzadas a la par con un conmutador SDN, pero en redes de área amplia, conectando aplicaciones o dispositivos que se encuentre en su red.

ZeroTier proporciona una consola web para software de administración de redes y de extremos para los clientes. Es una tecnología punto a punto cifrada, lo cual significa que a diferencia de lo que sucede con las soluciones de VPN tradicionales, las comunicaciones no tienen que pasar a través de un servidor o enrutador central y los mensajes se envían directamente de host a host.

En consecuencia, es muy eficiente y garantiza una latencia mínima. Entre otros beneficios, se incluyen el proceso de implementación y configuración sencillo de ZeroTier, su fácil mantenimiento y el hecho de que permite registrar y administrar de forma centraliza los nodos autorizados a través de la consola web. (github.com, act.2021)

.

Figura 16. ZeroTier, (Print Screen).
Fuente: https://www.zerotier.com/

2.3.1. Generalidades del Controlador Zerotier

El controlador de red ZeroTier (que actualmente se encuentra en el nodo / subcarpeta del repositorio ZeroTier One) es un motor de virtualización de red autónomo que implementa una capa de virtualización Ethernet similar a VXLAN en la parte superior de una red peer to peer cifrada global.

El protocolo ZeroTier es original, aunque sus aspectos son similares a VXLAN e IPSec. Tiene dos capas conceptualmente separadas, pero estrechamente acopladas en el sentido del modelo OSI: **VL1** y **VL2**.

*VL1 es la capa de transporte de igual a igual subyacente, el "cable virtual",
*VL2 es una capa de Ethernet emulada que proporciona a los sistemas operativos y aplicaciones un medio de comunicación familiar. Un centro de datos global requiere un armario de cableado global.

En las redes convencionales, L1 (capa 1 de OSI) se refiere a los UTP CAT5 / CAT6 o canales de radio inalámbricos reales por los que se transportan los datos y los chips transceptores físicos que los modulan y demodulan. VL1 es una red de igual a igual que hace lo mismo mediante el uso de cifrado, autenticación y muchos trucos de red para crear cables virtuales dinámicos según sea necesario.

2.3.2. Topología de Red Zerotier

VL1 está diseñado para ser de configuración cero. Un usuario puede iniciar un nuevo nodo ZeroTier sin tener que escribir archivos de configuración o proporcionar las direcciones IP de otros nodos. También está diseñado para ser rápido. Dos dispositivos en el mundo deberían poder ubicarse y comunicarse casi instantáneamente. Para lograr esto, VL1 está organizado como DNS. En la base de la red hay una colección de servidores raíz siempre presentes cuya función es similar a la de los servidores de nombres raíz DNS. Las raíces ejecutan el mismo software que los puntos finales normales, pero residen en ubicaciones rápidas y estables en la red y están designadas como MUNDO. Las definiciones de MUNDO vienen en dos formas: PLANETA y uno o más LUNAS.

El protocolo incluye un mecanismo seguro que permite actualizar el MUNDO en banda si cambian las direcciones IP de los servidores raíz o las direcciones ZeroTier.

Solo hay un planeta. Los servidores raíz de mundo son operados por ZeroTier, Inc. como un servicio gratuito. Actualmente hay doce servidores raíz organizados en dos clústeres de seis miembros distribuidos en todos los continentes principales y múltiples proveedores de red. Casi todo el mundo tiene uno con una latencia de red de menos de 100 ms desde su ubicación. Un nodo puede "orbitar" cualquier número de LUNAS. Una Luna es solo una forma conveniente de agregar servidores raíz definidos por el usuario al grupo. Los usuarios pueden crear satélites para reducir la dependencia de la infraestructura de ZeroTier, Inc. o para ubicar los servidores raíz más cerca para un

mejor rendimiento. Para el uso de SDN en las instalaciones, se puede ubicar un grupo de servidores raíz dentro de un edificio o centro de datos para que ZeroTier pueda continuar funcionando normalmente si se pierde la conexión a Internet.

Los nodos comienzan sin vínculos directos entre sí, solo aguas arriba de las raíces (planeta y lunas). Cada par en VL1 posee una dirección ZeroTier de 40 bits (10 dígitos hexadecimales) única a nivel mundial, pero a diferencia de las direcciones IP, estos son identificadores criptográficos opacos que no codifican información de enrutamiento. Para comunicarse, los pares primero envían paquetes "hacia arriba" del árbol y, a medida que estos paquetes atraviesan la red, desencadenan la creación oportunista de enlaces directos a lo largo del camino. El árbol está constantemente tratando de "colapsarse" para optimizarse al patrón de tráfico que transporta. (github.com, act.2021)

La configuración de la conexión de igual a igual es la siguiente:

1. Si quiere enviar un paquete a B, pero como no tiene una ruta directa, lo envía en sentido ascendente a R (una raíz).
2. Si R tiene un enlace directo a B, reenvía el paquete allí. De lo contrario, envía el paquete en sentido ascendente hasta que se alcanzan las raíces planetarias. Las raíces planetarias conocen todos los nodos, por lo que eventualmente el paquete llegará a B si B está en línea.
3. Si R también envía un mensaje llamado *encuentro* en A que contiene pistas sobre cómo podría llegar a B. Mientras tanto, la raíz que reenvía el paquete a B envía el encuentro informando a B cómo podría llegar en A.
4. A y B obtienen sus mensajes de *encuentro* e intentan enviarse mensajes de prueba entre sí, posiblemente logrando perforar cualquier NAT o firewall con estado que se encuentre en el camino. Si esto funciona, se establece un enlace directo y los paquetes ya no necesitan tomar la ruta panorámica.

.

Dado que las raíces reenvían los paquetes, A y B pueden alcanzarse instantáneamente. Luego, A y B comienzan a intentar establecer una conexión directa de igual a igual. Si esto tiene éxito, el resultado es un enlace de latencia más baja más rápido. A esto lo llamamos *aprovisionamiento de enlace activado por transporte,* ya que es el reenvío del paquete en sí lo que activa la red de igual a igual para intentar una conexión directa.

VL1 nunca se rinde. Si no se puede establecer una ruta directa, la comunicación puede continuar a través de la retransmisión (más lenta). Los intentos de conexión directa continúan para siempre de forma periódica. VL1 también tiene otras características para establecer conectividad directa, incluido el descubrimiento de pares de LAN, la predicción de puertos para el cruce de NAT IPv4 simétricas y el mapeo de puertos explícito utilizando uPnP y / o NAT-PMP si están disponibles en la LAN física local. (ZeroTier Inc, act 2021)

2.3.3. Antecedentes de Zerotier

Los objetivos y principios de diseño de ZeroTier se inspiran, entre otras cosas, en el documento original de Google BeyondCorp y en él Jericho Forum con su noción de “desperimetría”. El cual fue "Colaborar de forma segura en las nubes", que implica la aplicación de los conceptos de COA al paradigma emergente de Cloud Computing. La premisa básica es que un enfoque colaborativo es esencial para obtener el máximo valor de "la nube". Gran parte de este trabajo se transfirió a Cloud Security Alliance.

El Foro de Jericó declaró su éxito y finalizó en la conferencia de Londres del OpenGroup el 29 de octubre de 2013. El trabajo del Foro de Jericó sobre identidad ha sido llevado a cabo por Global Identity Foundation, una organización sin fines de lucro que trabaja para definir los componentes de un ecosistema de identidad digital global. (opengroup.org, 2019)

2.3.3.1. ZeroTier Edge. Un dispositivo de virtualización de redes, SD-WAN y VPN de nivel empresarial con tecnología de software de código abierto. Comercialmente se vendió como un dispositivo autónomo preconfigurado, que le permitió conectar dispositivos físicos a redes virtuales y conectar redes físicas en varios sitios con facilidad.

El dispositivo Edge en sí debía designarse como un puente Ethernet en todas las redes virtuales ZeroTier que desease puentear a los puertos físicos. Esto debía hacerse en ZeroTier Central , o si estaba ejecutando su propio controlador de red configurando el campo "activeBridge" en "verdadero" en el registro de miembros de la red de Edge. Si Edge no estaba autorizado para actuar como puente, no se le permitía reenviar paquetes ethernet hacia y desde dispositivos que no estén conectados a él. Era común configurar un nodo en la nube que actuaba como un enrutador de Internet y proporcionaba DHCP, DNS y otros servicios. En este caso varias ubicaciones y usuarios remotos, podrían compartir la misma puerta de enlace en la nube.

A partir de 2020, es el fin de su producción comercial y vida útil. Un repositorio del código fuente del Software ZeroTier Edge y sistemas de archivos raíz (AARCH64) que se ejecutan en él se puede encontrar aquí: https://github.com/zerotier/edge; la instancia electrónica se ejecutó en hardware ESPRESSObin v7.

2.3.3.2. Hardware ESPRESSObin. Marvell ARMADA 3700 Community Board (ESPRESSObin). Es una placa comunitaria para compartir proyectos útiles de software y hardware, sirve de soporte para varias configuraciones de placas electrónicas y configuraciones de red, resolución de problemas y todo lo demás relacionado con la electrónica de laboratorio.

La familia de dispositivos Marvell® ARMADA® 3700 ofrece soluciones integrales de sistema en chip (SoC) impulsadas por la tecnología de CPU de alto rendimiento dual Cortex-A53 ARMv8 que funciona hasta 1,2 GHz. (GitHub, 2017)

Versiones de hardware de ESPRESSObin se muestran en la siguiente tabla:

Tabla 5. Tabla Comparativa de Hardware ESPRESSObin v5 & v7.

Componente	ESPRESSObin v5 y anteriores	ESPRESSObin v7
SoC	Procesador Marvell Armada 3700LP (88F3720) de doble núcleo ARM Cortex A53 de hasta 1 GHz	Marvell Armada 3700LP (88F3720) de doble núcleo procesador ARM Cortex A53 hasta 1,2 GHz
Memoria	DDR3 de 1GB o DDR3 de 2GB	DDR4 de 1GB o 2GB
Almacenamiento	1x ranura para tarjeta microSD	1x ranura para tarjeta microSD
	Huella eMMC de 4GB opcional	Huella eMMC de 4GB opcional
	Conector fuente y SATAS 3.0	Conector SATA único
Conectividad de red	1x conmutador de red Topaz	1x conmutador de red Topaz
	2 LAN Ethernet GbE	2 LAN Ethernet GbE
	1x Ethernet WAN	1x Ethernet WAN
	1x ranura MiniPCIe para periféricos inalámbricos / BLE	1x ranura MiniPCIe para periféricos inalámbricos / BLE
USB	1x USB 3.0	1x USB 3.0
	1x USB 2.0	1x USB 2.0
	1x puerto micro-USB	1x puerto micro-USB
Expansión	Cabezales GPIO de 2x46 pines para accesorios y protectores con I2C, GPIO, PWM, UART, SPI, MMC, etc.	Cabezales GPIO de 2x46 pines para accesorios y protectores con I2C, GPIO, PWM, UART, SPI, MMC, etc.
Misc	Botón de reinicio	Botón de reinicio
	Interfaz JTAG	Interfaz JTAG (compatible con depurador Marvell JTAG Prove
Fuente de alimentación	Conector de 12V DC	Conector de 12V DC
Consumo de energía	Disipación térmica de menos de 1 W a 1 GHz	Disipación térmica de menos de 1

Fuente: http://wiki.espressobin.net/tiki-index.php?page=About+ESPRESSObin

A continuación, se enumeran algunos de los casos de uso en los que podría utilizar la placa ESPRESSObin:

- NAS-SATA-Wi-Fi -> Al conectar la interfaz SATA del dispositivo a las tarjetas SSD, puede transmitir vía inalámbrica o Ethernet.
- Monitor de cámara de video -> Reutilice cámaras USB antiguas y conviértalas en cámaras IP al conectarlas a ESPRESSObin, transmitirá contenido directamente a su computadora o consola.
- Servidor de medios Plex -> Transmisión de medios desde SSD, memorias, servidores en la WEB.
- Clúster de conmutadores apilables -> Cree unidades de almacenamiento energéticamente eficientes conectando almacenamiento y Ethernet. Clústeres de cómputo para aplicaciones de servidor Torrent.
- IoT Gateway -> Convierta el ESPRESSObin en un IoT Gateway conectando LORA, BLE, ZigBee, Zwave y otros protocolos a través de una variedad de interfaces compatibles con la placa. (Espressobin, s.f.)

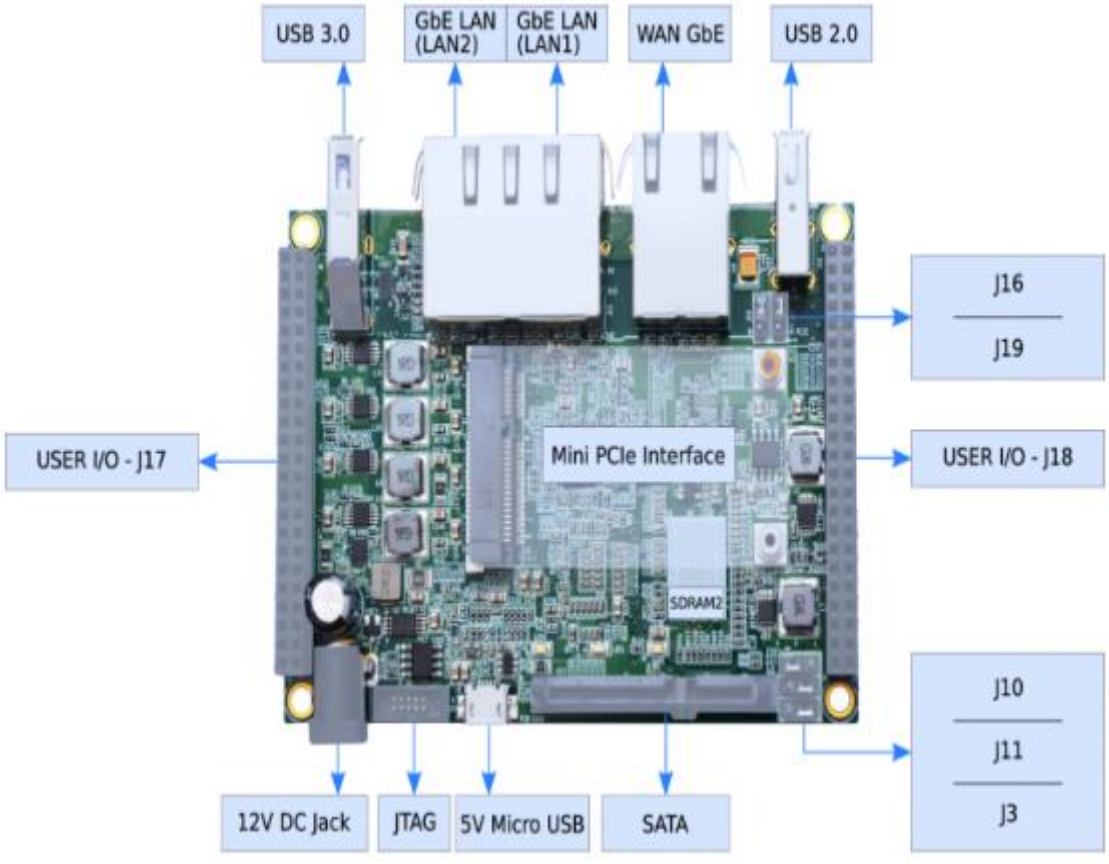


Figura 17. Hardware ESPRESSObin, (Print Screen).
Fuente: http://wiki.espressobin.net/tiki-index.php?page=Board+Distributors

2.3.4. Zerotier & SBC

Como beneficio adicional, el equipo de ZeroTier ha puesto su código a disposición en GitHub . Una de las ventajas de un proyecto de código abierto es que puede ver lo que se supone que debe hacer el código. En este contexto, el SBC facilita la resolución de los nuevos retos en seguridad que estos sistemas innovadores requieren. Es por ello por lo que se ha convertido en una herramienta imprescindible a la hora de utilizar de manera fiable un sistema de comunicación basado en SDN. Se debe enfatizar en cuanto a las funciones principales de un SBC:

Seguridad: Una de las funciones más importantes de un SBC es proporcionar una capa de seguridad adicional que mantenga la red fuera del alcance de hackers e infiltrados. Es decir, realiza la misma tarea que un firewall, pero adaptado de manera eficiente a las características concretas de una conexión de Protocolo de Internet (IP). De esta manera, dichas conexiones cuentan con una capa de seguridad adicional que permite preservar los datos y su confidencialidad.

Conectividad: El SBC permite la conectividad de diferentes partes de la red, de manera que se puedan comunicar mediante la combinación de diferentes técnicas. Algunas de ellas, como la NAT transversal, o la conectividad VPN elevan los estándares de seguridad, de manera que las conexiones sean eficaces, seguras y a prueba de fallas.

Encriptamiento: La transferencia de datos, en la mayoría de los casos confidenciales, que llevamos a cabo en todo momento, mediante redes, hace necesario la utilización de herramientas que permitan la encriptación de la información transmitida.

Figura 18. ZeroTier & SBC (Print Screen).
Fuente: https://loopup.com/us/solutions/cloud-telephony/

2.3.5. ZeroTier One

ZeroTier One hace que las redes virtuales ZeroTier estén disponibles como puertos de red virtual 'tap', de un modo de controlador tun/tap. La API de nodo zerotier es la API formada con programación en C simple y orientada para uso exterior, que envuelve la clase Nodo final en la carpeta node/conf. Proporciona una interfaz independiente de la plataforma para el motor central de virtualización de red ZeroTier.

Es entre otras palabras es el backplane virtual que trabaja como invitado he independiente del proveedor para implementaciones en la nube, es una forma segura de acceder a dispositivos integrados remotos, entre otros. (ZeroTier.org, 2021)

Figura 19. ZeroTier One, (Print Screen).
Fuente: https://www.zerotier.com/download/#

2.3.6. ZeroTier Uniendo Dispositivos - Uniendo Redes

ZeroTier One debe estar instalado y unido a la red a la que desea proporcionar el servicio DNS, descargue ZeroTier One en cualquier dispositivo para obtener una dirección de nodo única de 10 dígitos e ingrese su ID de red de 16 dígitos en el campo para unirse a la red en el dispositivo para solicitar acceso a su red.

Los usuarios de Windows pueden descargar el MSI (Microsoft Windows Installer) desde la página de descargas de ZeroTier. Las plataformas Mac y Windows tienen interfaces gráficas que proporcionan iconos de barra de tareas o de bandeja, asegúrese de aprobar la instalación del controlador durante el proceso de instalación.

Para entornos Unix existe la guía de descarga he instalación detallada y está disponible en la base de conocimientos, visite los foros de la comunidad para obtener ayuda (https://zerotier.com/download/).

ZeroTier One se puede compilar fácilmente desde la fuente para su uso en sistemas operativos distintos a los que admitimos a través de paquetes prediseñados. Esto incluye versiones menos comunes de Linux, FreeBSD, OpenBSD, NetBSD, y son soportados por la comunidad FreeBSD.

2.3.7. ZeroTier en Docker

El controlador tun / tap implementa la función de una tarjeta de red virtual. (Tun representa un dispositivo virtual punto a punto y tap representa un dispositivo Ethernet virtual). Para hacer esto dentro de un contenedor Docker, se requieren algunos permisos elevados y acceso al dispositivo / dev / net / tun.

ZeroTier lleva su red al espacio de usuario. Hemos emparejado nuestro núcleo del módulo de red virtual con un docker pull de la imagen de ZeroTier One para contenedores desde GitHub ya que ZeroTier no tiene soporte directo para redes en contenedores y se apoya únicamente en los foros de la comunidad de ZeroTier y de GitHub.

Para proporcionarle a él núcleo de Red Virtual (VXLAN), una interfaz de red virtual exclusiva y privada asegúrese de establecer correctamente las rutas iptables y las redes de Docker Host en la Raspberry Pi antes de cargar el contenedor de la red ZeroTier. La fuente abierta de GitHub (https://github.com/zerotier/ZeroTierOne), nos brinda la guía necesaria para realizar un docker pull a la imagen de ventana acoplable que ofrece la funcionalidad de Zerotier en una Raspberry Pi y adaptarla al contenedor de nuestro proyecto.

Ejecute la guía de instalación del Capítulo III, “Zerotier en la Raspberry Pi”, para instalar una red peer to peer de capa 2 en un sistema de contenedores. Una vez que tenga comunicación con el interruptor de ZeroTier Administrador podrá conectar dispositivos y redes superponiéndose (VXLAN) a cualquier red LAN que esté conectado, ahora puede cargar dispositivos a la red como se mostró en el subcapítulo anterior.

La red Zerotier es "privada", es decir. todos los nodos deben tener autorización para acceder a la red.

Resulta que hay 1 archivos clave en el contenedor de la ventana acoplable que deben conservarse desde Portainer, debe activar el contenedor de la Red Virtual para que no se desconecte la red de ZeroTier cada vez que se reinicie el sistema.

2.4. OpenMediaVault “OMV“

Es una solución de software diseñada para almacenamiento conectado a la red de próxima generación basada en Debian Linux con su sistema de archivos de código abierto y una sencilla interfaz web para su gestión, opera con una flexibilidad sin precedentes y un compromiso inquebrantable con la integridad de los datos.

OpenMediaVault está escrita en PHP usando el marco JavaScript y se caracteriza por cargar datos según sea necesario usando la tecnología Ajax sin recargar páginas, soporta configuraciones RAID, y admite varios sistemas de ficheros como ext4, JFS, y XFS. Además, incluye un sin número de plugins para añadir funcionalidades a la operación. (OpenMediaVault.org, 2021)

OpenMediaVault anticipa complementos de desarrollo como soporte para dispositivos integrados, mientras que la dirección clave del desarrollo de FreeNAS es utilizar las capacidades del sistema de archivos ZFS (Zeta File System).

ZFS es un sistema de archivos transaccionales diseñado para eliminar la gestión de volúmenes mediante la fusión de dispositivos en un grupo de almacenamiento para administrar el espacio de almacenamiento físico. Elimina la mayoría, si no todas, las deficiencias encontradas en los sistemas de archivos heredados y los dispositivos RAID de hardware.

OpenMediaVault contiene servicios como:

- SMB / CIFS: Common Internet File System – en español Sistema de Archivos de Internet Común. Son las siglas de. Forma parte del protocolo SMB y va a permitir conectar de forma remota múltiples plataformas, como pueden ser Windows, Linux o macOS. SMB son las siglas de Server Message Block. Este protocolo cliente-servidor se encarga de gestionar el acceso a archivos o directorios, de intercambiar información entre procesos de un sistema. Ha estado presente en los diferentes sistemas operativos de Windows para compartir SMB usando Samba como servidor independiente de forma predeterminada.

- SFTP: El protocolo SFTP (SSH File Transfer Protocol) o también conocido como transferencia de ficheros SSH, es un protocolo que no tiene nada que ver con el protocolo FTP. SFTP no es la versión segura del protocolo FTP, ya que está basado en el protocolo SSH por completo. Este protocolo SFTP permite autenticar y realizar transferencia de ficheros entre equipos como si fuera un servidor FTPES, pero utilizando criptografía del protocolo SSH que tenga instalado en el servidor de archivos. SFTP tampoco es un protocolo donde FTP utilice SSH para asegurar la conexión, es un protocolo completamente nuevo basado en SSH y no en FTP.

El protocolo SFTP hace uso del puerto TCP 22 por defecto, el mismo que el protocolo SSH. Si el servidor de archivos tiene un servidor SSH para ejecutar comandos

CLI, también tendrá la opción de habilitar el SFTP para conectarse con clientes SFTP como FileZilla, y comenzar a transferir archivos. La autenticación de SFTP es exactamente la misma que en SSH, es decir, si en SSH se utiliza una clave pública, en SFTP también deberá hacer uso de esta criptografía de clave pública.

- RSync: Demonio del servidor. Las carpetas compartidas se pueden definir como módulos rsyncd. Con tareas programadas, el cliente rsync se puede configurado para trabajos de empujar o tirar.

- SSH: acceso de shell remoto usando openssh. (Volker , 2021, pág. 10)

OpenMediaVault incluye ciertas características esenciales de registro como syslog, perro guardián, notificaciones de correo electrónico. Puede agregarse complementos de servicios adicionales como LVM, servicio de directorio LDAP, AFP, Cliente BitTorrent, servidor DAAP, destino iSCSI, antivirus.

Para quienes desconocen del proyecto OpenMediaVault, deben saber que se fundó en 2009 después de una división en el campo de los desarrolladores de la distribución FreeNAS, como resultado de lo cual, junto con el clásico FreeNAS basado en FreeBSD, se creó una rama, cuyos desarrolladores tenían como objetivo transferir la distribución al kernel de Linux y la base del paquete Debian.

Figura 20. OpenMediaVault, (Print Screen).
Fuente: https://www.openmediavault.org/

2.5. Docker

Docker es un proyecto de software de código abierto que le permite desarrollar, automatizar y administrar aplicaciones en estructuras de contenedores. Docker le permite separar sus aplicaciones de su infraestructura para que pueda entregar software rápidamente. Con Docker, puede administrar su infraestructura de la misma manera que administra sus aplicaciones al aprovechar las metodologías de Docker para enviar, probar e implementar código rápidamente, puede reducir significativamente la demora entre la escritura del código y su ejecución en producción.

2.5.1. Generalidades de Docker

2.5.1.1. Cómo Funciona Docker. El software le proporciona una manera estándar de ejecutar su código de manera similar a cómo una máquina virtual virtualiza (elimina la necesidad de administrar directamente) el hardware del servidor, los contenedores virtualizan el sistema de operación de un servidor, se instala en cada uno y proporciona comandos sencillos que puede utilizar para crear, iniciar o detener contenedores. (Amazon Web Services, Inc., 2021)

2.5.1.2. Por qué usar Docker. Docker le permite entregar código con mayor rapidez, estandarizar las operaciones de las aplicaciones, transferir el código con facilidad. Con Docker, obtiene un solo objeto que se puede ejecutar de manera fiable en cualquier lugar. La sintaxis sencilla y simple de Docker le aporta un control absoluto y por su amplia adopción significa que existe un gran ecosistema de herramientas y aplicaciones listas para su uso que puede utilizar con Docker.

2.5.1.3. Cuando usar Docker. Puede utilizar los contenedores de Docker como bloque de construcción principal a la hora de crear aplicaciones y plataformas modernas. Docker facilita la creación y la ejecución de arquitecturas de microservicios distribuidos, la implementación de código con canalizaciones de integración y entregas continúas estandarizadas, la creación de sistemas de procesamiento de datos altamente escalables y la creación de plataformas completamente administradas para sus desarrolladores. (Docker Inc., 2021)

Figura 21. Escenario de Comunicaciones Docker, (Print Screen).
Fuente: https://www.udemy.com/course/debian-desktop-server/learn/lecture/

2.5.2. Arquitectura de Docker

Docker utiliza una arquitectura cliente-servidor. El *cliente de* Docker habla con el *demonio de* Docker, que hace el trabajo pesado de construir, ejecutar y distribuir sus contenedores Docker. El cliente y el demonio de Docker pueden ejecutarse en el mismo sistema, o puede conectar un cliente de Docker a un demonio de Docker remoto. El cliente y el demonio de Docker se comunican mediante una API REST, a través de sockets UNIX o una interfaz de red. (Zafra CPR, 2021)

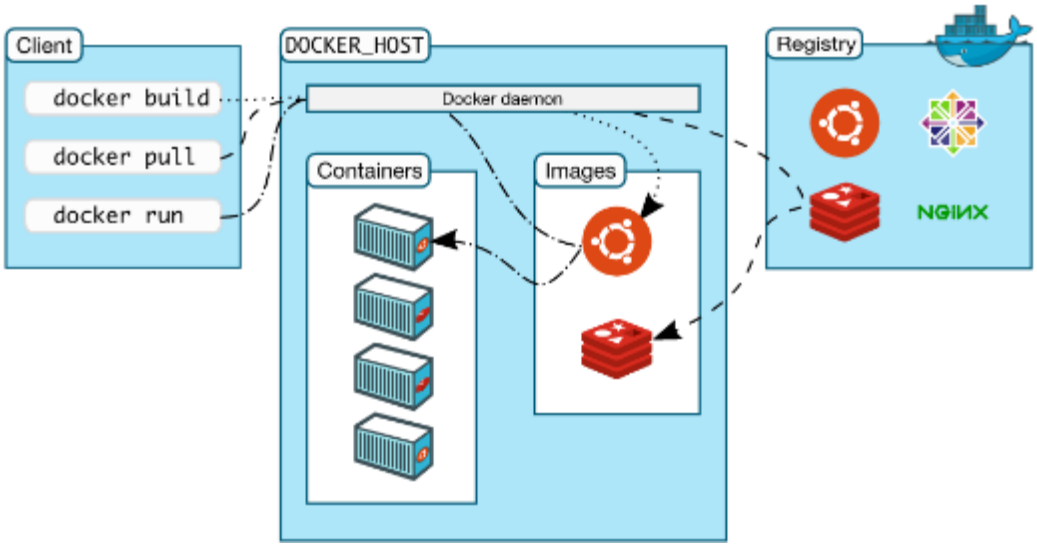


Figura 22. Descripción General de la Arquitectura de Docker, (Print Screen).
Fuente: https://iesgn.github.io/curso_docker_2021/sesion2/dockerhub.html

2.5.2.1. El Demonio de Docker. El demonio de Docker (dockerd) escucha las solicitudes de la API de Docker, administra objetos de Docker como imágenes, contenedores, redes y volúmenes. Un demonio también puede comunicarse con otros demonios para administrar los servicios de Docker. (Docker Inc., 2021)

2.5.2.2. El Cliente Docker. El cliente de Docker (docker) es la forma principal en que muchos usuarios de Docker interactúan con Docker. Cuando utiliza comandos como docker run, el cliente envía estos comandos a dockerd, que los ejecuta. El docker comando usa la API de Docker. El cliente de Docker puede comunicarse con más de un demonio. (Docker Inc., 2021)

2.5.2.3. Registros de Docker. Un registro de Docker almacena imágenes de Docker. *Docker Hub* es un registro público de libre acceso. El demonio de Docker está configurado para buscar imágenes en el repositorio de Docker Hub de forma predeterminada. Incluso puede ejecutar su propio registro privado.

Para extraer una imagen de Docker se debe realizar un pull o run al registro, cuando usa un push, su imagen se envía al registro configurado. (Zafra CPR, 2021)

2.5.3. Objetos Docker para-ARM

Cuando usa Docker, está creando y usando imágenes, contenedores, redes, volúmenes, complementos y otros objetos. Esta sección es una breve descripción general de algunos de esos objetos de Docker aplicados a nuestro proyecto que será para arquitecturas ARM.

2.5.3.1. Contenedores para ARM. Un contenedor es una instancia ejecutable de una imagen. Puede crear, iniciar, detener, mover o eliminar un contenedor mediante la API o la CLI de Docker. Puede conectar un contenedor a una o más redes, adjuntarle almacenamiento o incluso crear una nueva imagen basada en su estado actual que para nosotros será la arquitectura de contenedores para arm.

De forma predeterminada, un contenedor está relativamente bien aislado de otros contenedores y de su máquina host. Se puede controlar qué tan aislados están de la red, el almacenamiento u otros subsistemas subyacentes de un contenedor de otros contenedores o de la máquina host.

Un contenedor se define por su imagen, así como por las opciones de configuración que le proporcione al crearlo o iniciarlo. Cuando se quita un contenedor, cualquier cambio en su estado que no esté almacenado en el almacenamiento persistente desaparece. (Docker Inc., 2021)

2.5.3.2. Imágenes Docker para ARM. Docker parte, en realidad, de un sistema x64, el habitual en la mayoría de los ordenadores modernos. Sin embargo, Raspberry Pi utiliza la tecnología ARM. Esto significa que las imágenes normales de Docker no son compatibles con la instancia en el Pi. Con todo, hoy ya se encuentran cada vez más contenedores preparados para Raspberry Pi.

Es importante descargar imágenes de fuentes fiables, como se indicó antes en Docker Hub encontrará un repositorio de imágenes disponibles de libre acceso para revisar su código, de lo contrario, es grande el riesgo al poder comprometer la seguridad de su equipo. Aunque la elección es limitada (y probablemente seguirá siéndolo en otros sistemas frente a Docker), aún es posible disfrutar de todo el potencial del sistema de contenedores desarrollándolos por ti mismo. (Zafra CPR, 2021)

2.5.3.3. Volúmenes Docker. Es un directorio/fichero en el docker engine que se monta directamente en el contenedor. Podemos montar varios volúmenes en un contenedor y en varios contenedores podemos montar un mismo volumen.

Cuando un contenedor es borrado, toda la información contenida en él desaparece. Para tener almacenamiento persistente en nuestros contenedores, que no se elimine al borrar el contenedor, es necesario utilizar volúmenes de datos.

2.5.4. Redes en Docker

Cada vez que creamos un contenedor, esté se conecta a una red virtual y docker hace una configuración por defecto del sistema (usando interfaces bridge, e iptables) para que la máquina tenga una ip interna, tenga acceso al exterior, podamos mapear (DNAT) puertos.

Docker usa este bridge ethernet para permitir al Daemon comunicarse con el dispositivo de red de la maquina a partir de ahora vamos a utilizar maquinas o host para referirnos a los mismos en cuestión.

Este bridge network se llama docker 0 ó Bridge y se crea cuando instalamos el Docker Engine.

bridge = docker 0

Este dispositivo siempre se activa cuando arranca la máquina, docker instala el dispositivo dentro del kernel de Linux para habilitar y configurar esta network, a continuación, docker crea una subnet virtual a la máquina para permitir la transmisión de paquetes entre contenedores para ver esta docker bridge Network puedes hacerlo utilizando el comando. (Geekflare, 2019)

Ip a ó ifconfig.

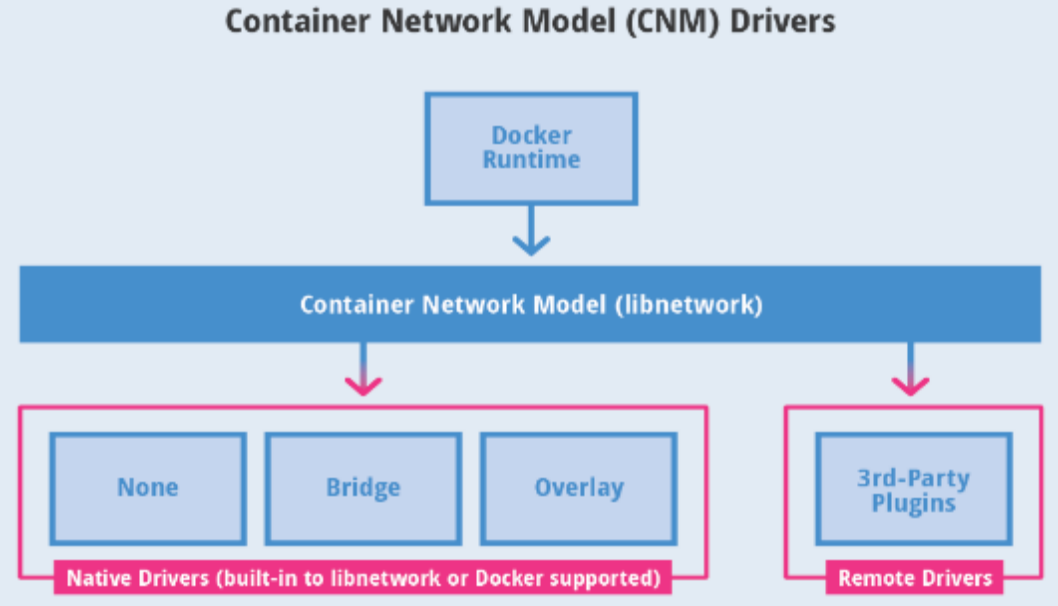


Figura 23. Redes con Docker, (Print Screen).
Fuente: https://sites.google.com/site/wwwvirtualizacioneducom/

2.5.4.1. Enrutamiento Docker. Por defecto, los contenedores están protegidos por el firewall del anfitrión y no abren ninguna ruta a sistemas externos. Esto puede cambiarse manipulando un poco con la bandera – network con las siguientes opciones:

- Bridge: red puente en docker por defecto y creadas por el usuario.
- Host: red de anfitrión (sin firewall).
- NONE: no hay red.
- Overlay: red superpuesta
- Macvlan: asigna una dirección MAC a la interfaz virtual de cada contenedor.

Debemos hacer énfasis en el concepto de los tipos de redes y la gestión que en ellos se puede crear. Es importante detallar el concepto de redes Bridge, hacer paréntesis en este tipo de red por defecto y que pueden ser creadas por el usuario:

2.5.4.1.1. *Redes Bridge por Defecto en Docker.* La red "bridge" por defecto, que es la usada por los contenedores tiene los siguientes aspectos:

- Para conectar en caliente a los contenedores en la red por defecto tendrá que parar previamente el contenedor.
- Brinda poco aislamiento de los contenedores, ya que al arrancar éste se incluye en la red por defecto donde pueden convivir servicios abiertos.
- Los contenedores dentro de la red "bridge" por defecto comparten todos ciertas variables de entorno lo que puede provocar ciertos conflictos.
- Aislar los distintos contenedores que están en distintas subredes docker, de tal manera que desde cada una de las subredes solo se podrá acceder a los equipos de esa misma subred.
- Publicar servicios de los contenedores mediante redirecciones que docker implementará con las pertinentes reglas de iptables. (Zafra CPR, 2021)

2.5.4.1.2. *Redes Bridge Creadas por Usuario*. Es la red que se genera en el ámbito de desarrollo de proyectos en contenedores creados por usuarios.

- Las redes que en desarrollo se definan proporcionarán resolución DNS entre los contenedores, cosa que la red por defecto no hace.
- Puedo conectar en caliente a los contenedores redes "bridge" definidas por el usuario. Si uso la red por defecto tengo que parar previamente el contenedor.

- Brinda seguridad en el aislamiento de los contenedores, ya que si no se indica una red al arrancar un contenedor éste se incluye en la red por defecto donde pueden convivir servicios abiertos.

- Supervisión de control fiable sobre la configuración de las redes que define el desarrollador. Los contenedores de la red por defecto comparten toda la misma configuración de red (MTU, ip tables etc).

En definitiva: Es importante que nuestros contenedores en producción se estén ejecutando sobre una red definida por el usuario.

2.5.4.1.3. ***Redes Host en Docker.*** Si conecto un contenedor a la red host, el contenedor ofrece el servicio que tiene configurado en el puerto de la red del anfitrión. No tiene ip propia, sino es cómo si tuviera la ip del anfitrión. Por lo tanto, los puertos son accesibles directamente desde el host. (Zafra CPR, 2021)

Para comprobar que funciona deberías de acceder a http://localhost:80.

2.5.4.1.4. ***Redes NONE en Docker.*** La red none no configurará ninguna IP para el contenedor y no tiene acceso a la red externa ni a otros contenedores. Tiene la dirección loopback y se puede usar para ejecutar trabajos por lotes. (Zafra CPR, 2021)

2.5.4.1.5. ***Redes Overlay en Docker***. Para el tipo overlay, necesitas trabajar en modo clúster. Si intentas crear una red de este tipo, sin estar en este modo, obtendrás el siguiente mensaje: Use "docker Swarm init" o "docker Swarm join".

Sin embargo, la experiencia para ti sería la misma: la comunicación entre los diferentes contenedores podrá hacerse tanto a través de su IP como de su nombre, actuando como una única red, aunque haya diferentes máquinas ejecutando nuestros contenedores. (Udemy.com, 2021)

2.5.4.1.6. ***Macvlan en Docker.*** Macvlan nos permite asignar IPs de una red a cada uno de los contenedores que viven en nuestro host. Sin embargo, para que este tipo de redes funcione necesitamos que la tarjeta de red del host esté en modo promiscuo, lo cual significa que necesita estar a la escucha de todos los paquetes que viajan por dicha red, simplemente para poder reconocer aquellos que potencialmente pueden ser para uno de sus contendores. La forma de configurar una red de este tipo sería así:

```
docker network create -d macvlan \
    --subnet=172.16.86.0/24 \
    --gateway=172.16.86.1 \
    -o parent=eth0 \
    my-macvlan-net
```

2.5.5. Publicar Puertos en Docker

Para que un puerto responda a peticiones en un servidor, se lo debe publicar. Para esto se debe utilizar la variable “-p” al crear un contenedor.

“docker run -itd -p 8080:80 nombre_imagen “.

De este modo el puerto 80 del contenedor estará publicado en el puerto 8080 de la máquina host. Es así como podemos determinar de un modo arbitrario el puerto de la máquina host que estará respondiendo al puerto del contenedor. (Udemy.com, 2021)

El siguiente comando verificar los puertos que se encuentran publicados

“docker port nombre_imagen Puerto“.

dcalbo@dcalbo-NTB ~ $ docker port web80 80

0.0.0.0:8080

2.6. Portainer

La administración de contenedores Docker puede no ser amigable a la hora de trabajar con la terminal. Así nació la necesidad de crear una interfaz web de administración la cual puede tener una visión global más clara de contenedores en ejecución y facilitar su gestión.

Portainer es una herramienta de administración con todas las funciones para Docker. Se ejecuta localmente, lo que brinda a los desarrolladores una interfaz de usuario rica para crear y publicar imágenes de contenedores, implementar y administrar aplicaciones y aprovechar la persistencia de datos y el escalado horizontal para sus aplicaciones.

Una vez que una aplicación se implementa en un contenedor, la GUI de Portainer facilita a los usuarios asegurar, monitorear y medir el desempeño de la plataforma. La herramienta niega la necesidad de que los desarrolladores aprendan Infraestructura como código y les facilita maximizar su eficiencia de una interfaz gráfica. (Portainer.io, act.2020)

2.6.1. Interfaz Web de Portainer (GUI)

2.6.1.1. Versiones de Portainer. Portainer está disponible en: Portainer CE y Portainer Business. Portainer CE es de código abierto, gratuito para siempre y adecuado para su uso en entornos de laboratorio con soporte de la comunidad de *Portainer.io*, que para nuestro proyecto nos brindara los recursos necesarios y suficientes para la administración de nuestros contenedores.

2.6.1.2. Entornos de Portainer. Portainer se puede implementar en entornos Docker, Docker Swarm o Kubernetes. Para nuestro proyecto escogeremos la implementación de Portainer en docker.

Portainer en docker se compone de dos elementos, *Portainer Server y Portainer Agente*. Ambos elementos se ejecutan como contenedores Docker ligeros en un motor Docker. Debido a la naturaleza de Docker, existen muchos escenarios de implementación posibles, sin embargo, detallamos Docker Host que es el escenario que coincide con nuestra configuración.

De forma predeterminada, Portainer expondrá la interfaz de usuario a través del puerto 9000 y expondrá un servidor de túnel TCP a través del puerto 8000. Este último es opcional y solo es necesario si planea usar las funciones de cómputo de Edge con agentes de Edge. Para obtener más información sobre los requisitos, visite la página de requisitos. https://www.portainer.io/solutions/docker; (github.com, act.2021)

2.6.2. Gestión de Acceso Portainer

2.6.2.1. Seguridad de Portainer. Portainer está diseñado para usarse de manera centralizada, a escala, en organizaciones enteras. Para lograr esto de manera segura, Portainer proporciona una serie de características relacionadas con la seguridad que garantizan que solo los usuarios autorizados puedan acceder a Portainer y solo puedan operar dentro de los límites de sus privilegios asignados.

Portainer ayuda a las organizaciones a proteger su entorno controlando quién puede hacer qué, registrando quién hace qué y brindando la capacidad de realizar copias de seguridad y restaurar la base de datos de configuración del administrador de Portainer.

2.6.2.2. Gestión de Identidad de Portainer. Portainer admite tres mecanismos de autenticación:

- Interno (Para implementaciones autónomas más pequeñas).
- LDAP (Para conexión a LDAP / Active Directory).
- Auth (Para conexión a fuentes de autenticación basadas en la nube).

Para acceder a Portainer se debe autenticar antes de que se les permita iniciar sesión mediante credenciales externas, se puede habilitar el aprovisionamiento automático, lo que crea un usuario de Portainer en el inicio de sesión exitoso.

2.6.3. Funciones Claves de Portainer

- La interfaz de usuario cubre el 100% de las acciones de la CLI de Docker.
- Concéntrese en entregar aplicaciones y no en la línea de comandos.
- Inspeccione aplicaciones, volúmenes y configuraciones con unos pocos click.
- Las plantillas de aplicaciones predefinidas instalan software basado en la industria con un click.
- Inspeccione fácilmente los registros de sus aplicaciones.
- Consola remota con visor de rendimiento de procesos.
- Elimina el riesgo de errores en cadenas de línea de comandos complejas.
- Mayor seguridad, solo los usuarios autorizados pueden realizar tareas.
- Reducir el tiempo para diagnosticar y resolver problemas.

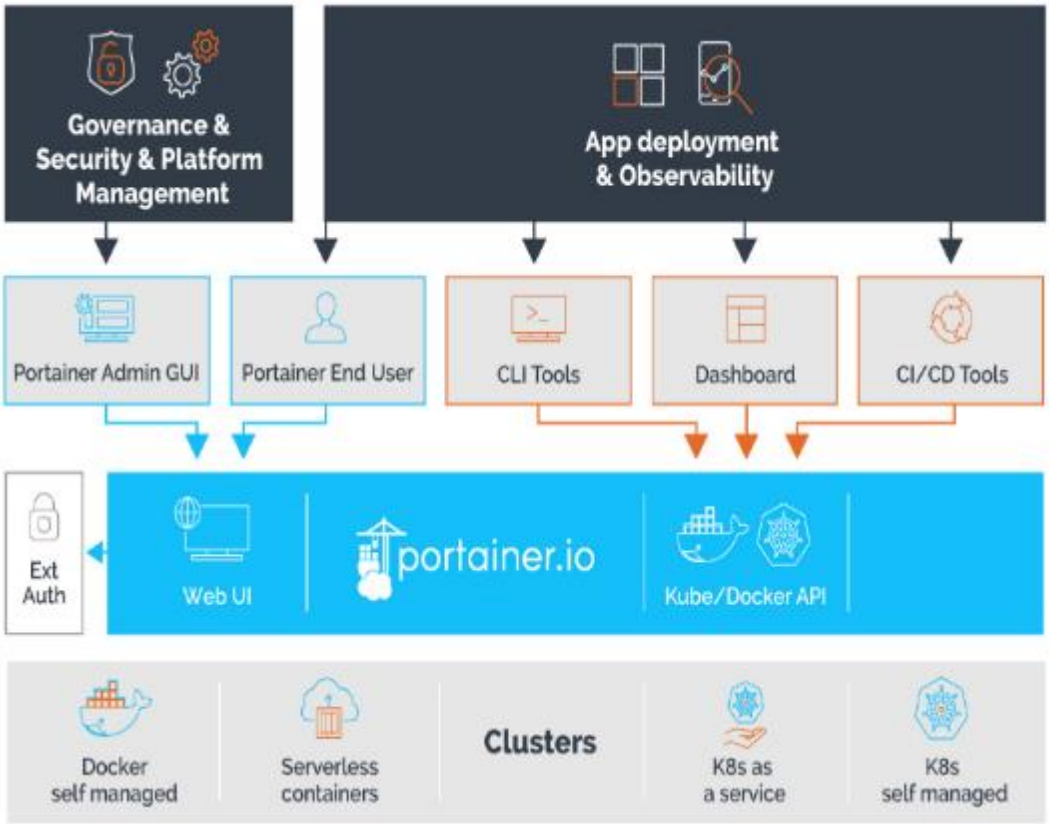


Figura 24. Infraestructura de Administración de Portainer - Basada en la Nube.
Fuente: https://www.portainer.io/features/platform-management

3. DISEÑO E IMPLEMENTACIÓN DEL PROYECTO

3.1. Planificación del Desarrollo

En este capítulo se va a describir el desarrollo de la implementación de la Red Virtual, con el objetivo de que se pueda leer, interpretar y analizar las partes que conforman el sistema desarrollado, a fin de comprender el armado de la estructura, las configuraciones y de poder realizar actualizaciones o modificaciones a futuro para mejorar su eficiencia y de ser posible sirva como base de proyectos futuros.

3.1.1. Estructura del Diseño

En este proyecto se genera la implementación de una red virtual, desarrollando un sistema basado en Docker que permita desempaquetar y programar un software llamado ZeroTier en arquitecturas ARM que es la arquitectura de las raspberry pi. Zerotier con encriptación P2P crea una WAN virtual (Red de Área Global) segura y encriptada a través de Internet. Esto significa que puedo conectarme a mi red ZeroTier a través de las aplicaciones iOS / Android / Windows / iOS y estar efectivamente en cualquier red, por lo tanto, estar disponible para todos los dispositivos siempre que estén conectados a la red ZeroTier.

Por otra parte, se conectará un cliente y un servidor juntos en una red punto a punto simple. Debido a que la red definida por software no utiliza el diseño tradicional de cliente y servidor, no se debe instalar ni configurar un servidor VPN central; esto agiliza la implementación de la herramienta y la adición de nodos complementarios. Una vez establecida la conectividad, tendrá la oportunidad de utilizar la capacidad VPN de ZeroTier empleando algunas funcionalidades inteligentes de Linux para permitir que el tráfico salga de un nodo de red ZeroTier desde nuestro servidor e indicar a un cliente que envíe su tráfico en esa dirección. Separando así la necesidad de configurar manualmente routers o firewalls habituales, optimizando la conectividad del servicio de redes LAN – oficinas centrales y la nube.

3.1.1.1. Selección de Equipos y Elementos.

Para el diseño del Proyecto Técnico se deberá tener presente lo siguiente:

- Raspberry PI 3 o superior.
- Conexión a red de internet (ethernet o inalámbrica).
- MicroSD mínima de 32GB.
- Disco Sólido SDD.
- Convertidor Serial ATA Delta-USB/fuente 110V/5-12VDC
- Porta Fusible y fusible tipo cuchilla de 5A.
- Porta Fusible tipo cartucho de vidrio de 15A.

3.1.1.2. Diagrama de Circuito Eléctrico del Proyecto.

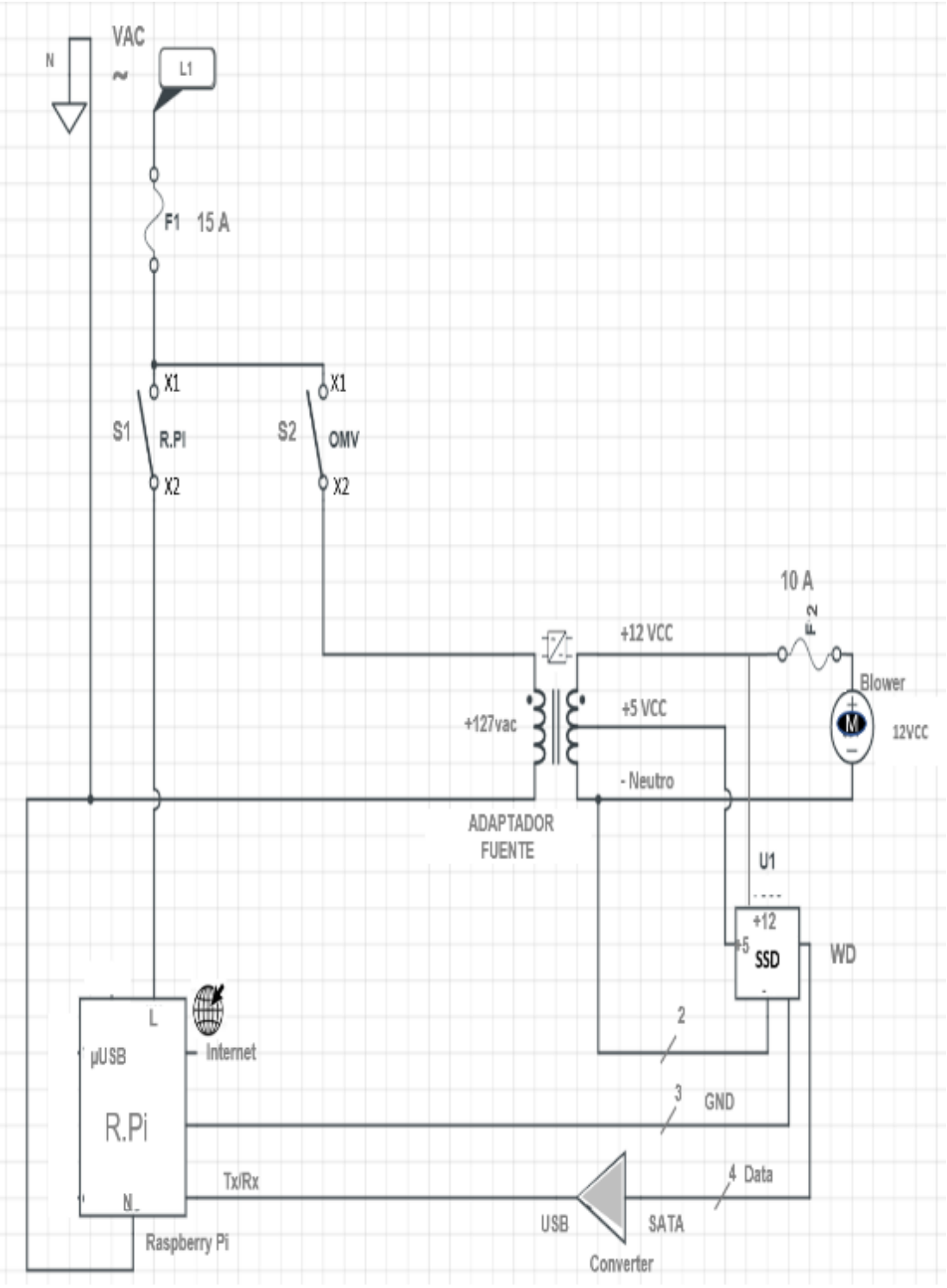


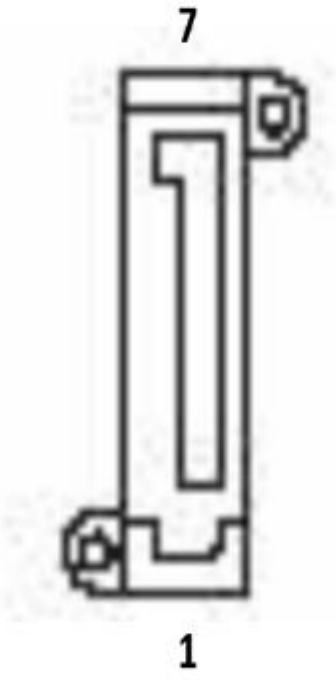
Figura 25. Diagrama de Circuito Eléctrico del Proyecto.
Fuente: Elaboración Propia del autor

3.1.1.3. Referencia de Conexiones en Unidad de Almacenamiento.

ALIMENTACION UNIDAD SSD

Pin	Nombre	Color	Descripción
1	+12V	Amarillo	+12 VCC
2	Masa	Negro	Masa +12 V
3	Masa	Negro	Masa +5 V
4	+5V	Rojo	+5 VCC

CABLE SATA DATOS



CONECTOR SATA

Pin #	Definición
1	GND
2	TXP
3	TXN
4	GND
5	RXN
6	RXP
7	GND

Figura 26. Referencia de Conexiones en Unidad SSD.
Fuente: Adaptado de http://recursostic.educacion.es/observatorio/web/eu/equipamiento-tecnologico/hardware/280

3.1.1.4. Diseño de la Estructura Base del Proyecto.

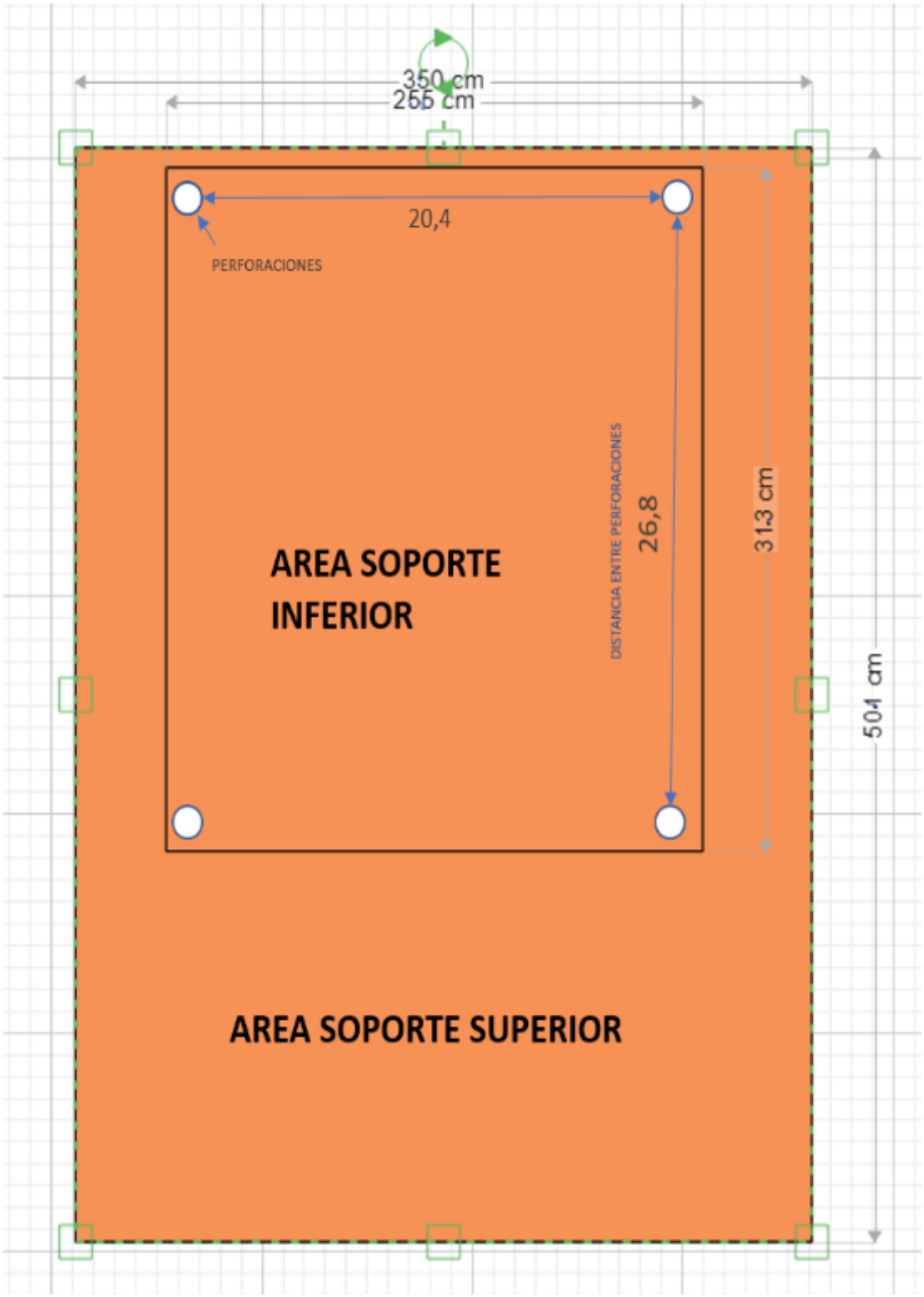


Figura 27. Diseño de la Estructura Base del Proyecto.
Fuente: Elaboración Propia del autor

3.1.2. Descripción Técnica de los Dispositivos para el Hardware

Para el diseño del proyecto se seleccionaron los dispositivos en base a las prestaciones técnicas y operacionales propias de cada uno para lograr cumplir con el objetivo y garantizar su correcto funcionamiento. Cabe mencionar que este tipo de tecnología está disponible en el mercado técnico orientado a la electrónica de redes y computadoras, asegurando así el mantenimiento del equipo.

A continuación, se detallan técnicamente:

3.1.2.1. Convertidor Serial ATA Delta. Plug and Play UASP SATA a USB compatible para-USB 3.0 a Hard Drive Converter, le permite conectar cualquier disco SSD-SATA a velocidad de hasta 480 Mbps. Con Indicadores LED para mostrar el estado y los procesos de los medios intercomunicados, es compatible con SATA III y la velocidad de transmisión será de acuerdo con el SSD conectado. (Mediaprice, 2021)

La estructura de la fuente de poder será adaptada a las condiciones de armado para soporte de los elementos técnicos del Host de Servicio de la Red Virtual.

Figura 28. Convertidor Sata Delta – Fuente de Poder, (Print Screen).
Fuente: https://www.mediaprice.com.ec/producto/convertidor-ide-sata-externo-delta-2/

3.1.2.2. SSD Western Digital - WD Green 2,5". Diseñados para ofrecer un alto rendimiento y confiabilidad, capacidad 120 GB, factor de forma 2.5 en interfaz sata III, dimensiones (l x w x h) 3.94 x 2.75 x 0.28, rendimiento secuencial de lectura 560mb / s rendimiento secuencial de escritura 530mb / S. (westerndigital.com, n.d.)

Figura 29. Disco Sólido Western Digital, (Print Screen).
Fuente: https://shop.westerndigital.com/products/internal-drives/wd-green-sata-2-5-ssd#WDS

3.1.2.3. Tablet Dragon EET1 K10,1´s OS. Tablet Notepad Android con cuatro núcleos y 16 GB de almacenamiento interno expandible hasta 64GB, perfecta para aplicaciones multimedia, entre sus principales características tenemos: (ganacell.com, act.2021)

- CPU: A33 Quad Core, ARM Cortex A7 @ 1.3Ghz.
- Pantalla Tipo: 10 inch LCD de visualización táctil capacitiva, 1024 x 600.
- RAM: 1.5 GB DDR3, Memoria interna, 16 GB NAND flash.
- Sistema operativo: Android A33S 8S. compatible con Google Play Store.
- Cámara dual / HDMI Wi-Fi 802.11, Bluetooth 4, Dongle, EVDO/WCDMA.

Figura 30. Tablet Dragon EET1 K10.1´s, (Print Screen).
Fuente: https://www.ganacell.com/tablet-eet1--10-pulg--16gb--1.5gb-ram--quad-core

3.1.3. Topología de Comunicación de la Red Virtual ZT_VXLAN

Se muestra el escenario de comunicación de la red de forma global. El módulo de la Red Virtual está diseñado para recibir la alimentación de internet desde cualquier XLAN, se puede personalizar según sus necesidades. La Topología proporciona una representación visual de los clientes y la red administrada por el controlador en (R. Pi).

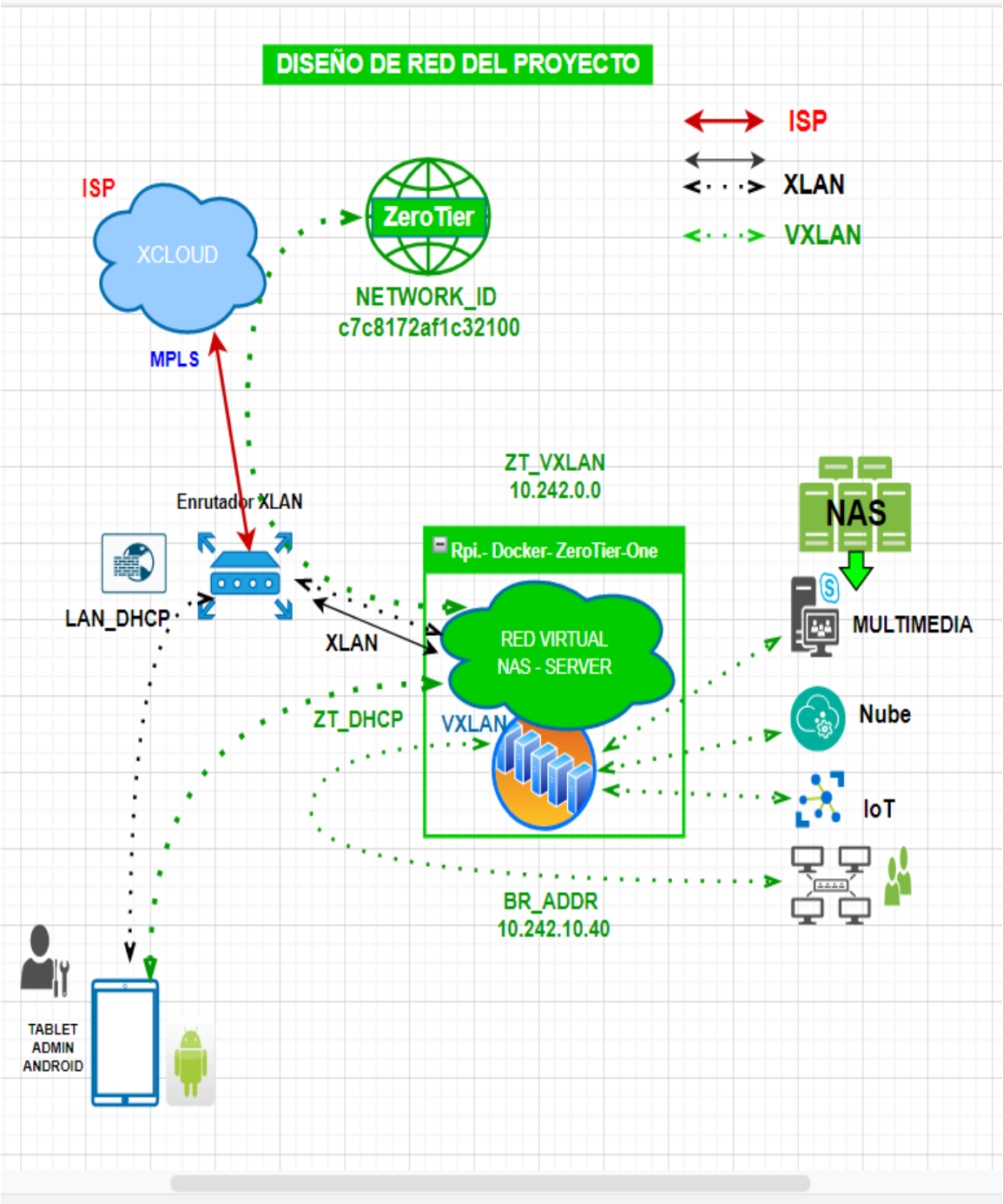


Figura 31. Topología de Comunicación de la Red ZT_VXLAN.
Fuente: Elaboración propia del Autor en https://app.diagrams.net/

3.1.4. Requerimientos Previos a la Implementación del Proyecto

Para la parte inicial del proceso de elaboración del proyecto debemos tener presente ciertos requerimientos previos al desarrollo.

- Está haciendo esto en una red controlada, puede iniciar sesión en su enrutador, administrar la configuración y tener control del firewall.
- Tiene comunicación ethernet conectado a internet. Es muy probable que rompamos la red y perdamos el acceso a la Raspberry pi.
- Estar familiarizado con la línea de comandos Linux, (Protocolos).
- Estar familiarizado con la contenerización de sistemas Docker Swarn.
- Vamos a utilizar la red systemd. Probablemente podría adaptar los conceptos a un sistema de configuración de red Linux diferente.
- Usamos una Raspberry Pi 3 +b en el proyecto, pero debería funcionar bien en cualquier procesador SBC que ejecute una distribución basada en Debian 10. No tiene que ser una raspberry pi, pero algunas de estas instrucciones pueden ser específicas de Rasbian.

Por último, para iniciar prepare esta información, registre los datos en la tabla 6:

- Conexión a LAN ethernet (Gbit/s). Si es Wireless necesariamente requerirá el SSID y password de la red.
- Rango de DHCP de LAN física.
- Rango de asignación de protocolo IP gestionada para el software de ZeroTier (Evitar conflictos de enrutamiento con LAN).
- Dirección IP de puerta de enlace predeterminada de la LAN (el enrutador).
- Identificación del Nodo Puente de red ZEROTIER (un numero de 10 dígitos).
- Cree una Red de ZeroTier Administrador y obtenga la ID de 16 dígitos.
- Prepare la Tabla de Enrutamiento para no superponer la red (conflicto).

IMPORTANTE: El rango de DHCP de la LAN y el rango de asignación automática de ZeroTier pueden estar en la misma subred, pero no superponerse.

Tabla 6. Gestión de Enrutamiento para el Proyecto de Red Virtual.

Nombre	Valor	Denominado
Subred - LAN IPV4	192.168.0.0/24	LAN
Puerta de enlace predeterminada	192.168.0.1	GW_ADDR
Rango DHCP ROUTER LAN	192.168.0.2 hasta 192.168.0.254	LAN_DHCP
Rango de asignación DHCP ZeroTier	10.242.0.2 hasta 10.242.255.254	ZT_DHCP
Ruta gestionada de Red ZeroTier	10.242.0.0/16	ZT_VXLAN
Dirección IP del Puente	10.242.10.40/16(o use DHCP)	BR_ADDR
ID de Red ZeroTier (16 dígitos)	c7c8172af1c32100	NETWORK_ID
ID del Nodo de Equipo (10 dígitos)	1########0	NODO_ID
Nombre de la interfaz de Red ZeroTier	ztmjfejtrx	ZT_IF

Fuente: Propio del autor

3.1.5. Guía de Operación para el Desarrollo de la Implementación

Para el desarrollo de la implementación del proyecto seguiremos los siguientes pasos.

1. Adaptar la fuente de poder para tener la estructura técnica del proyecto, cortes y acople de elementos a utilizar para el armado.

2. Conexión de los elementos eléctricos y electrónicos del proyecto.

3. Armado de la estructura soporte del proyecto. Sujeción de las capas superior e inferior de vidrios biselados en las bases de aluminio.

4. Descarga, instalación y configuración del sistema Operativo Raspberry pi, ajuste de las reglas de iptables.

5. Instalar y configurar el Software FreNAS OpenMediaVault.

6. Preparar el entorno de Docker como orquestador host en el OMV.

7. Instalar y Configurar el Administrador de contenedores “Portainer”.

8. Realizar el pull de la imagen del Interruptor ZeroTier desde la fuente de repositorios reconocidos por ZeroTier.Inc en Git Hub.

9. Preparar la red de ZeroTier, Configuración general de sus interfaces.

10. Desarrollar el escenario de Comunicación de los contenedores con los servicios anexos del proyecto.

Para el desarrollo de la implementación de servicios anexos cargados en el Proyecto, seguiremos los siguientes pasos.

Nota. - Tener en cuenta que los directorios de configuración del proyecto estarán cargados en dos rutas, en directorios de la Raspberry y en el SSD, al crear el sistema de contenedores de servicios ciertos registros estarán comunicados desde el controlador de R.pi y otros desde la base de Datos (SSD), en resumen, quedarán así:

/home/pi = ruta absoluta.
/var/lib/ = estado y archivos de copia de seguridad del editor.
/var/run/ = datos variables en tiempo de ejecución.

OMV : directorios = home/pi/.
Docker : directorios = home/pi; adm = var/lib.
Portainer : admin = var/lib; directorios= /var/run/ (R.pi)
ZeroTier One : adm=/var/lib/zerotier-one; dir= /dev/net/tun (crear carpeta R.pi).
Home Assistant: dir = home/pi/docker/Assistant [crear carpeta R.pi]
NextCloud : adm = /var/lib/mysql; dir=home/pi/docker
volumes = - /srv/dev-disk-by-uuid-/Cloud/ [OMV]
Jellyfin : home/pi; adm= var/lib; volumes: /srv/dev-disk-by-uuid- [OMV]
volumes= - /srv/dev-disk-by-uuid-/Cloud/ [OMV]

Crear una carpeta de Medios en OMV para almacén de directorios.

1. Desarrollo del Software Zerotier One en un contenedor de red host.
2. Desarrollar Home Assistant como contenedor de IOT.
3. Desarrollar NextCloud como contenedor de Nube privada.
4. Desarrollar Jellyfin como contenedor Multimedia de VXLAN.
5. Aplique prácticas de Comunicación de red y registre las actividades.

La ruta completa de OMV saldrá según su propia comunicación con el disco solido que desea utilizar, que para nuestro caso es:

- /srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/config
Interpretación de Software

NOTA: Cuando se indica desarrollar, se enuncia seguir por la línea de comandos de la terminal (CLI). Cuando se indica Instalar, se enuncia realizar un pull de una imagen del repositorio de GibHub. (PULL)

3.2. Armado del Hardware en el Equipo

Esta sección tiene como objetivo mostrar el proceso cómo se llevó a cabo el armado del proyecto, se detallan los recursos utilizados para la implementación además de una breve explicación de su acoplamiento e instalación.

1. Vista general del inventario de las partes y piezas que conformarán el Proyecto de Red Virtual.

Figura 32. Elementos del Proyecto.
Fuente: Propia del autor

2. Preparación de la estructura metálica del Proyecto, brindará el alojamiento de todo el equipo eléctrico y electrónico, realizar ciertas adaptaciones de la estructura, necesarias para el armado correcto.

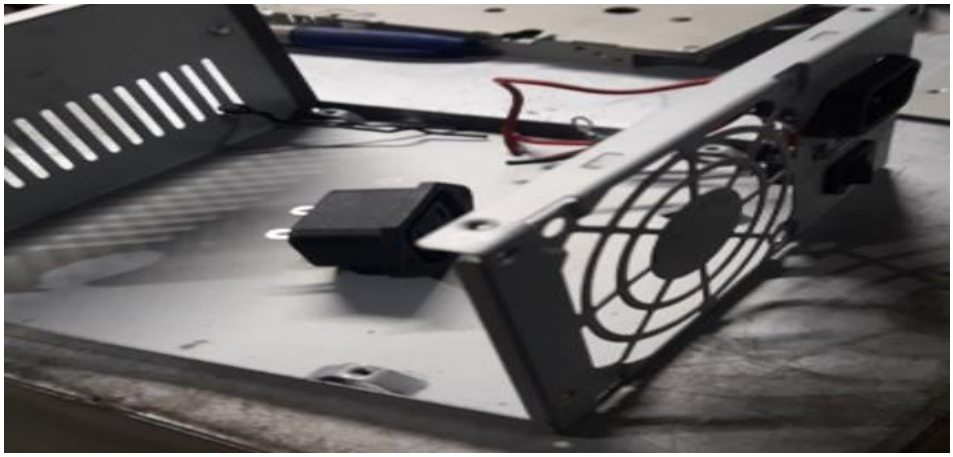


Figura 33. Preparación de la Estructura Metálica del Proyecto.
Fuente: Propia del autor.

3. Los cortes requeridos en la caja metálica fueron realizados en un taller artesanal garantizando las medidas de seguridad que así lo requería la labor.

Figura 34. Cortes de la Estructura Metálicas en Taller Artesanal.
Fuente: Propia del autor.

4. Acoplamiento de estructuras metálicas, soldadura y prensado.

Figura 35. Acoplamiento de las Partes Metálicas de la Estructura.
Fuente: Propia de Autor

5. Pulido de la estructura previo a realizar la pintura de acabado.

Figura 36. Pulido y Limpieza de la Estructura Metálica.
Fuente: Propia del autor.

6. Preparación de las bases y soporte para el Proyecto, el cual se realizó en dos capas de vidrio templado que se detallan así:

   Superior: Área de 50.1cm por 35.0cm, con el borde biselado a 3 cm.
   Inferior: Área de 25.5cm por 31.3cm longitudinalmente.

Figura 37. Construcción de la Estructura Base del Proyecto.
Fuente: Propia del autor.

7. Acople de capas de vidrios con soportes de aluminio de 1’ de Ø por 10 cm de largo, perforadas a 20,4cm en lo ancho y 26,8cm a lo largo de la base respectivamente (ilustración en Figura#27).

Figura 38. Acoplamiento de Soportes de Aluminio entre las Capas de Vidrio.
Fuente: Propia del autor.

8. Pintura y acabado final de la estructura metálica.

Figura 39. Pintura y Acabado Final de la Estructura Metálica.
Fuente: Propia del autor.

9. Acoplamiento de interruptores, ventilador, porta fusibles y punto de corriente.

Figura 40. Acoplamiento de los Elementos Eléctricos .
Fuente: Propia del autor

10. Soldadura de las conexiones de interruptores, protección tipo cuchilla del ventilador, protección tipo cartucho de vidrio para los circuitos de control y la fuente del circuito de datos a 110VAC.

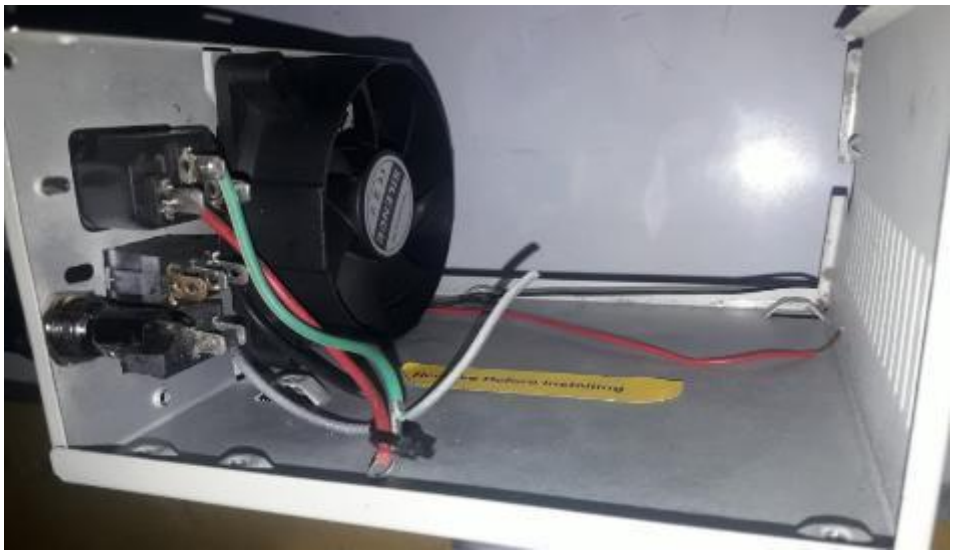


Figura 41. Soldadura de Conexiones en Interruptores y Protecciones.
Fuente: Propia del autor.

11. Conexión de la fuente de alimentación de 5/12VCC, conecte desde la fuente VCC del Convertidor Serial ATA a la entrada SSD.

Figura 42. Conexión de la Fuente de Alimentación del Disco Sólido SSD.
Fuente: Propia del autor.

12. Montaje y sujeción de la tarjeta Raspberry Pi en el soporte metálico.

Figura 43. Sujeción de la Raspberry Pi en la Estructura Metálica.
Fuente: Propia del autor.

13. Montaje y sujeción de la tarjeta SSD Wester Digital en el soporte metálico.

Figura 44. Sujeción del SSD Wester Digital en la Estructura Metálica.
Fuente: Propia del autor.

14. En la tarjeta SSD, conectar cables de datos del convertidor serial ATA he intercomunicador a una entrada usb de la raspberry pi.

Figura 45. Conexiones de la SSD - Comunicación del Convertidor Serial ATA.
Fuente: Propia del autor.

15. Conexión de la toma de 110VAC. Armado de la cubierta externa y alimentación eléctrica de la Raspberry Pi.

Figura 46. Armado de la Cubierta Externa del Equipo.
Fuente: Propia del autor.

16. Proyecto de Red Virtual Pre armado. Para finalizar el armado estaríamos acoplando las capas como soporte de la estructura final y conectando la Tablet a la Interfaz de red del equipo.

Figura 47. Acoplamiento de las Bases y Soportes del Proyecto
Fuente: Propia del autor.

17. Puesta en funcionamiento del Proyecto, alimentación eléctrica y comunicación de red a internet.

Figura 48. Hardware del Proyecto de Red Virtual Armado.
Fuente: Propia del autor.

Infraestructura técnica del Proyecto de Red Virtual armado, Raspberry Pi lista para precisar el arranque operativo. El sistema de almacenamiento SSD instalado en los puertos sata correspondientes del adaptador convertidor Serial ATA (Sata-USB) comunicará hacia los puertos USB de la Raspberry Pi toda la información que en el SSD se almacene. (Directorios y archivos de Servicio).

En el siguiente subcapítulo encontrará la configuración de los diferentes sistemas como el Free NAS, Docker, y la propia configuración de la red virtual de Zerotier. El Sistema Operativo (SO), que desee emplear queda a criterio del administrador, cualquier SO estará bien para la administración ya que la Red Virtual no precisa de depender de un tipo de Sistema Operativo en Particular, si el acceso SSH de la Raspberry Pi está habilitado bastará en cualquier terminal de comunicación-CLI.

Finalmente, seleccionar la opción de arranque ideal. Las opciones disponibles son las siguientes:

- Console: La Raspberry Pi arrancará en modo consola. Justo al finalizar el arranque deberemos indicar el usuario que queremos usar y su contraseña.
- Console Autologin Text console: Nuestro dispositivo arrancará de forma automática en modo consola con el usuario Pi y sin introducir contraseña.
- Desktop: Nuestra Raspberry Pi arrancará en modo gráfico. Una vez arrancada tendremos que seleccionar el usuario e introducir su contraseña.
- Desktop Autologin: La Raspberry Pi arrancará en modo gráfico. Justo al finalizar el arranque seleccionar el usuario y contraseña a utilizar.

3.3. Configuraciones del Software

Esta publicación tiene como objetivo dar a conocer la configuración que se realizó en el proyecto, lo que dará como resultado la utilidad de un controlador de Red Virtual. Iniciamos con la operación y puesta en funcionamiento de la Raspberry Pi.

3.3.1. Puesta a Punto de Operación de la Raspberry Pi

3.3.1.1. Preparar la Rasberry Pi.

1. Descargar Imagen de Raspberry Pi desde el sitio web oficial de Raspberry Pi. https://www.raspberrypi.org/software/.

Figura 49. Descargando Imagen del Sistema Operativo Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

2. Descargar la herramienta de Raspberry Pi Imager desde sitio web oficial de Raspberry Pi para el software recomendado del equipo.

◙ Coloque la tarjeta microSD en el Equipo donde grabará el SO.

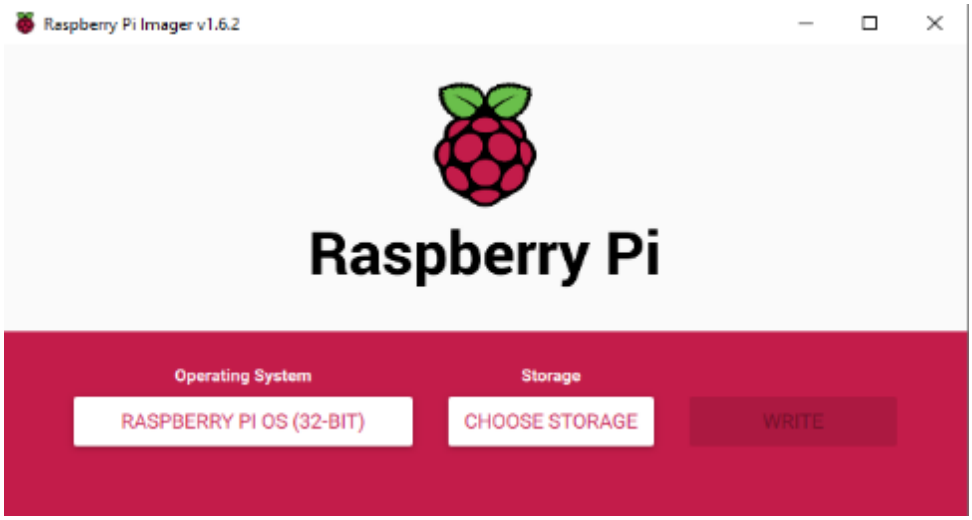


Figura 50. Herramienta Raspberry Pi Imager, (Print Screen).
Fuente: Propia del autor.

3. El sistema operativo Rasbian con imagen de escritorio y software recomendado es el escogido para nuestro proyecto, en archivo ZIP luego de descomprimir tendrá un tamaño superior a 4 GB.

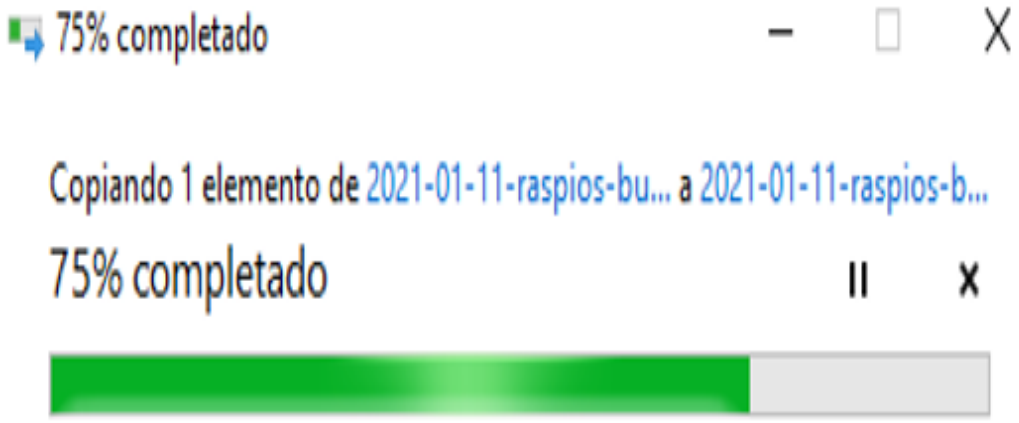


Figura 51. Descomprimiendo el Archivo ZIP de la Imagen, (Print Screen).
Fuente: Propia del autor.

4. Instalación del Sistema Operativo Raspberry Pi.

◙ Ejecute Raspberry. Pi Imager y Cargue el SO en microSD.

- ✓ Tenga en cuenta que las rutas de la instalación estén correctas, la herramienta de instalación formateará la SD.
- ✓ Haga clic en "ESCRIBIR" para iniciar.
- ✓ Ejecute la herramienta del propio sistema operativo para verificar la integridad de la imagen iso descargada

Figura 52. Medio de Instalación de la Imagen Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

5. Al finalizar la escritura de la imagen en la microSD por seguridad la herramienta procede a verificar que está todo correcto. El resultado será uno de estos dos resultados:

    - ✓ Error Verifying write filed. Contents of SD card is different then what was writen to it. Algo ha salido mal y debes repetir el proceso. Podría ser imagen o microSD dañada.
    - ✓ Write Successful “You can now remove the SD card from the reader. Escritura exitosa, "Ahora puede quitar la tarjeta SD del lector".

Figura 53. Verificación de la Instalación del Sistema Operativo, (Print Screen).
Fuente: Propia del autor.

6. Sistema Operativo Listo, ahora podrás llevar la tarjeta microSD a la Raspberry Pi para empezar a usarla.

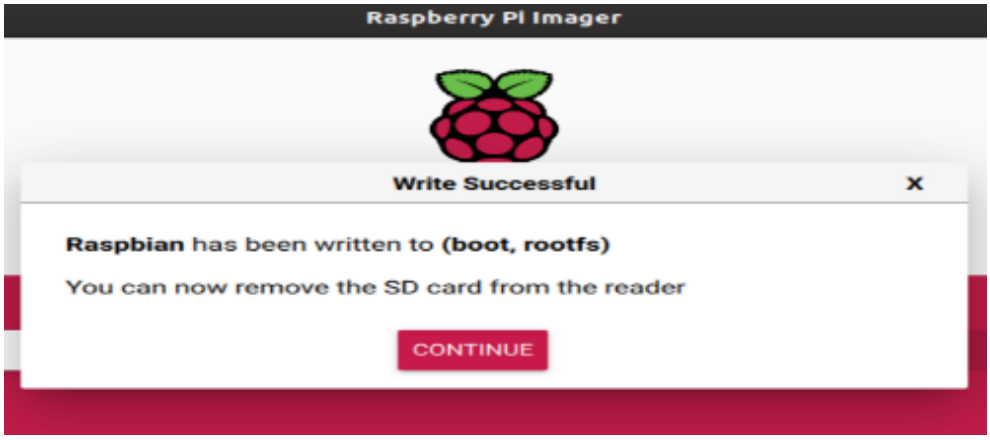


Figura 54. Sistema Operativo Raspberry Pi Instalado Satisfactoriamente.
Fuente: Propia del autor.

3.3.1.2. Arranque y Configuración de Raspberry Pi. Ahora puede insertar la tarjeta SD en la Raspberry Pi y encenderla.

Inicio de sesión del Sistema Operativo, con credenciales predeterminadas.

- Usuario: pi
- Contraseña: raspberry.

Nota: Debe considerar cambiar la contraseña predeterminada de inmediato para asegurarse que su Raspberry Pi sea segura.

USO DE LA TERMINAL PARA CONFIGURAR LA RASPBERRY PI

1. Actualizar el sistema Operativo de Raspberry Pi.

Código Fuente 1. Actualización de Raspberry Pi.

```
# El proceso de instalación exige la utilización de sudo en la terminal
        sudo-su

# Actualizar cache de raspberry pi.
        apt-get update && apt-get upgrade.

#Reiniciar
        sudo reboot

# El proceso de actualización y reinicio en una sola línea de comando.
        sudo apt update && sudo apt -y full-upgrade && sudo reboot
```

Importante = pi@raspberrypi: ~ $

- ✓ **pi:** indica el usuario conectado a la terminal
- ✓ **@:** significa "en"
- ✓ **raspberrypi:** indica el nombre de la máquina a la cual estamos conectados
- ✓ **~:** indica la ruta en la cual nos encontramos, en este caso ruta de inicio predeterminada
- ✓ **$:** indicador para comenzar a escribir nuevas órdenes o comandos

Fuente: Propia del autor.

Nota: Otra forma de configurar la raspberry Pi, en forma gráfica, con la ventana de configuración de Raspbian. sudo raspi-config.

2. Seleccionar System Options - Entrar a la configuración de sistema.

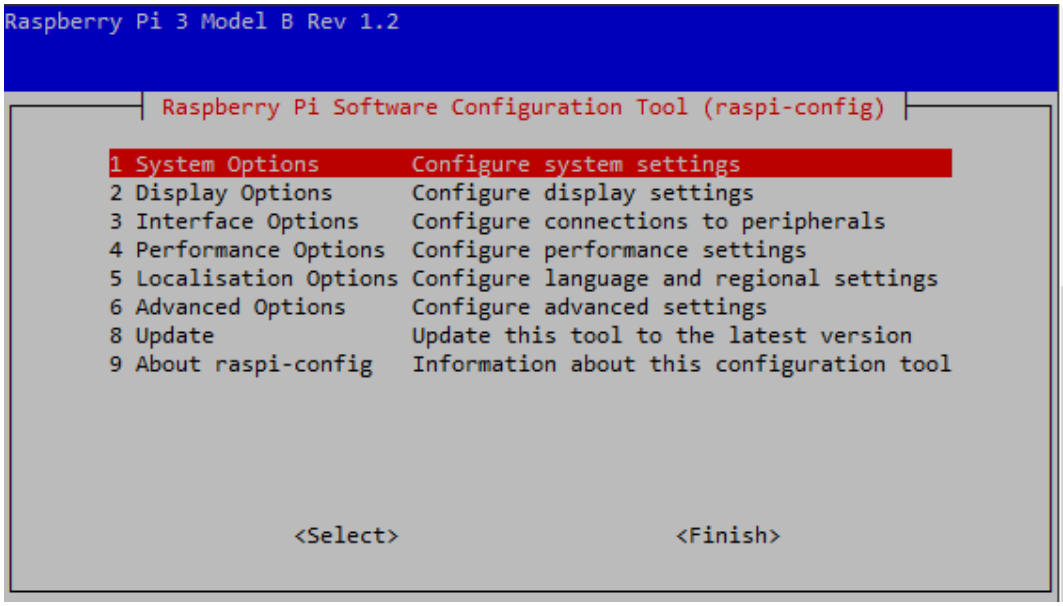


Figura 55. Acceso a la Configuración del Sistema de la Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

3. Seleccionar S4 Hostname – Renombrar el proyecto.

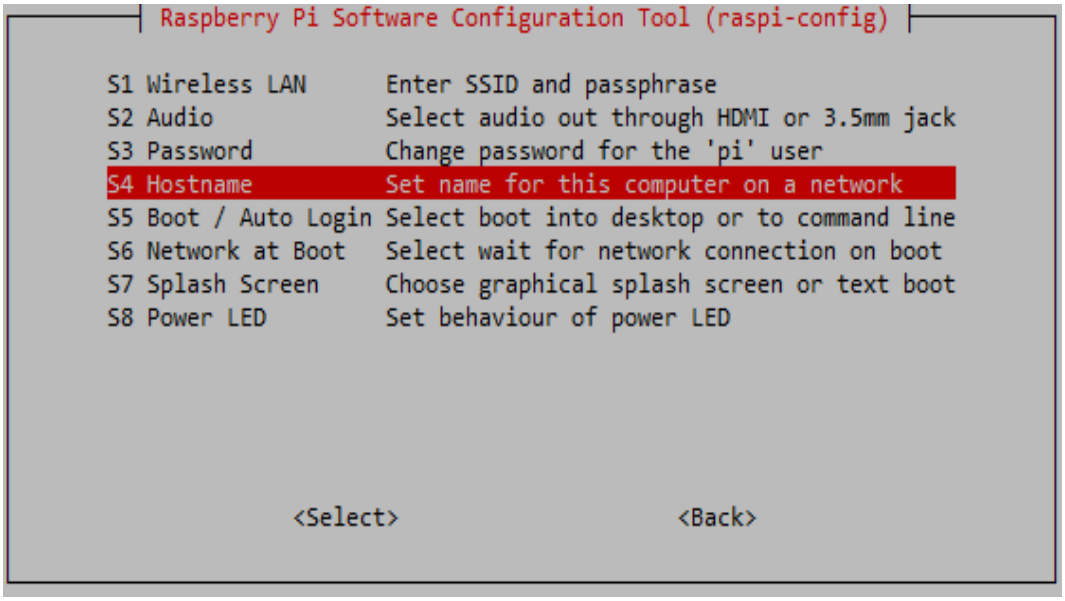


Figura 56. Acceso a la Configuración del Host de Servicio de la Raspberry Pi.
Fuente: Propia del autor.

4. Observaciones RFCS sobre el etiquetado de un nombre de Host.

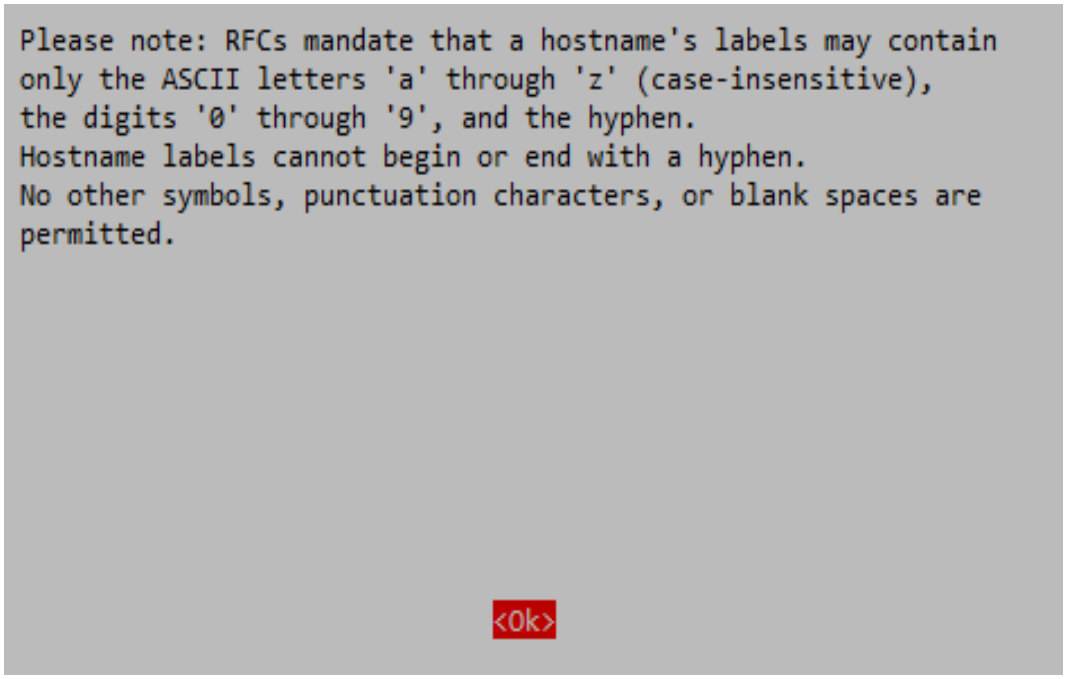


Figura 57. Nota RFCS para Renombrar el Host de Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

5. Renombrar el Host de servicio de la Raspberry Pi.

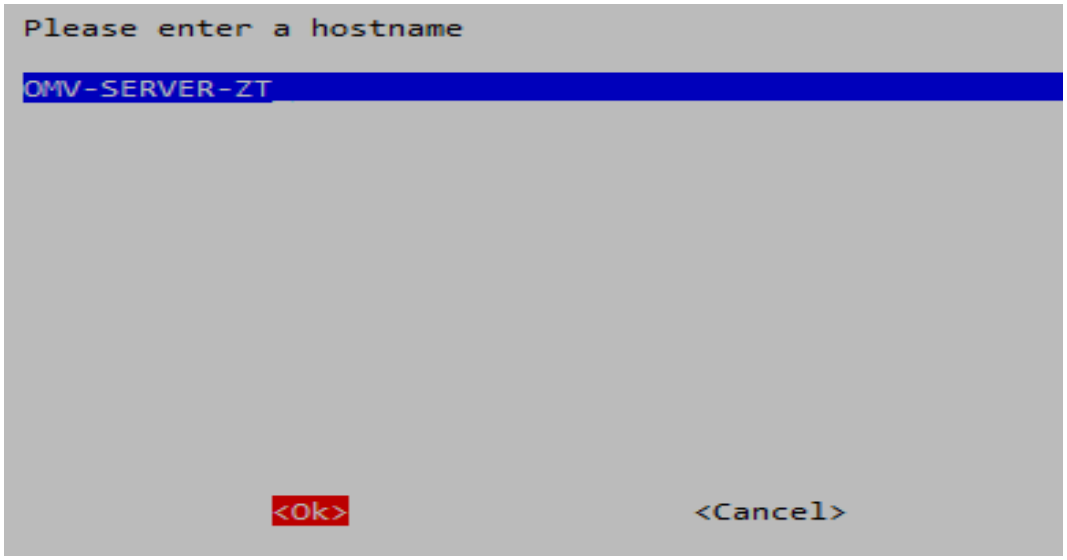


Figura 58. Renombrar el Hostname de la Raspberry Pi, (Print Screen)
Fuente: Propia del autor.

6. Activar/desactivar comunicación de protocolo SSH.

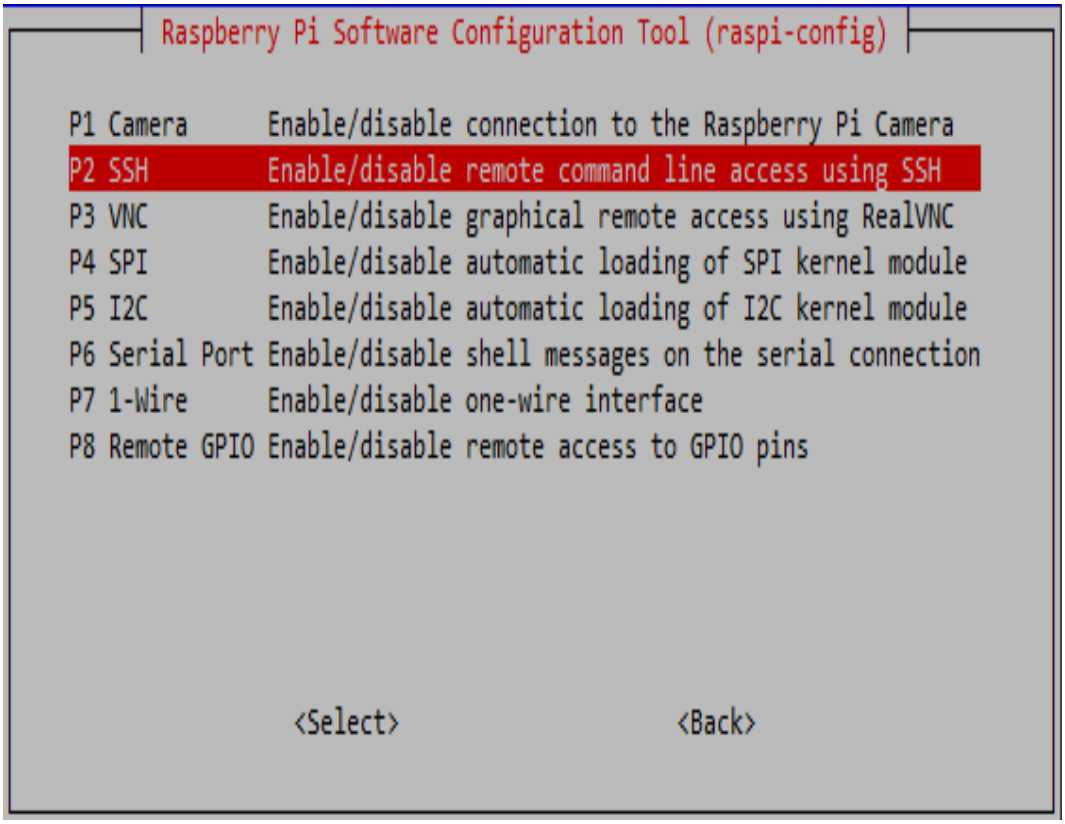


Figura 59. Activación de Comunicación SSH en Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

7. Confirmación de activación de protocolo SSH.

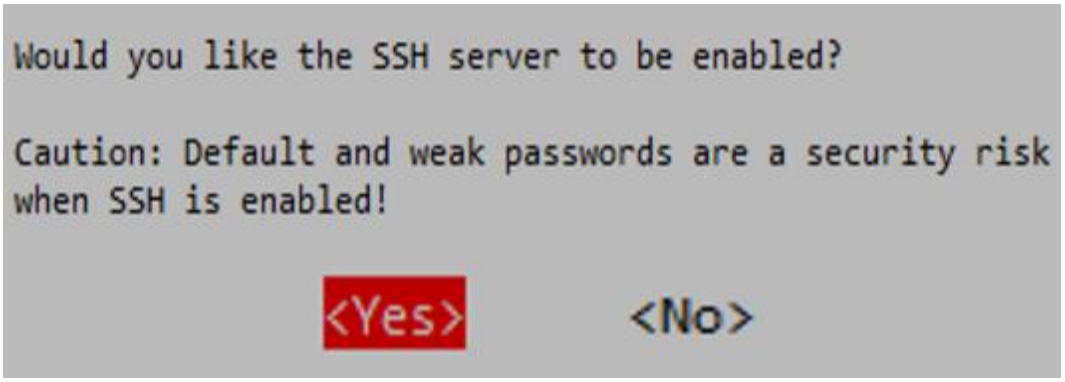


Figura 60. Confirmación para la Activación del Protocolo SSH, (Print Screen).
Fuente: Propia del autor.

8. Protocolo activado, ahora puede comunicarse al Host de la Raspberry Pi, desde cualquier equipo en la red, posteriormente lo podrá realizar desde una red virtual.

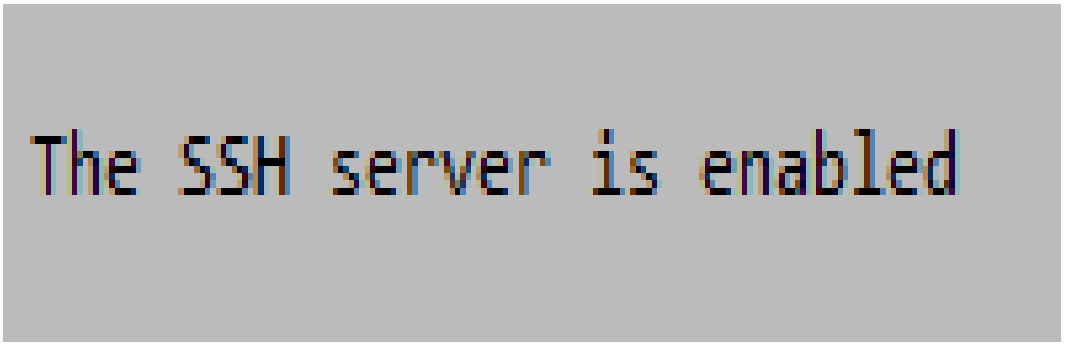


Figura 61. Protocolo SSH en servicio para la Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

9. Configuración Regional del Host de Servicio – Lenguaje y Localización.

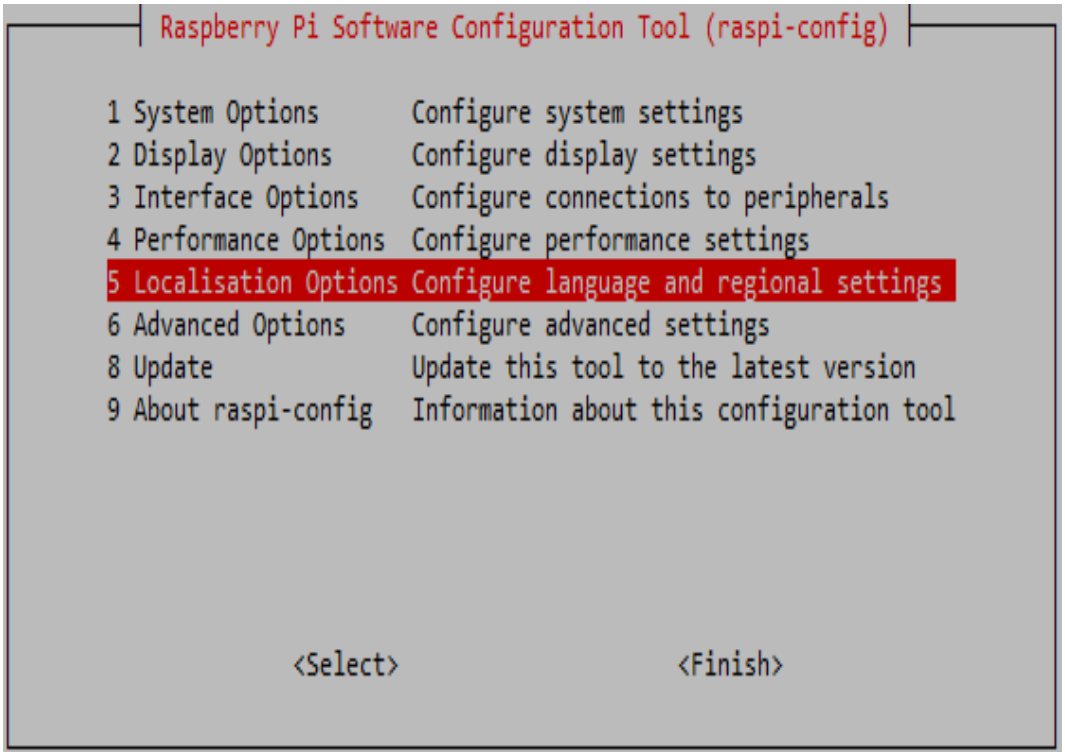


Figura 62. Acceso a Configuración de Zona Horaria Regional del Host de Servicio.
Fuente: Propia del autor.

10. Configuración de Zona horaria automática.

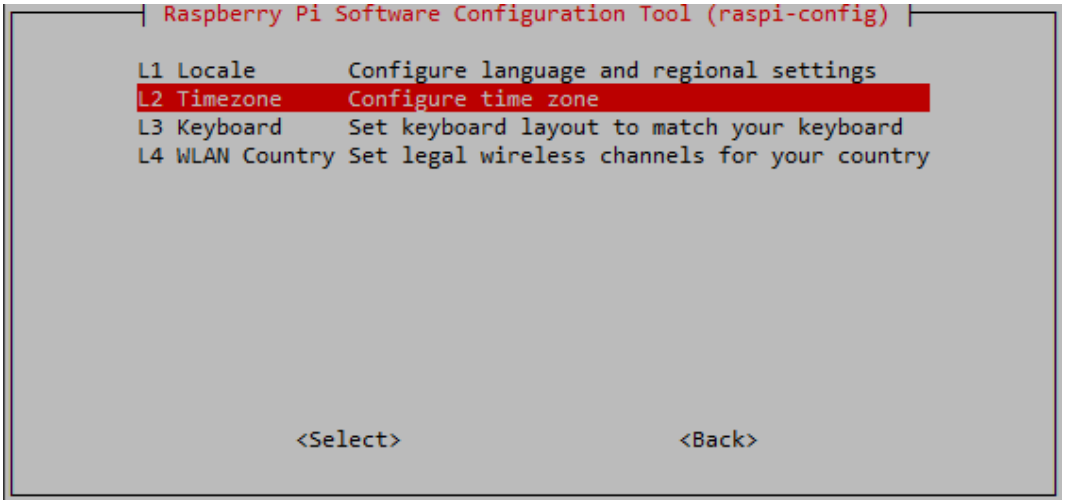


Figura 63. Ajuste de Zona Horaria Regional del Host de Servicio, (Print Screen)
Fuente: Propia del autor.

11. Área Geográfica de la Instalación.

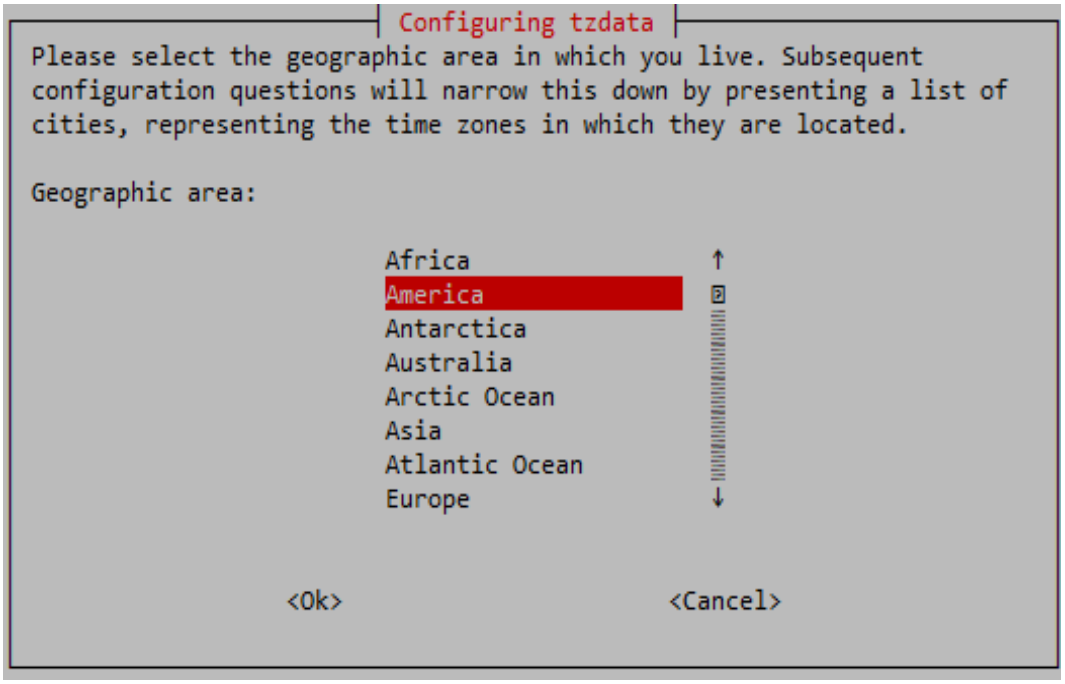


Figura 64. Fijación de Área Geográfica de la Instalación, (Print Screen)
Fuente: Propia del autor.

12. Finalizar configuración.

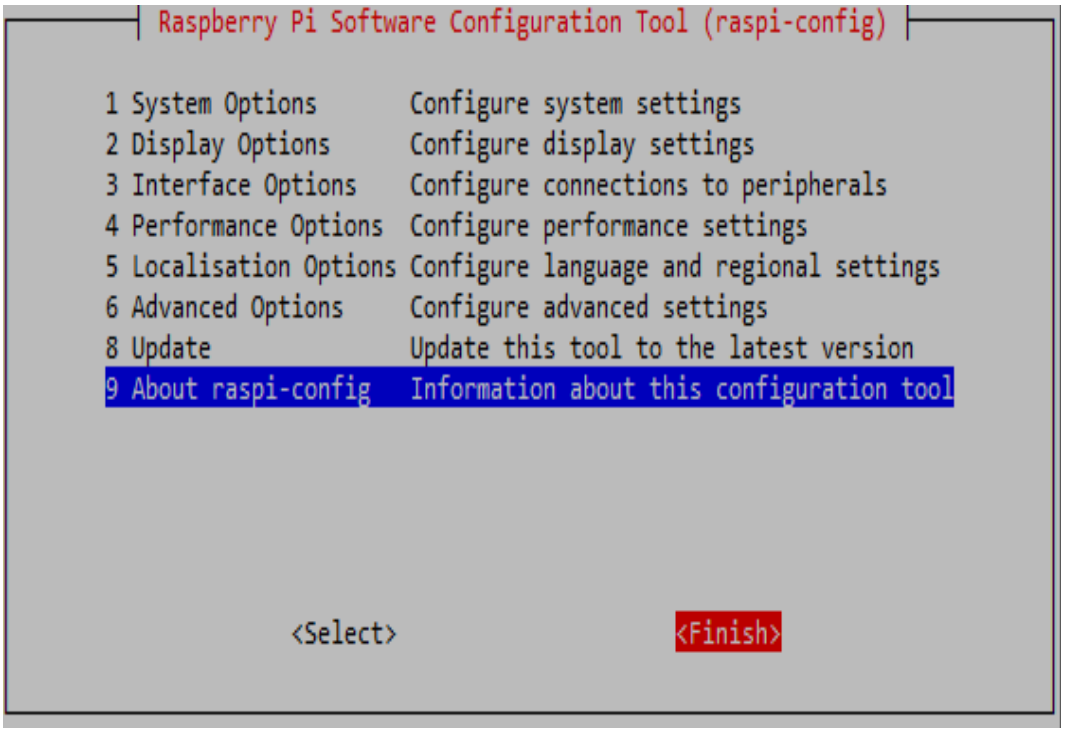


Figura 65. Finalizar la Configuración del Sistema de la Raspberry Pi, (Print Screen).
Fuente: Propia del autor.

13. Reiniciar equipo para actualizar la configuración.

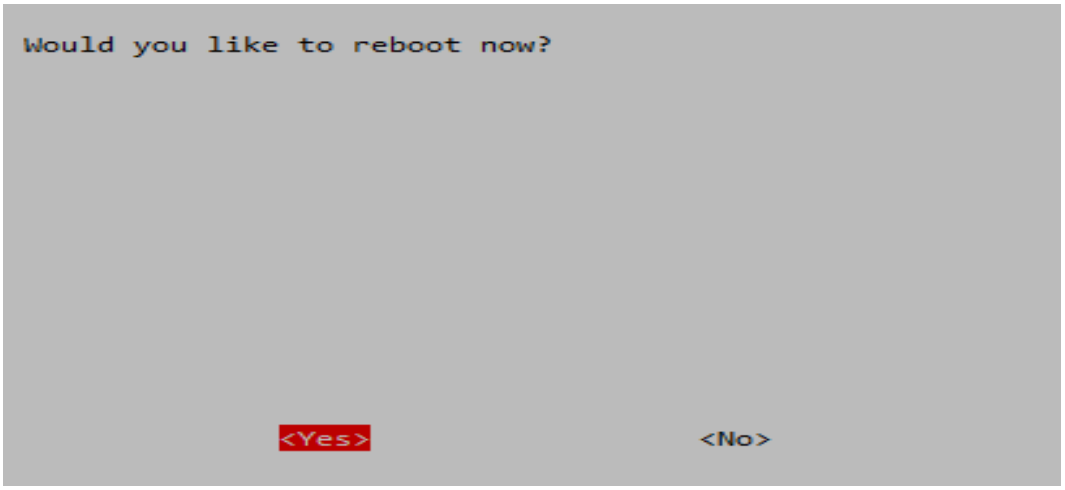


Figura 66. Reiniciar el Host desde la Herramienta de Configuración, (Print Screen)
Fuente: Propia del autor.

3.3.1.3. Configuración de Iptables en Raspberry Pi. Una vez que las rutas están configuradas, es hora de jugar con iptables. Honestamente, es mejor conectarlo al Pi a través de un shell en serie, o simplemente conectar un teclado / mouse y una pantalla. Vas a estropear esto al menos una vez y romperás la conectividad. (James , 2020), (Hashemian, 2021), (hackaday.io, act.2020)

Código Fuente 2. Configuración de Iptables en la Raspberry Pi.

```
# El proceso de instalación exige la utilización de sudo en la terminal
# Comience habilitando el reenvío de paquetes en la Pi:
# Descomente la línea net. ipv4.ip_forward = 1

        $ sudo-su

        $ sudo sysctl -w net.ipv4.ip_forward=1


#Reemplace ZT_INTERFACE con el nombre de la interfaz ZeroTier local; puede obtenerlo en ifconfig, la interfaz se llama ztrtastctq.

        $ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

        $ sudo iptables -A FORWARD -i eth0 -o ZT_INTERFACE -m state --state RELATED, ESTABLISHED -j ACCEPT

        $ sudo iptables -A FORWARD -i ZT_INTERFACE -o eth0 -j ACCEPT


#Una vez hecho esto, debería poder realizar la prueba nuevamente. Su tráfico debería llegar a la interfaz a través de su conexión doméstica, así que conéctese a través de datos móviles y vea cuál es su dirección IP .


#Si esto funciona, es hora de guardar las reglas de iptables para que persistan durante el próximo reinicio. Para hacer esto, ejecute los siguientes comandos:

        $ sudo apt install iptables-persistent

        $ sudo iptables-save > /etc/iptables/rules. v4


#Detalles de la red.

        $ docker network inspect bridge


# Realizar el proceso de actualización y reinicio de sesión en una sola línea.

        $ sudo apt update && sudo apt -y full-upgrade && sudo reboot
```

Fuente: El autor - Adaptado de https://www.jamesleighton.com/2020/05/16/vpn

Interfaces de red del Host. Uso de **ifconfig**

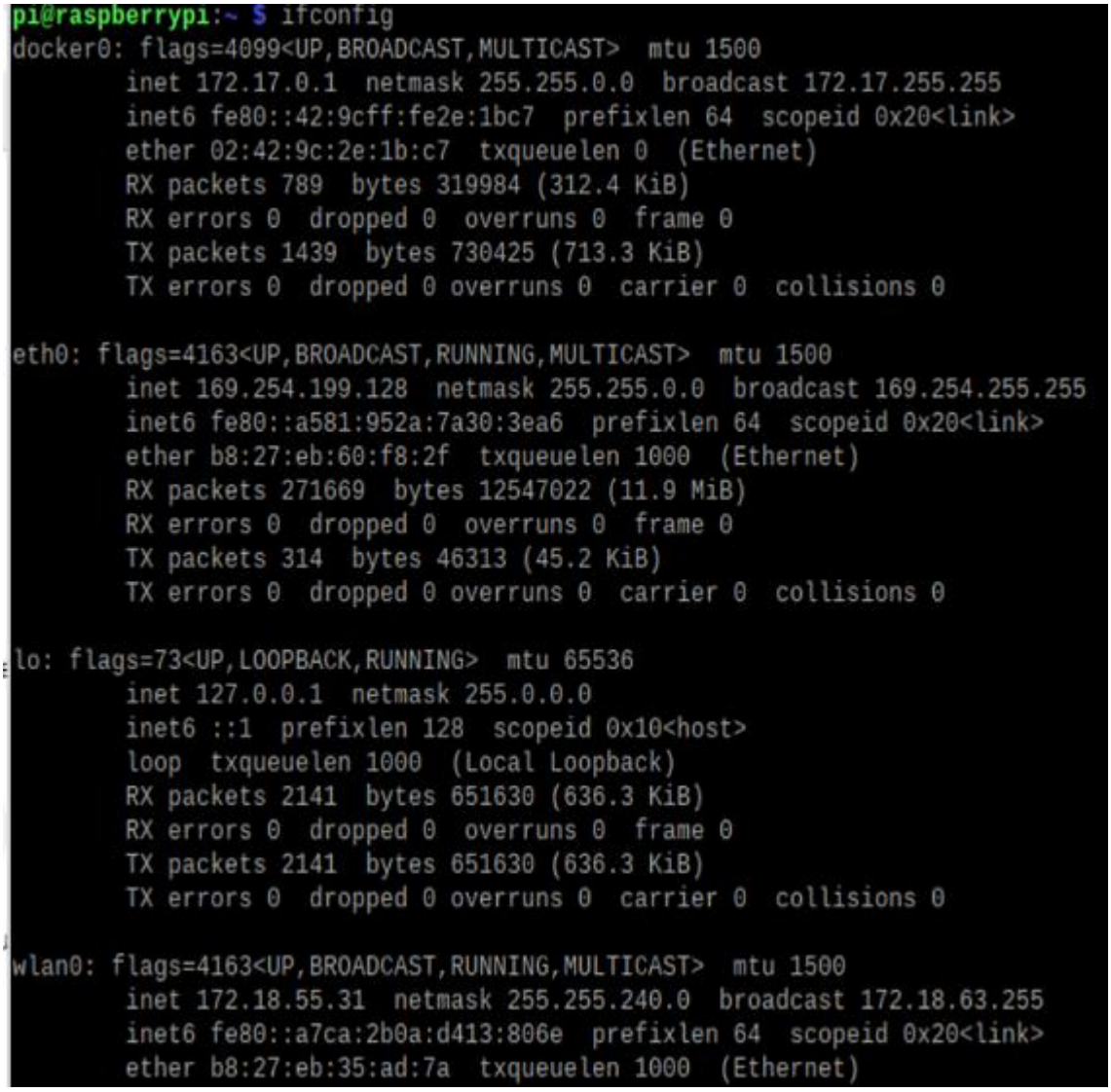


Figura 67. Interfaces de Red del Host de Servicio de la Raspberry Pi. (Print Screen)
Fuente: Propia del autor.

Cambiar contraseña en host de R. Pi, uso del comando: **passwd**.

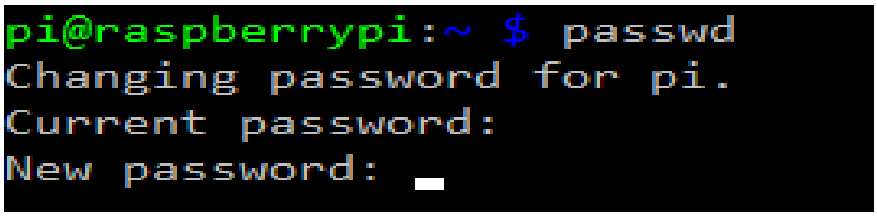


Figura 68. Cambio de Contraseña de Usuario Host en Raspberry Pi. (Print Screen)
Fuente: Propia del autor.

3.3.2. Protocolos de Comunicación

Protocolos de Comunicación SSH, uso en la terminal y asistencia por Putty.

1. Contamos con la herramienta de comunicación mediante el protocolo ssh llamada Putty, disponible para diversos sistemas operativos.

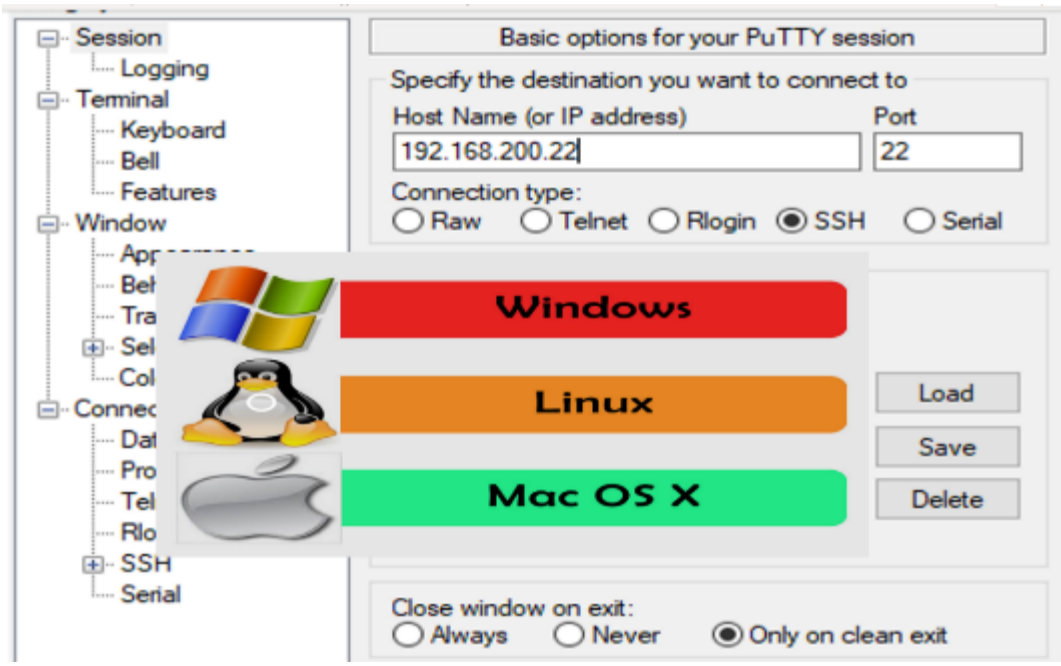


Figura 69. Herramienta de Comunicación SSH - Putty. (Print Screen)
Fuente: Propia del autor.

2. Comunicación ssh en red (LAN) desde Putty, a local host de R.Pi.

Figura 70. Comunicación SSH al Host de la Raspberry Pi con Putty. (Print Screen)
Fuente: Propia del autor.

3. Comunicación de Protocolo SSH en equipos Windows.

Código Fuente 3. Ejecutar Protocolo SSH en la Terminal de Windows.

```
# Para windows desde la terminal (cmd)

# ssh pi@<ip-address-of-pi>
# ssh pi@ ip.local


          $ ssh pi@192.168.200.22
```

Fuente: Propia del autor.

4. Protocolo ssh en red Local (LAN) desde equipos Windows.

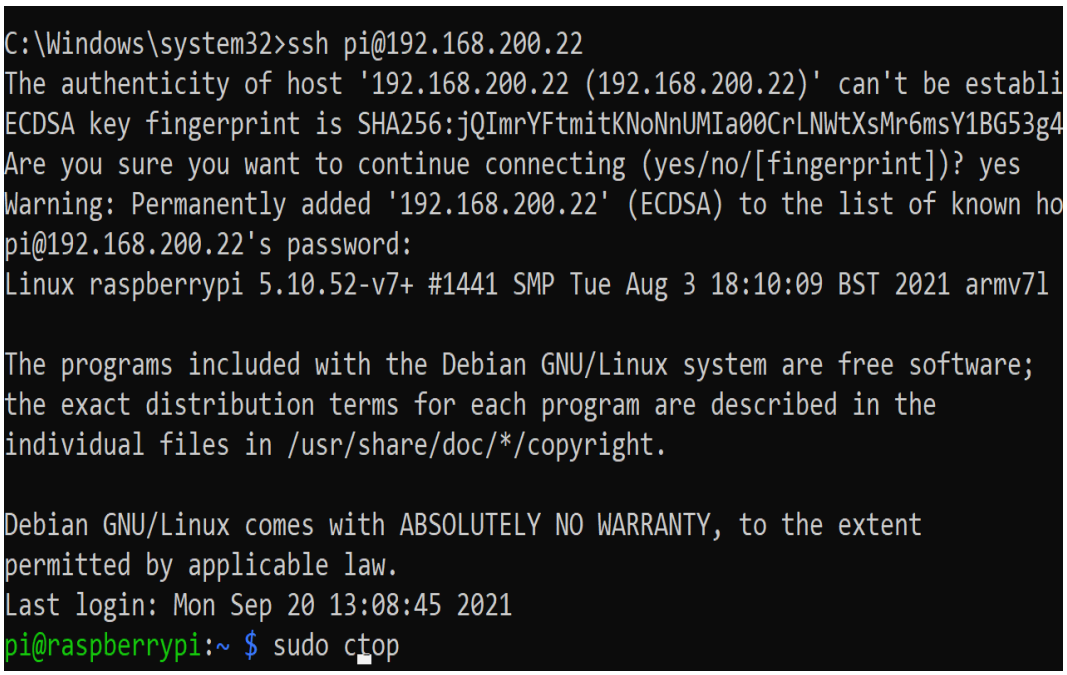


Figura 71. Comunicación SSH en la Terminal de Windows. (Print Screen)
Fuente: Propia del autor.

5. Acceso y Operación de Raspberry Pi, con las credenciales actualizadas

. Host: OMV-SERVER-ZT
Login: pi
Password: *********

3.3.3. Implementación del Software OpenMediaVault

Antes de configurar y ejecutar el software de código abierto se requiere la imagen del software, la misma que se migrara desde los repositorios de GitHub primero debe descargarlo. La instalación y configuración de los atributos necesarios para OMV 5.5 se proporciona en el script de instalación. (OpenMediaVault.org, 2021)

3.3.3.1. Instalar OpenMediaVault (OMV).

Código Fuente 4. Instalación de OpenMediaVault.

```
# El proceso de instalación exige la utilización de la terminal y actualización de la cache
# Conceder privilegios de administrador con sudo su

        $ sudo su

        $  apt-get update && apt-get upgrade

        $ sudo rm -f /etc/systemd/network/99-default.link

        $ wget -O - https://github.com/OpenMediaVault-Plugin- Developers/install Script/raw/master/install |  sudo bash


# Si tiene inconsistencia de privilegios ejecute.

        $ sudo curl -sSL https://github.com/OpenMediaVault-Plugin-Developers /installScript/raw/master/install | sudo bash


#Si no desea utilizar los pasos de configuración de red del script, utilice las siguientes líneas en el terminal.

        $ chmod +x install


# verificar los permisos

        $ ls -lh  %


#instalar OMV

        $ sudo ./install -n


#Reiniciar el Host

        $sudo reboot
```

Fuente: https://github.com/OpenMediaVault-Plugin-Developers/installScript

3.3.3.2. Arranque de OpenMediaVault.

1. Inicio de sesión en la consola web

   - ✓ Para poder acceder a la GUI de OpenMediaVault, deberá conocer la dirección IP del Local Host – (Raspberry Pi).
   - ✓ Escriba la dirección IP proporcionada por la primera pantalla de arranque. Si no la recuerda haga uso de la CLI.

Código Fuente 5. Mostrar Local Host de Raspberry Pi.

```
#En la terminal ejecute.

        hostname -I

        $ Ifconfig
```

Fuente: Propia del autor.

En el portal Web de su elección ejecute https://**R.Pi IPADDRESS]**

Las credenciales predeterminadas se muestran a continuación, se recomienda cambiar el password inmediatamente para mantener la integridad de la operación.

Usuario: admin
Contraseña: openmediavault

Haciendo clic en el icono del ojo, se muestra la contraseña sin mascara.

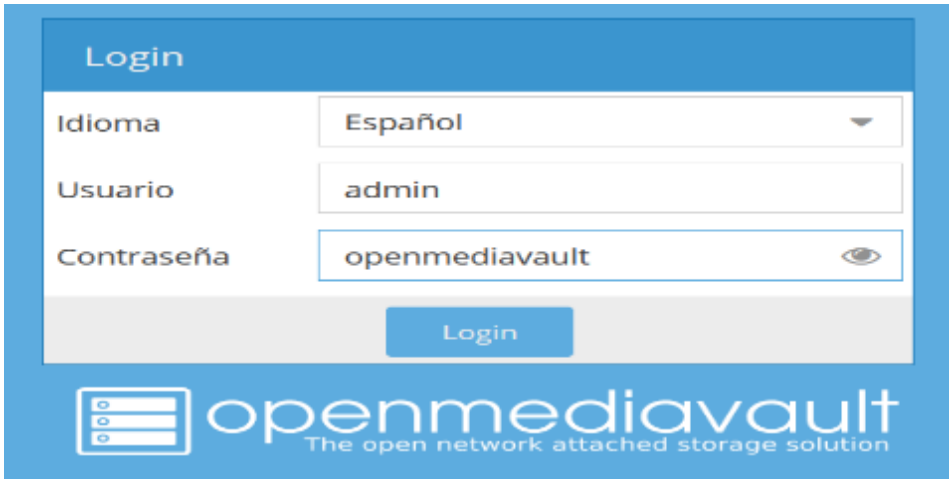


Figura 72. Inicio de Sesión en OpenMediaVault "OMV", (Print Screen)
Fuente: Propia del autor.

2. Panel de control de OpenMediaVault.

Iniciada la sesión, será recibido por el panel de Control de OMV.

Figura 73. OpenMediaVault – Panel de Control, (Print Screen)
Fuente: Propia del autor.

3.3.3.3. Configuración de Sistema del OpenMediaVault.

1. Cambiar contraseña de usuario administrador por seguridad.

◙ Click en Sistema >> Opciones Generales >> Administrador Web.
✓ Gestione el Cambio de contraseña.
◙ Salvar.

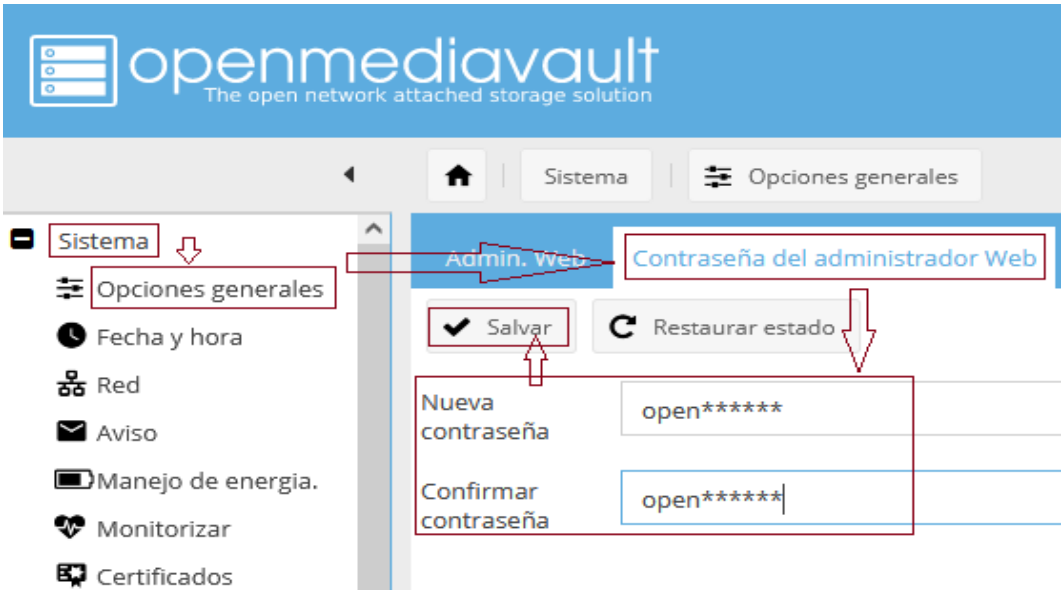


Figura 74. OpenMediaVault – Configuración de Contraseña, (Print Screen)
Fuente: Propia del autor.

2. Control de tiempo sin Interacción.

◙ Click en Sistema >> Opciones Generales >> Administrador Web.
✓ Seleccionar el tiempo a su elección (30 minutos).
◙ salvar.

Figura 75. OpenMediaVault – Control de Interacción, (Print Screen)
Fuente: Propia del autor.

3. Gestión de Redes en OMV.

◙ Click en Sistema >> Red
◙ Click en Interfaces.
✓ Gestionar sus redes de internet.
◙ Click en salvar.

Figura 76. OpenMediaVault – Gestión de Redes, (Print Screen)
Fuente: Propia del autor.

4. Actualizar Nombre de Equipo.

- ◙ Click en Sistema >>Red
- ◙ Click en General.
  - ✓ Cambiar el nombre de host de OMV.

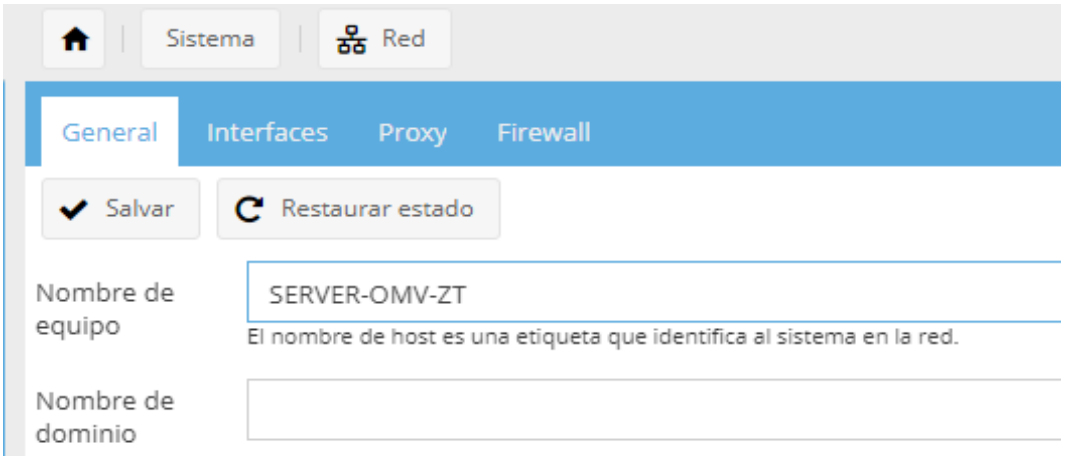


Figura 77. OpenMediaVault – Actualizar el Nombre del Host, (Print Screen)
Fuente: Propia del autor.

5. Crear Certificados.

- ◙ Click en Sistema >> Certificados.
- ◙ Click en SSL.
  - ✓ Crear un certificado de seguridad SSL.

Figura 78. OpenMediaVault – Crear Certificados de Seguridad, (Print Screen)
Fuente: Propia del autor.

6. Actualización de plugins en repositorios de paquetes

- ◙ Click en Sistema >> Gestión de Actualizaciones
- ◙ Click en Refrescar
  - ✓ Seleccionar la casilla de todos los paquetes.
- ◙ Click en Instalar >> Cerrar >>

Una vez que se actualice el repositorio debe salvar para guardar los cambios.

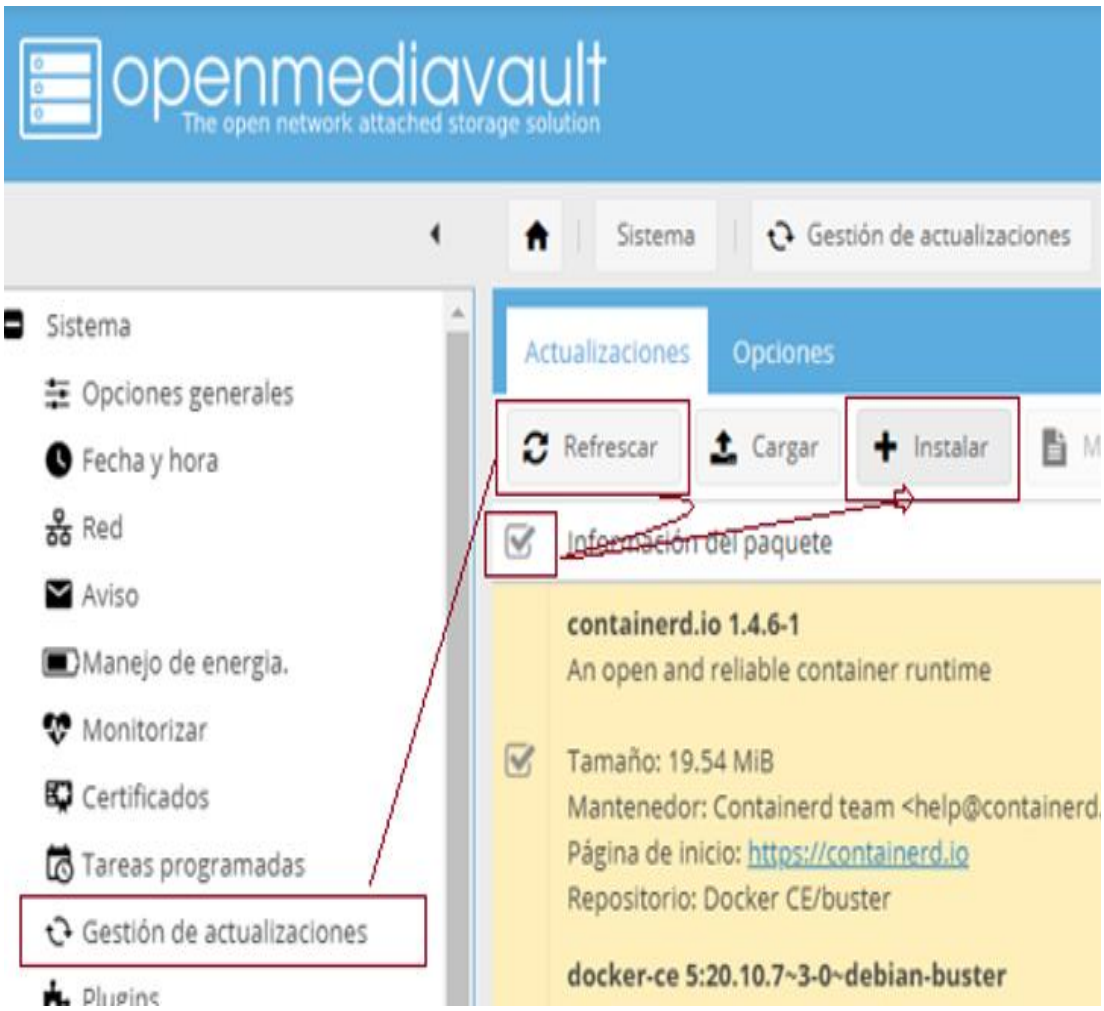


Figura 79. OpenMediaVault – Actualización de Plugin, (Print Screen)
Fuente: Propia del autor.

La configuración en sistemas ha cambiado. Debe aplicar los cambios para que tenga efecto. Aparecerá una ventana con Franja Amarilla.

- ◙ Click en Aplicar >>Actualizando los cambios.

Configuración de Sistema OpenMediaVault OK.

3.3.3.4. Configuración de Almacén OpenMediaVault.

1. Preparar Discos de Almacenamiento.

◙ Click en Almacenamiento >> Discos
- ✓ Verificar la comunicación del SSD Debe mostrarlo en la pantalla de DISCOS.

◙ Click en Scanner.
- ✓ Seleccione el nuevo hardware destinado al almacenamiento del NAS. (SSD detectado)

◙ Click en Borrar.
- ✓ Confirmar la limpieza del medio.>>yes
- ✓ Haga click en rápido como método de limpieza.

◙ Click en Borrar.
◙ Cerrar >> Salvar >> Reinicio.

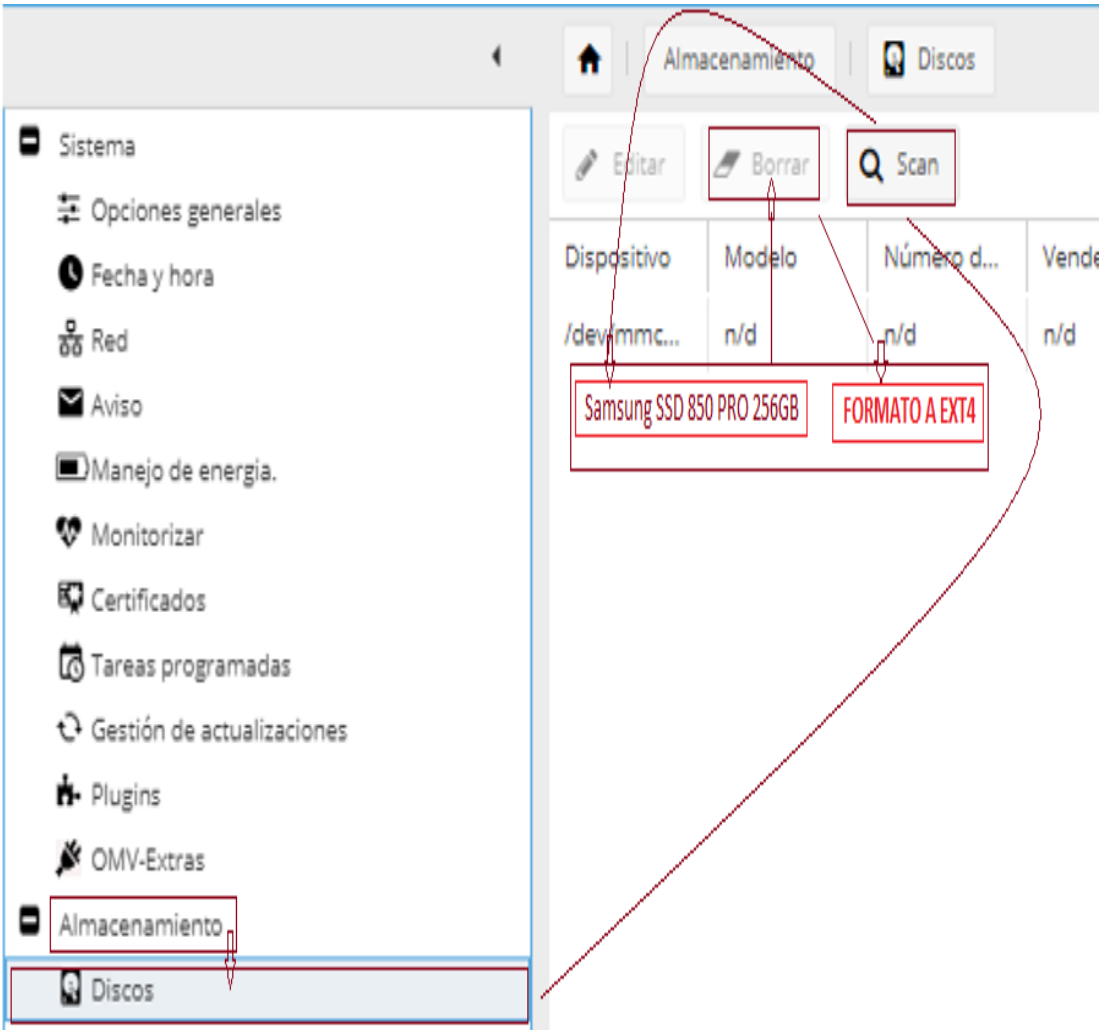


Figura 80. OpenMediaVault – Configuración de Discos, (Print Screen)
Fuente: Propia del autor.

2. Crear Sistema de Archivos en el medio de almacenamiento.

◙ Almacenamiento >> Sistema de Archivos >> Crear.
- ✓ Seleccione el nuevo hardware destinado al almacenamiento.
- ✓ Añada una etiqueta de Identificación y seleccione formato EXT4.

◙ OK
◙ Cerrar >> Salvar >> Reinicio

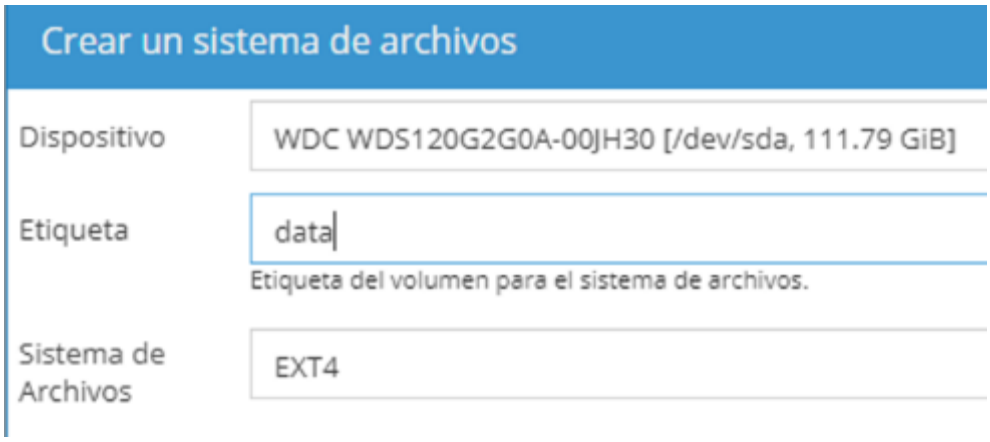


Figura 81. OpenMediaVault – Sistema de Archivos, (Print Screen)
Fuente: Propia del autor.

2.1 Montar el Medio de Almacenamiento NAS.

◙ Almacenamiento >> Sistema de Archivos.
- ✓ Seleccione el hardware de almacenamiento.

◙ Montar >> Aplicar.

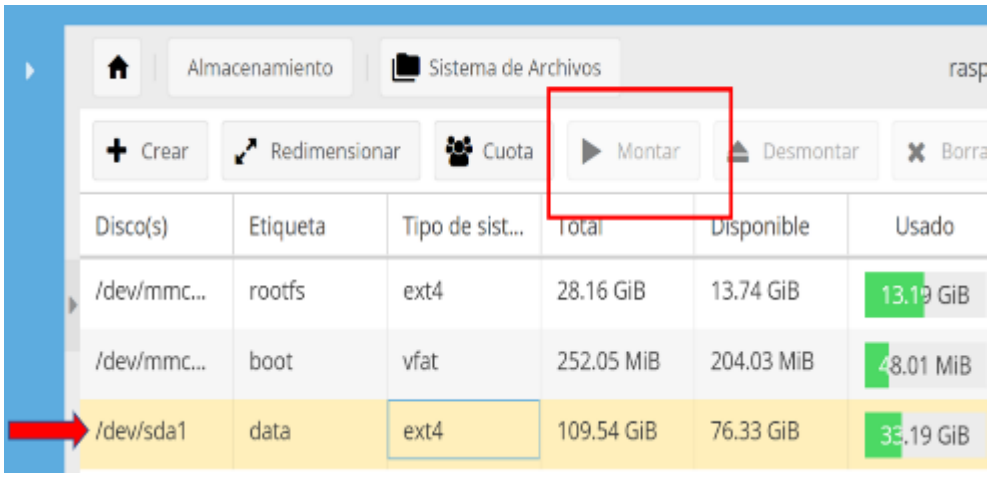


Figura 82. OpenMediaVault – Montar Discos, (Print Screen)
Fuente: Propia del autor.

3.3.3.5. Configuración Permisos de Acceso OpenMediaVault.

1. Configuración de usuarios.

◙ Click en Permisos de acceso >> Usuarios.
- ✓ Prepare la siguiente información para crear usuario nuevo: (Nombre, contraseña, email).

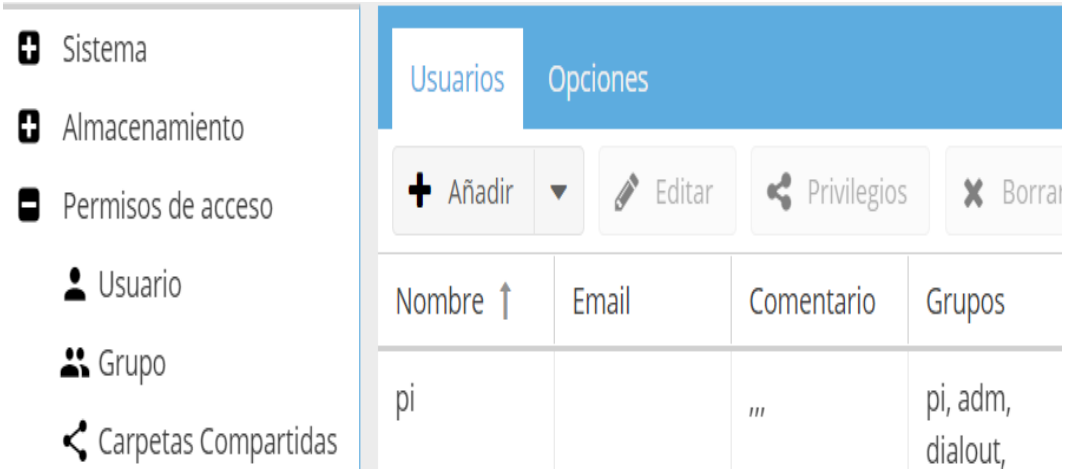


Figura 83. OpenMediaVault – Configuración de Usuarios, (Print Screen)
Fuente: Propia del autor.

1.1 Añadir nuevo usuario.

◙ Click en Usuarios >> + Anadir.
◙ Click Añadir usuario >> General.
- ✓ Ingrese Nombre de usuario, email y contraseña.
- ✓ Puedes agregar varios usuarios a OMV, donde puedes establecer diferentes niveles de permiso para cada uno.
- ✓ Considere si admite permisos de control al usuario.

Figura 84. OpenMediaVault – Crear Nuevo Usuario, (Print Screen)
Fuente: Propia del autor.

1.2 Añadir Grupo de Usuarios y Accesos.

◙ Click en Usuarios >> + Anadir.
◙ Click en Anadir Usuario" >> Grupos.
- ✓ Seleccione los usuarios que formarán el grupo.
- ✓ Seleccione los créditos para la cuenta asignada a los grupos.

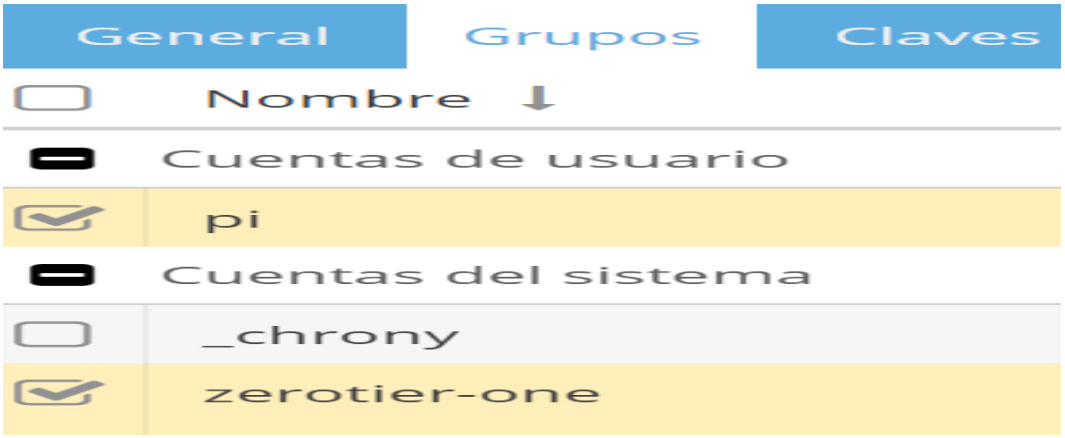


Figura 85. OpenMediaVault – Añadir Grupo de Usuarios, (Print Screen)
Fuente: Propia del autor.

2. Permisos de acceso a Carpetas Compartidas.

◙ Click en Carpetas Compartidas.
◙ Click en Añadir >> prepare la siguiente información.
- ✓ Identificación de directorios compartidos y privados.
- ✓ Etiquete los directorios de sistema.
- ✓ Considere crear rutas de accesos privados a futuro.

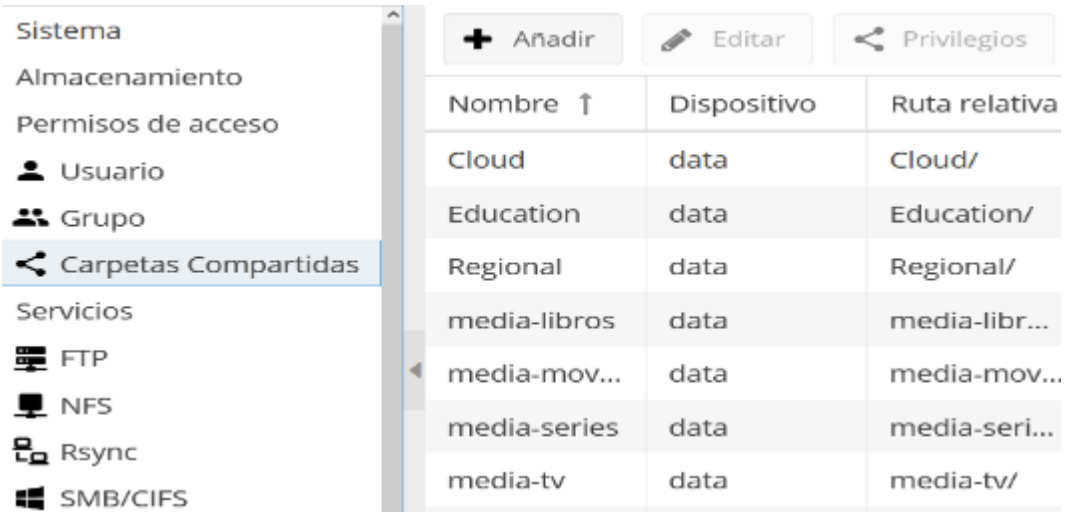


Figura 86. OpenMediaVault – Mostrar Carpetas Compartidas, (Print Screen)
Fuente: Propia del autor.

2.1 Crear Nueva Carpeta Compartida.

- ◙ Click en Carpetas Compartidas.
- ◙ Click en Añadir >> Añadir Carpeta Compartida.
  - ✓ Escriba el Nombre de la nueva carpeta. “Sistema”.
  - ✓ Seleccione el Dispositivo de Almacenamiento disponible.
  - ✓ Verifique que la ruta / tenga la misma etiqueta del nombre.
  - ✓ Conceda permisos de administración y acceso.
- ◙ Salvar >> Aplicar >> Guardar Cambios.

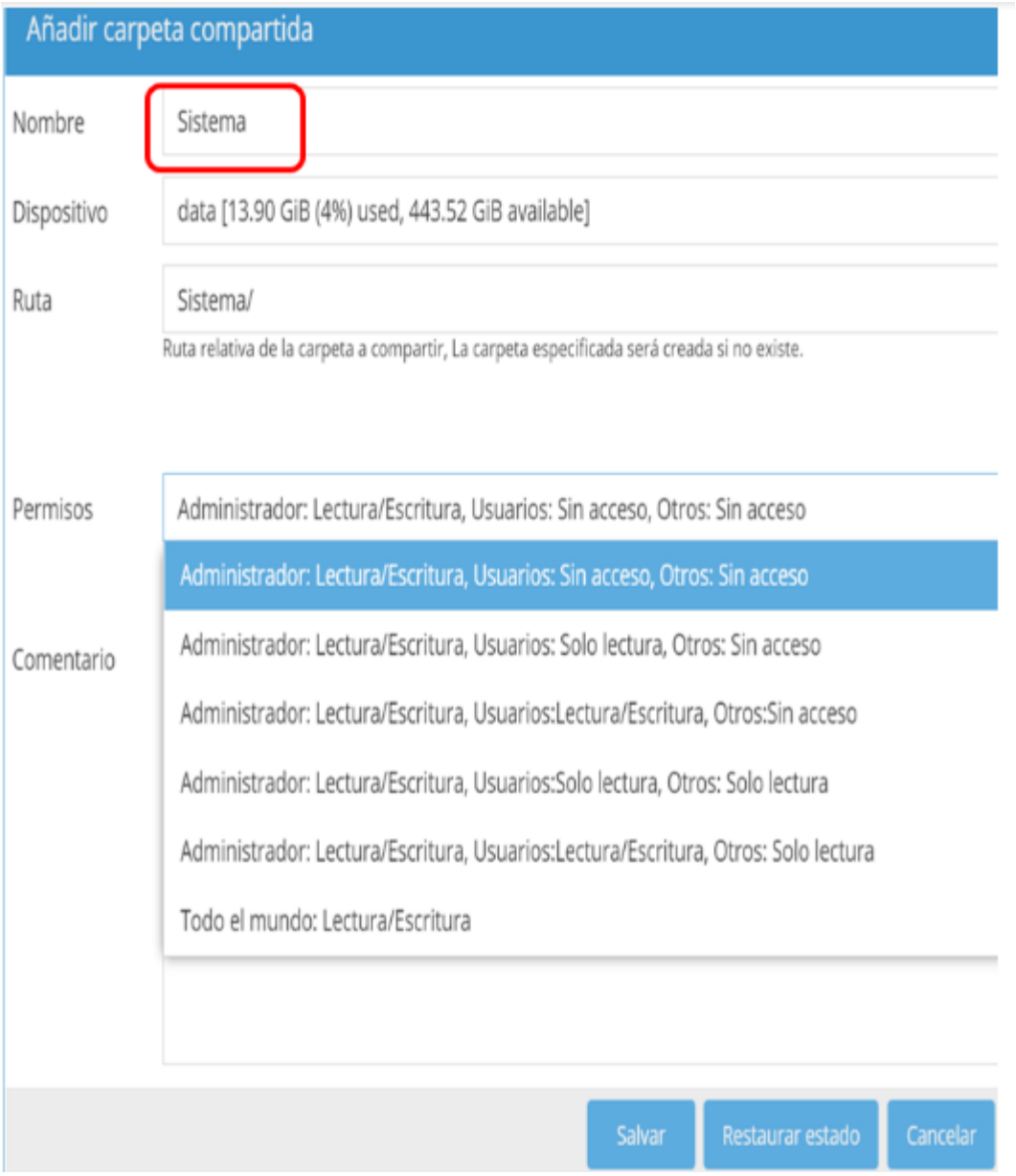


Figura 87. OpenMediaVault – Añadir y Compartir Carpetas, (Print Screen)

Fuente: Propia del autor.

2.2 Añadir Privilegios de las Carpetas Compartidas.

- Click en Carpetas Compartidas.
- Click en Privilegios
  - ✓ Administre los privilegios de usuarios.

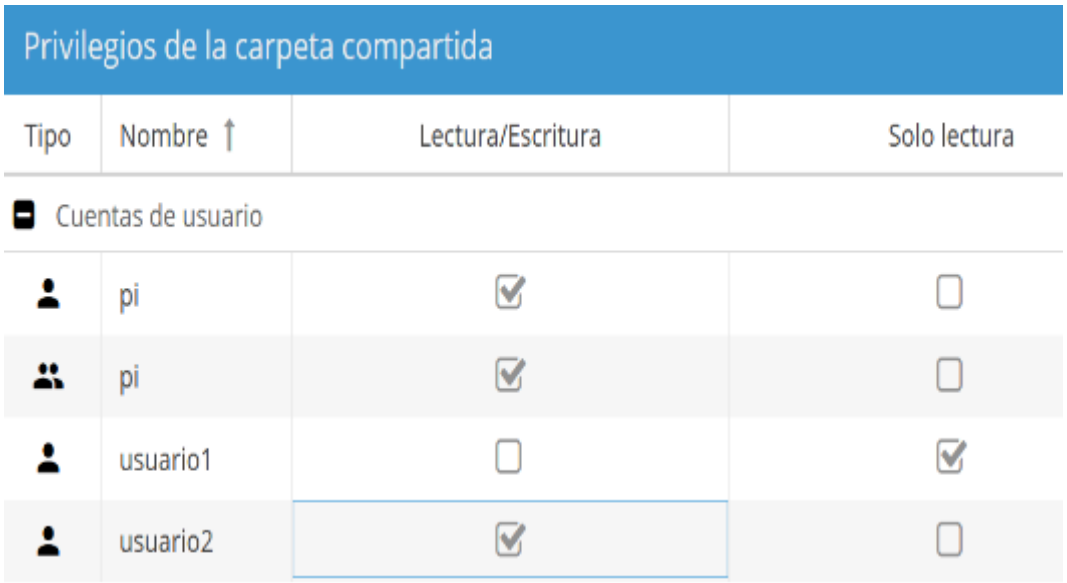


Figura 88. OpenMediaVault – Privilegios de Carpetas Compartidas, (Print Screen)
Fuente: Propia del autor.

2.3 Ruta de acceso absoluto avanzado de Carpetas Compartidas.

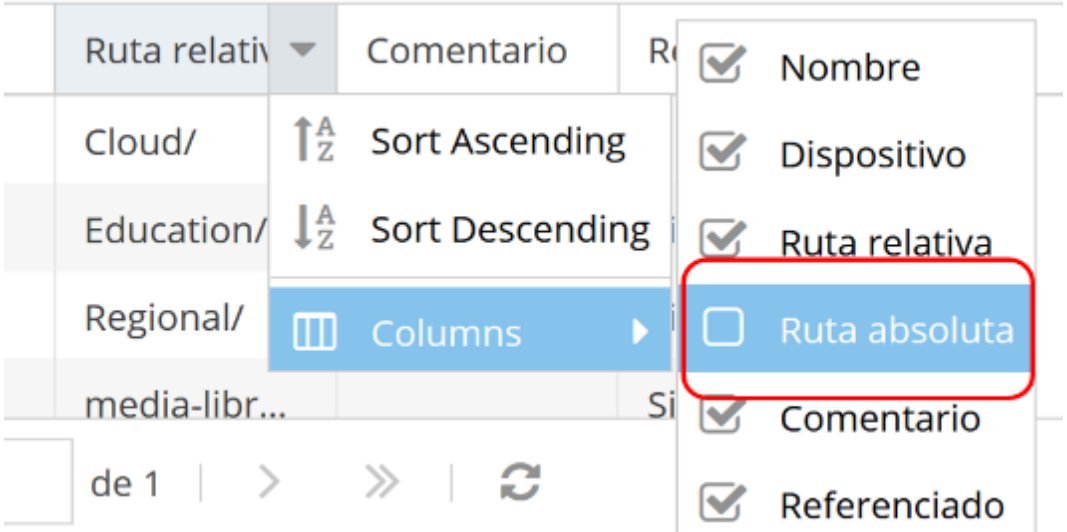


Figura 89. OpenMediaVault – Gestión de Ruta Absoluta de Carpetas, (Print Screen)
Fuente: Propia del autor.

2.4 Nombre de Ruta Absoluta avanzado de Carpetas Compartidas.

Figura 90. OpenMediaVault – Ruta Absoluta de Carpetas, (Print Screen)
Fuente: Propia del autor.

3.3.3.6. Configuración de Servicios OpenMediaVault.

1. Guía de Configuración de carpetas con protocolo Samba.

- ◙ Click en Servicios >> SMB/CIFS.
- ◙ Click en Usuarios.
  - ✓ Antes de que cualquier dispositivo pueda acceder a las carpetas compartidas, deberás habilitar **SMB / CFIS,** que son servicios compartidos por OMV.
  - ✓ Si lo deseamos también podemos habilitar los directorios home de usuario se compartan por Samba, siempre y cuando los tengamos activados... Los archivos creados previamente deben cambiar su modo de permiso. Compruebe también que no tiene habilitado solo lectura. Esta opción anula los privilegios y POSIX.
  - ✓ Estas directivas NO son ACL. Lo siguiente sería elegir qué carpetas queremos compartir mediante este servicio, para ello, simplemente tenemos que añadirlas en la pestaña Compartidos,

haciendo click en Añadir y seleccionando la carpeta que deseemos.

- Guardar.

Figura 91. OpenMediaVault – Configuración de Aplicación Samba. (Print Screen)
Fuente: Propia del autor.

2. Disponibilidad de Carpetas compartidas.

Las carpetas compartidas estarán animadas con el botón verde.

Figura 92. OpenMediaVault – Carpetas Compartidas con Protocolo Samba.
Fuente: Propia del autor.

3. Servicios disponibles compartidos

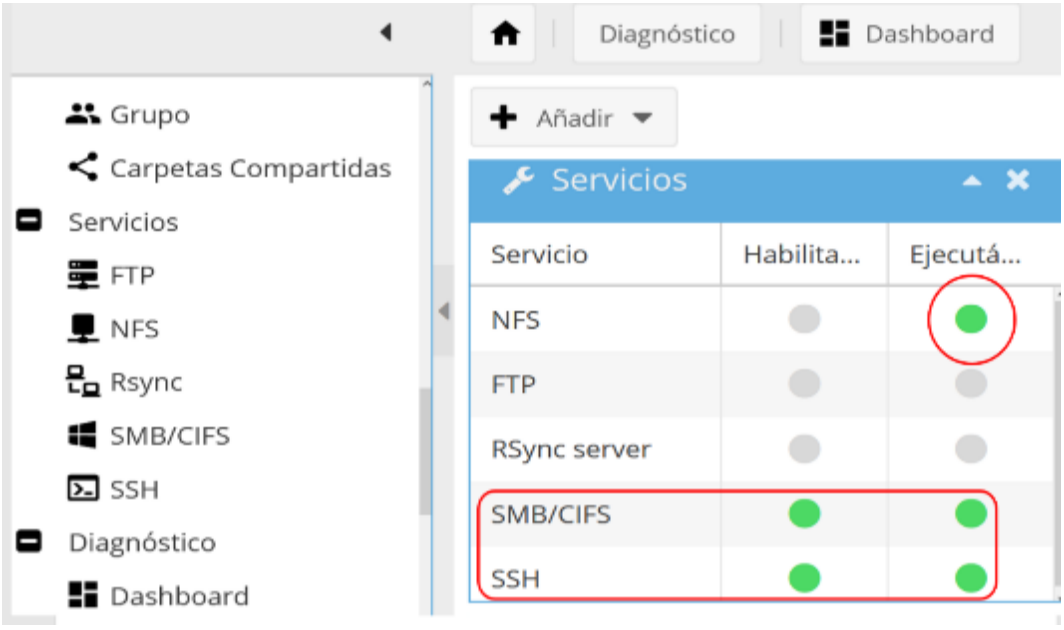


Figura 93. OpenMediaVault – Disponibilidad de Servicios Activos, (Print Screen)
Fuente: Propia del autor.

4. Guardar los cambios del OpenMediaVault

- Click en Salvar.
- Click Aplicar >> Confirmación de aplicar cambios.
  - ✓ Una vez hecho esto se abran guardado todos los cambios realizados en el Sistema.
- Salvar

¡Sistema de Almacenamiento en la Red Listo!

Figura 94. OpenMediaVault – Aplicar Cambios en la Configuración, (Print Screen)
Fuente: Propia del autor.

Al aplicar la configuración de OMV tendrá todas las características detalladas en este Proyecto. Un sistema de administración web, un sistema sencillo de actualización e instalación de paquetes, administrador de volúmenes, S.M.A.R.T. Wake on lan, notificaciones por correo electrónico, un sistema FreeNAS para cargar todo tipo de servicios y protocolos para compartir datos, un interesante sistema de extensiones o plugins que otorgan un gran potencial para comunicaciones a nivel de IoT, y muchas más funciones avanzadas a disposición de las TIC.

Por último, es recomendable dejar desactivado el acceso root y crear un usuario específico para SSH añadiéndolo, como ya hemos mencionado la seguridad del sistema responde a la configuración que aplique el administrador.

Podremos acceder a esta carpeta mediante un navegador, o accediendo mediante la ruta IP/hostname/nombre.

3.3.4. Implementación del Contenedor de Docker Host

1. Script de instalación de Docker para arquitectura arm, encontramos una guía detallada en, (Shahriar , 2018) Linuxhint.

Código Fuente 6. Instalación de Docker.

```
METODO 1:  EJECUTANDO EL SCRIPT

# Crear carpeta para directorios en home/pi/docker
# Conceder privilegios de administrador con sudo su

$ sudo su

# Usar apt para instalar Docker.

$ sudo apt-get update # actualizar

# Instalar paquetes de los que depende HTTPS

$ sudo apt-get install apt-transport-https
    \ ca-certificates \software-properties-common

# Agregar la clave GPG de Docker

$ curl -fsSL https://yum.dockerproject.org/gpg | sudo  apt-key add –

# Verifique la identificación de la clave:

$ apt – key fingerprint 58118E89F3A9128############

# Configure un repositorio estable:

$ sudo add-apt-repository \"deb https://apt.dockerproject.org/repo/ \
  raspbian-$(lsb_release -cs) \   main"

 # Nota: Si encuentra problemas con el comando add-apt-repository, puede intentar agregar la siguiente línea a sources.list de la fuente del software Raspberry Pi, de la siguiente manera:

$ sudo nano /etc/apt/sources.list
```

Agregue una línea:

$ deb https://apt.dockerproject.org/repo/ Raspbian- RELEASE main

Ajuste el RELEASE anterior según la versión de su sistema

$ lsb_release -cs

Actualizar repositorios y continuar con la instalación de Docker

$ sudo apt-get update

$ sudo apt-get -y install docker-engine

METODO 2: EJECUTANDO UN DOCKER PULL (GitHub, Git Lab):

Instalación a través de compilaciones OpenSource.

$ curl -fsSL https://get.docker.com -o get-docker.sh

$ sh get-docker.sh

$ curl -fsSL https://test.docker.com -o test-docker.sh

$ sh test-docker.sh

NOTA: Asegúrese de comprobar la imagen que ha descargado, debe coincide con el contenido de #install.sh ubicado en https://github.com/docker/docker-install antes de ejecutar.

desde git conociendo la dirección de una imagen segura a #través de:

$ sudo docker pull "nombre"

Instalar la imagen descargada

$ ls -lh % verifica los permisos

$ sudo ./install -n

Prueba de Docker Anfitrión, creamos un contenedor de prueba (hello-word).

$ sudo docker run hello-world

RESPUESTA: ¡Hello from Docker!

Fuente: https://get.docker.com/; https://linuxhint.com/install_docker_raspberry_pi/#

2. Comandos para el Administrador de Docker, (Geekflare, 2019)

Código Fuente 7. Uso de la CLI en la Administración de Contenedores.

```
        IMAGENES

 # Verificar imágenes de contenedores creados

            $ docker images -a
            $ sudo docker image ls


# Eliminar una imagen añadiendo el indicador -q para pasar el ID de la imagen a docker rmi:

            $ docker images -a
            $ docker rmi $(docker images -a -q)


        ENUMERAR CONTENEDORES

# Enumerar y detallar contenedores que se están ejecutando.

            $ sudo docker ps


# Puede enumerar todos los contenedores que ha creado:

            $ sudo docker ps -a


# Iniciar el contenedor 01234568547 de nuevo, con el siguiente comando:

            $ sudo docker start 01234568547


# Para entrar en el contenedor pudiendo elegir el Shell o el usuario ejecute el siguiente comando:

            $ docker container exec 01234568547


        ELIMINAR UN CONTENEDOR

# Detenga el contenedor ID 01234568547 con el siguiente comando.

            $ sudo docker stop 01234568547


# Elimine el contenedor ID 01234568547 de la ventana acoplable con.
```

$ sudo docker rm 01234568547

VOLUMENES

Crear volumen para almacén, que utilizará el contenedor de ventana acoplable.

$ docker volume create

Listar los volúmenes creados en Docker.

$ docker volume ls

puede eliminar uno o más volúmenes con el comando

$ docker volume rm (nombre)

INFORMACION DE DOCKER

Obtenga información detallada sobre el contenedor; versión del kernel, el #número de contenedores e imágenes, etc.

$ sudo docker info && docker inspect

Ver los logs del contenedor

$ docker logs.

Comprobar cuánta memoria, CPU, E/S de disco, E/S de red, etc.
Ver las estadísticas del contenedor con el siguiente comando:

$ sudo docker stats

HISTORIAL

Muestra el historial de una imagen de la ventana acoplable con el nombre de la imagen mencionado en el comando.

$ docker history (nombre)

ACTUALIZACION DE DOCKER

Actualice las configuraciones - Esto muestra todas las opciones de actualización.

$ docker update –help

Reiniciar contenedores

$ sudo docker restart (nombre)

Fuente 1: Adaptado de https://geekflare.com/es/docker-networking/, [Geekflare,2019]

3. Administrador De Redes en Docker.

Código Fuente 8. Uso de la CLI en el Administrador de Redes en Docker.

 # REDES DE DOCKER

#El siguiente comando en Docker enumera los detalles de toda #la red en el clúster**.**

$ sudo docker network ls

#Administrar redes en docker

#Comandos:

$ docker network inspect: *:* Nos da información de la red

$ docker network connect: Conectar/desconectar el container con una red.

$ docker network create: Crear una red.

$ docker network prune: Eliminar todas las redes no utilizadas

$ docker network rm: Eliminar una o más redes

$ ip a: puente de red específico para cada red

'docker network --help' soporte del comando.

 # REDES VIRTUALES EN DOCKER

#Comandos:

$ iptables -L verificar reglas iptables

$ iptables -L docker Definir reglas iptables de docker.

$ ip_forward=true Habilita el tráfico exterior

#Comando para borrar todas las redes virtuales utilizadas por #docker

$ docker stats –format "table {{.Container}}\t{{.CPUPerc}}\t{{.MemUsage}}\t{{.MemPerc}}"

Fuente 1: Adaptado de https://geekflare.com/es/docker-networking/, [Geekflare,2019]
Fuente 2: Adaptado de https://linuxhint.com/install_docker_raspberry_pi/#; [Shahriar,2018]
Fuente 3: Adaptado https://iesgn.github.io/curso_docker_2021/sesion4/[Udemy.com,2021]

3.3.5. Implementación del Software Portainer

A continuación, se detalla en pasos numerados la instalación de Portainer para facilitar la guía descriptiva de configuración y puesta en operación del software.

3.3.5.1. Instalar Portainer.

1. Instalar el Script de Portainer, desde los repositorios de DockerHub.

Código Fuente 9. Instalación de Portainer.

```
# El proceso de instalación exige la utilización de la terminal y actualización de la cache

# Conceder privilegios de administrador con sudo su

        $ sudo su

        $ apt-get update && apt-get upgrade


    #Crearemos un volumen al cual denominaremos Portainer

        $ docker volume create portainer_data


      #Ejecutamos el siguiente script de instalación.

        $ docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock -v
portainer_data:/data portainer/portainer


    # Lo que hace es escuchar en el puerto 9000. Allí es donde lo tendremos arrancado. Por
tanto, para acceder a él, simplemente debemos ir a http://localhost:9000 en nuestro navegador.

    # Instalar Portainer realizando un Pull a la imagen de docker Portainer.

        $ sudo docker pull portainer / portainer: linux-arm


    #El nuevo contenedor que ejecutará Portainer está utilizando el puerto 9000 en su
Raspberry Pi, si ya lo está usando (esto es poco #probable), deberá cambiar los puertos a
continuación

        $ sudo docker run—restart always -d -p 9000: 9000 -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data: / data portainer / portainer: linux-
arm
```

Fuente: https://github.com/portainer/portainer, [github.com,act2021]

2. Repositorios DockerHub

DockerHub es el repositorio de donde descargamos la imagen de Portainer, (portainer/portainer).

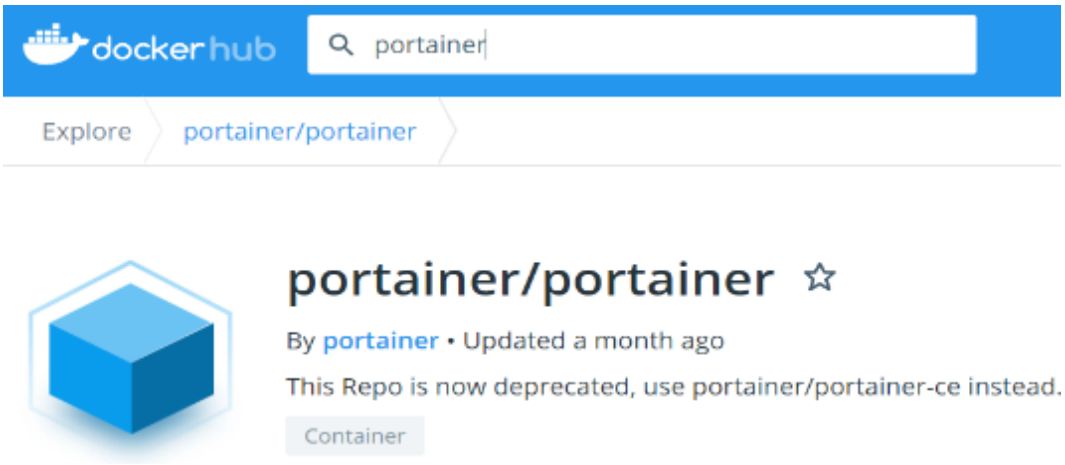


Figura 95. Docker Hub, Pull Portainer, (Print Screen).
Fuente: Propia del autor.

3.3.5.2. Arranque de Portainer.

1. Inicio de sesión y crear usuario admninistrador.

Ahora debería poder navegar hasta la dirección IP del host en el puerto 9000 para acceder a Portainer. Cuando se cargue la pantalla de bienvenida, cree un nombre de usuario y una contraseña segura. https://[RASPBERRY_PI_IP_ADDR]: 9000

Nombre de Usuario: admin
Password: open******

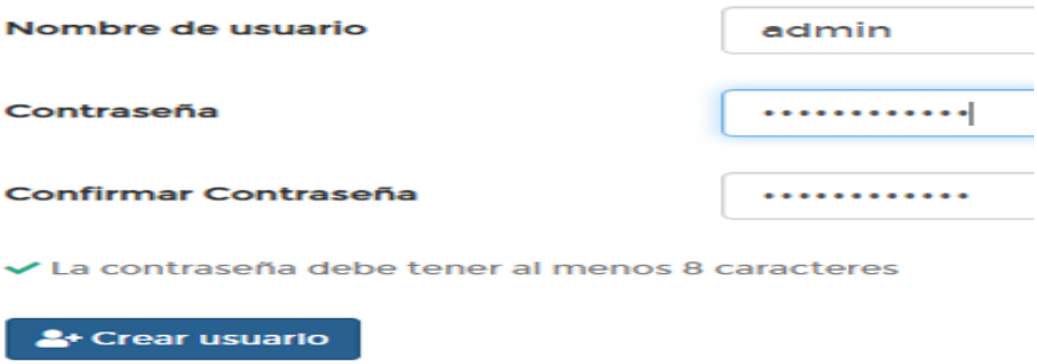


Figura 96. Portainer – Crear Usuario Administrador, (Print Screen).
Fuente: Propia del autor.

2. Selección de Entorno Administrador de Docker

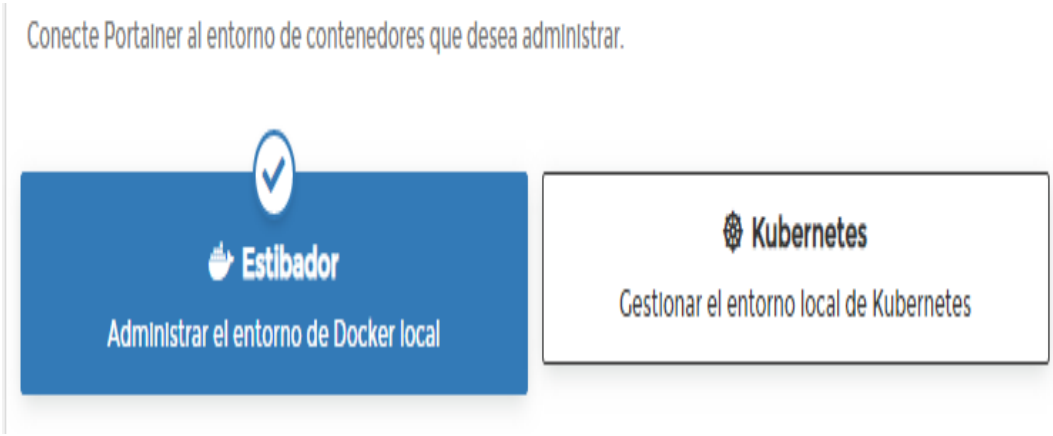


Figura 97. Portainer – Entorno para Administración del Anfitrión, (Print Screen).
Fuente: Propia del autor.

3.3.5.3. Configuración del Administrador Portainer.

1. Configuraciones para el Administrador

Iniciada la sesión, será recibido por el tablero de Control de Portainer.

Figura 98. Portainer – Configuraciones del Administrador, (Print Screen).
Fuente: Propia del autor.

1.1 Crear usuarios.

- ◙ Click en Usuarios.
  - ✓ Escriba el Nombre de usuario y confirmar contraseña.
  - ✓ Conceder permisos de acceso al usuario.
- ◙ Crear equipos de trabajo, designar roles de operación de usuarios.
- ◙ Actualizar >> Guardar Cambios.

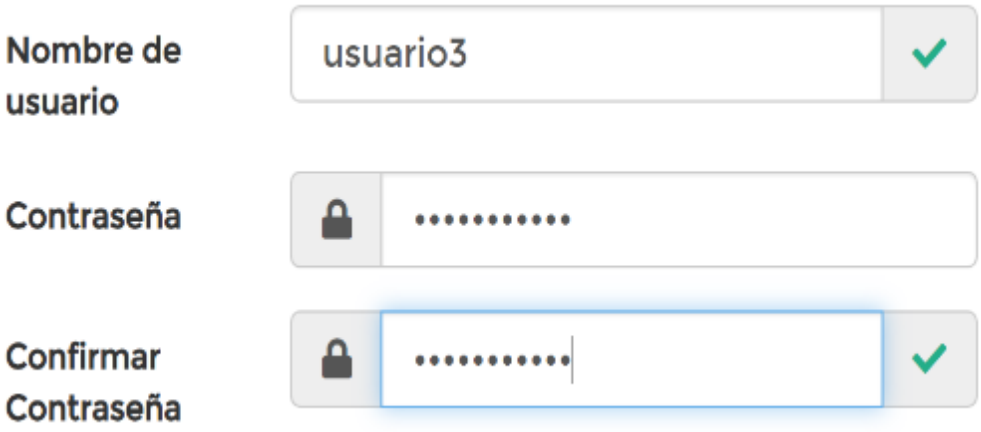


Figura 99. Portainer – Crear Usuarios de Acceso, (Print Screen).
Fuente: Propia del autor.

1.2 Puntos finales, (administración de múltiples clústeres de Docker)

- ◙ Click en Puntos finales.
  - ✓ Identificar los nodos o puntos finales de estibador de sistemas.
- ◙ Crear grupo de estibadores con permisos o denegación de servicios.
- ◙ Asignar equipos de trabajo a grupo de estibadores.
- ◙ Gestionar etiquetas de identificación.

Figura 100. Portainer – Armar Grupos de Trabajo, (Print Screen).
Fuente: Propia del autor.

1.3 Autenticación

La mayoría de los puntos finales de la API requieren autenticación, así como cierto nivel de autorización de parte de los administradores para su operación. La API de Portainer utiliza JSON Web Token para administrar la autenticación y, por lo tanto, requiere que proporcione un token en el encabezado de autorización de cada solicitud.

- ◙ Click en Ajustes.
  - ✓ Señalar el método de autenticación interno y el tiempo de permanencia de la sesión.
- ◙ Intervalo de instantáneas, frecuencia de sondeo es opcional en la administración de su contenedor.
- ◙ Actualizar >> Guardar cambios.

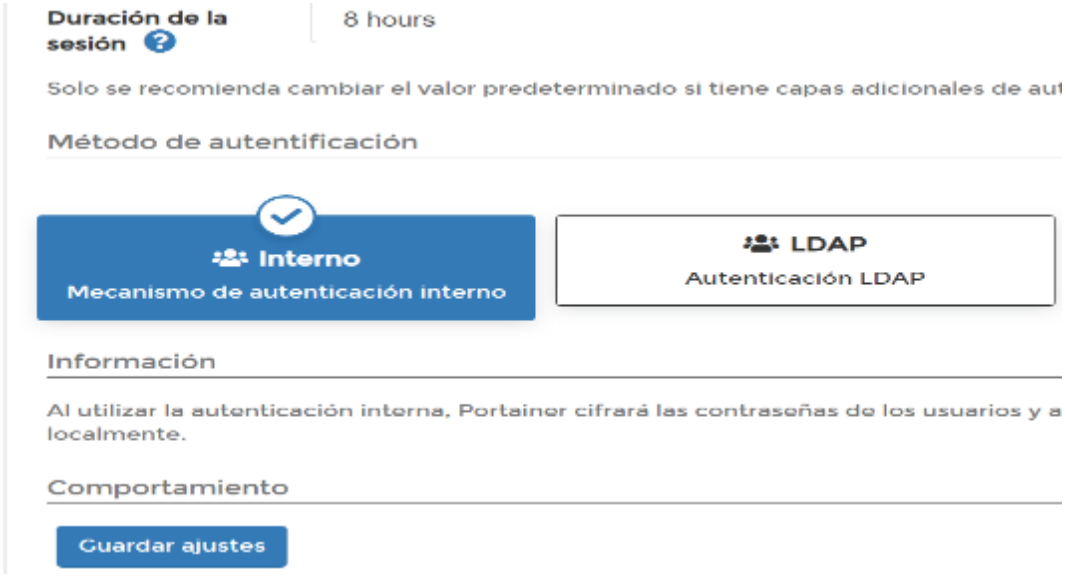


Figura 101. Portainer – Pines de Autenticación, (Print Screen)
Fuente: Propia del autor.

1.4 Información general del anfitrión.

Es una abstracción para identificar el contenedor de docker.

- ◙ Click en Home.
- ◙ Click en anfitrión.>> configuración.
  - ✓ Identifique la información relevante para las configuraciones posteriores de los servicios.

- ✓ Tome nota del Directorio raíz de la carpeta de configuraciones.
- ✓ Identificar los complementos de red para armar la red de su orquestador.

◙ Actualizar >> Guardar Cambios.

Figura 102. Portainer – Información del Anfitrión Docker, (Print Screen).
Fuente: Propia del autor.

1.5 Actualizar el anfitrión (encontrará una guía en el código fuente #7).

Figura 103. Portainer – Actualizar Contenedores, (Print Screen)
Fuente: Propia del autor.

2. Configuración Home – Iniciada la Sesión

2.1 Panel.

Es el escritorio gráfico donde se encuentran las herramientas de administración de los contenedores de forma global, acceso directo a las herramientas.

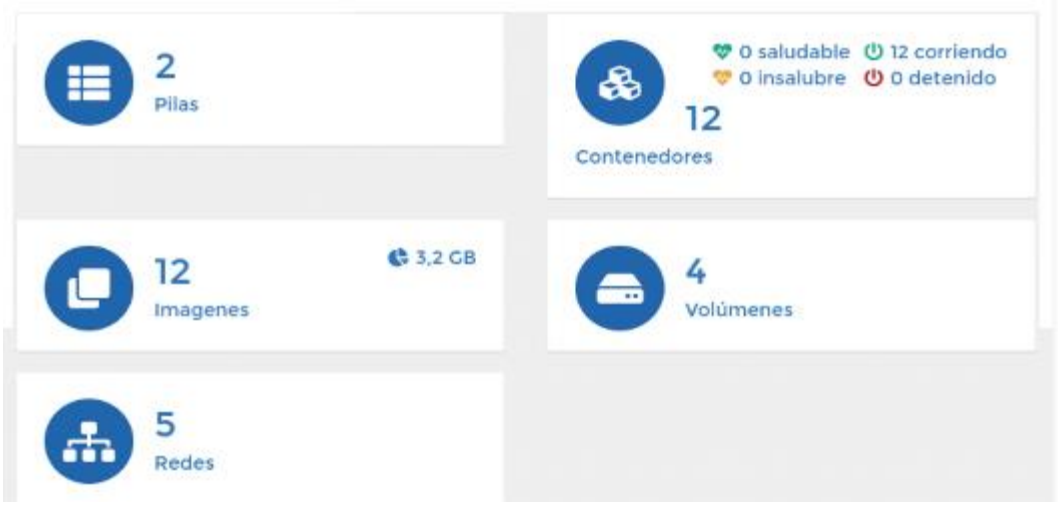


Figura 104. Portainer – Panel de Control, (Print Screen).
Fuente: Propia del autor.

2.2 Plantillas de aplicación.

- ◙ Click en Home >> Plantilla de aplicación.
  - ✓ Identificaremos una serie de contenedores precargados en el anfitrión a disposición del admninistrador.
- ◙ Crear contenedores con permisos o denegación de servicios.
- ◙ Actualizar >> Guardar Cambios.

Figura 105. Portainer – Plantillas de Contenedores Disponibles, (Print Screen)
Fuente: Propia del autor.

2.3 Stack

La forma más rápida de crear, probar y compartir de forma segura aplicaciones modernas listas para su entorno de producción.

- Click en Home >> Stack (Pila de Contenedores)
  - ✓ Utilice el editor web para definir los servicios de la pila contenedor utilizando un formato de composición de ventana acoplable.
  - ✓ Si tiene un archivo stack.yml, puede cargarlo desde su admninistrador y usarlo para implementar la pila.
- Implementar contenedor desde una fuente OpenSource.
  - ✓ Puede utilizar un archivo de formato docker-compose alojado en el repositorio de GitHub.
- Al actualizar Portainer se guardarán los cambios.

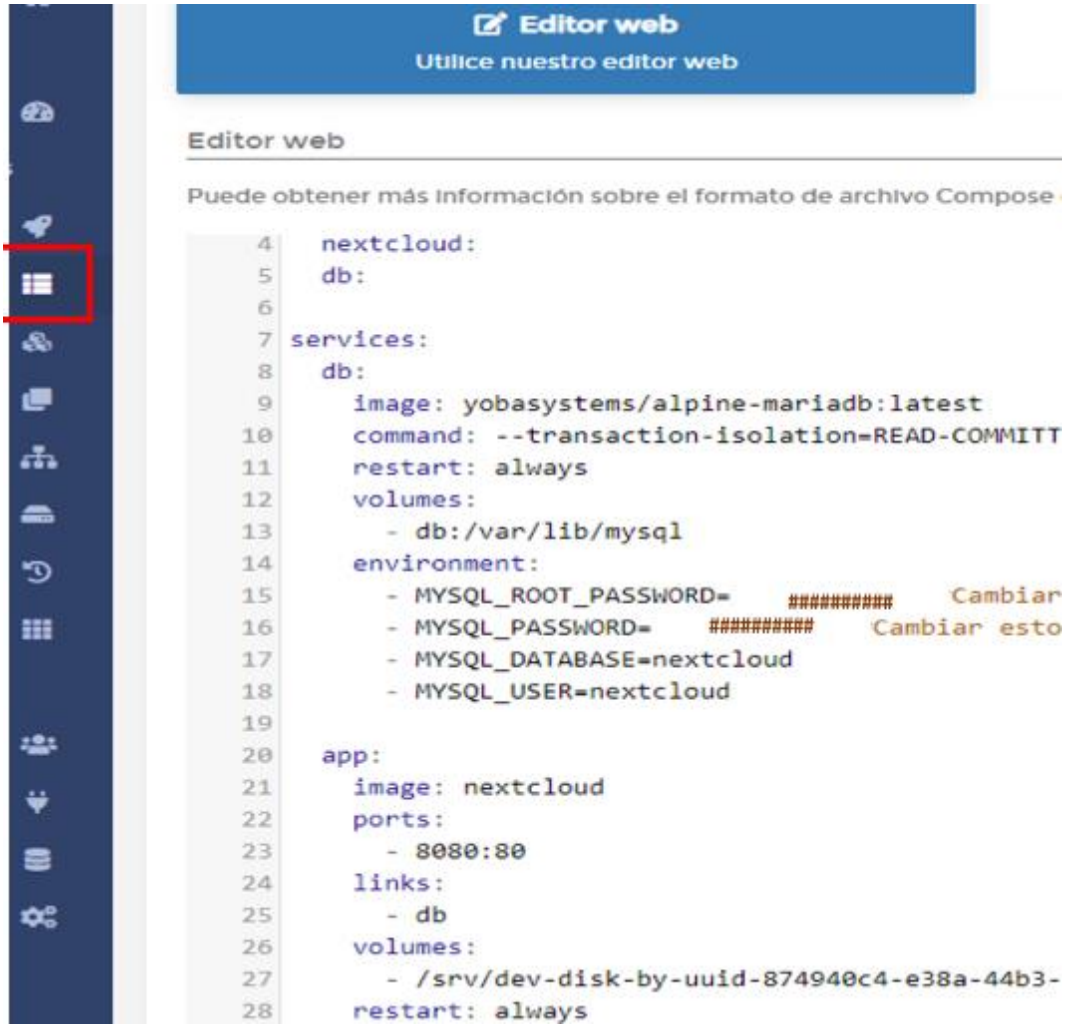


Figura 106. Portainer – Administración de una Pila de Docker, (Print Screen)

Fuente: Propia del autor.

3.3.5.4. Administración del Anfitrión de Docker Host.

1. Contenedores.

Estamos en el escritorio principal de administración de los contenedores, aquí podemos crear, enlistar, iniciar, detener, pausar, mover o eliminar un contenedor mediante la API o la CLI de Docker. Puede conectar un contenedor a una o más redes.

◙ Click en Home >> Contenedores

- ✓ Identifique información relevante y controle su sistema.
- ✓ Tome control de operaciones más habituales como parar, pausar, matar o borrar un contenedor.
- ✓ Ver informaciones del contenedor (docker inspect).
- ✓ Crear una imagen nueva desde el mismo contenedor y añadirla a un registro (docker commit).
- ✓ Ver los logs del contenedor (docker logs).

- ✓ Ver las estadísticas del contenedor (docker stats).
- ✓ Entrar en el contenedor pudiendo elegir el shell o el usuario (docker exec).
- ✓ Conectar/Desconectar el contenedor con una red (docker network connect)

◙ Al actualizar Portainer se guardarán los cambios

Tiene una guía de Administración de contenedores en Código fuente #7.

Figura 107. Portainer – Gestión de Contenedores, (Print Screen)
Fuente: Propia del autor.

2. Imágenes

En un entorno de varios nodos, la imagen extraída solo estará disponible en el nodo que seleccione en la sección Implementación. Para que la imagen esté disponible para todos los nodos, considere agregar un registro a Portainer en el anfitrión.

Para extraer una imagen desde cualquier registro de imágenes.

◙ Click en Home.
◙ Click en Imágenes >> Identificar las imágenes del orquestador.
- ✓ Podrá eliminar, bloquear, brindar privilegios, guardar todo desde este escritorio gráfico, facilitando así la operación de imágenes de contenedores.
- ✓ Seleccione el registro y luego ingrese el nombre de la imagen.

◙ Click modo avanzado.

- ✓ Con el modo avanzado, puede definir una URL, un puerto y una imagen de registro personalizados. Agregue el registro, el puerto y la imagen.
- ✓ En un entorno de varios nodos, seleccione el nodo en el que se implementará .

◙ Actualizar >> Guardar cambios.

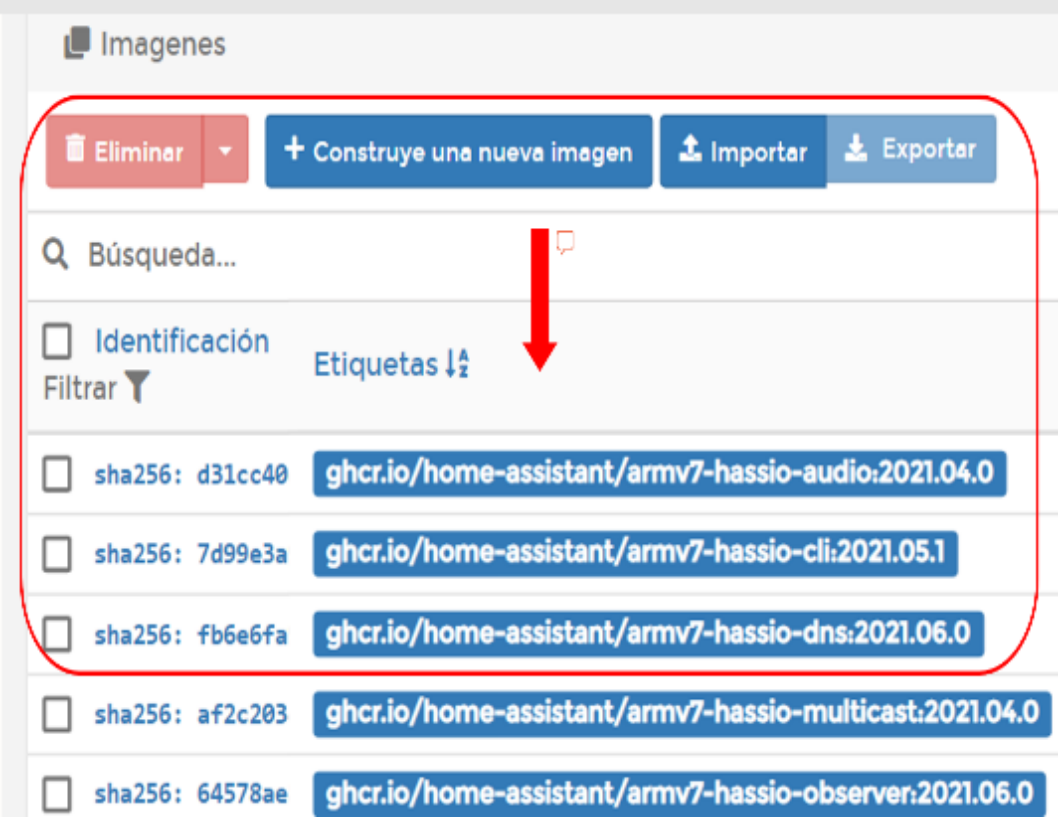


Figura 108. Portainer – Gestión de Imágenes Docker, (Print Screen)
Fuente: Propia del autor.

3. Volúmenes

Un volumen es una abstracción para guardar información de forma persistente **(persistent storage**). La información almacenada en un volumen perdura más allá del ciclo de vida de un contenedor.

En nuestro Proyecto será la manera sencilla y predefinida para almacenar todos los ficheros (salvo unas pocas excepciones irán almacenarse en el disco SSD) de un contenedor, usará el espacio de nuestro equipo real y en "/var/lib/docker/volumes" creará una carpeta para cada contenedor que se cargue a nuestro docker.

◙ Click en Home.
◙ Click en Volúmenes.
- ✓ Identifique los volúmenes activos del orquestador.

- ✓ Inicie las tareas de control de sus volumenes. Salvo que tenga una limitación de espacio usando el modificador “tmpfs-size=999bytes”, el espacio que pueden ocupar los ficheros es ilimitado (o limitado por la capacidad disponible de RAM), Este es el motivo por el cual las prestaciones de servicios de la Red Virtual se provisionan en el disco SSD.
- ✓ Este tipo de almacenamiento puede ser usado para almacenar ficheros de sesiones web, temporales o contenido que nos interese que se borre en cada arranque del contenedor.

◙ Actualizar >> Guardar Cambios.

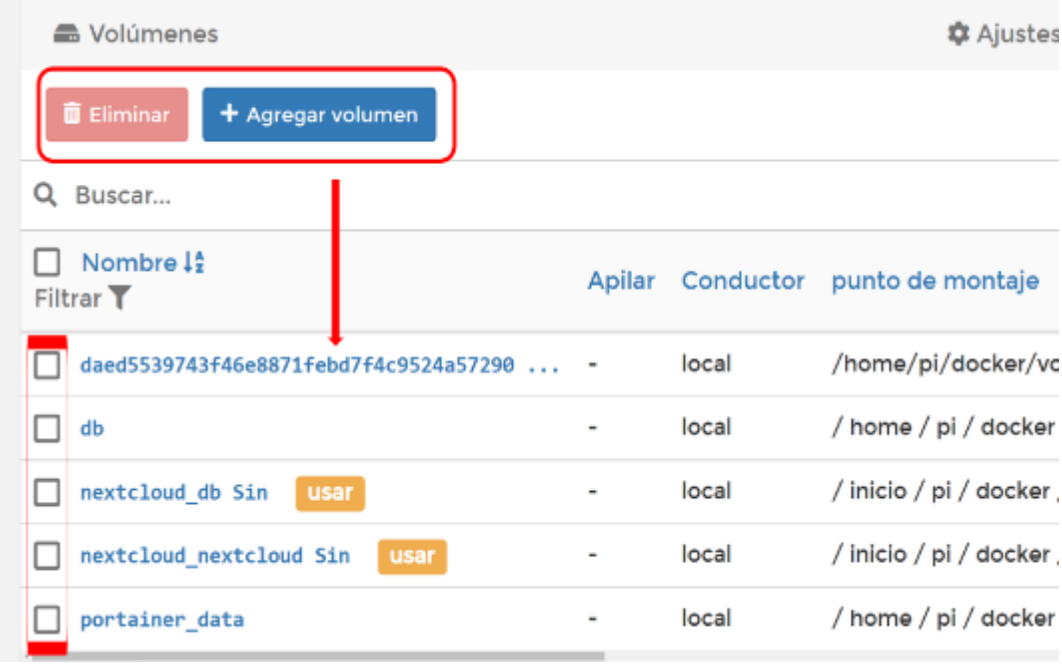


Figura 109. Portainer – Gestión de Volúmenes Docker, (Print Screen)

Fuente: Propia del autor.

3.3.5.5. Administración de Redes en Docker.

Debemos ajustar solo 3 opciones en la configuración de la red del contenedor de Docker, estos parámetros de configuración se especifican cuando arrancamos al tocar el Daemon (demonio de Docker) y son los siguientes:

- Que no haya reglas he iptables ya definidas en la máquina, que puedan bloquear el tráfico hacia y desde nuestros contenedores sean estas forward o input, podemos verificar con el comando **iptables -L**

- Cuando los contenedores quieren comunicarse entre sí, docker necesita definir reglas iptables o delegarlo entre otros sistemas, para permitir a docker crear las reglas iptables entre contenedores que se comunican entre sí y es necesario habilitar el parámetro **iptables=true** de este modo docker engine añadirá las reglas a la filter change de docker, por cierto,

ten en cuenta que docker no modifica ninguna regla. Para esto podemos ejecutar el siguiente comando **iptable -L docker**

- Puede afectar la comunicación entre contenedores (ip_forward), habilita el tráfico entre contenedores y el mundo exterior cuando está habilitado sí ejecutamos un **ip_forward = true** puedes usar el siguiente comando para comprobarlo, si no habilitas el ip_forward=true tus contenedores no podrán comunicarse entre sí, esto por un lado protegería los contenedores y el host de cualquier vulnerabilidad de red pero por otro lado sin ip_fprward=true los contenedores podrían comunicarse entre ellos de manera arbitraria.

Más detalles de Iptables del proyecto en el subcapítulo 3.3.1.3.

Figura 110. Portainer – Administración de Redes con Docker, (Print Screen).
Fuente: Propia del autor.

3.3.6. Crear Servicios de Red Virtual Utilizando ZeroTier

La plataforma ZeroTier proporciona el punto de conmutación central de control para su red definida por software. En ella, podrá autorizar y desautorizar clientes, elegir un esquema de red configurable con reglas de tráfico. Lo primero que vamos a hacer es crear un ID de Red ZeroTier administrador (16 dígitos), al cual puede dirigir a sus clientes cuando los configure, para lograr esto se debe registrar con una cuenta en la página principal de zerotier.com, será el mismo portal para acceder a la administración de la red de ZeroTier en adelante.

Para respaldar el uso de ZeroTier como protocolo SDN/NFV de alto rendimiento en redes físicamente seguras, el protocolo admite una función denominada rutas de confianza. (Iptables y configuración de red Privada).

Importante: Es posible configurar todos los dispositivos ZeroTier de una red determinada (Pública) para que omitan el cifrado y la autenticación del tráfico en una ruta física designada. Esto puede reducir notablemente el uso de la CPU en escenarios de alto tráfico, pero a costa de perder prácticamente toda la seguridad en el transporte de información.

Bienvenidos a la configuración de la red de ZeroTier.

3.3.6.1. Autenticación del Administrador de Red ZeroTier.

1. Portal principal de Acceso Web a ZeroTier.Inc en my.zerotier.com.

Figura 111. ZeroTier – Portal de Acceso WEB, (Print Screen).
Fuente: Propia del autor.

2. Registro de Cuenta Administrador de Red Zerotier.

◙ Click en Inscribirse >>Registrarse
- ✓ Registrar los nombres del administrador de red.
- ✓ Registrar una cuenta mail activa para verificación.
- ✓ Registre una contraseña segura que pueda recordar.
- ✓ Puede iniciar sesión directamente con sus cuentas de Google, GitHub, o Microsoft.

◙ Click en registrarse.

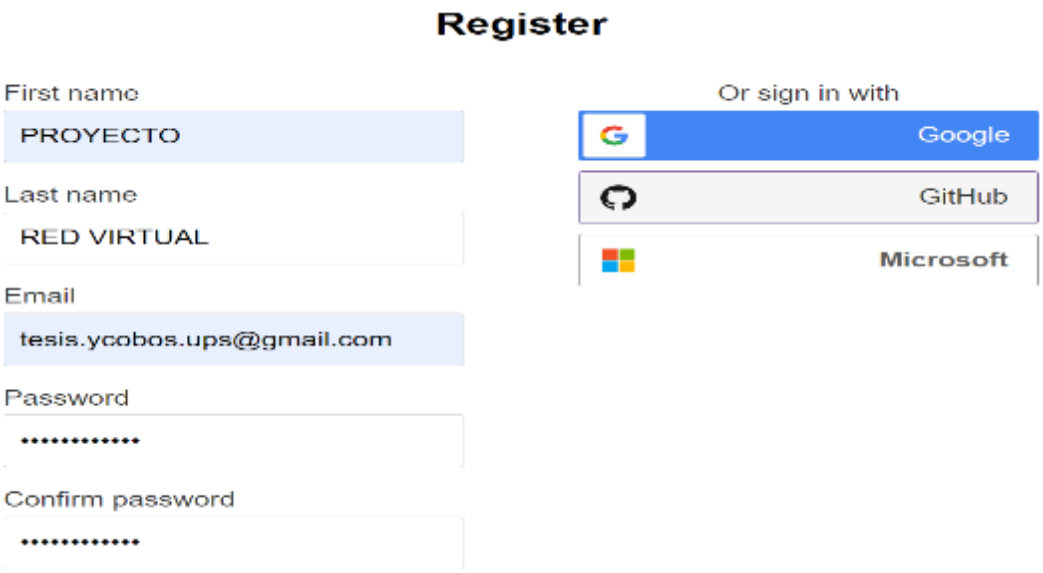


Figura 112. ZeroTier – Registro de Cuenta Administrador, (Print Screen).
Fuente: Propia del autor.

3. Verificación de cuenta Administrador de Red ZeroTier.

   - ✓ Diríjase a la cuenta de correo con que se registró.
   - ✓ Acceda al correo - verifique para activar la cuenta de administrador ZeroTier.

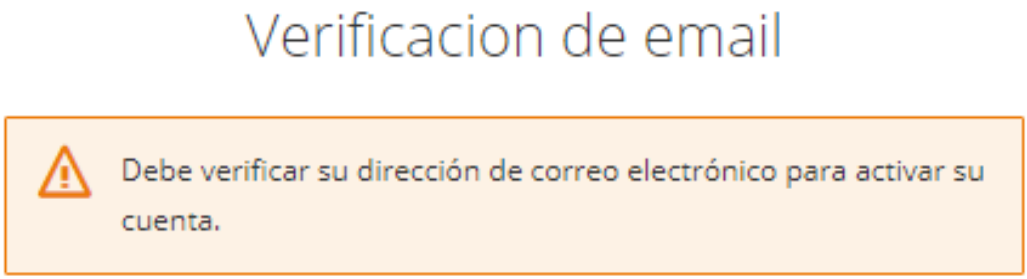


Figura 113. ZeroTier – Verificación de la Cuenta Administrador, (Print Screen).
Fuente: Propia del autor.

4. Portal de Bienvenida al Administrador de Red ZeroTier

Ya está registrada su nueva cuenta luego de verificar el correo de confirmación, cabe indicar que la confirmación solo estará activa por un tiempo máximo de 10 minutos para confirmar. Ahora puede gestionar el controlador de la red.

Nota. Al no tener actividad en el controlador de la Red de ZeroTier por más de 60 días, la compañía dará de baja a la red.

Figura 114. ZeroTier – Administrador de Red Activado, (Print Screen).
Fuente: Propia del autor.

3.3.6.2. Configuración de la Red ZeroTier.

1. Interfaz de Inicio de Sesión.

Figura 115. ZeroTier – Interfaz de Inicio de Sesión, (Print Screen).
Fuente: Propia del autor.

2. Credenciales de inicio de Sesión del Administrador.

◙ Click en iniciar sesión.
  ✓ Coloque el correo electrónico y la contraseña que registró.
◙ Click en Iniciar sesión.

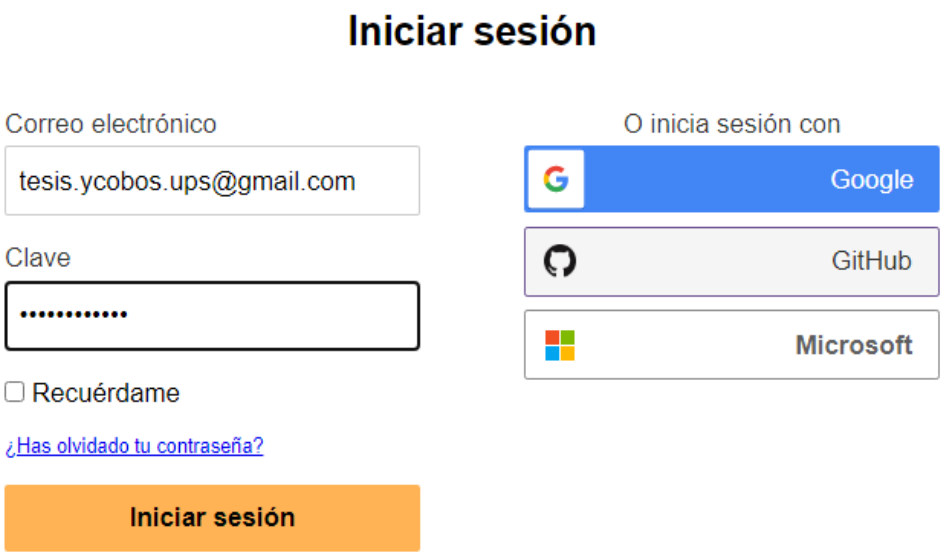


Figura 116. ZeroTier – Credenciales de Inicio de Sesión, (Print Screen).
Fuente: Propia del autor.

3. Crear Red de ZeroTier. (Coloque el nombre y una breve descripción).

Figura 117. ZeroTier – Descripción General de la Red, (Print Screen).
Fuente: Propia del autor.

4. Identificación de Red ZeroTier.

El código hexadecimal de 16 dígitos es la Identificación de la RED, (c7c8172af1c32100). Esta ID de red es la que utilizará para unir sus dispositivos a la red virtual desde un cliente de red. Si está siguiendo la guía de este documento, reemplace el ID c7c8172af1c32100.

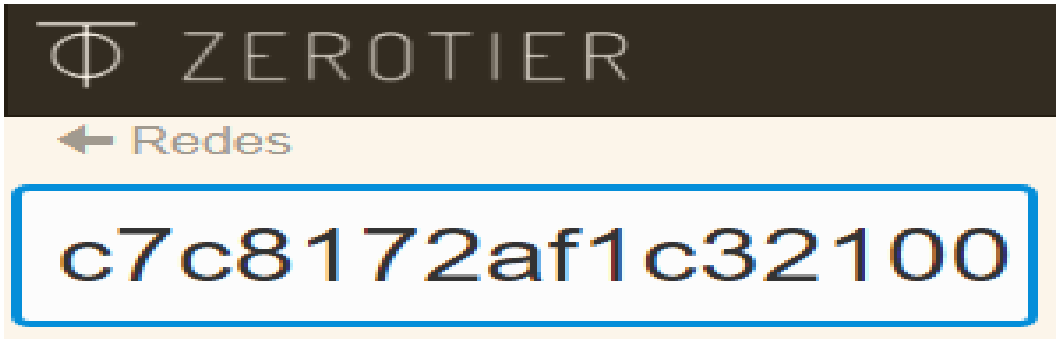


Figura 118. ZeroTier – NETWORK_ID de Red Virtual , (Print Screen).
Fuente: Propia del autor.

5. Configuración Básica de la Red ZeroTier.

Configuración de la red ZeroTier Central, a la que nombramos "RED VIRTUAL ZT-SDWAN", realice los siguientes pasos para blindar su red.

- ✓ Identificación de la red.
- ✓ Seleccionar control de acceso privado para la red, esto garantiza que solo los dispositivos aprobados puedan conectarse a la red, y que no lo haga cualquiera que conozca su ID.

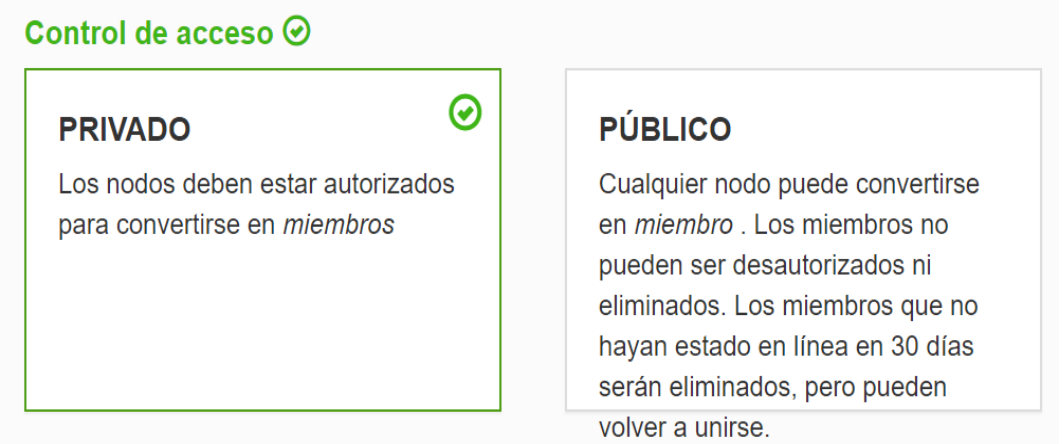


Figura 119. ZeroTier – Control de Acceso Privado a la Red, (Print Screen).
Fuente: Propia del autor.

6. Guía Configuración Avanzada de la Red ZeroTier.

◙ Rutas Managed >> Asignación automática de IPv4.

- ✓ Tomar en cuenta que la asignación de Ipv4 no debe generar conflicto de enrutamiento en el Host de la Raspberry Pi.
- ✓ Designaremos una dirección IPv4 (10.242.x.x).

◙ Ruta Asignada VXLAN – IPV4 10.242.0.0/16; IPv6 OFF.

Figura 120. ZeroTier – Gestión de Ruta (Direccionamiento IP), (Print Screen).
Fuente: Propia del autor.

7. Unir equipos a la red

La unión ocurre desde un dispositivo, después de instalar ZeroTier One en los equipos que se unirán a la red. Puede unirse a sus propias redes y a las redes de otras personas. Podría utilizar la línea de comando de la consola, pero como ya hemos visto la interfaz de ZeroTier Cliente nos brinda un acceso en la cual se interactúa de forma sencilla con la interfaz.

Formas en las que podríamos unirnos a una red:

- **Unirse por correo electrónico**. Consiste en una invitación mediante un enlace de correo electrónico de un cliente final el cual obtendrá el ID de red de invitado, se unirá a la red al confirmar el acceso.
- **Agregar miembro manualmente:** Se podrá agregar un cliente de red con el ID de ZeroTier One de la instalación.
- **Línea de código**: La línea de código será la utilizada cuando quiera agregar un equipo Linux o derivaciones, esto se ampliará más adelante en el subcapítulo 3.4.2 al cargar equipos linux a la red virtual.

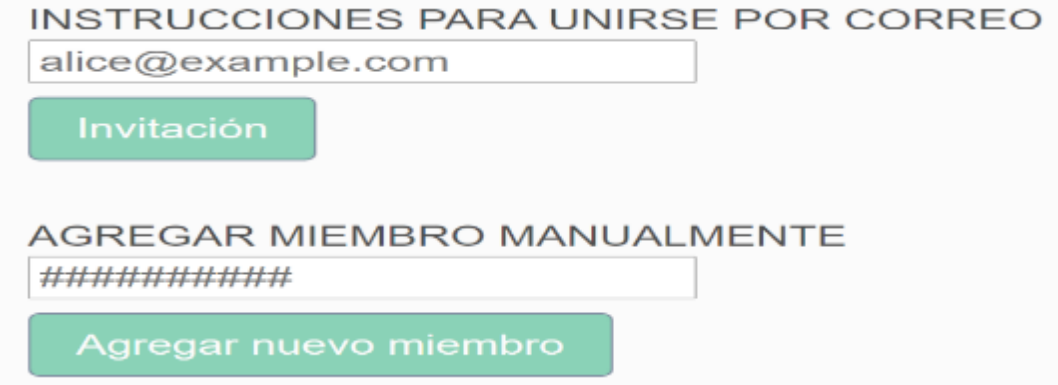


Figura 121. ZeroTier – Agregar Equipos a la Red, (Print Screen).
Fuente: Propia del autor.

8. Reglas de flujo

El lenguaje de definición de reglas de ZeroTier está diseñado para ser simple de leer tanto para humanos como para máquinas. Tenga en cuenta que las reglas de flujo irán cambiando a medida que se vaya configurando la red, El software creará una copia de las reglas. (LaDuke, 2021)

<tipo>	Código de tipo de Ethernet (16 bits, use 0x ###
scara> <inicio [-end]>	Coincidir con el rango de valores de campo IP
l <protocolo>	Valor del campo de protocolo IP (por ejemplo,
aleatorio	Coincidencias con probabilidad dada, rango de
MAC>	Dirección MAC de Ethernet de origen (se pued
<MAC>	Dirección MAC de Ethernet de destino (se pue
bits>	Dirección IP de origen y bits de máscara de re
/ bits>	Dirección IP de destino y bits de máscara de re
o> <código>	Tipo y código ICMP (use el código -1 para los t cualquier código)
inicio [-end]>	Intervalo de puertos IP de origen (TCP, UDP, S
cio [-end]>	Intervalo de puertos IP de destino (TCP, UDP,

Figura 122. ZeroTier – Reglas de Flujo de la Red, (Print Screen).
Fuente: Propia del autor.

3.3.6.3. Conmutador ZeroTier Script de Instalación

1. Script de instalación de ZeroTier en Docker para Raspberry Pi, (ugeek.github.io, 2020).

Código Fuente 10. Instalación del Puente de Red ZeroTier.

```
#Antes de iniciar la instalación debemos actualizar el Sistema operativo de la Rasp.Pi.

        $ sudo apt update && sudo apt -y full-upgrade


# Instalación desde Repositorios Zerotier One

        $ sudo apt install zerotier-one


#Instalación de la última Versión

        $ curl -s https://install.zerotier.com/ | sudo bash


# Instalación para arquitectura ARM (Rasp.Pi), desde repositorio de Github.

        $ docker run --name zerotier-one --device=/dev/net/tun --net=host --cap-add=NET_ADMIN --cap-add=SYS_ADMIN -v $HOME/docker/zerotier one:/var/lib/zerotier-one ugeek/zerotier:arm


#Conectarnos el contenedor a la red ZeroTier.

        $ sudo zerotier-cli join c7c8172af1c32100


#Si queremos desconectarnos de la red, escribiremos los siguiente.

        $ sudo zerotier-cli leave c7c8172af1c32100


#Si Queremos ver el contenedor podemos listar los activos en el del Host con:

        $ sudo docker -ps

#Si queremos listar las redes del Host

        $ sudo zerotier-cli listnetworks
```

Fuente 1: https://ugeek.github.io/blog/post/2020-04-09-zerotier-funcionamiento-e-instalacion-por-docker-o-repositorios.html
Fuente 2: https://github.com/docker-projects/docker-zerotier

2. Opcional - Rutas Estáticas en la Red ZeroTier.

Código Fuente 11. Opcional - Rutas Estáticas en la Red ZeroTier.

En la terminal establecemos las nuevas variables de Shell

NETWORK_ID=<c7c8172af1c32100>

BR_IF="br0"

BR_ADDR=<10.242.10.40>

GW_ADDR=<192.168.0.1>

Si no queremos que ZeroTier administre direcciones o rutas en $ ZT_IF. Lo estamos haciendo estáticamente a continuación, en la interfaz del puente.

$ sudo zerotier-cli set $NETWORK_ID allowManaged=0

Copie el nombre `dev` de la salida de` listnetworks` para $ ZT_IF. Será algo como: ztmjfejtrx

ZT_IF=<ztmjfejtrx>

Verifique el dispositivo en my.zerotier.com

- Vaya a la sección Miembros de la Red
- Abra el ícono de llave inglesa para configuraciones avanzadas y verifique
    - Marque; Permitir puente
    - Marque; No asignar automáticamente
- Autorizar al miembro

Eliminar material de red existente:

$sudo apt remove --purge --auto-remove dhcpcd5 fake-hwclock ifupdown isc-dhcp-client isc-dhcp-common openresolv

Reiniciamos el anfitrión para guardar los cambios.

$ sudo reboot

Fuente: Aplicando Rutas Estáticas al Contenedor de Red ZeroTier.

3. Cambiar redes systemd. (atlassian.net, 2021); (systemd.network, act.2021)

Código Fuente 12. Redes Systemd del Contenedor de Red ZeroTier.

```
# Habilitar systemd-networkd

        $ sudo ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
        $ sudo systemctl enable systemd-networkd
        $ sudo systemctl enable systemd-resolved
        $ sudo systemctl enable systemd-timesyncd



# Configurar interfaces

         $ sudo zerotier-cli set $NETWORK_ID allowManaged=0

        Escriba archivos de configuración de red. Pone ethernet y ZeroTier en el puente, configura el puente con una IP estática. Consulte a continuación la configuración de DHCP en el puente.

        cat << EOF | sudo tee /etc/systemd/network/25-bridge-br0.network
        [Match]
        Name=$BR_IF


        [Network]
        Address=$BR_ADDR
        Gateway=$GW_ADDR (192.168.0.1)
        DNS=          % resolvconf guardar en (/etc/resolv.conf)
        EOF


        cat << EOF | sudo tee /etc/systemd/network/br0.netdev
        [NetDev]
        Name=$BR_IF
        Kind=bridge
        EOF

        cat << EOF | sudo tee /etc/systemd/network/25-bridge-br0-zt.network
        [Match]
        Name=$ ztmjfejtrx

        [Network]
        Bridge=$BR_IF
        EOF

        cat << EOF | sudo tee /etc/systemd/network/25-bridge-br0-en.network
        [Match]
        Name=eth0 # might be en*

        [Network]
        Bridge=$BR_IF
        EOF

# Debería poder, desde la LAN física, conectarse al Pi a través de $ BR_ADDR
```

Fuente: https://zerotier.atlassian.net/

4. Utilidad de la CLI de ZeroTier

En sistemas Unix, deberá conceder privilegios de administrador con sudo su, mientras que en Windows deberá usar el símbolo del sistema en modo administrador.

Código Fuente 13. Uso de la CLI en el Administrador de la Red ZeroTier.

1. Obtener dirección ZeroTier y verificar el estado de servicio

   $ zerotier-cli status

2. Unirse a una red.

   $ zerotier-cli join c7c8172af1c32100

3. Abandonar una red

   $ zerotier-cli leave c7c8172af1c32100

4. enumerar redes

   $ zerotier-cli listnetworks c7c8172af1c32100

5. Si un miembro no tiene una IP física en la lista, es posible que esté retransmitiendo
¿Es "tcpFallbackActive" verdadero? La reserva de TCP es la forma más lenta de retransmisión.

   $ zerotier-cli info -j

6. Buscar el ID del nodo <ztaddr> del dispositivo con el que intenta comunicarse y vea si tiene una dirección IP en <ruta>. Si no hay una dirección IP, se está transmitiendo

   $ zerotier-cli listpeers

   .

7. Hay una segunda forma de retransmisión en la que los paquetes rebotan en la infraestructura de ZeroTier. Esto es mejor que la retransmisión TCP, pero aún puede estar sujeto a pérdida de paquetes y alta latencia.

   $ zerotier-cli peers

8. Revisar la configuración de redes systemd.

   $ tail -n+0 /etc/systemd/network/*

Fuente: https://zerotier.atlassian.net/wiki/spaces/SD/pages/29065282/Command+Line+I

3.3.6.4. Problemas de red ZeroTier.

3.3.6.4.1. *Fallo - Excesivo tiempo esperando la red durante el arranque*:

La interfaz física resulta ser un largo "nombre de interfaz predecible" como: "enb827eb0d4176", a veces es simplemente "eth0", según la versión de Rasbian.

Posible solución: Conecte un teclado y un monitor al R.pi, verifique ip addr **(ifconfig)** luego edite "/ etc / systemd / network / 25-bridge-br0-en network" para que coincida.

3.3.6.4.2. ***Fallo - Comunicación de red avanzado (10.242.0.0):*** En ocasiones los paquetes de configuración no fueron cargados correctamente, originando una falla en la comunicación de la central zerotier y el administrador,

Posible solución: Desde el navegador diríjase al portal administrador en my.zerotier.com/network/$NETWORK_ID >> Configuración>Avanzado.

- ✓ Elimine la ruta administrada predeterminada. Agregar la nueva ruta administrada $ ZT_ROUTE
- ✓ Cambie la asignación automática de IPV4 >> Avanzado,
- ✓ Eliminar la piscina existente. Crear una nueva piscina con inicio y finalización desde $ ZT_POOL
- ✓ Para fines de documentación, asigne $ BR_ADDR al miembro del puente ZeroTier.

3.3.6.4.3. ***Fallo - Puedo hacer ping al puente, pero no detrás de él:*** No tengo comunicación entre equipos, realizo tracerouter en la red y resulta en error.

Posible solución: A veces, se aplican las reglas de iptables.

- ✓ echo "0" > /proc/sys/net/bridge/bridge-nf-call-iptables
- ✓ iptables -A FORWARD -p all -i br0 -j ACCEPT

3.3.6.4.4. ***Fallo - Acceso denegado:*** Su nodo debe estar autorizado en esta red (a través de my.zerotier.com).

Posible solución: Habla con tu amigable administrador de red zerotier.

¡Informe los problemas a contact@zerotier.com e intentaremos solucionarlo lo antes posible!

3.3.6.4.5. ***Fallo - La conexión no es privada.*** Falta la certificación SSL, tiene un certificado que ya no es válido o expiró, esto se da cuando tiene un certificado que el navegador no lo tiene reconocido.

Posible solución: Actualice los paquetes del fichero directorio, dentro de los paquetes deben actualizarse los certificados, ciertos certificados tienen vigencia.

3.3.6.4.6. ***Fallo – Reinicio el Host y pierdo la comunicación de la Red ZeroTier.*** El contenedor de la red de ZeroTier no está configurado para realizar el levantamiento del estado en inicio de red, provocando que se pierda la comunicación cuando se reinicie o prenda el conmutador.

Posible solución: En el administrador de docker deberá actualizar la configuración de inicio de sesión del contenedor de Red ZeroTier, reconocimiento del último estado migrado de inicio y apagado del contenedor, o en la terminal.

- ✓ # Persistent configuration folder (for ZT controller mode)
- ✓ # Option config path '/etc/zerotier
- ✓ # Options port '9993'

3.3.6.4.7. ***Fallo - Falta el token de autenticación y authtoken, secret no encontrado (o legible):*** Si recibe este mensaje de error, es probable que esté intentando ejecutar **zerotier-cli** desde una cuenta no administrativa. En macOS, Linux u otros sistemas basados en Unix, use **sudo zerotier-cli** . En Windows, use un símbolo del sistema de administrador.

3.3.6.4.8. ***Fallo - Algo salió mal con el instalador de ZeroTier*****:** Podría tratarse de un error de puerto.

Generate secret on first start option secret ''

Posible solución: Asegúrese de que el “TUN” controlador esté cargado.

3.3.6.4.9. ***Fallo – Mi dispositivo está retransmitiendo***. Consulte la lista de miembros de la red en ZeroTier Central. Si un miembro no tiene una IP física en la lista, es posible que esté retransmitiendo**.**

Ejemplo de Red ZeroTier

Network ID: c7c8172af1c32100
IPv4 Auto-Assign (advanced) Privada
[x] Auto-Assign from Range: 172.28.28.1-172.28.28.255
Managed Routes:
172.28.28.0/24 (VXLAN)
192.168.1.0/24 (172.28.28.1) LAN

3.4. Conectar Equipos en la Red Virtual

A continuación, se detallará el procedimiento para conectar equipos Windows, Linux y dispositivos móviles. ZeroTier debe estar instalado (Raspberry Pi) y unido a la red a la que desea proporcionar el servicio DNS.

Antes de iniciar debemos garantizar lo siguiente:

- ✓ La correcta comunicación entre los equipos, el sistema de almacenamiento en red (OMV) como NAS y el administrador de contenedores Portainer.

- ✓ Que las rutas de redes LAN y VXLAN estén correctamente configuradas (ver tabla de la sección 3.1.4), nos faltaría conectar equipos, conectar redes y practicar un poco con iptables. Para iniciar la configuración podemos conectarnos a través de un shell en serie.

- ✓ Para cargar equipos móviles a la red se debe tener presente que los equipos deberán tener mínimo la versión de Android 6, ya que la aplicación ZeroTier One está disponible para versiones Android 6 en adelante.

Descargas desde el Portal web de ZeroTier en my.zerotier.com

Figura 123. ZeroTier One – Opciones Disponibles para Clientes, (Print Screen).
Fuente: Propia del autor.

3.4.1. Conectar Clientes Windows

1. Descargar el archivo MSI desde el Portal de ZeroTier en my.zerotier.com.

Figura 124. ZeroTier One en Windows – APK_MSI de Instalación, (Print Screen).
Fuente: Propia del autor.

2. Utilice el asistente de instalación del ZeroTier One.

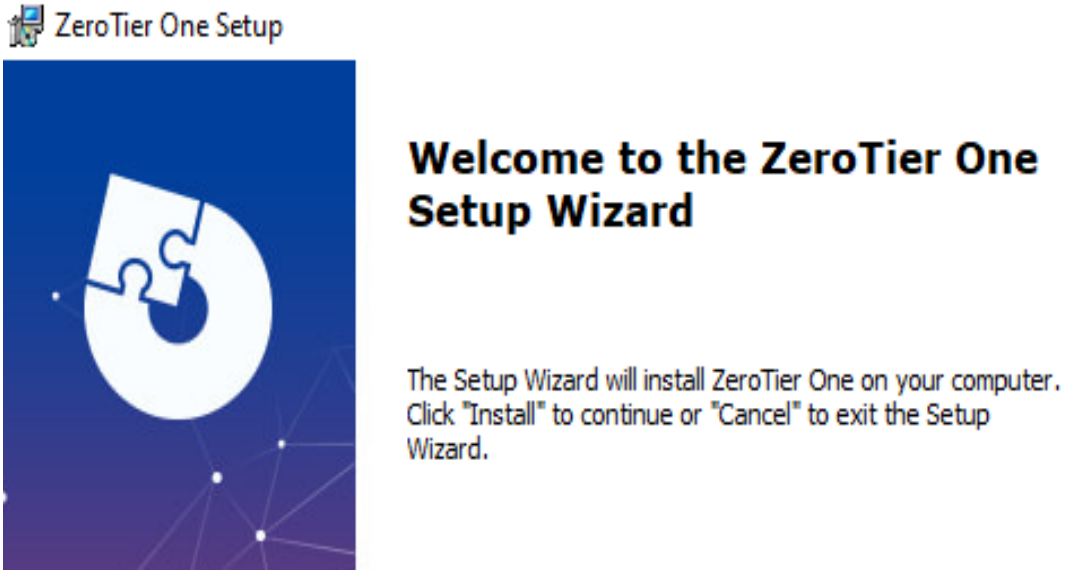


Figura 125. ZeroTier One en Windows – Asistente de Instalación del apk msi.
Fuente: Propia del autor.

3. Ejecutamos la instalación del apk msi, conceder derechos de administrador.

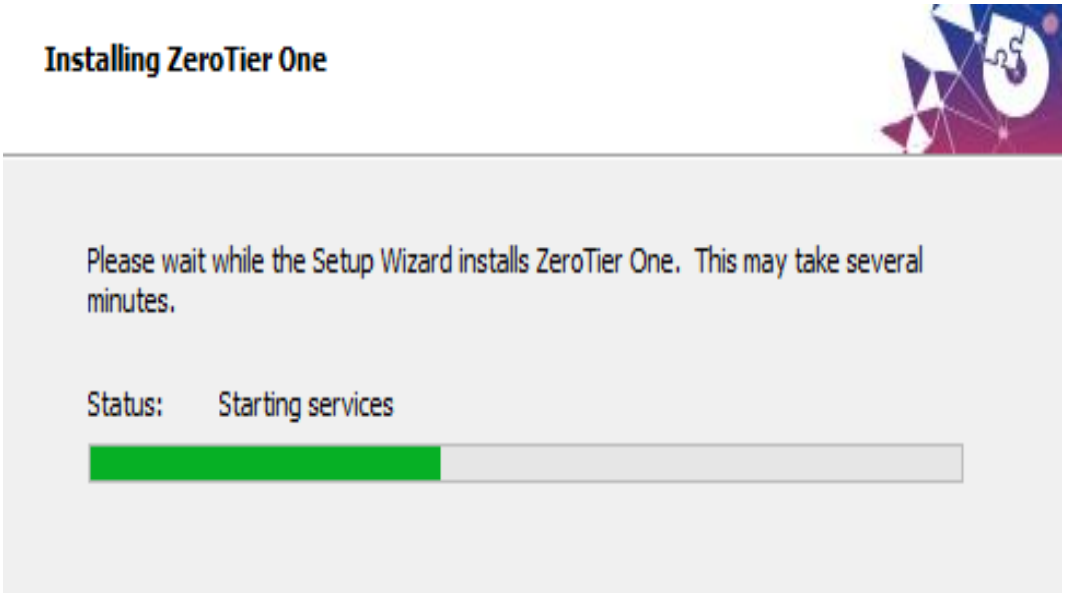


Figura 126. ZeroTier One en Windows – Instalación del apk msi, (Print Screen).
Fuente: Propia del autor.

4. Portal asistente de configuración. Ahora podrá conectarse a una red.

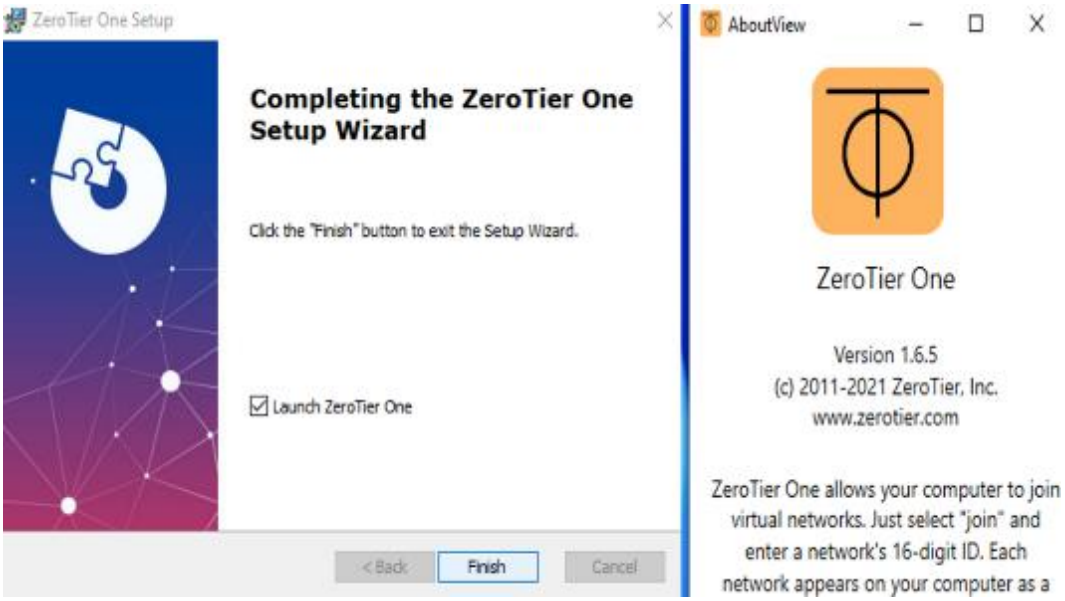


Figura 127. ZeroTier One en Windows – Asistente de Configuración del apk.
Fuente: Propia del autor.

5. Conceder Permisos de Comunicación entre equipos.

Figura 128. ZeroTier One en Windows – Conceder Permisos de Comunicación.
Fuente: Propia del autor.

6. La identificación del Nodo_ID del equipo en la red se detallará con un código de 10 dígitos.

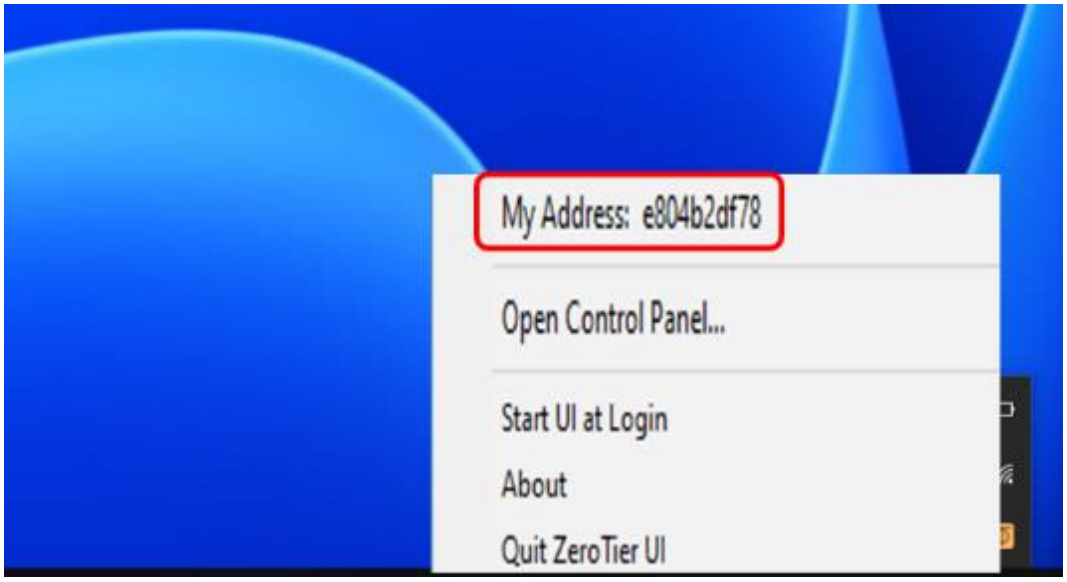


Figura 129. ZeroTier One en Windows – Nodo_ID "Dirección del Equipo".
Fuente: Propia del autor.

7. Para unirnos a una red tenemos dos opciones.
   - ✓ 1. Mediante el Nodo_ID de 10 dígitos en ZeroTier Central.

✓ 2. Mediante el NETWORK_ID de 16 dígitos de la Red.

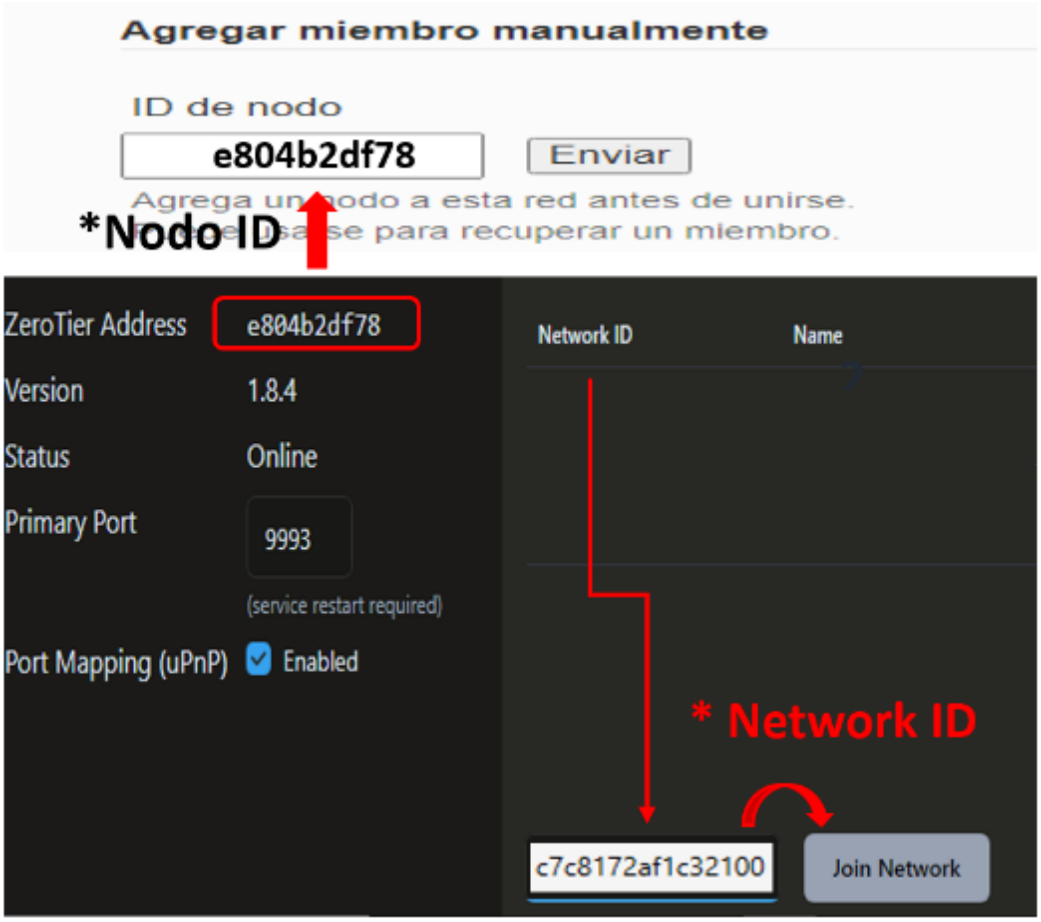


Figura 130. ZeroTier One en Windows – Unirse a la Red de ZeroTier Central.
Fuente: Propia del autor.

8. Luego de conectarnos a la red deberemos autenticar el equipo.

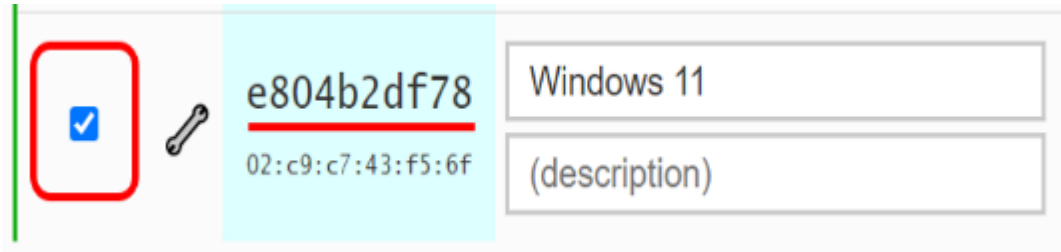


Figura 131. ZeroTier One en Windows – Autenticar Equipo en ZeroTier Central.
Fuente: Propia del autor.

9. Adaptadores de equipos en red habilitados.

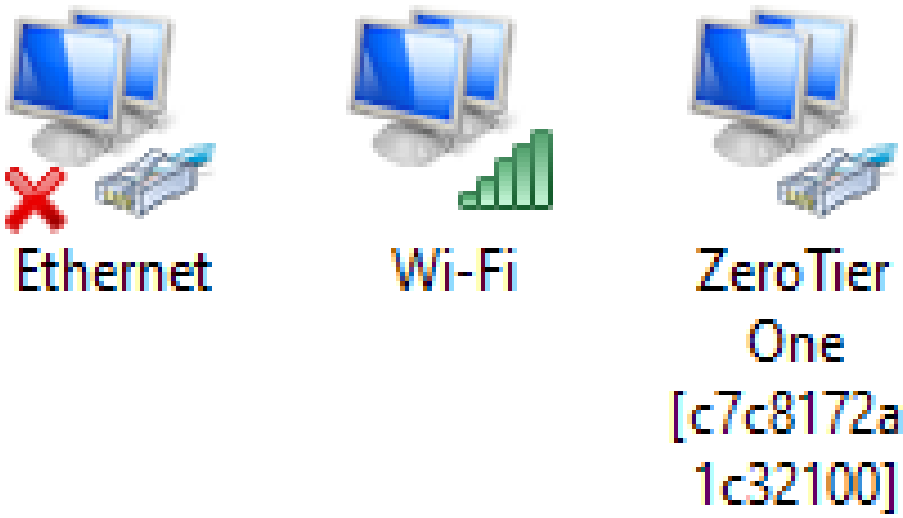


Figura 132. ZeroTier One en Windows – Adaptador de Red ZeroTier Habilitado.
Fuente: Propia del autor.

10. Equipo en red – Nodo_ID de Equipo Windows en comunicación.

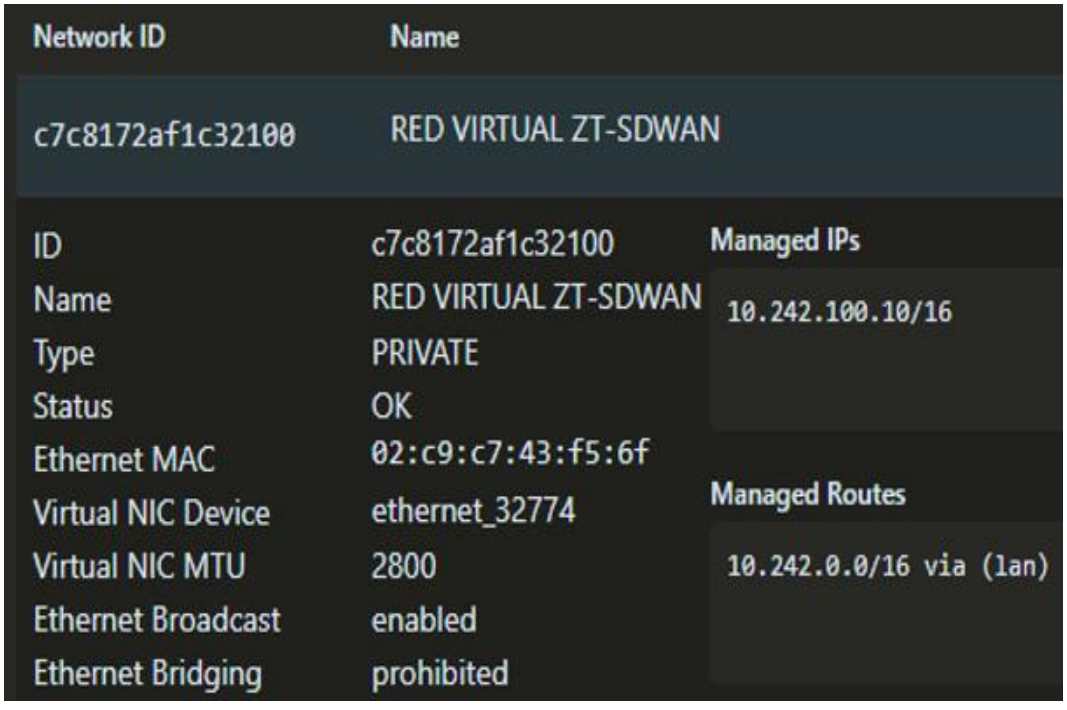


Figura 133. ZeroTier One en Windows – Comunicación Exitosa del Equipo.
Fuente: Propia del autor.

3.4.2. Conectar Clientes Linux

ZeroTier One para Linux en repositorios de my.zerotier.com y GitHub. Necesitará un usuario no root con privilegios sudo, (github.com, act.2021)

1. Para DEBIAN, UBUNTU, CENTOS, RHEL, FEDORA

Código Fuente 14. Ejecutar ZeroTier One en Equipos Linux.

```
# Si está dispuesto a confiar en SSL para autenticar el sitio,

        $sudo apt install zerotier-one
        $sudo curl -s https://install.zerotier.com | sudo bash


# Si tiene GPG instalado, hay una opción más segura disponible:

$ sudo curl -s 'https://raw.githubusercontent https.com/zerotier/ZeroTierOne /master/doc/contact%40zerotier.com.gpg' | gpg --import && \ if z=$(curl -s 'https://install.zerotier.com/' | gpg); then echo "$z" | sudo bash; fi


        CONECTARNOS A UNA RED

# Tanto la primera vez, como todas aquellas veces que queremos conectar nuestros dispositivos Linux a ZeroTier, utilizaremos el siguiente comando para unirnos a la red Central utilizando el ID de la red ZeroTier Central:

        $docker exec -it zerotier-one bash

        $sudo zerotier-cli join %NETWORK_ID   %%c7c8172af1c32100


# En un único comando (cambie a su propio ID)

        $docker exec -it zerotier-one zerotier-cli join "ID NETWORK"


# Listar las redes

        $sudo zerotier-cli listnetworks


# Información del nodo de Red instalado.

        $sudo zerotier-cli info


# Verificar la conectividad.

       $ip addr sh zt0 | grep 'inet'
```

Fuente 1: https://github.com/docker-projects/docker-zerotier
Fuente 2: https://github.com/zerotier/ZeroTierOne, [github.com, act.2021]

2. ZeroTier One en Contenedores de Docker para amd64 – arm32v7

Código Fuente 15. Ejecutar ZeroTier One en Contenedores de Docker.

```
#Preparar la instalación.

        $ sudo apt install zerotier-one


#ZeroTier One en docker para amd64

        $ docker run --name zerotier-one --device=/dev/net/tun --net=host --cap-add=NET_ADMIN --cap-add=SYS_ADMIN -v $HOME/docker/zerotier-one:/var/lib/zerotier-one ugeek/zerotier:amd64


#ZeroTier One en docker para arm32v7

        $ docker run --name zerotier-one --device=/dev/net/tun --net=host --cap-add=NET_ADMIN --cap-add=SYS_ADMIN -v $HOME/docker/zerotier-one:/var/lib/zerotier-one ugeek/zerotier:i386


        CONECTARNOS A UNA RED

#Utilizaremos el siguiente comando:

        $ docker exec -it zerotier-one bash
        $ zerotier-cli join c7c8172af1c32100
```

Fuente 2: https://github.com/zerotier/ZeroTierOne, [github.com, act.2021]

3. Se debe autenticar el equipo para tener acceso a la red.

Figura 134. ZeroTier One en Linux – Autenticación del Equipo en ZeroTier Central.
Fuente: Propia del autor.

3.4.3. Conectar Clientes Móviles

Disponible en las tiendas de las plataformas móviles.

3.4.3.1. Conectar Equipos Android.

1. Disponible desde la play store de equipos Android.

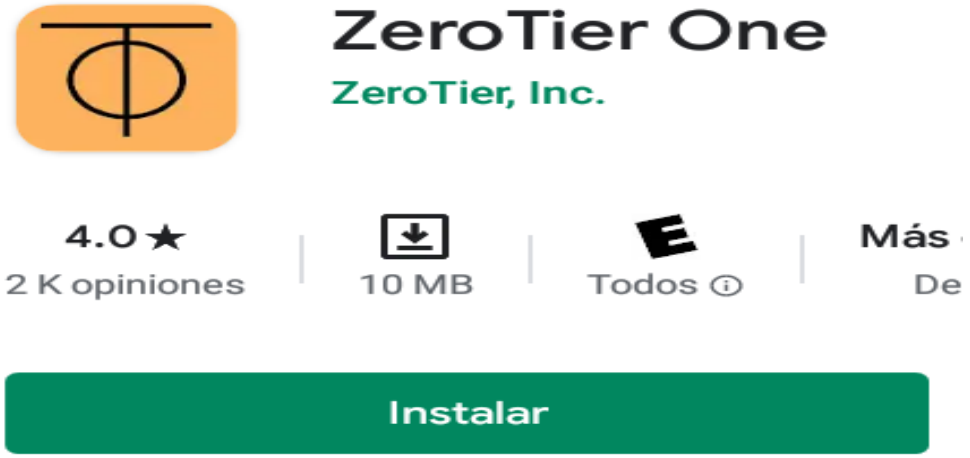


Figura 135. ZeroTier One en Dispositivo Android, (Print Screen)
Fuente: Propia del autor.

2. Portal de bienvenida de la app, inicio de configuración de red.

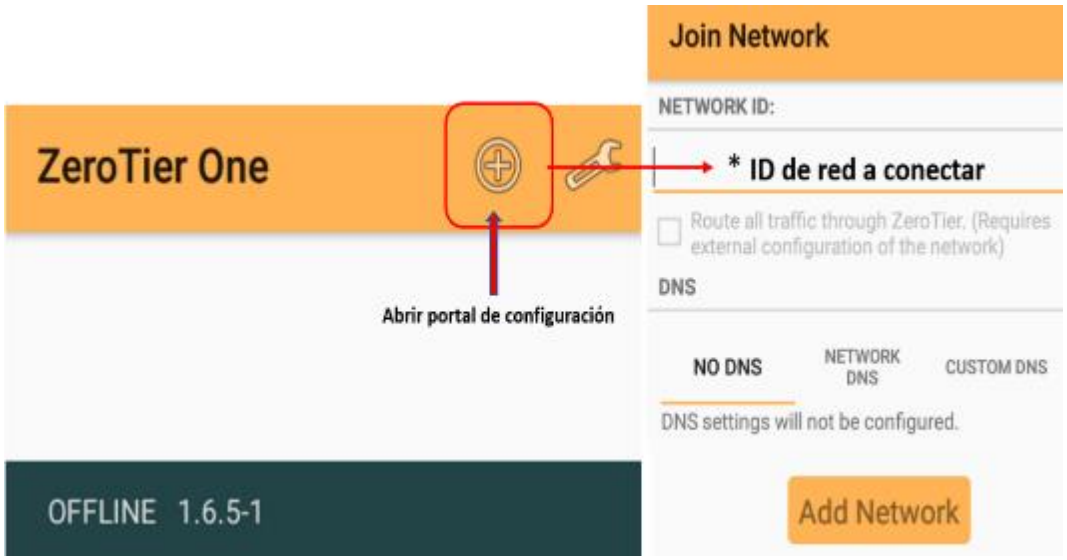


Figura 136. ZeroTier One en Android – Configuración de la Instalación.
Fuente: Propia del autor.

3. Cargar el Id de red a la cual se va a conectar el dispositivo móvil.

Figura 137. ZeroTier One en Android – Unirse a la Red de ZeroTier Central.
Fuente: Propia del autor.

4. Activar permisos para instalar Túnel VPN entre el equipo y la red.

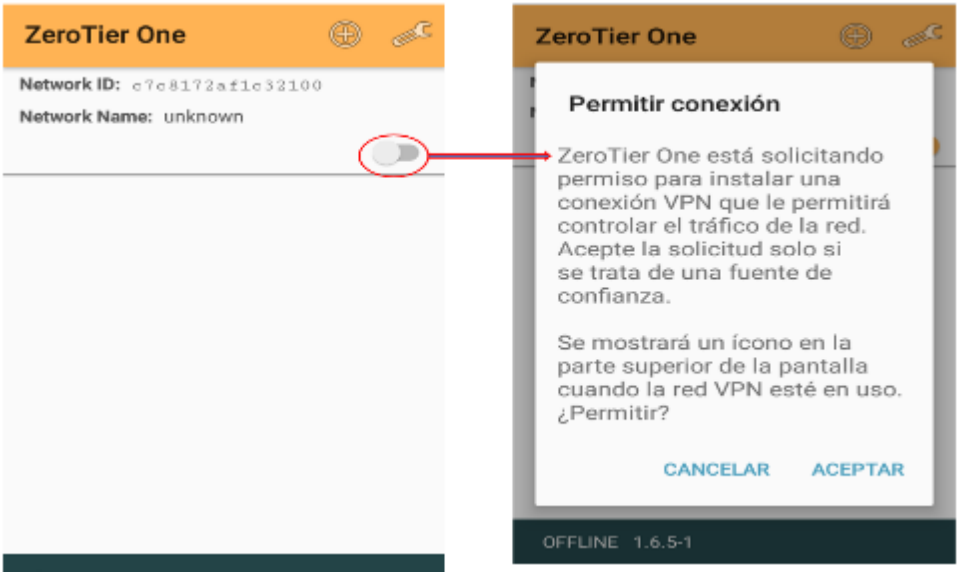


Figura 138. ZeroTier One en Android – Conceder Permisos de Comunicación.
Fuente: Propia del autor.

5. Se debe autenticar el equipo para tener acceso a la red.

Figura 139. ZeroTier One en Android – Autenticación de Equipo en ZeroTier Central.
Fuente: Propia del autor.

6. Equipo conectado a la red, podemos activar datos y conectarnos desde cualquier lugar.

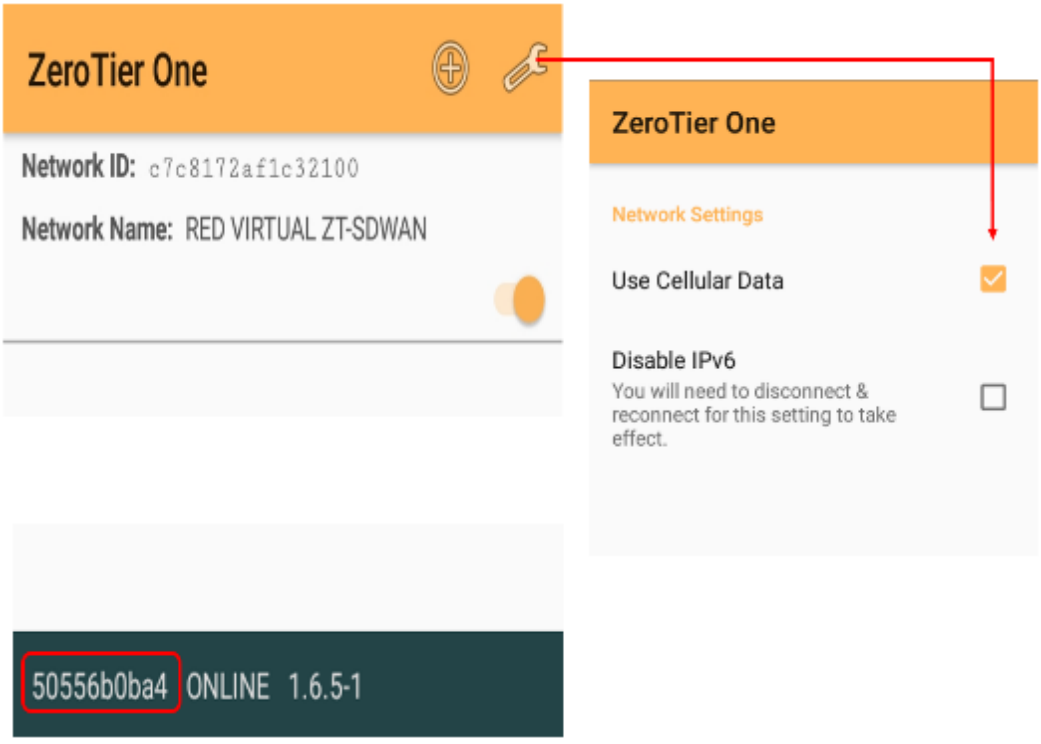


Figura 140. ZeroTier One en Android – Comunicación Exitosa del Equipo.
Fuente: Propia del autor.

3.4.3.2. Conectar Equipos IOS.

1. Disponible desde la App Store para iPhone y iPad.

Figura 141. ZeroTier One en Dispositivo IOS, (Print Screen).
Fuente: Propia del autor.

2. Portal de bienvenida del nodo app, inicio de configuración de red.

Figura 142. ZeroTier One en IOS – Configuración de la Instalación, (Print Screen)
Fuente: Propia del autor.

3. Cargar el Id de la red a la cual se va a conectar el dispositivo IOS.

Figura 143. ZeroTier One en IOS – Unirse a la Red de ZeroTier Central.
Fuente: Propia del autor.

4. Activar permisos para instalar Túnel VPN entre el equipo y la red.

Figura 144. ZeroTier One en IOS – Conceder Permisos de Comunicación.
Fuente: Propia del autor.

5. Se debe autenticar el equipo para tener acceso a la red.

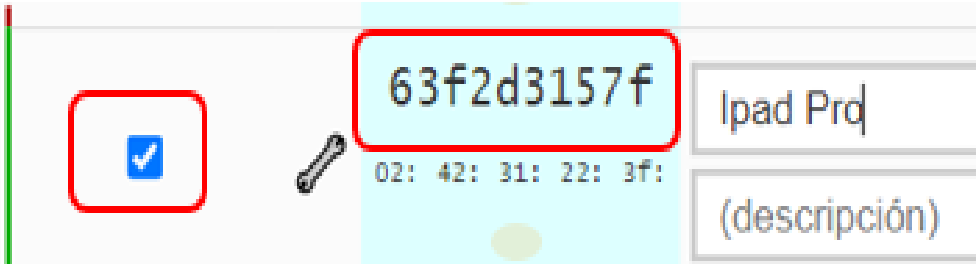


Figura 145. ZeroTier One en IOS – Autenticación de Equipo en ZeroTier Central.
Fuente: Propia del autor.

CAPÍTULO IV

4. ASIGNACIÓN DE EJERCICIOS PRÁCTICOS

El propósito de este capítulo es evaluar el proyecto, que utiliza los conceptos de SDN, si puede implementarse en un centro de datos pequeño / mediano, y la necesidad de conectar dispositivos activos de red de una manera rápida, confiable y segura, caso que se tiene cada vez más en cuenta al planificar entornos informáticos.

Entonces se trata de buscar una solución de red de alta capacidad y rentable para aplicaciones de interconexión de centros de datos DCI (Data Center Interconnect) en distancias metropolitanas, regionales y de larga distancia, proporcionando conexiones estables, fiables y de banda ultra ancha. Este dispositivo ofrece soluciones de red optimizadas para la era de la nube El enfoque SDN que utiliza intenciones y flujos subyacentes se pondrá en juicio con el mundo tradicional sin SDN, que depende en gran medida de la conmutación L2. Antes de la descripción de la metodología real y la evaluación posterior, se describirá en detalle el entorno del laboratorio que se va a realizar.

Parte de este capítulo son asignaciones creadas para la aplicación de comunicaciones en redes definidas por software, estas asignaciones individuales están diseñadas para guiar a través de los problemas de SDN ya que no tenemos demasiada libertad para establecer varias topologías de laboratorio. Además, nos gustaría gestionar y configurar los dispositivos a través del controlador SDN de ZeroTier para establecer la conectividad del canal óptico de extremo a extremo.

Los ejercicios responden a prácticas ya formalizadas en la TIC de forma tradicional, pues en este medio se incluirá la comunicación de equipos y servicios por medio de tecnología SDN brindada por el software zerotier.

4.1. Método de Enseñanza

El sistema para la enseñanza se centra en la parte práctica del problema. Al comienzo del capítulo, se da una breve introducción del tema en las prácticas individuales, que en lo posterior se explicará en cada configuración de los ejercicios resueltos en este capítulo. Las asignaciones se crean en forma de prácticas que se suceden de forma lógica y sistemática. Al final, cada evento se complementa con comentarios sobre la configuración realizada en cada paso.

Ciertas preguntas responden a las secciones críticas de configuración que el autor encontró durante el trabajo. Si se descubre un problema durante la configuración, se recomienda seguir las mejores prácticas o encontrar una solución alternativa adecuada para el problema en cuestión pues la información está a disposición de quien la solicite en repositorios o comunidades representantes de publicaciones y entidades referentes.

4.2. Definición de Asignaciones Prácticas

Se trata de aplicar la técnica de enseñanza utilizada es la instrucción directa y usa tareas definidas, sustentadas por el autor, que tiene el rol de control, disciplina y es responsable de la evaluación, La asignación de tareas disminuye el tiempo de aprendizaje de las habilidades, y permite mejorar el tiempo útil de implementación práctica al atender a grandes grupos de alumnos/as.

Cada asignación plasmada en este trabajo se familiarizará con la configuración del conmutador SDN ZeroTier y redes virtuales, que contienen dispositivos terminales, conmutadores y conexiones entre ellos. Cada una de las asignaciones está asociada con una topología específica de red como ya se indicó. Cada una responde a un problema diferente en el campo de las TIC y de las redes definidas por software.

4.3. Estructura de Asignaciones Prácticas

Las Prácticas están estructuradas de la misma manera para que quede claro de un vistazo qué se debe configurar y cómo proceder. Los capítulos de las prácticas están diseñados de la siguiente manera:

- Datos Informativos.
  Recursos utilizados.
  Preparación.

- Datos de la Práctica
  Objetivos generales y específicos.
  Metodología.
  Tabla de enrutamiento
  Marco Procedimental.
  Desarrollo.

4.4. Procedimiento para la Preparación de Asignaciones

Se va a utilizar el conmutador SDN de ZeroTier que es implementado como se detalla en el Capítulo III, se puede configurar de varias formas. La primera y más sencilla forma de configurar es utilizar la interfaz de Administrador de los contenedores (Portainer). La GUI de ZeroTier para la red y la GUI de Portainer para la administración de los contenedores a menudo es mediada por el usuario que usa un navegador.

Otra posibilidad es utilizar una GUI/CLI interactiva (terminal/pantalla), en los dispositivos conectados a la red, para interactuar tanto con el conmutador SDN (Zerotier), como para la red virtual y servicios en general, gracias a la cual se abre paso a la administración en todos los dispositivos virtuales que estén en ejecución en la red. Es decir, se puede acceder a cada componente de la red mediante una CLI independiente de donde se encuentre físicamente.

4.5. Ejercicios Prácticos

Este capítulo presenta la ejecución de las prácticas desarrolladas en la Red Virtual. Para ello se realizarán tres prácticas en base a cumplir los objetivos específicos del proyecto.

- Práctica 1. Generar un Escenario de Comunicaciones entre Dispositivos Portátiles y PC.
- Práctica 2. Analizar el Funcionamiento de un Servidor NAS con Tecnología SDN – Escenario SD-WAN.
- Práctica 3. Documentar los Resultados de la Conectividad y Tráfico de Datos entre Equipos Internos y Remotos de la Red.

Nota.- En cuanto a los servicios cargados en la red, en el apéndice de anexos encontrará la guía completa de instalación y configuración (ver anexo).

4.5.1. Práctica 1: Generar un Escenario de Comunicaciones entre Dispositivos Portátiles y PC

DATOS INFORMATIVOS

Recursos Utilizados

- Proyecto de Red Virtual.
- Raspberry Pi 3B+ o superior.
- Equipos de Cómputo PC o Portátiles.
- Equipos móviles Android, IPhone o iPad.
- Tres redes de internet con conexión ethernet, preferible de operadores diferentes.
- Enrutador que sirve como puerta de enlace para la conexión a Internet y la administración de SSH.

Preparación

- Comunicación de red virtual - Sistema instalado y operacional.
- Raspberry pi 3B + con sistema operativo Raspbian y servidor SSH funcional.
- Crear usuario Administrador de red ZeroTier.
- Equipos de Cómputo con cliente ZeroTier One instalado y cliente SSH (por ejemplo, Putty).
- Equipos móviles con apk ZeroTier One instalado.
- Disponibilidad de conexión a Internet.
- Identificar el segmento de red de las redes LAN.

Procedimiento

1. Habilitar la administración y el acceso de la Red ZeroTier.
2. Configurar tabla de enrutamiento en los equipos invitados de la red según el diseño (Topología de Red).
3. Ejecución del escenario de comunicación de redes.
4. Pruebas de Comunicación de equipos en las redes VXLANs.
5. Gestión de Archivos en la Red.

DATOS DE LA PRÁCTICA

Objetivo General.

El objetivo de la Práctica 1 es familiarizarse con el principio de funcionamiento de la Red Virtual y la inmersión de ciertos protocolos de comunicación, para habilitar la comunicación entre dispositivos y redes, es necesario configurar la red superpuesta de ZeroTier y así lograr salir a internet permitiendo la comunicación entre diferentes espacios de direcciones en los dispositivos de red.

Objetivos Específicos

- Familiarizarse con las posibilidades de las redes SDN.
- Realizar configuraciones básicas entre dispositivos de red y el Proyecto de Red Virtual.

Glosario

LAN: Red de Área Local o LAN (Local Área Network). Es una red que conecta equipos en un área relativamente limitada.

SDN: Redes Definidas por Software o (Software Define Network). Conjunto de técnicas relacionadas con el área de redes computacionales basadas en software.

P2P: Red de Pares o P2P (Peer to Peer). Modelo de comunicación de igual a igual, comparte arquitectura, tareas, privilegios y equipotentes en la aplicación.

VXLAN:– Redes Locales Virtuales Extensibles o VXLAN (Virtual Extensible Local Área Network). Es una red superpuesta, partiendo de una infraestructura de red LAN/WAN que funcionará como base para la comunicación de máquinas virtuales.

Marco Metodológico

Demuestre la posibilidad de implementar una red basada en SDN en los dispositivos disponibles de la red. Se proporcionan dos topologías que se muestran en la Figura 146 (Práctica 1-A) que sustenta la comunicación directa de equipos en la red virtual y Figura 161 (Práctica 1-B) que sustenta la comunicación de equipos y gestión de archivos en varias redes virtuales.

Para una comunicación adecuada, la tabla de enrutamiento (Tabla 7) debe configurarse correctamente en la red virtual (VXLANs), solo así el enrutamiento del tráfico a otra dirección de red se enviará correctamente.

La red virtual estará disponible para los participantes que se conecten a la red, pero para tener servicio deberá estar autenticado por el administrador de la red ZeroTier. Utilizamos el conmutador basado en software (ZeroTier), para hacer un uso más eficiente de la potencia informática que brinda la red superpuesta a la red LAN a la que está conectado su Host.

La red Virtual creada funciona en el protocolo ipv4 (Protocolo de Internet versión 4) para todos los equipos, la comunicación se realiza mediante un enlace punto a punto (P2P) en la red y entre redes al estar configurado con las rutas como se observan en las gráfica de la topologías correspondientes.

Al estar operativa las redes VXLANs podemos comunicarnos hacia y desde cualquier dispositivo, nos podemos conectar a través de internet y estar conectados como si estuviéramos en el mismo lugar, aunque físicamente estemos distantes como una VPN en particular, pero saltándonos configuraciones de administradores de redes sin importar las configuraciones de firewall que en su red estén configuradas.

El alcance de la red va desde una comunicación directa entre equipos hasta el aprovisionamiento y respaldo de información sin importar los GB que tenga que enviar gracias a SAMBA (SMB/CIFS), que es una suite de software de código abierto, permitiendo la interoperabilidad entre sistemas basados en Linux y sistemas basados en Windows.

Tablas de Direccionamiento IP

Tabla 7. Gestión de Enrutamiento de las Redes – Práctica 1.

Nombre	Valor	Denominado
ISP 1 – AP-Verizon	192.168.43.0/24	LAN 1
ISP 2 – CLARO AVENDANO	192.168.200.0/24	LAN 2
ISP 3 – UPS-GYE	172.18.142.0	LAN 3
RED VIRTUAL (VXLAN 1)		
Ruta gestionada de Red ZeroTier	10.242.0.0/16	ZT_VXLAN 1
Rango de asignación DHCP de ZeroTier	10.242.0.2 hasta 10.242.255.254	ZT_DHCP 1
Dirección IP del Puente ZeroTier	10.242.10.40/16(o use DHCP)	BR_ADDR 1
ID de Red ZeroTier (16 dígitos)	c7c8172af1c32100	NETWORK_ID 1
Nombre de la interfaz de Red ZeroTier	ztmjfejtrx	ZT_IF1
RED SOPORTE (VXLAN2)		
Ruta gestionada de Red ZeroTier	172.24.0.0/16	ZT_VXLAN 2
Rango de asignación DHCP de ZeroTier	172.24.0.2 hasta 172.24.255.254	ZT_DHCP 2
Dirección IP del Puente ZeroTier	172.24.10.40/16(o use DHCP)	BR_ADDR 2
ID de Red ZeroTier	8056c2e21c862c2b	NETWORK_ID 2
Nombre de la interfaz de Red ZeroTier	ztmj2i6twb	ZT_IF2

Fuente: Propia del autor

Tabla 8. Direccionamiento IP de los Equipos Conectados en Red – Práctica 1.

DISPOSITIVOS	XLAN_DHCP	VXLAN1/DHCP	VXLAN2/DHCP	AP
ANDROID J7	192.168.43.1	10.242.234.242	172.24.234.242	ISP 1
TABLET ADMIN	192.168.200.32			ISP 2
DESKTOP V7AVHRA	192.168.200.29	10.242.100.10	172.24.237.143	ISP 2
PORTATIL YILDER	192.168.200.34	10.242.3.3		ISP 2
GYECLABTELPC08	172.18.142.9	10.242.65.88		ISP 3
GYECLABTELPC11	172.18.142.10	10.242.144.81		ISP 3
IP PUENTE BR_ADDR	192.168.200.15	10.242.10.40	172.24.10.40	ISP 2

Fuente: Propia del autor

MARCO PROCEDIMENTAL

Comunicación de Equipos en Red Virtual – Escenario A

1. Topología de Comunicación de equipos en Práctica 1 – Escenario A.

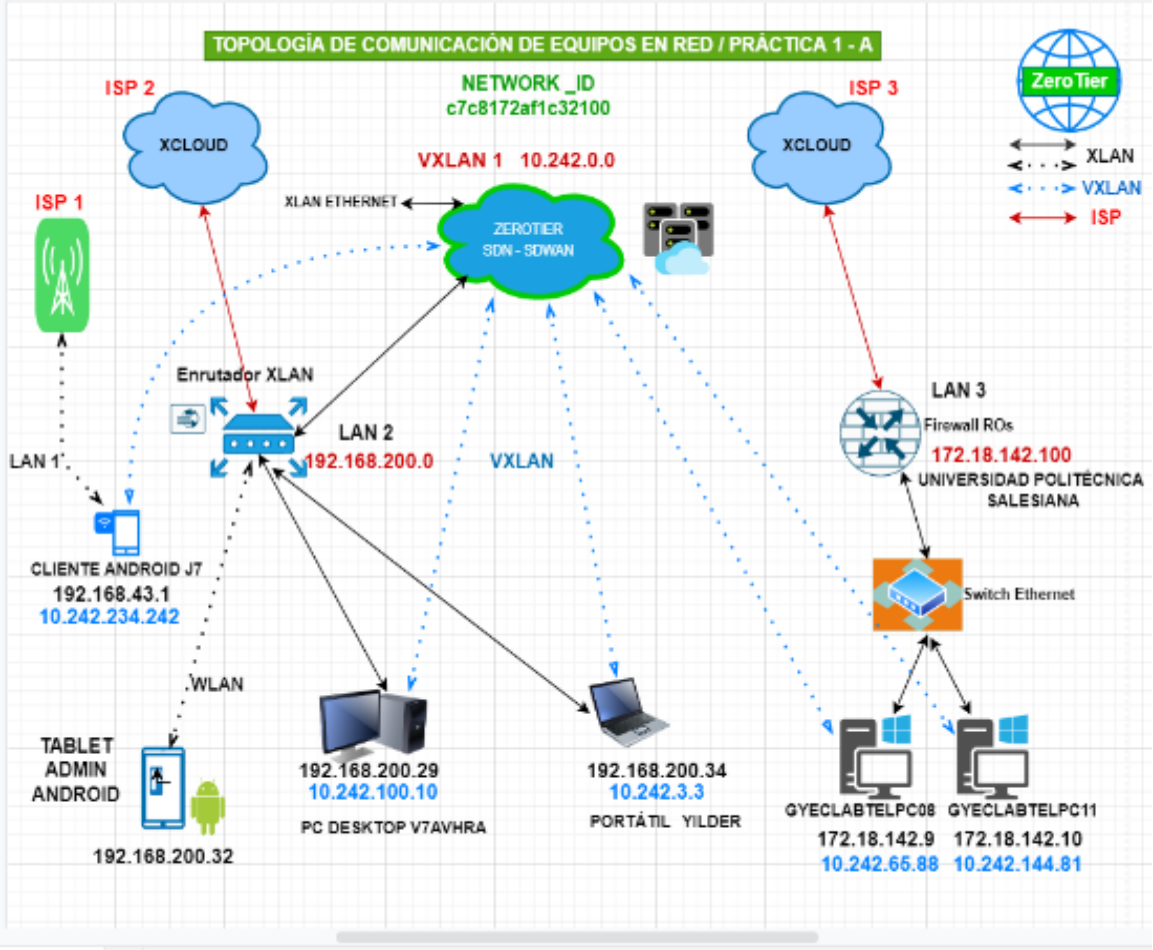


Figura 146. Topología de Red Práctica 1 – Escenario A.
Fuente: Elaboración propia del Autor en https://app.diagrams.net/

2. Identificar los equipos que se van a conectar a la red virtual.

Tipo de sistema:	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponibl

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo:	GYECLABTELPC08
Nombre completo de equipo:	GYECLABTELPC08

Figura 147. Identificar Equipos a Conectar en la Red VXLAN 1, (Print Screen).
Fuente: Propia del autor.

3. Crear cuenta de administrador y red ZeroTier, para esto siga las instrucciones detalladas en el literal 3.3.6.1 para autenticarse y 3.3.6.2 para crear la Red de ZeroTier, la red tendrá un ID de 16 dígitos.

Figura 148. Crear Red ZeroTier Central y la Cuenta Administrador, (Print Screen).
Fuente: Propia del autor.

4. Descargar el nodo ZeroTier One en cada equipo que va a conectar como cliente de red, para esto siga las instrucciones detalladas en el subcapítulo 3.4 para cada equipo.

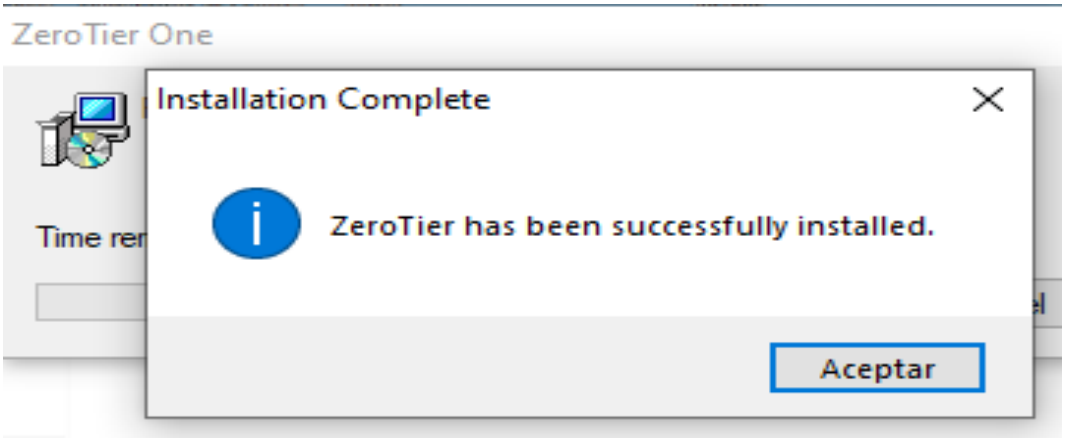


Figura 149. Descarga he Instalación de los NODO_ID Clientes en cada Equipo.
Fuente: Propia del autor.

5. Una vez instalado el nodo cliente en cada equipo únase a la red de su preferencia con el Network_ID de la Red a la que desea conectarse, para esto siga las instrucciones detalladas en el literal 3.4.1 para equipos Windows y 3.4.3 para equipos móviles respectivamente.

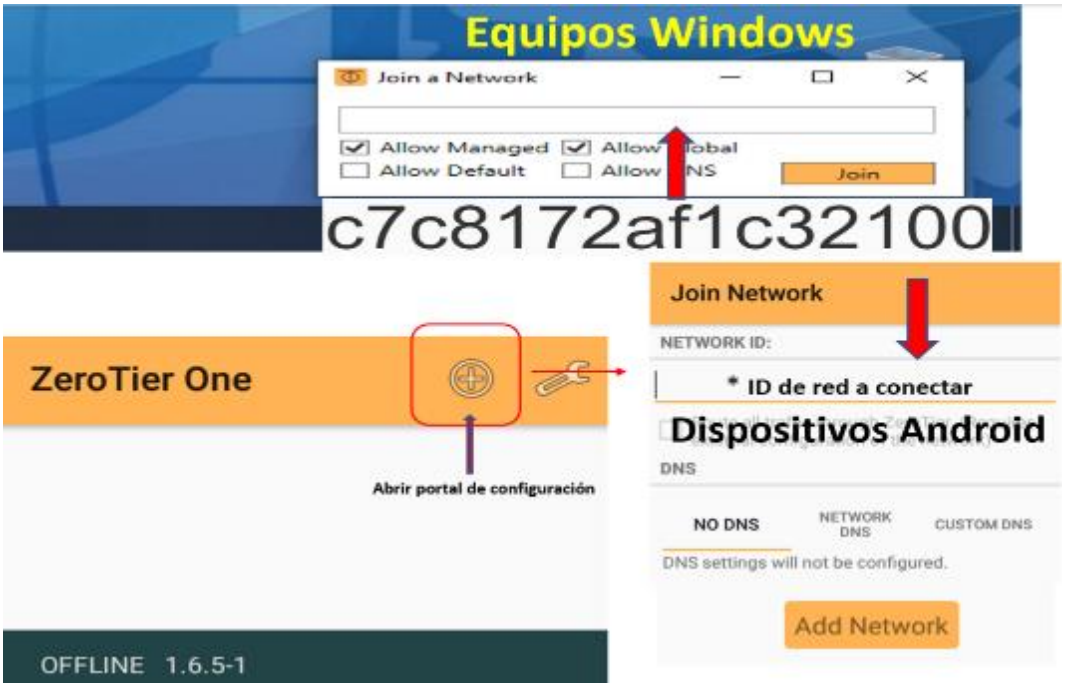


Figura 150. Uniendo Equipos a la Red Virtual – Ingrese NETWORK_ID 1.
Fuente: Propia del autor.

6. En el administrador central de red ZeroTier, autenticará todos los equipos que formarán parte de la red, deberá observar esta pantalla.

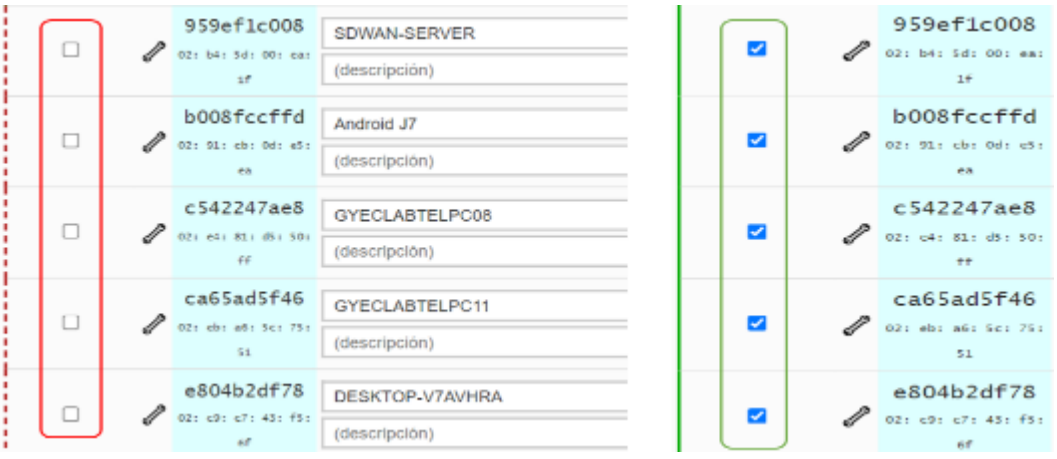


Figura 151. Autenticar los Equipos de la Red en ZeroTier Central – VXLAN 1.
Fuente: Propia del autor.

7. A continuación, verificamos las interfaces de red virtual gestionado por (ZT_DHCP1), en los equipos conectados en la Red, en esta instancia podemos gestionar la ruta IP virtual y el nombre que recibirá como identificación cada equipo host de la red.

Figura 152. Direccionamiento IP de los Equipos Conectados en VXLAN 1.
Fuente: Propia del autor.

8. Una vez gestionadas las rutas, los equipos deberán aparecer en red con su protocolo IP asignado por (ZT_DHCP1), o por el administrador y EN LINEA en el panel de ZeroTier Central.

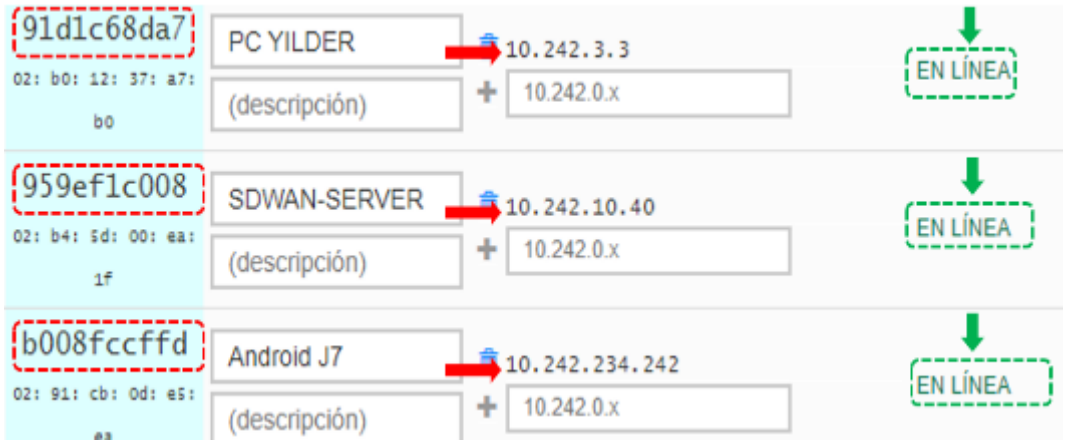


Figura 153. Verificación de Equipos EN LINEA – VXLAN 1, (Print Screen).
Fuente: Propia del autor.

9. Comunicación entre VXLAN 1 y NODO_ID e804b2df78 » Equipo PC DESKTOP -V7AVHR conectado en LAN 2.

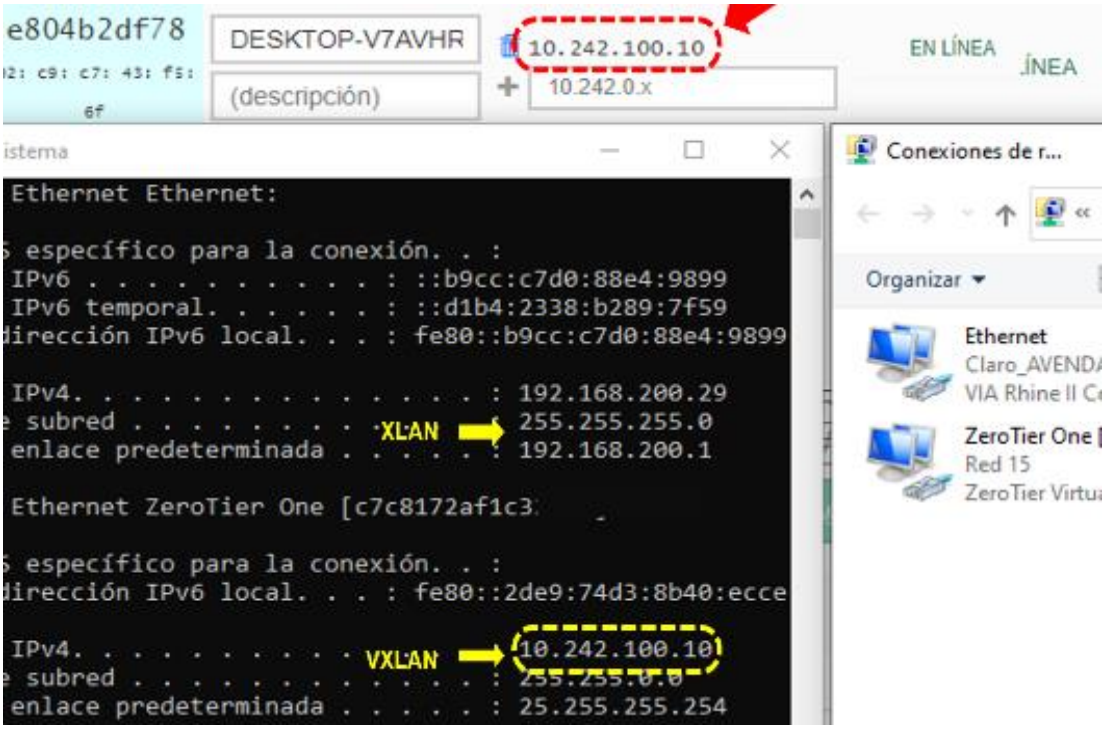


Figura 154. Comunicación entre VXLAN 1 y Equipo PC DESKTOP V7AVHR.
Fuente: Propia del autor.

10. Comunicación entre VXLAN 1 y NODO_ID 91d1c68da7 » PORTATIL.

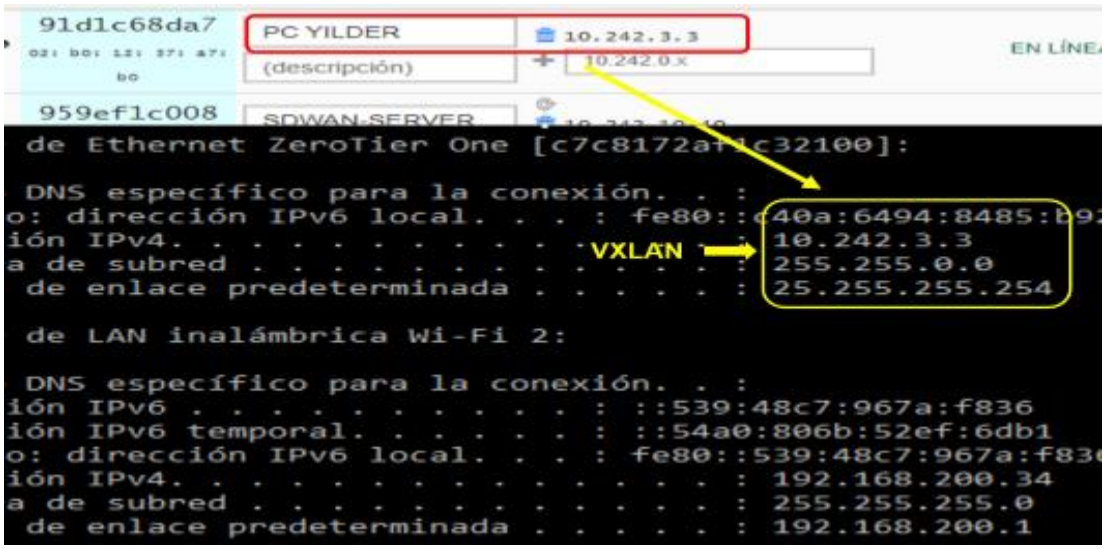


Figura 155. Comunicación entre VXLAN 1 y Equipo PORTÁTIL YILDER.
Fuente: Propia del autor.

11. Comunicación entre VXLAN 1 (LAN 2), y NODO_ID c542247ae8 » Equipo PC GYECLABTELPC08 conectado en LAN 3.

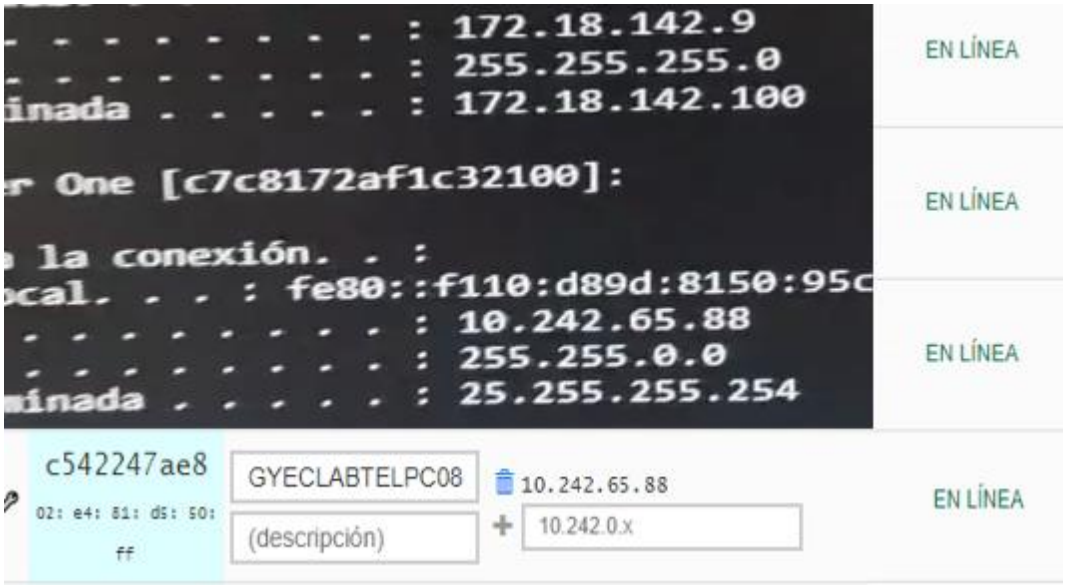


Figura 156. Comunicación entre VXLAN 1 y Equipo PC GYECLABTELPC08.
Fuente: Propia del autor.

12. Comunicación entre VXLAN 1(LAN 2) y NODO_ID ca65ad5f46 » Equipo PC GYECLABTELPC11 conectado en LAN 3.

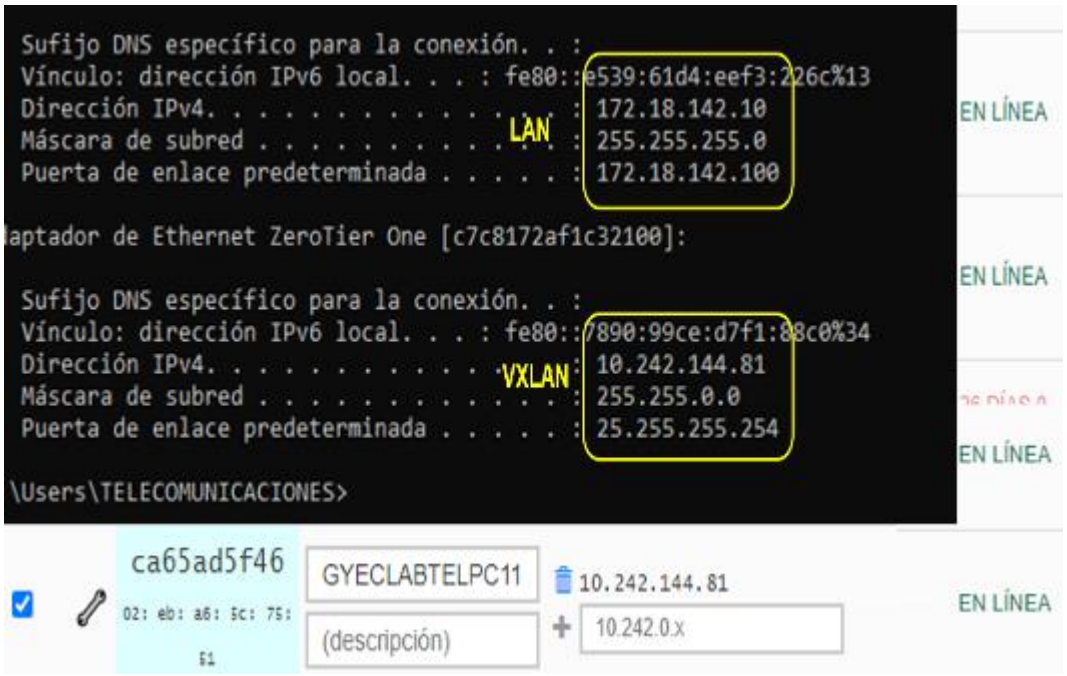


Figura 157. Comunicación entre VXLAN 1 y Equipo PC GYECLABTELPC11.
Fuente: Propia del autor.

13. Comunicación entre VXLAN 1 en (LAN 2), y NODO_ID b008fccffd » Equipo Android J7 conectado en (LAN 1).

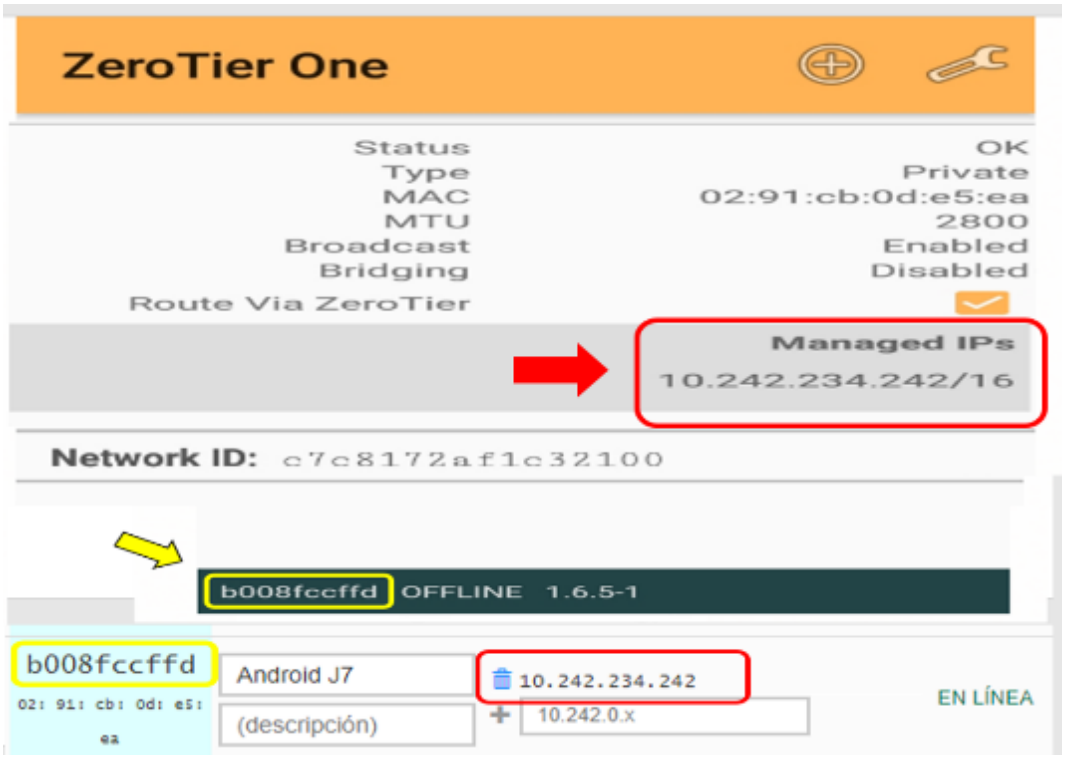


Figura 158. Comunicación entre VXLAN 1 y Equipo Android J7, (Print Screen).
Fuente: Propia del autor.

14. Escenario de comunicación de equipos conectados en red VXLAN 1, equipos físicos situados en ubicaciones remotas.

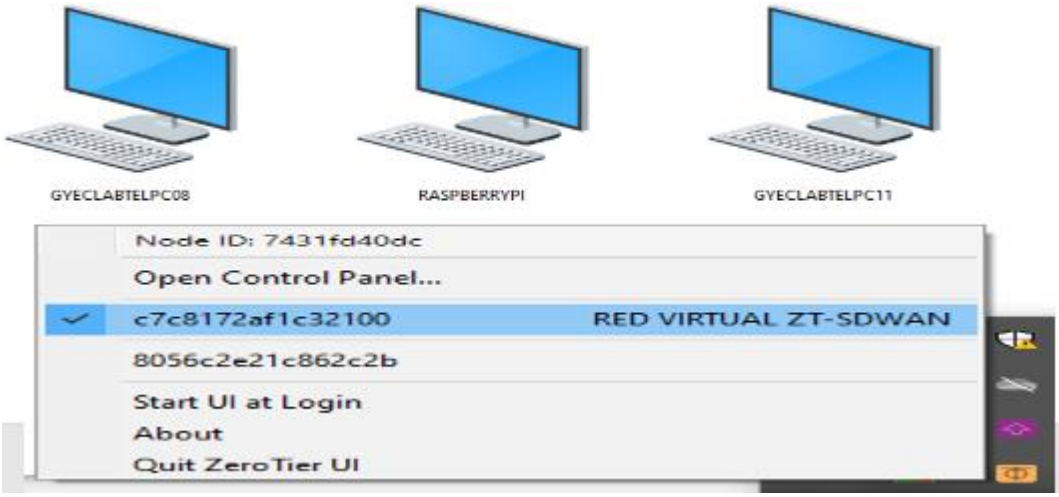


Figura 159. Escenario de Comunicación de Equipos Conectados en Red VXLAN 1.
Fuente: Propia del autor.

15. Se realiza diagnóstico de transmisión mediante protocolo ICMP al Host de la Red (BR_ADDR = 10.242.10.40), y en equipo TABLET ADMIN (LAN_DHCP=192.168.200.32) desde » DESKTOP – V7AVHR.

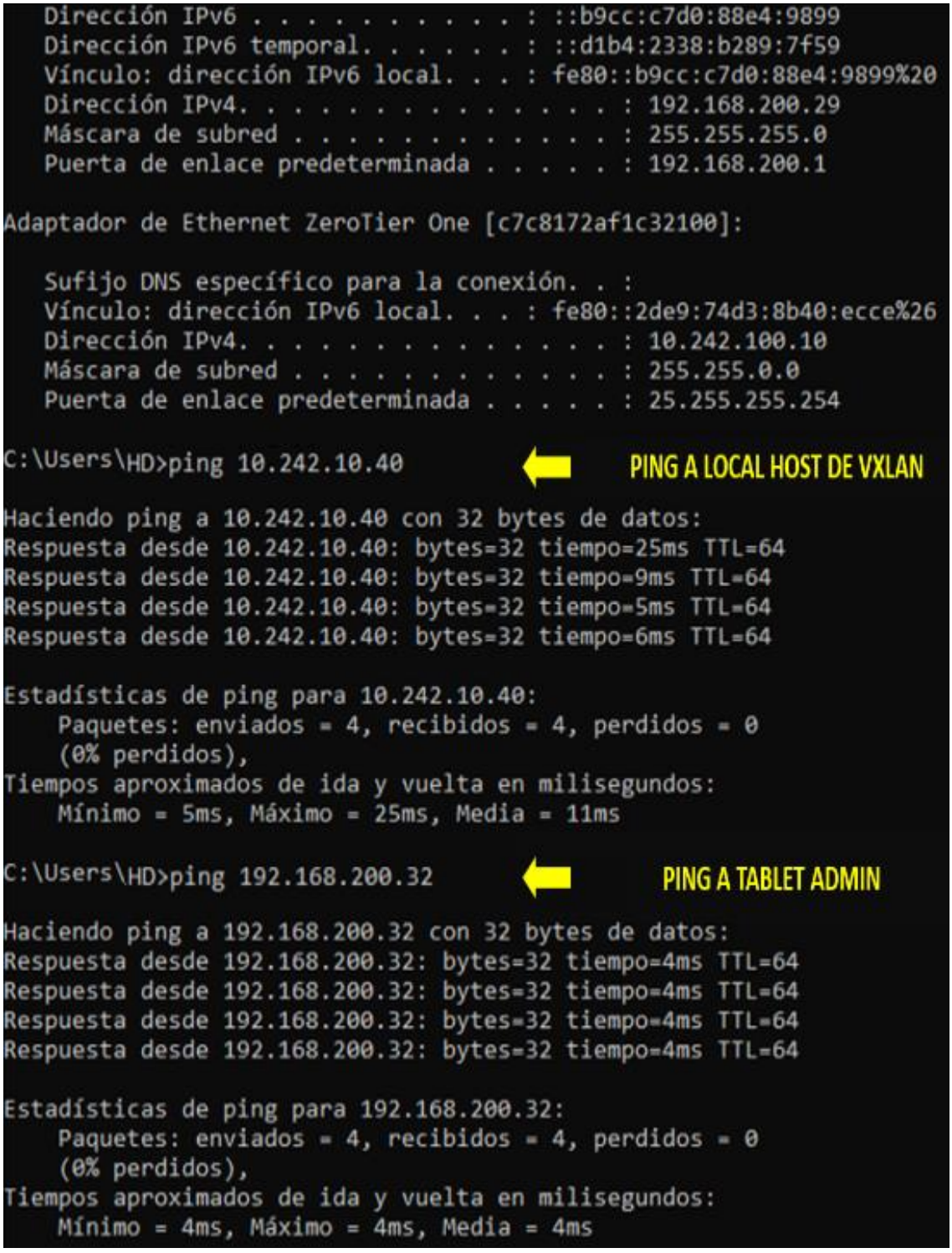


Figura 160. Diagnóstico de Protocolo ICMP en LAN y VXLAN 1, (Print Screen).
Fuente: Propia del autor.

Comunicación de Equipos en Redes Virtuales – Escenario B

1. Topología de Comunicación de Equipos en Práctica 1 – Escenario B.

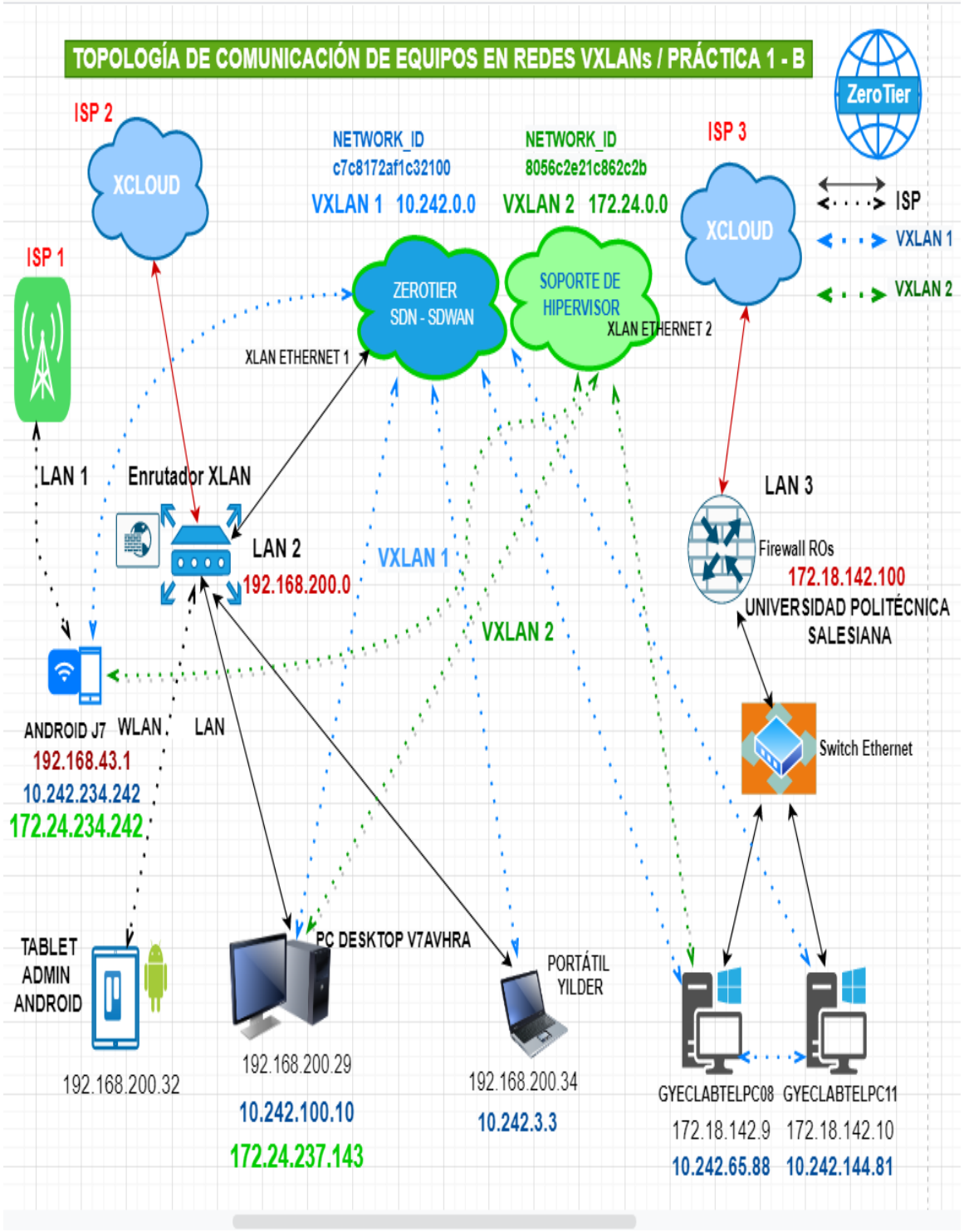


Figura 161. Topología de Red Práctica 1 – Escenario B.
Fuente: Elaboración propia del autor en https://app.diagrams.net/

2. Creamos un escenario de conexión de equipos en dos redes, para esto deberá crear una segunda red (Soporte de Red Virtual–VXLAN 2) (figura 162), o cree una subred en VXLAN 1 a la que llamaremos Sucursal (figura 163). Comunicaremos el NODO_ID e804b2df78 » DESKTOP-V7AVHR. Para lograr esto siga las instrucciones detalladas en los subcapítulos 3.3.6.2. y 3.4.1 respectivamente.

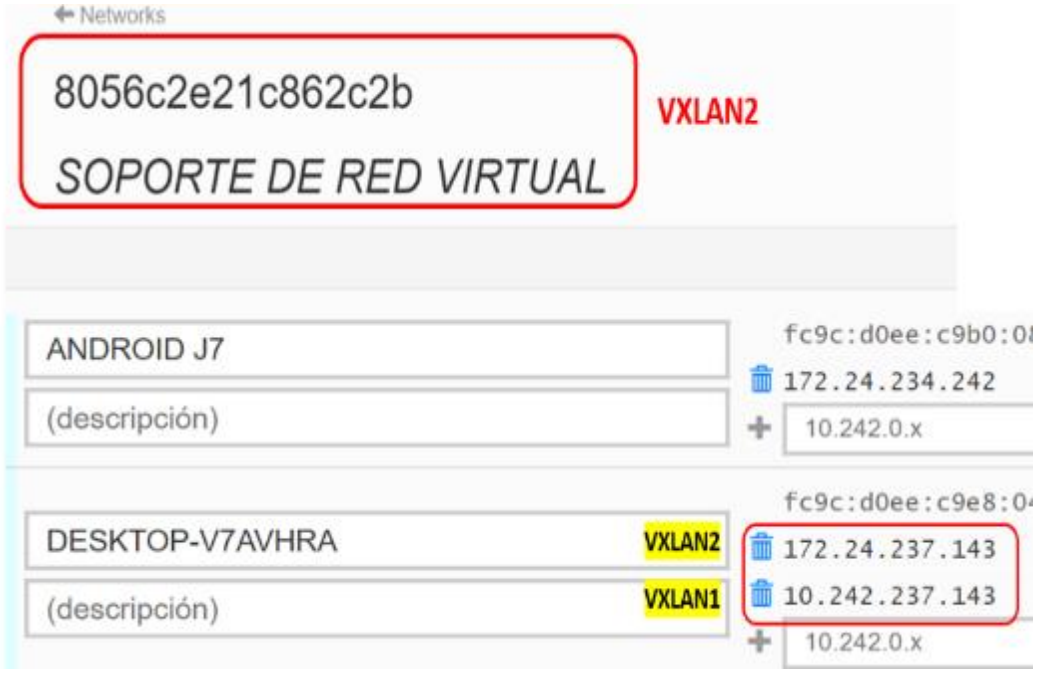


Figura 162. Ilustración de Red Soporte de Red Virtual (VXLAN 2).
Fuente: Propia del autor.

3. Cabe destacar que la red Sucursal pertenece al mismo grupo de redes del administrador del Proyecto de Red Virtual. Sin embargo, deberá plasmar su propia configuración siguiendo los pasos detallados en el subcapítulo 3.3.6.2.

NETWORK ID	NAME↑	DESCRIPTION	SUBNET
c7c8172af1c32100	RED VIRTUAL ZT-SDWAN	Red Virtual VXLAN1	10.242.0.0/16
8286ac0e47d298b6	Sucursal		172.22.0.0/16

Figura 163. Ilustración de Subred Virtual Sucursal, (Print Screen).
Fuente: Propia del autor.

4. Integrar los nodos clientes de los equipos a la red Virtual de la cual requerimos obtener comunicación. En esta práctica comunicaremos dos redes (VXLAN 1 – VXLAN 2), con el NODO_ID e804b2df78 » Equipo DESKTOP – V7AVHR, para esto siga las instrucciones detalladas en el literal 3.4.1 para instalar ZeroTier One.

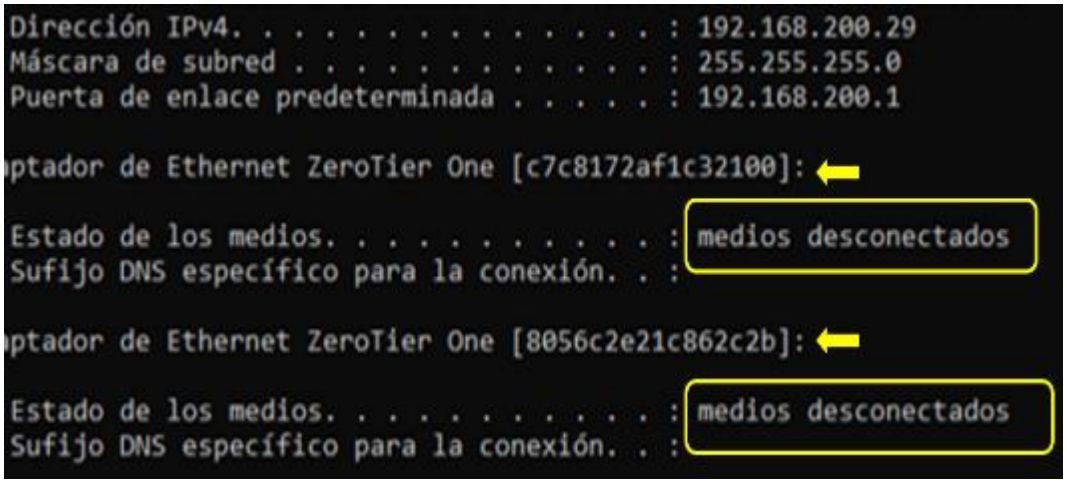


Figura 164. Equipo DESKTOP-V7AVHR – Gestión de Ruta de las Redes Virtuales.
Fuente: Propia del autor.

5. Active las VXLANs en los nodos de red ZeroTier One de los equipos como ya se ha indicado anteriormente. Recuerde que estamos trabajando en el NODO_ID e804b2df78 » DESKTOP-V7AVHR.

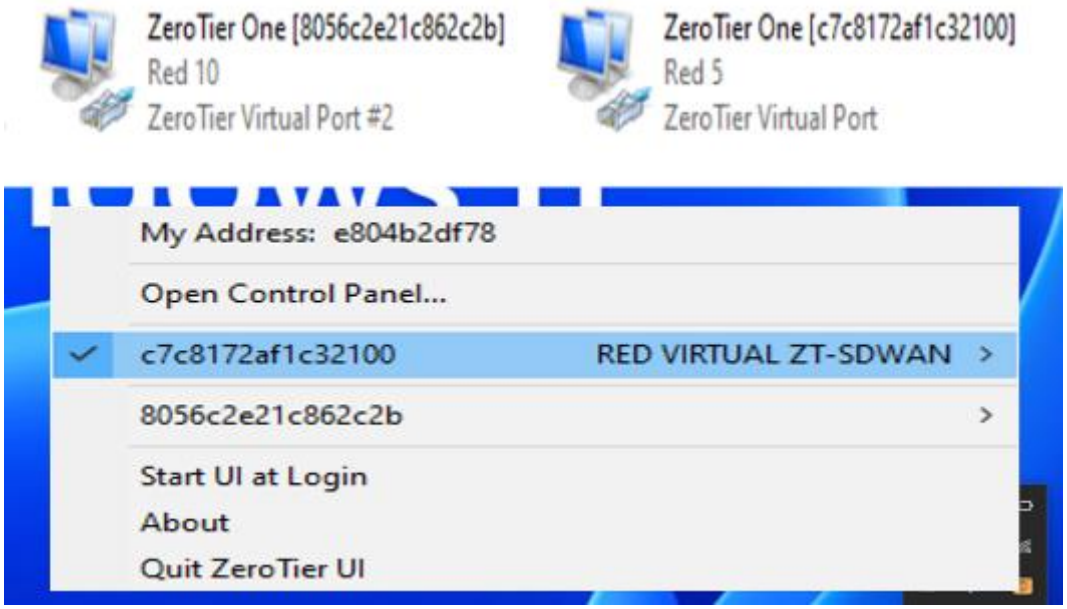


Figura 165. Equipo DESKTOP-V7AVHR – NETWORK_ID Habilitados ; ZT_IFs OK
Fuente: Propia del autor.

6. Una vez gestionadas las rutas, los equipos deberán comunicarse por las direcciones de protocolo IP asignadas por ZT_DHCP 1 y ZT_DHCP 2 de las VXLANs respectivamente en DESKTOP-V7AVHR.

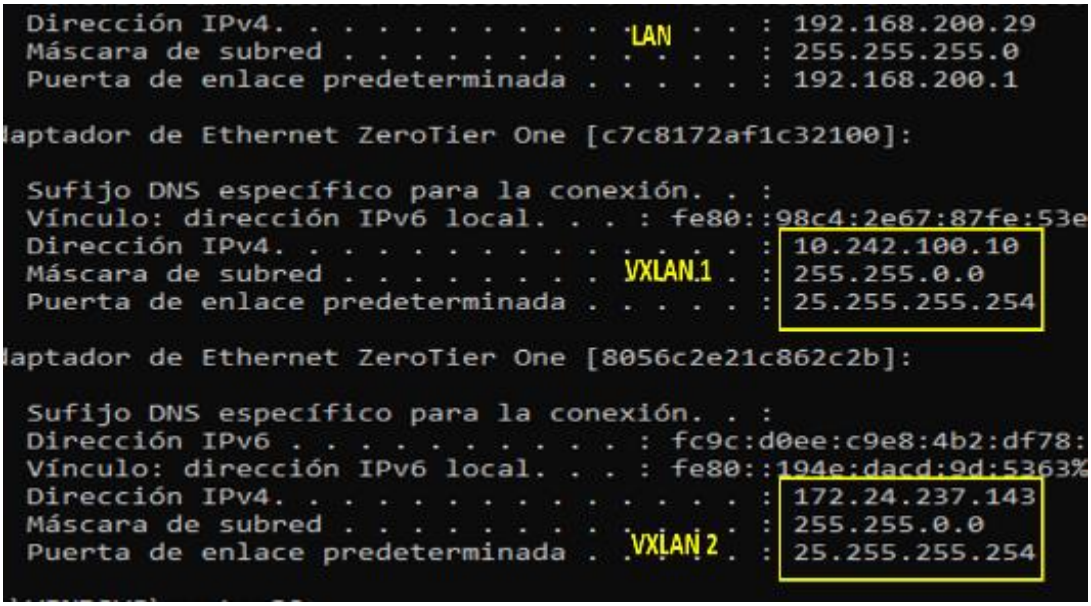


Figura 166. Equipo DESKTOP-V7AVHR Direccionamiento IP de Redes VXLANs.
Fuente: Propia del autor.

7. Se muestran los Adaptadores y NETWORK_ID de VXLANs activados – Verificación en el NODO_ID e804b2df78 » DESKTOP-V7AVHR.

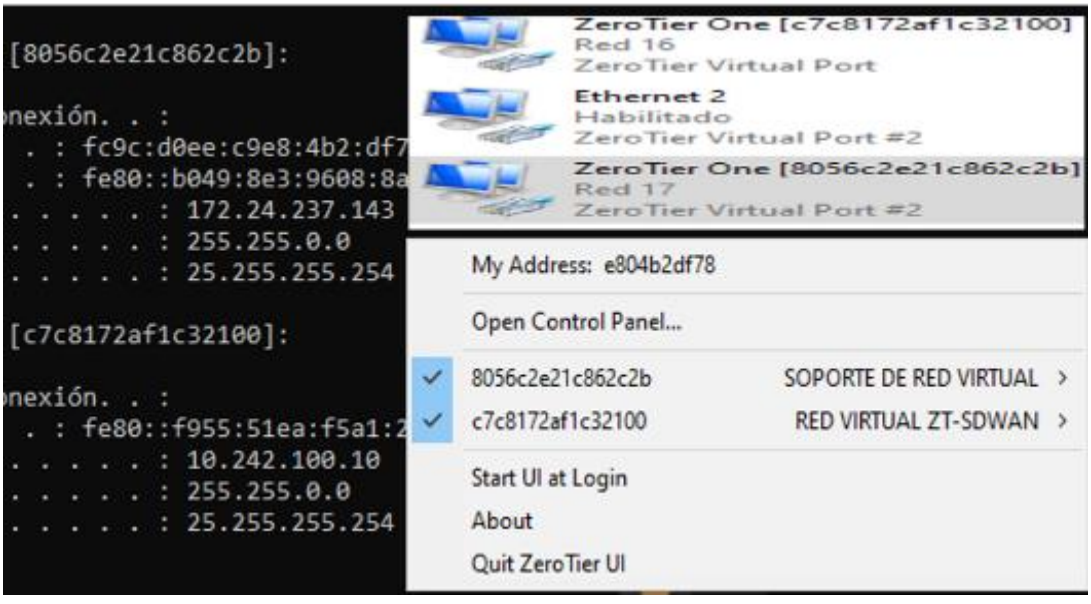


Figura 167. Equipo DESKTOP-V7AVHR en Comunicación con VXLAN1 Y VXLAN 2.
Fuente: Propia del autor.

8. Para integrar equipos móviles en las VXLANs deberá integrar los NODOS_ID de los equipos a la red Virtual de la cual requiera obtener servicio de comunicación. En esta práctica conectaremos dos redes (VXLAN 1 – VXLAN 2), con el NODO_ID b008fccffd » Android J7, para lograr esto siga las instrucciones detalladas en el subcapítulo 3.4.3 para integrar equipos móviles con ZeroTier One.

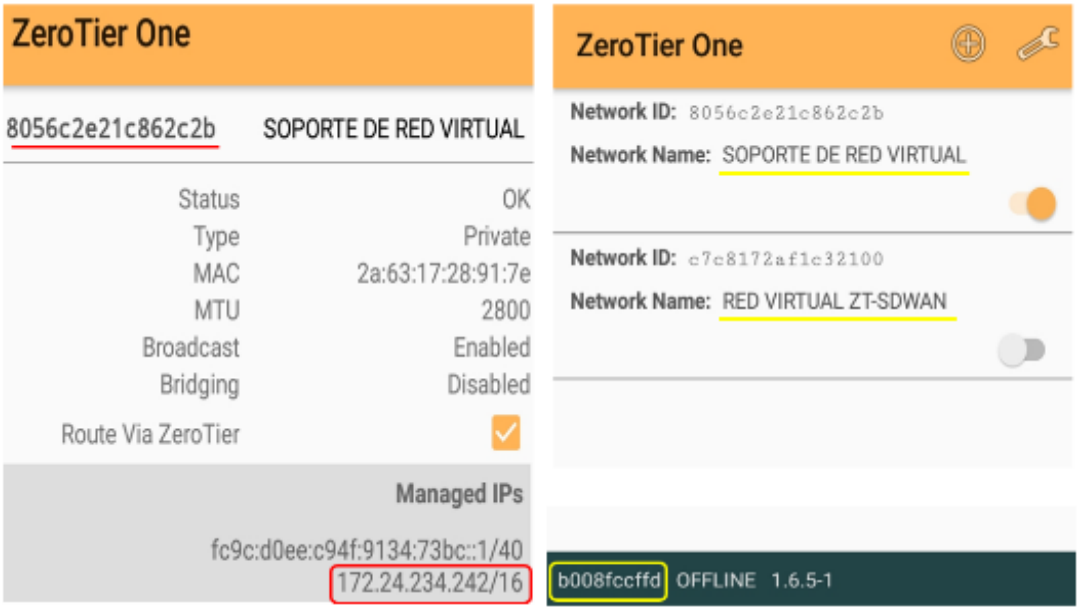


Figura 168. Equipo Android J7 en Comunicación con VXLAN1 Y VXLAN 2.
Fuente: Propia del autor.

9. El Problema más común por encontrarse en la gestión de Ruta es el error en el direccionamiento ip de los equipos.

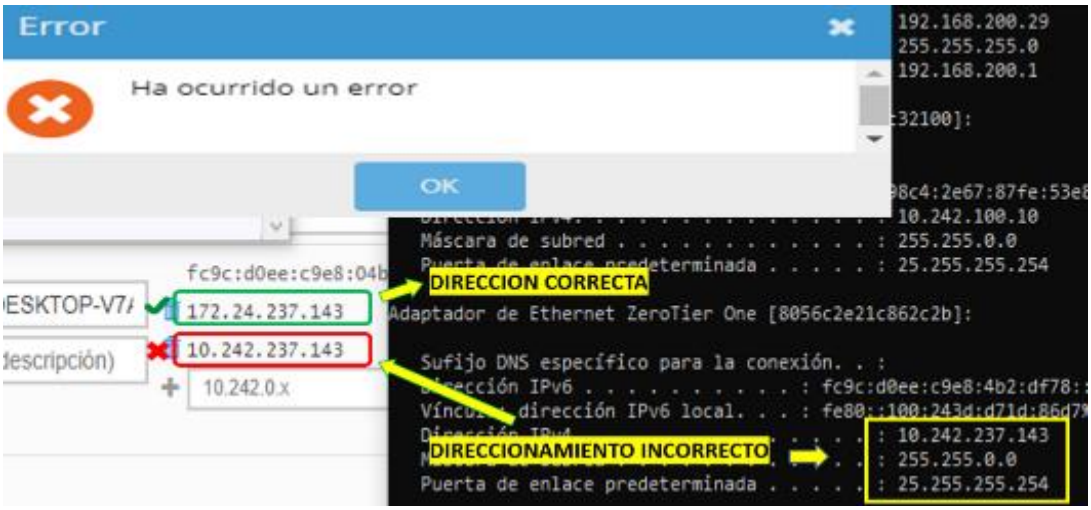


Figura 169. Error de Comunicación – Incorrecto Direccionamiento IP, (Print Screen)
Fuente: Propia del autor.

4.5.2. Práctica 2: Prestaciones del Proyecto como Servidor NAS con Tecnología SDN de ZeroTier – Escenario SD-WAN

DATOS INFORMATIVOS

Recursos Utilizados

- Proyecto de Red virtual.
- Raspberry Pi 3B+ o superior.
- Equipos de Cómputo PC o Portátiles.
- Equipos móviles Android, IPhone o iPad.
- Tres redes de internet con conexión ethernet, preferible de operadores diferentes.
- Enrutador que sirve como puerta de enlace para la conexión a Internet y la administración de SSH.
- Software de código abierto ZeroTier.

Preparación

- Comunicación de la red virtual - Nodo ZeroTier instalado en el host. Sistema de contenedores operacional.
- Raspberry 3B + con sistema operativo Raspbian y servidor SSH funcional.
- Sistema FreeNAS OpenMediaVault con Servidor SSH funcional.
- Crear usuario Administrador de red ZeroTier.
- Equipos de Cómputo con cliente ZeroTier One instalado y cliente SSH (por ejemplo, Putty).
- Equipos móviles con apk ZeroTier One instalado.
- Disponibilidad de conexión a Internet.
- Identificar el segmento de red de las redes LAN.

Procedimiento

1. Habilitar la administración de los contenedores.
2. Habilitar la administración y el acceso de la Red ZeroTier.
3. Configurar tabla de enrutamiento en los equipos invitados de la red según el diseño (Topología de Red).
4. Ejecución del escenario de comunicación de redes.
5. Pruebas de Comunicación de equipos en las redes VXLANs.
6. Gestión de Servicios cargados en el NAS de la Red Virtual.

DATOS DE LA PRÁCTICA

Objetivo General

Familiarizarnos con el principio de funcionamiento de la Red Virtual, la inmersión de ciertos protocolos de comunicación y las garantías sobre las prestaciones de servicio y comunicación total en un escenario de red SDWAN, se obtendrá un sistema de comunicación centralizado en nuestro propio servidor, en la oficina, en nuestro hogar, en fin, en el lugar que deseemos. Para habilitar la comunicación entre dispositivos y redes, es necesario configurar la red superpuesta de ZeroTier y así lograr salir a internet permitiendo la comunicación entre diferentes espacios de direcciones en los dispositivos de red.

Objetivos Específicos

- Familiarizarse con las posibilidades de las redes SDN-SDWAN.
- Realizar configuraciones básicas de administrador entre los dispositivos de red, y el Proyecto de Red Virtual.

Glosario

LAN: Red de Área Local o LAN (Local Área Network). Es una red que conecta equipos en un área relativamente limitada.

SDN: Redes Definidas por Software o (Software Define Network). Conjunto de técnicas relacionadas con el área de redes computacionales basadas en software.

SD-WAN: Las redes de área amplia definidas por software o SD-WAN (Software-Defined Networking Área Network), esta red proporciona acceso definido por software al enrutamiento de redes WAN. Funciona entre redes o centro de datos en regiones diferentes, a menudo muy remotas.

VXLAN: Redes Locales Virtuales Extensibles o VXLAN (Virtual Extensible Local Área Network). Es una red superpuesta, partiendo de una infraestructura de red LAN/WAN que funcionará como base para la comunicación de máquinas virtuales.

NAS: Almacenamiento Conectado en Red o NAS (Network Attached Storage).

APK: Es un archivo extensión .apk (Android Application Package, significado en español: Paquete de Aplicación Android) es un paquete para el sistema operativo Android.

Marco Metodológico

Demuestre la posibilidad de implementar una red SDN-SDWAN en los dispositivos disponibles. Se proporciona una topología simple que se muestra en la Figura 170, que sustenta la comunicación de equipos y gestión de servicios en varias redes virtuales.

Para una comunicación adecuada, la tabla de enrutamiento (Tabla 9) debe configurarse correctamente en la VXLAN, solo así el enrutamiento del tráfico a otra dirección de red se enviará correctamente. Utilizamos un nodo conmutador ZeroTier para hacer un uso más eficiente de la potencia informática que brinda la red superpuesta a la red LAN a la que está conectado el Host.

Presentamos un escenario de Servicios con Multimedia, Nube Personal y Comunicación centralizada para plataformas de IoT, para lograr esto debemos gestionar el repositorio de los servicios open source que vamos a probar, cierta data de configuración y directorios necesitan ser cargados en el controlador (Raspberry Pi), en anexos capítulo IX encontrará la guía de instalación de los servicios, allí encontrará de forma detallada los pasos en cuanto a la implementación de cada servicio cargado en este proyecto.

Luego tenemos como sabemos las nubes públicas y las nubes privadas, pues podemos crear nuestra propia nube privada para poder acceder desde cualquier sitio, en consecuencia, podemos replicar cualquier dato de manera automática, podemos hacer un backup de nuestros datos soportados en las nubes públicas, o simplemente tener la privacidad de nuestros datos con nuestra nube propia.

Finalizando esta sección se proporciona un escenario de validación de conectividad de canal óptico de extremo a extremo entre equipos y entre redes.

Desde la interfaz de administración o de cualquier equipo conectado a la red VXLAN, podemos tomar el control de la Red Virtual y gestionar base de Datos como un NAS Server, para lo cual deberá ingresar la dirección IP por defecto a la cual esté conectada la Raspberry Pi en la red, garantizando así la interoperabilidad entre sistemas basados en Linux y sistemas basados en Windows, en redes LAN, WAN por Software.

Tablas de Direccionamiento IP

Tabla 9. Gestión de Enrutamiento de las Redes – Práctica 2.

Nombre	Valor	Denominado
ISP 1 – AP-Verizon	192.168.43.0/24	LAN 1
ISP 2 – CLARO AVENDANO	192.168.200.0/24	LAN 2
ISP 3 – UPS-GYE	172.18.142.0	LAN 3
RED VXLAN		
Ruta gestionada de Red ZeroTier	10.242.0.0/16	ZT_VXLAN
Rango de asignación DHCP de ZeroTier	10.242.0.2 hasta 10.242.255.254	ZT_DHCP
Dirección IP del Puente ZeroTier	10.242.10.40/16(o use DHCP)	BR_ADDR
ID de Red ZeroTier (16 dígitos)	c7c8172af1c32100	NETWORK_ID
ID del Nodo de Equipo (10 dígitos)	1########0	NODO_ID
Nombre de la interfaz de Red ZeroTier	zt5u451c7y	ZT_IF

Fuente: Propia del autor.

Tabla 10. Direccionamiento IP de los Equipos Conectados en Red – Práctica 2.

DISPOSITIVOS	XLAN_ DHCP	ZT_VXLAN/DHCP	OPERADOR
EQUIPO ANDROID J7	192.168.43.1	10.242.234.242	ISP 1
TABLET ADMIN	192.168.200.32		ISP 2
PC DESKTOP V7AVHRA	192.168.200.29	10.242.100.10	ISP 2
PORTATIL YILDER	192.168.43.222	10.242.3.3	ISP 1
GYECLABTELPC08	172.18.142.9	10.242.65.88	ISP 3
GYECLABTELPC11	172.18.142.10	10.242.144.81	ISP 3
IP LAN PUENTE ZT – R.PI	192.168.200.15	10.242.10.40	ISP 2
IP WLAN PUENTE ZT – R.PI	192.168.200.36	10.242.10.40	ISP 2

Fuente: Propia del autor.

MARCO PROCEDIMENTAL

Análisis del Proyecto como Servidor NAS

1. Topología de Comunicación de Equipos en Práctica 2.

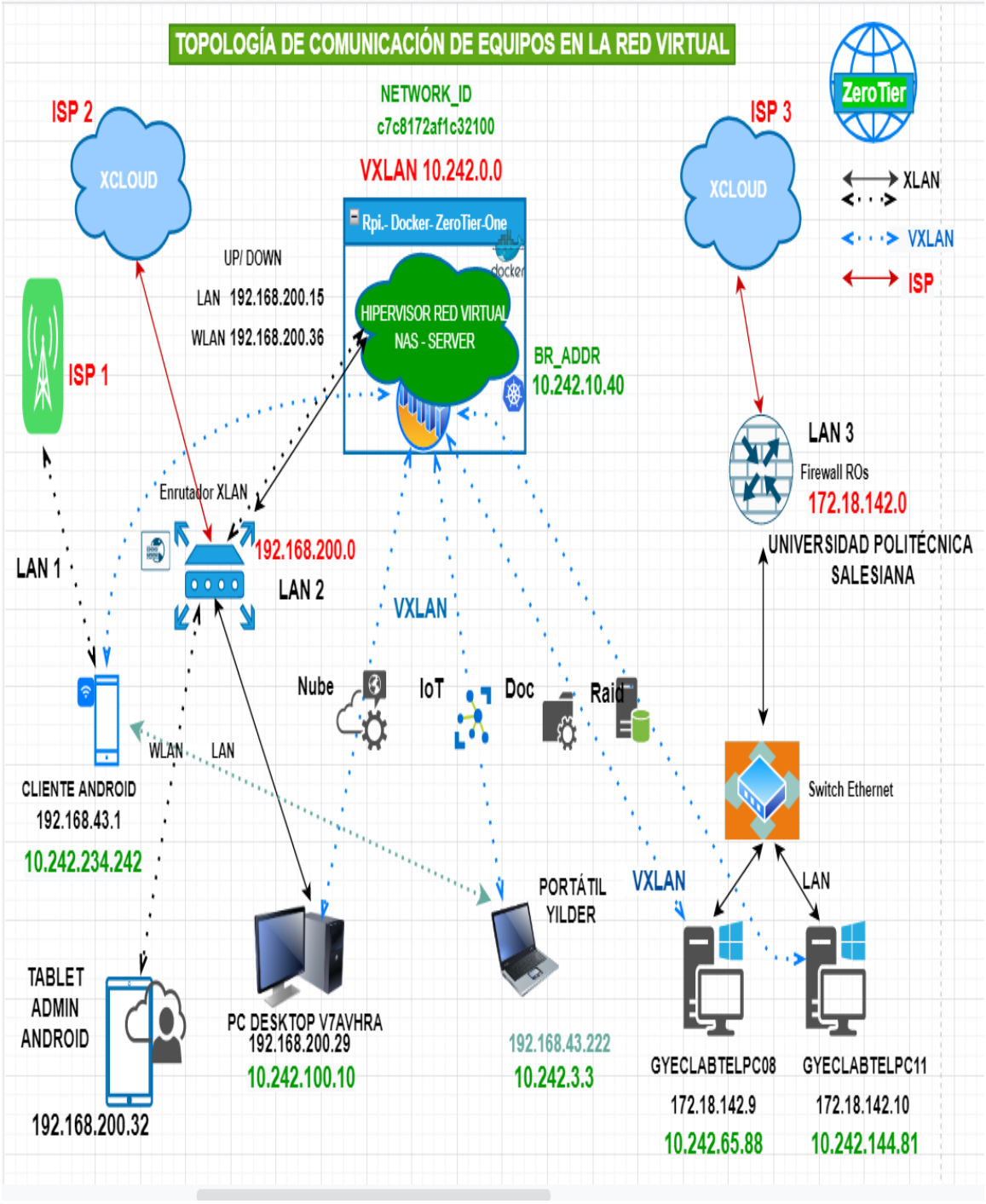


Figura 170. Topología de Red Práctica 2, (Print Screen).
Fuente: Elaboración propia del Autor en https://app.diagrams.net/

Gestión de Archivos en la Red Virtual

2. Integrar equipos a la red Virtual de cual requerimos obtener servicio de comunicación. Verificar que cada equipo tenga dirección IP desde ZT_DHCP, A continuación, ilustramos la comunicación en NODO_ID ca65ad5f46 » Equipo PC GYECLABTELPC11. Para cargar los equipos a la red siga las instrucciones del subcapítulo 3.4.

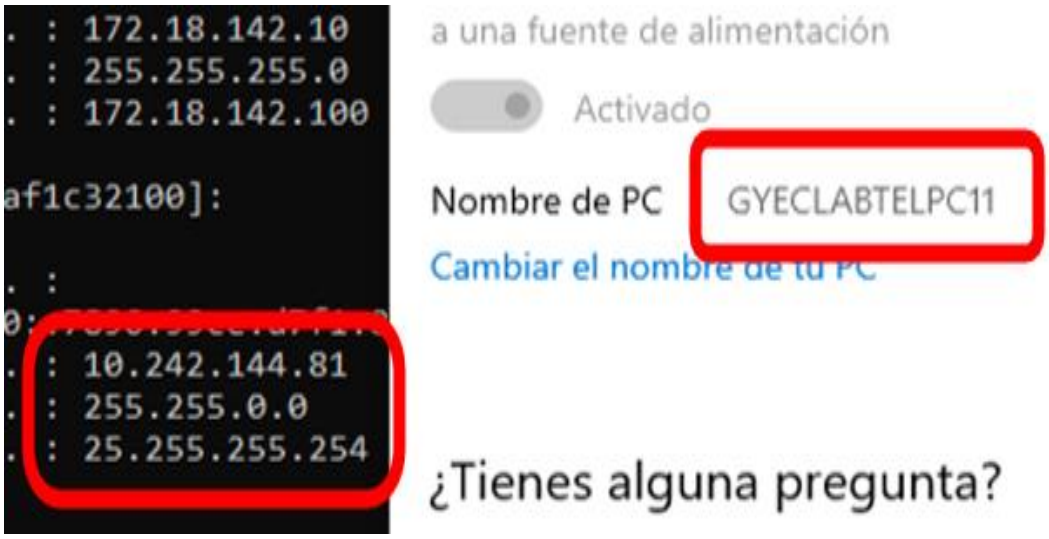


Figura 171. Comunicación entre VXLAN y Equipo PC GYECLABTELPC11.
Fuente: Propia del autor.

3. Ilustramos la comunicación en NODO_ID 91d1c68da7 » PORTATIL.

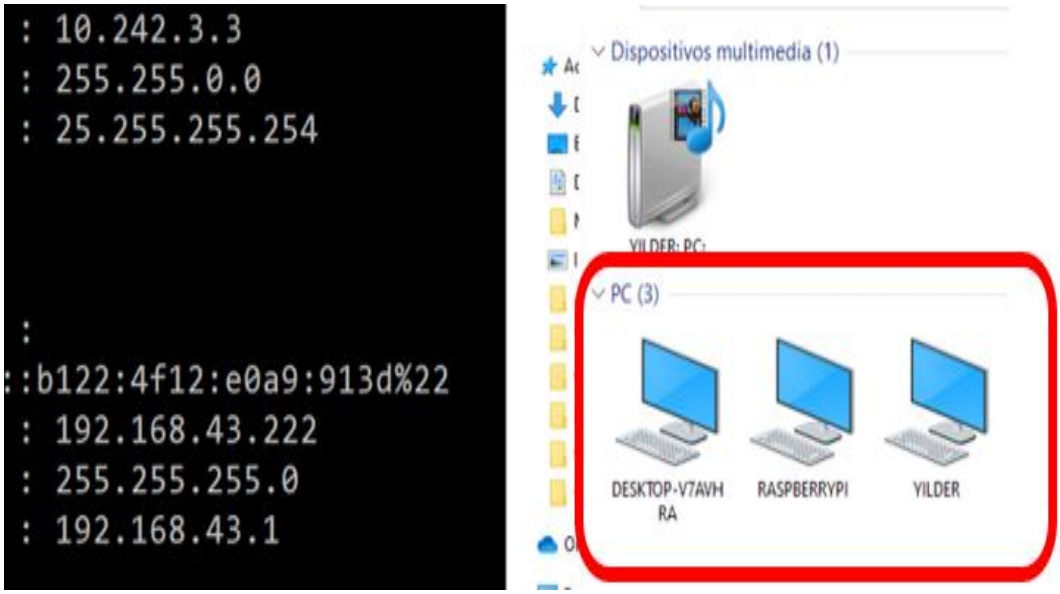


Figura 172. Comunicación entre VXLAN y Equipo PORTÁTIL YILDER
Fuente: Propia del autor.

4. Escenario de comunicación de equipos en VXLAN, podemos verificar esto en mis sitios de red; usuarios y grupos tienen permiso de acceso a este recurso siempre que estén en red ZT_VXLAN, equipos físicos situados en ubicaciones remota de LAN 1, LAN 2, LAN 3.

Figura 173. Escenario de Comunicación de Equipos Conectados en Red VXLAN.
Fuente: Propia del autor.

5. Recursos compartidos en VXLAN (LAN 2), acceso desde el equipo NODO_ID 91d1c68da7 » YILDER. IP 10.242.3.3 conectado en LAN1.

Figura 174. Recursos Compartidos por VXLAN – Acceso desde PORTÁTIL YILDER.
Fuente: Propia del autor.

6. Recursos compartidos por VXLAN (LAN 2), acceso desde el equipo NODO_ID e804b2df78 » DESKTOP - V7AVHR. IP 10.242.100.10

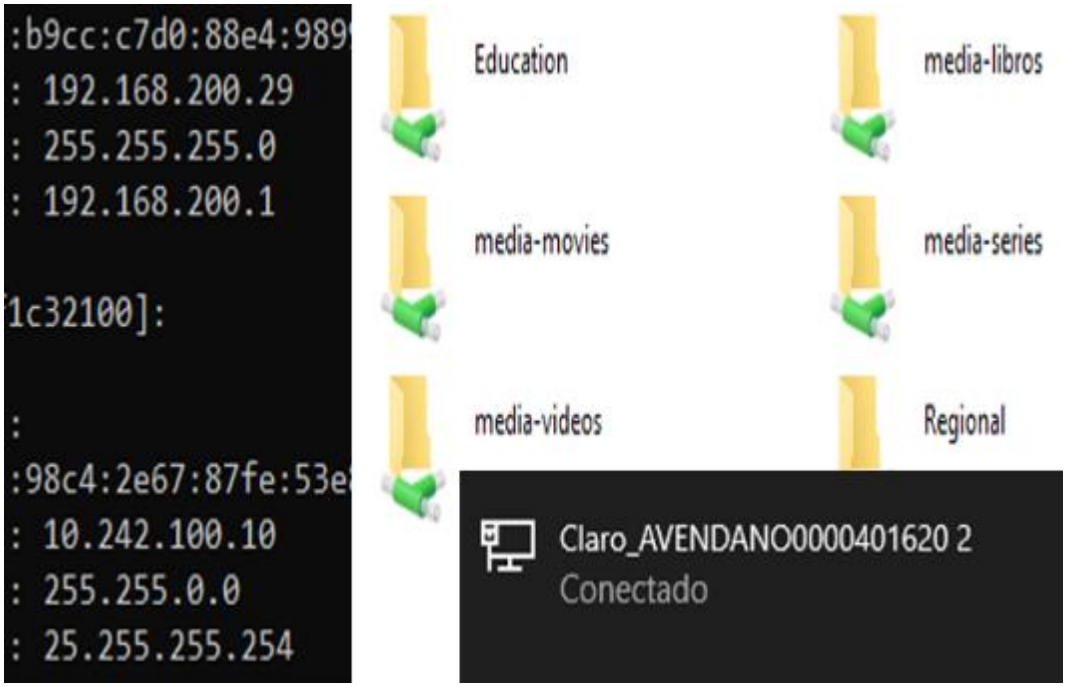


Figura 175. Recursos Compartidos por VXLAN – Acceso desde PC-V7AVHR.
Fuente: Propia del autor.

7. Recursos compartidos por VXLAN (LAN 2), acceso desde el equipo NODO_ID c542247ae8 » GYECLABTELPC08. IP 10.242.65.88, conectado en LAN 3 – UPS ESTUDIANTES.

Figura 176. Recursos Compartidos por VXLAN – Acceso desde GYECLABTELPC08
Fuente: Propia del autor.

8. Respaldo de (5GB) en base de datos desde Equipo DESKTOP-V7AVHR hacia el Host del Servidor NAS, carpeta de destino “media”.

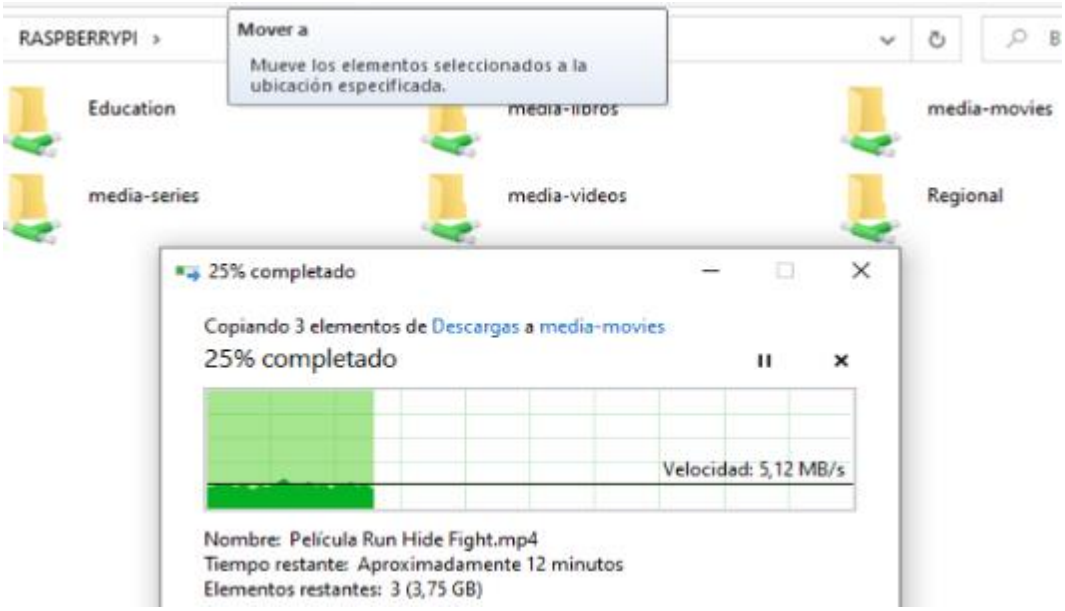


Figura 177. Copia de 5 GB de Datos del Equipo PC-V7AVHR, hacia Servidor NAS.
Fuente: Propia del autor.

9. Copia de base de datos (30GB), desde el Equipo NODO_ID 91d1c68da7 » PORTÁTIL YILDER hacia el Host del Servidor NAS, carpeta destino “Regional”. (Conectado en LAN 1).

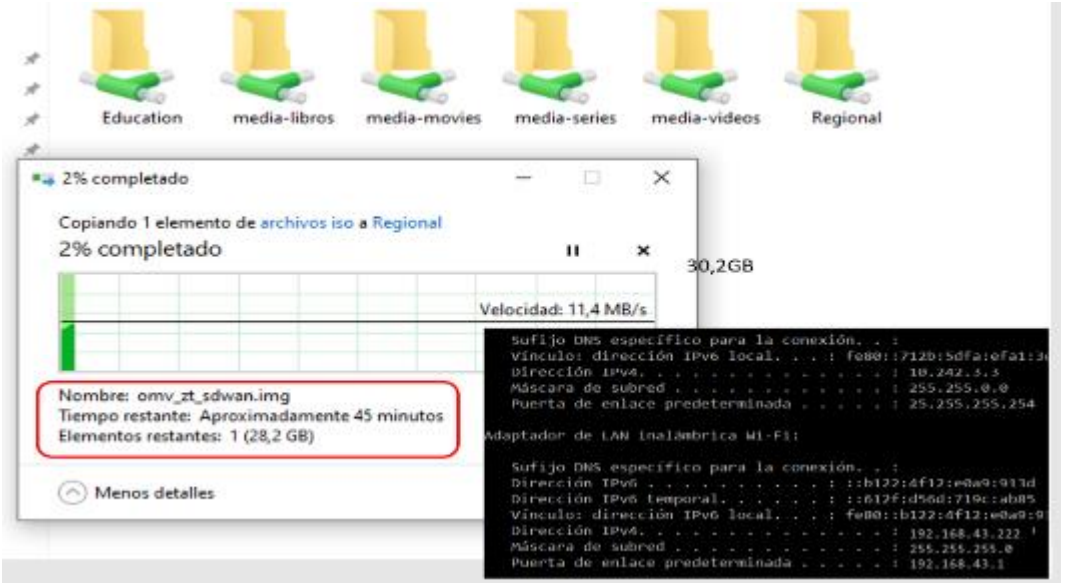


Figura 178. Copia de 30,2GB de Datos del Equipo PORTÁTIL, hacia Servidor NAS.
Fuente: Propia del autor.

10. Podremos acceder al OpenMediaVault desde cualquier dispositivo conectado en la red LAN o en la red superpuesta VXLAN, tome el control de la Red y de la base de datos, dirigiéndose al local Host del Servidor (192.168.200.15), o BR_ADDR (10.242.10.40).

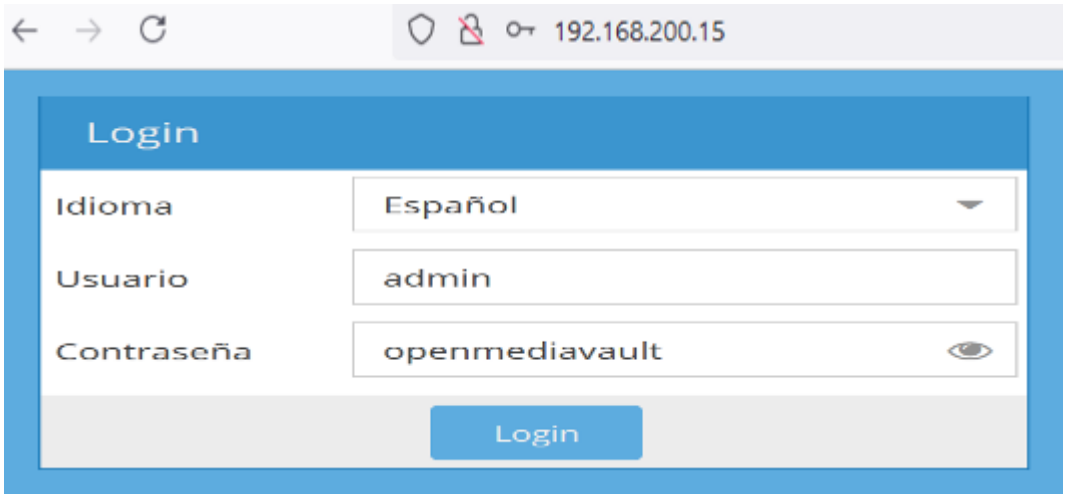


Figura 179. GUI de OpenMediaVault, (Print Screen).
Fuente: Propia del autor.

11. La GUI de OpenMediaVault nos permite interactuar con un amplio portafolio de opciones para poner en práctica múltiples opciones en operación NAS, Interfaz vista en equipo GYECLABTELPC08.

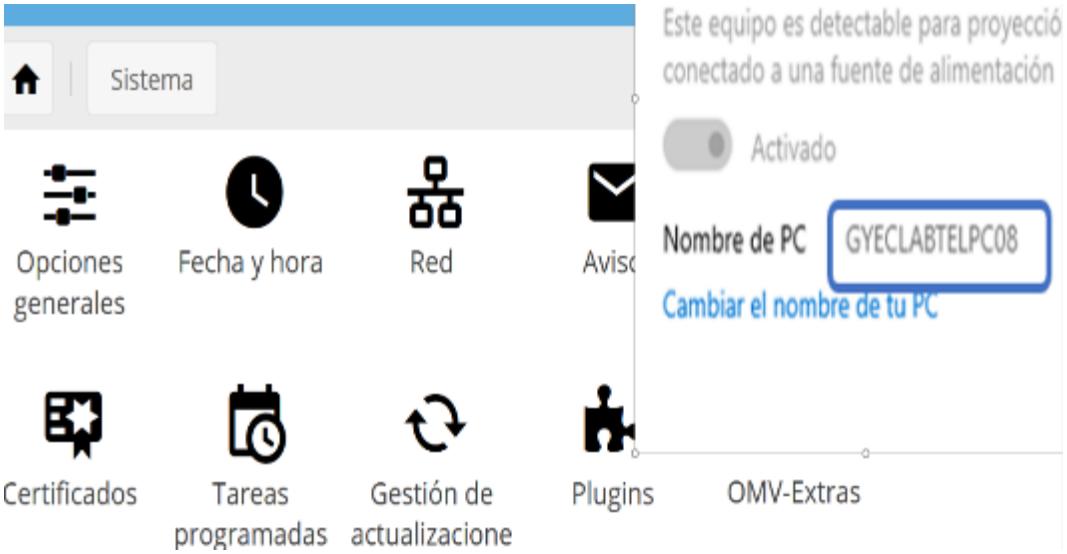


Figura 180. Portafolio de Opciones en OpenMediaVault, (Print Screen).
Fuente: Propia del autor.

12. Realizamos un breve repaso de las prestaciones que nos brinda el software, como se muestra en la figura tenemos los servicios de compatibilidad con archivos NFS, FTP, Rsync server, Samba y SSH, lo que nos garantiza un amplio abanico para la gestión de archivos con múltiples plataformas, interfaz vista en equipo Portátil Yilder.

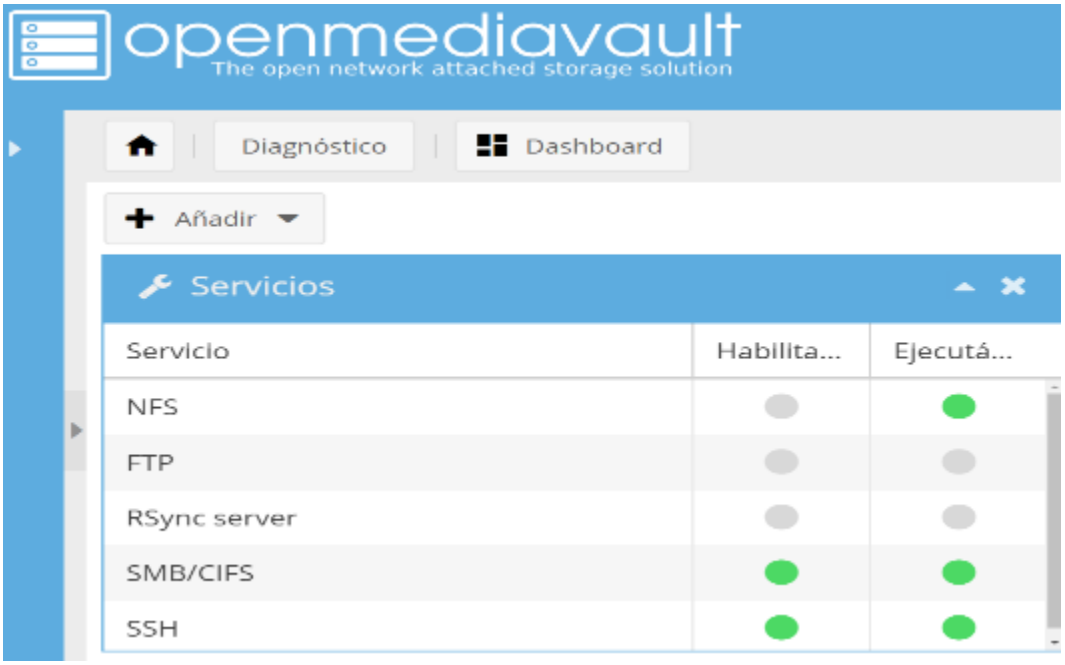


Figura 181. Archivos Compatibles en OpenMediaVault, (Print Screen).
Fuente: Propia del autor.

13. El Almacén de OpenMediaVault tiene gestión de administración con todas las características que podríamos necesitar. Un sistema de administración web, un sistema de instalación y administración de paquetes y volúmenes con sistema S.M.A.R.T. Wake on lan, notificaciones por correo electrónico, un interesante sistema de extensiones o plugins que otorgan gran potencial.

Figura 182. Gestión de Almacenamiento OpenMediaVault desde VXLAN.
Fuente: Propia del autor.

14. Escenario de administración del Almacenamiento NAS desde diferentes puntos de acceso en red del ISP o en la red Virtual.

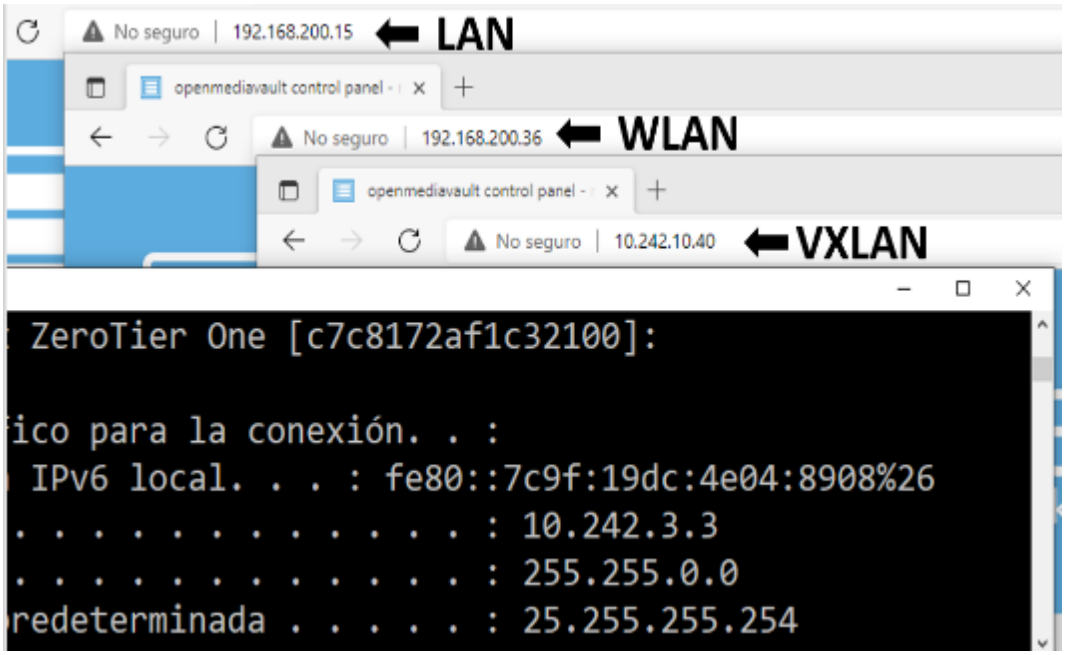


Figura 183. Acceso a OMV desde Sitios Remotos de la Red, (Print Screen).
Fuente: Propia del autor.

Análisis del Proyecto como Servidor NAS

Al implementar esta capacidad es necesario hacer énfasis en que no existe una sola forma de hacerlo, para implementar el servidor nos valimos de las prestaciones que tienen los programas open source con que se construyó el Proyecto, desde agregar y montar una unidad de destino en la red con OpenMediaVault y un SSD, hasta jugar con la reglas de flujo iptables de la red en un escenario de contenedores de acuerdo con la sección denominada A Basic Data Drive.

Cuando se formateó, el disco SSD utilizado en este proyecto, se etiquetó para indicar su función en carpetas para diferentes directorios de servicios y aplicaciones. Esta es una buena práctica que ayudará a los nuevos usuarios a identificar fácilmente las unidades y evitar errores de administración. Se pueden utilizar unidades de tamaño diferente, siempre que la unidad de destino sea lo suficientemente grande para contener los datos de la unidad de origen.

Es importante que, si estás instalando paquetes en OpenMediaVault, cambies el puerto 80 y si lo usas, también el 443, por otros. Ahora que hemos instalado todos los paquetes necesarios (Anexo IX), veremos a continuación la capacidad de nuestro Proyecto como servidor NAS en tres prestaciones de servicios cargados en contenedores como se ha indicado antes, sobre una nube personal, una plantilla de comunicaciones para IoT y un centro multimedia.

15. La imagen a continuación muestra la ruta absoluta de los directorios, es la ruta reconocida de las carpetas donde se guardará la data de la implementación de paquetes, directorios de aplicaciones o contenedores de docker. Mas detalle en el subcapítulo 3.3.3.5

Ruta relativa	Ruta absoluta
Cloud/	/srv/dev-disk-by-uuid-d539ecf7-afb0-4924-9efe-252000754eb3/Cloud
media-tv/	/srv/dev-disk-by-uuid-d539ecf7-afb0-4924-9efe-252000754eb3/media-
multimedia/	/srv/dev-disk-by-uuid-d539ecf7-afb0-4924-9efe-252000754eb3/multim

Figura 184. Ruta Absoluta de Directorios y Carpeta de Servicios Compartidos.
Fuente: Propia del autor.

16. En la interfaz GUI del equipo ADMIN accedemos al Administrador de Contenedores docker, en un portal WEB nos dirigimos a la ip local o virtual del equipo.

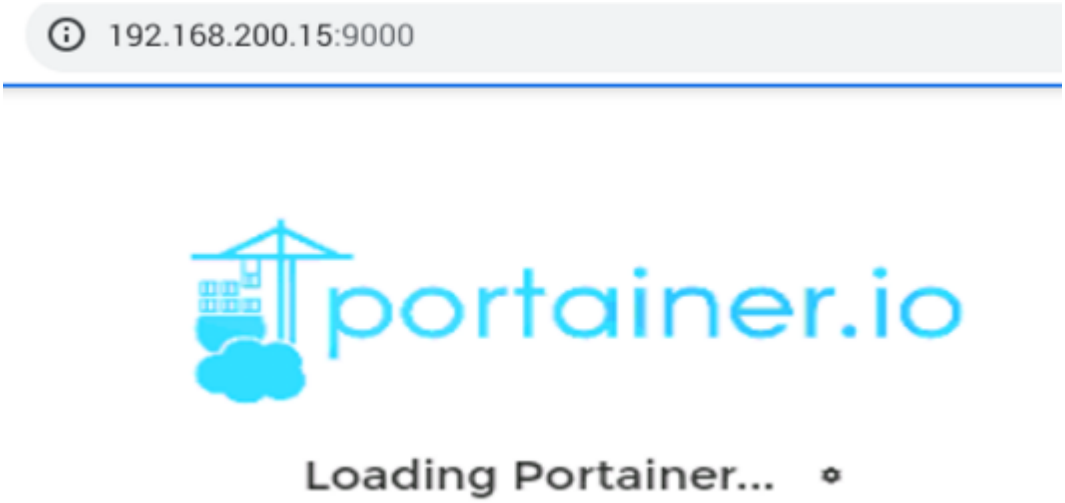


Figura 185. Portal de Administración de Contenedores Portainer, (Print Screen).
Fuente: Propia del autor.

17. Administraremos los contenedores para nuestros clientes en Docker. Nuestros clientes son los servicios que hemos cargado como anexos como ya se ha mencionado anteriormente. Administrar los contenedores del host es un poco complicado si lo hacemos por línea de comandos. Al buscar una herramienta todo en uno, descubrimos Portainer y decidimos implementarlo en nuestro proyecto. Encontrará una guía detallada en el apartado 3.3.5 de la configuración de la Red Virtual.

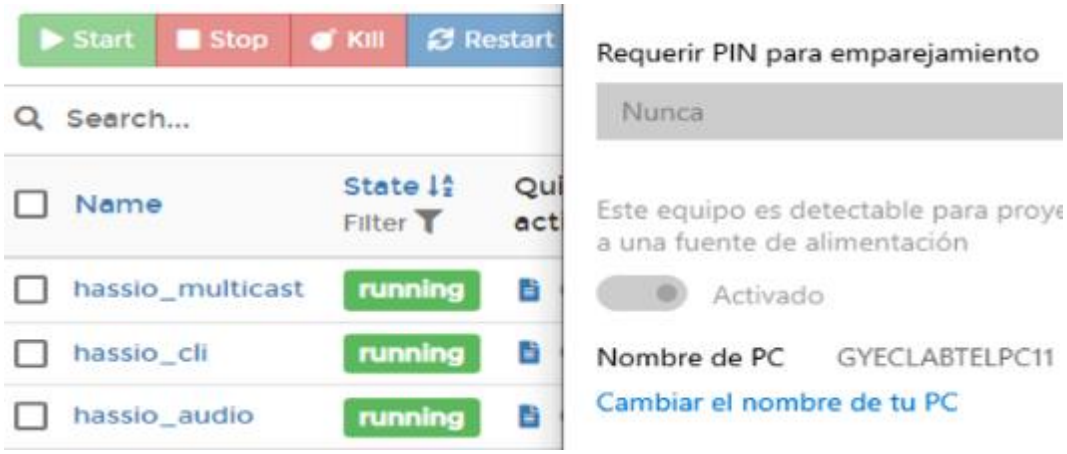


Figura 186. Sistema de Contenedores Docker, (Print Screen).
Fuente: Propia del autor.

18. Escenario de administración de Portainer desde equipos remotos conectados a la Red Virtual, red LAN en 192.168.200.36 o virtual 10.242.10.40.

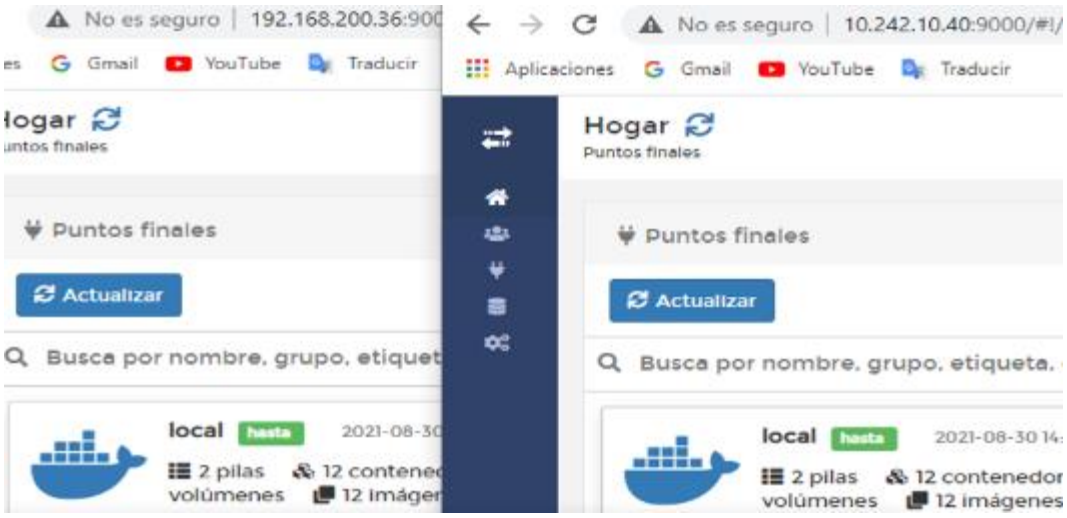


Figura 187. Acceso a Portainer desde Sitios Remotos de la Red, (Print Screen).
Fuente: Propia del autor.

Nube Personal con NextCloud

La guía detallada de instalación de NextCloud la encontrará en el anexo 9.5

19. Desde el interfaz administrador ingresar al Portal de acceso de la nube personal, en el navegador direccionamos el buscador a la ip local o virtual del Servidor en el puerto 8080. Puede acceder desde los clientes de escritorio o móviles que estén en la red.

Figura 188. Inicio de Sesión en la Nube Personal. (Print Screen).
Fuente: Propia del autor.

20. El servicio de Nube Personal nos permite generar copias de seguridad de varios dispositivos creando automáticamente la funcionalidad de nuestra propia nube privada. Con ello, podremos sincronizar copias de seguridad de archivos en varios dispositivos, pudiendo acceder desde cualquiera de ellos y en cualquier lugar.

Figura 189. Sincronización de Cuentas Privadas con su Nube, (Print Screen).
Fuente: Propia del autor.

21. La sencilla interfaz web le permite compartir archivos con otros usuarios o permitir que otros carguen archivos en su nube personal y recibir notificaciones directamente en su teléfono y escritorio cuando comparte archivos con usted.

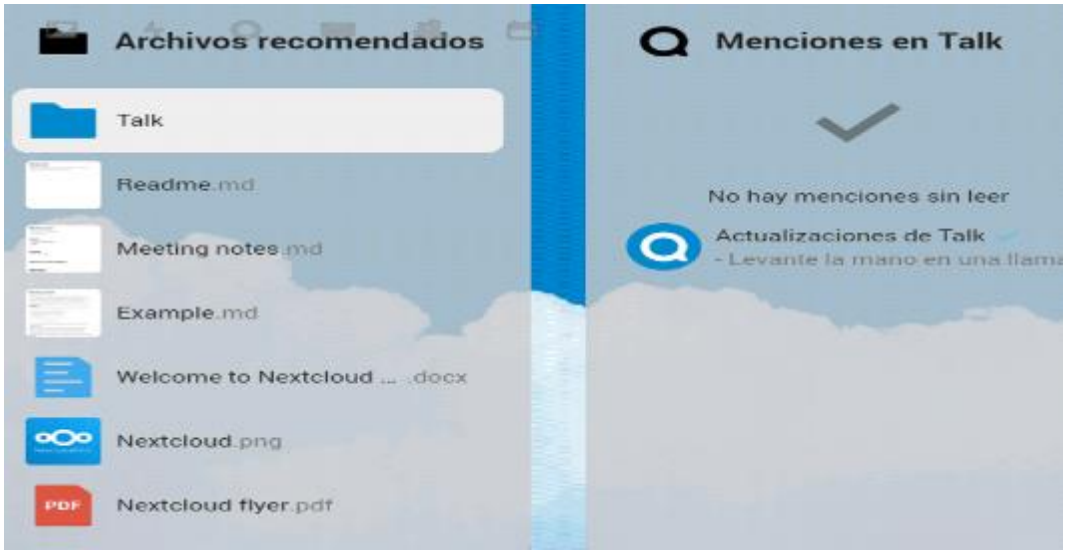


Figura 190. Recursos Compartidos entre Usuario Externos, (Print Screen).
Fuente: Propia del autor.

22. La capacidad de Gestión NextCloud nos brinda acceso a los datos donde sea que se encuentren, así como también puede acceder a archivos almacenados con una amplia variedad de proveedores de servicios en la nube populares como Amazon, Google y Dropbox, pero también puede acceder a ellos utilizando protocolos estándar como NFS, SAMBA, (S) FTP.

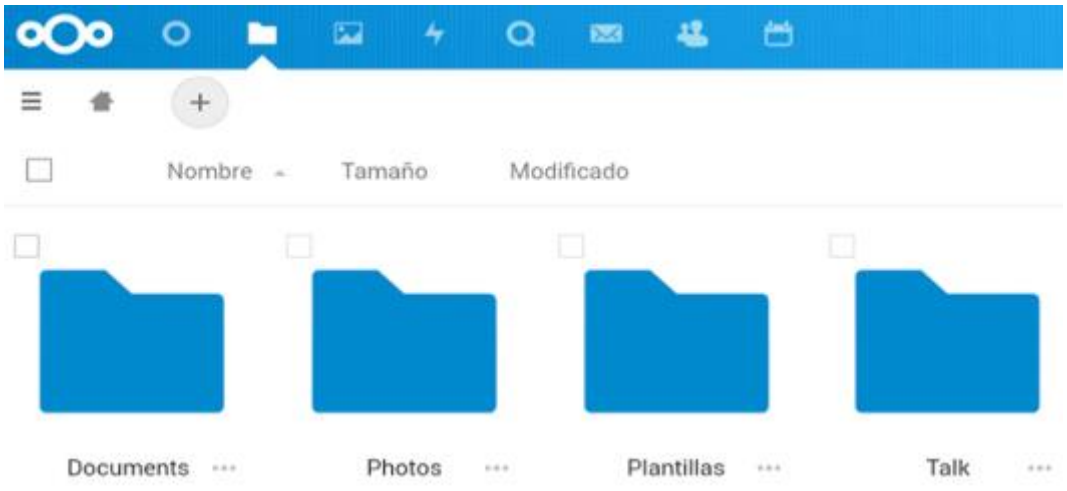


Figura 191. Gestión de Archivos con Nubes Privadas, (Print Screen).
Fuente: Propia del autor.

La guía detallada de instalación Home Assistant la encontrará en el anexo 9.5

23. Desde el interfaz administrador ingresar al portal de acceso de Home Assistant, en el navegador direccionamos el buscador a la ip local o virtual del Servidor en el puerto 8123. Puede acceder desde los clientes de escritorio o móviles que estén en red.

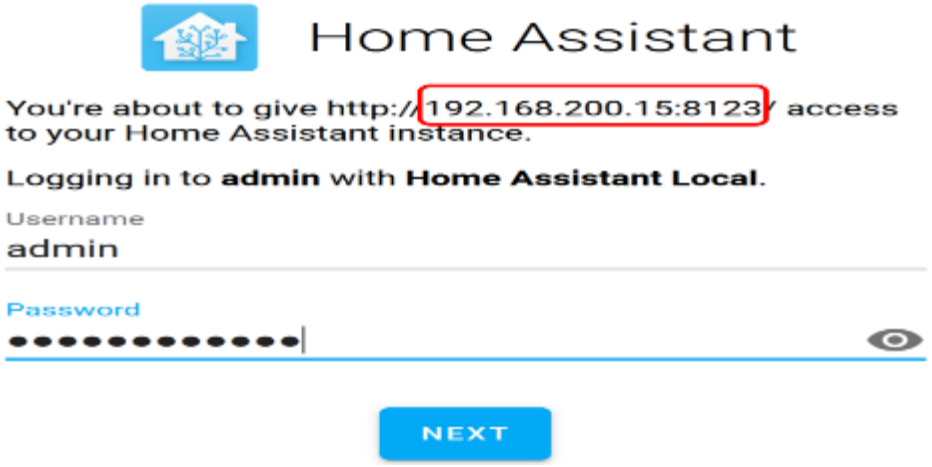


Figura 192. Inicio de Sesión en Home Assistant, (Print Screen).
Fuente: Propia del autor.

24. Home Assistant es una plataforma abierta, por lo que la gestión no se limita a un hardware específico. Cualquier hardware de monitoreo y de función autómata que se integre con Home Assistant se puede utilizar como fuente de datos.

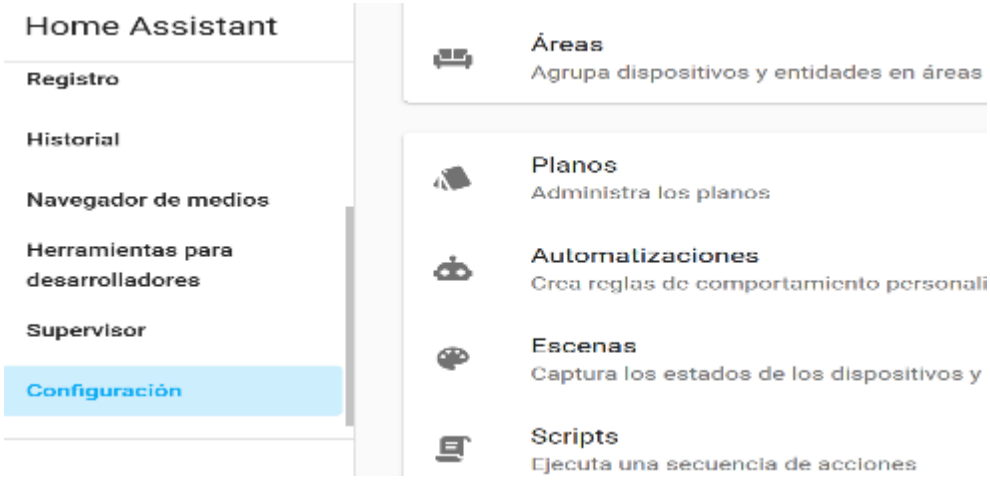


Figura 193. Panel de Control Home Asistant, (Print Screen).
Fuente: Propia del autor.

25. En la pantalla resumen, podrá ver el estado de operación y monitoreo de su entorno y el de aplicaciones instaladas.

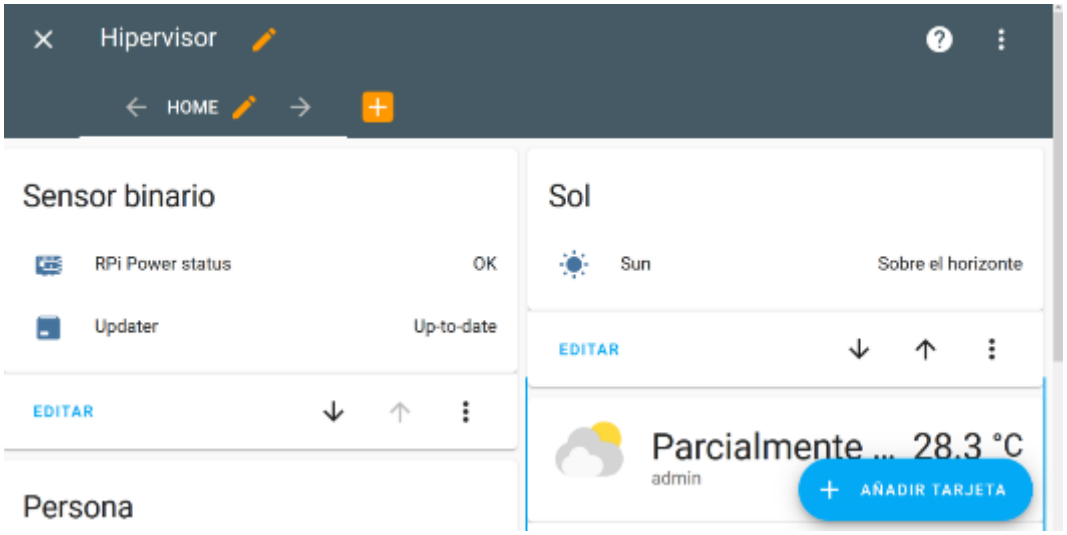


Figura 194. Interfaz de Monitoreo del Entorno IoT, (Print Screen).
Fuente: Propia del autor.

26. Como parte del proceso de incorporación predeterminado, Home Assistant puede detectar su ubicación a partir de la geolocalización de la dirección IP, Puede cargar integraciones las cuales automatizará el control de la unidad de temperatura y una zona horaria en función de esta ubicación.

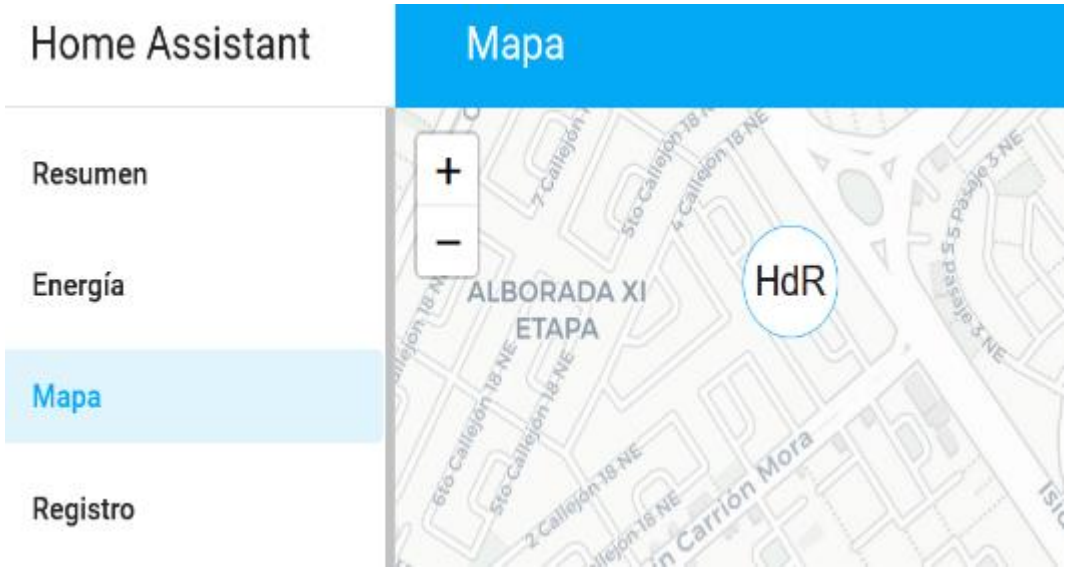


Figura 195. Detección de Ubicación del Servidor, (Print Screen).
Fuente: Propia del autor.

27. Home Assistant podrá descubrir automáticamente muchos dispositivos y servicios disponibles en su red. La mayor parte de la integración se puede configurar completamente a través de la interfaz de usuario.

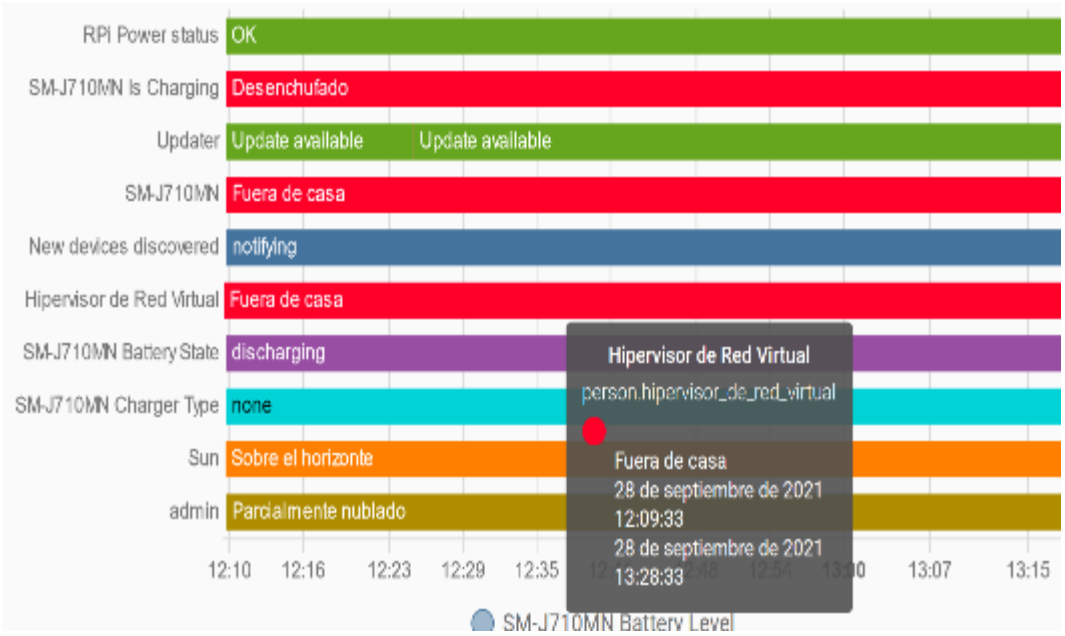


Figura 196. Integración de Dispositivos al Entorno IoT – Equipo Android J710MN.
Fuente: Propia del autor.

28. Los planos de automatización son automatizaciones prefabricadas que puede agregar fácilmente a su instancia de Home Assistant. Cada plano se puede agregar tantas veces como desee.

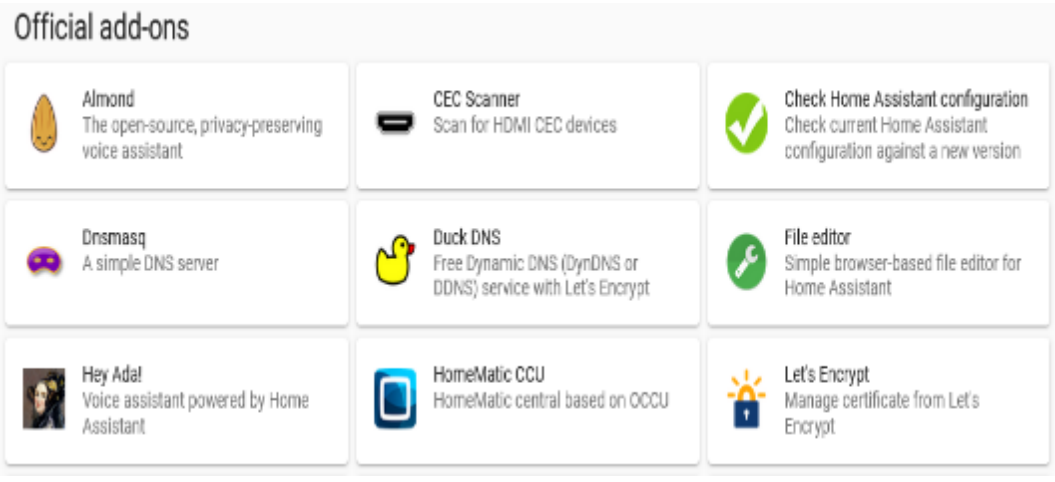


Figura 197. Plugin Disponible para IoT en Home Assistant, (Print Screen).
Fuente: Propia del autor.

29. Accesos remotos a Home Assistant desde la red LAN y la red Virtual.

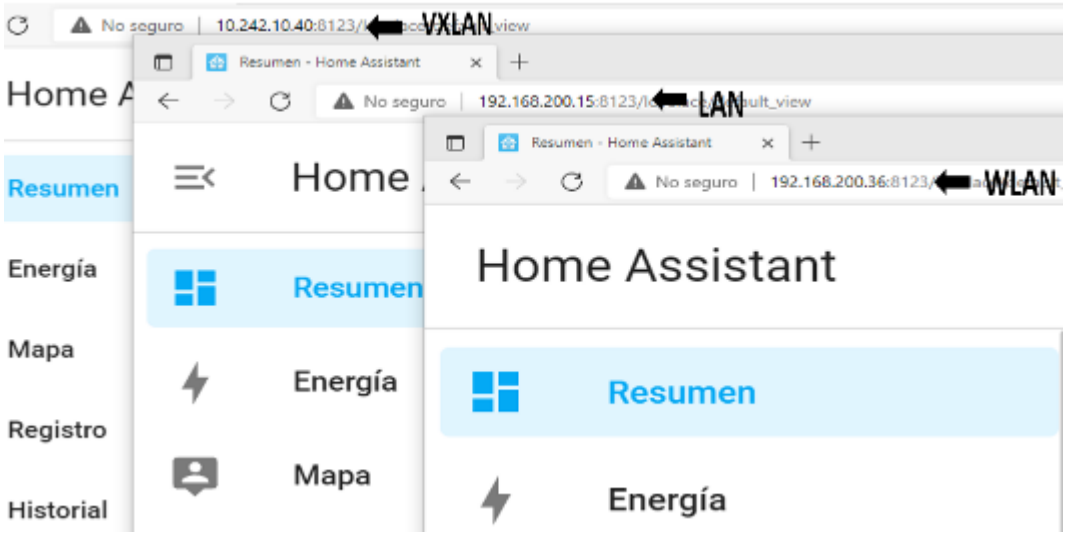


Figura 198. Acceso Remoto a Home Assistant, (Print Screen).
Fuente: Propia del autor.

30. Plantilla de aplicación para la Asistencia de la red eléctrica del entorno, desde Equipo ID 91d1c68da7 » PC YILDER (LAN 1).

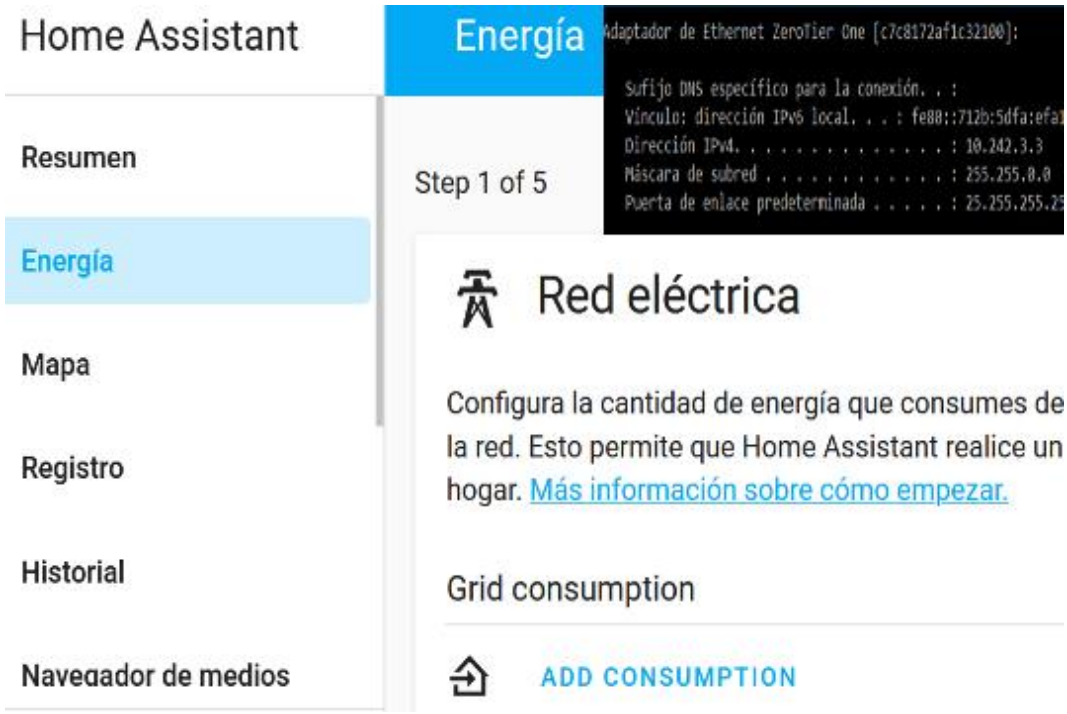


Figura 199. Aplicación de IoT – Asistencia de la Red Eléctrica del Entorno.
Fuente: Propia del autor.

Centro Multimedia con Jellyfin

La guía detallada de instalación de Jellyfin la encontrará en el anexo 9.5

31. Desde el interfaz administrador ingresar al portal de acceso de Jellyfin, en el navegador direccionamos el buscador a la ip local o virtual del Servidor en el puerto 8096. Puede acceder desde los clientes de escritorio o móviles que estén en red.

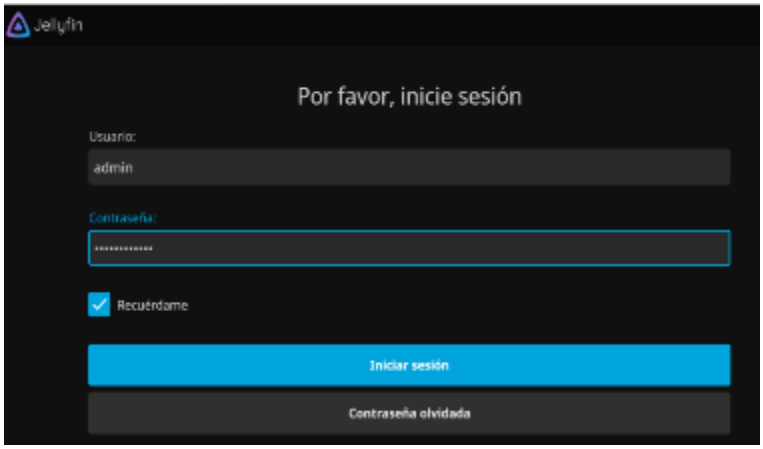


Figura 200. Portal de Inicio de Sesión del Servidor de Medios, (Print Screen).
Fuente: Propia del autor.

32. Los contenidos cargados en el servidor estarán disponibles para acceder desde cualquier tipo de dispositivo, Jellyfin es un sistema de medios de software libre que le permite controlar la gestión y la transmisión de sus medios.

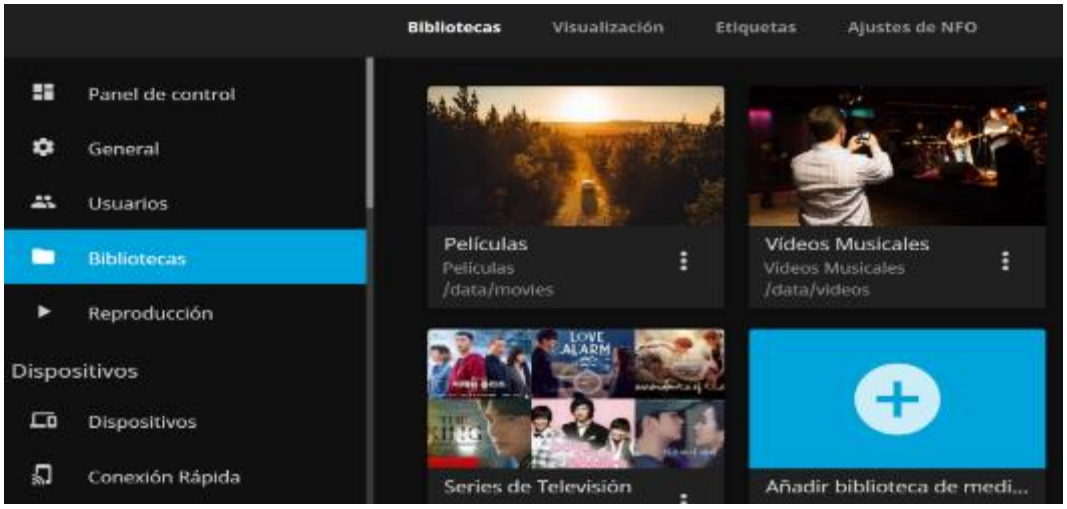


Figura 201. Contenido Multimedia Disponible en el Servidor, (Print Screen).
Fuente: Propia del autor.

33. Acceso al centro Multimedia desde equipo ca65ad5f46 » GYECLABTELPC11 .

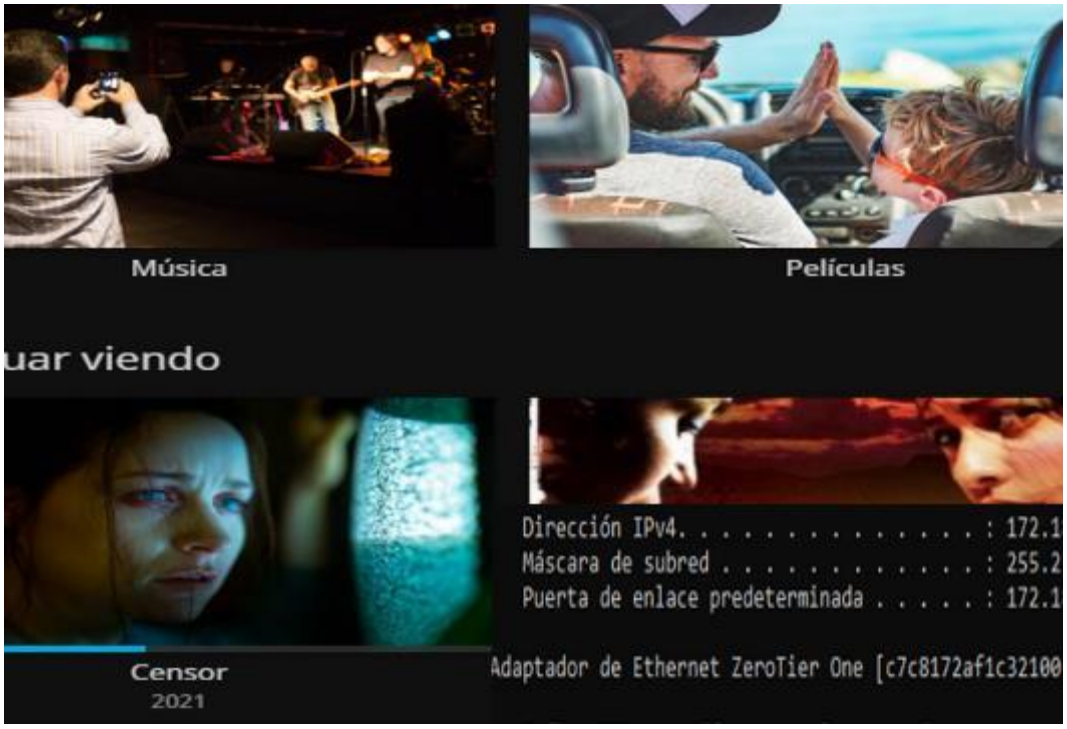


Figura 202. Acceso al Servidor Jellyfin desde el Equipo GYECLABTELPC11.
Fuente: Propia del autor.

34. Los ajustes de servicio estarán habilitados solo en el plano de administrador, donde tendrá que configurar la plantilla de aplicación de la app.

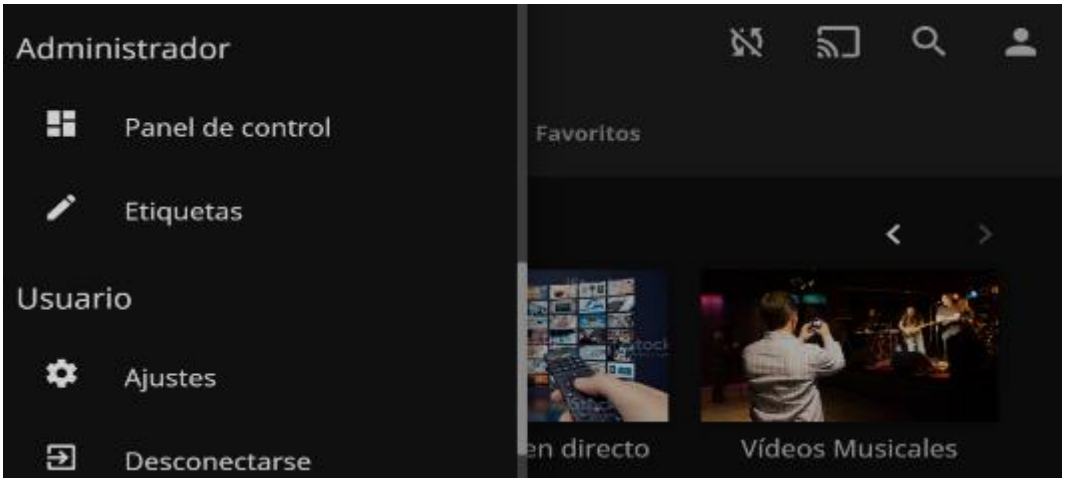


Figura 203. Plantilla para Configuración de Administrador, (Print Screen).
Fuente: Propia del autor.

35. La función principal del Panel de Control es permitir a un administrador controlar el escenario operacional de los dispositivos, vista gráfica de los registros del servidor en la siguiente imagen.

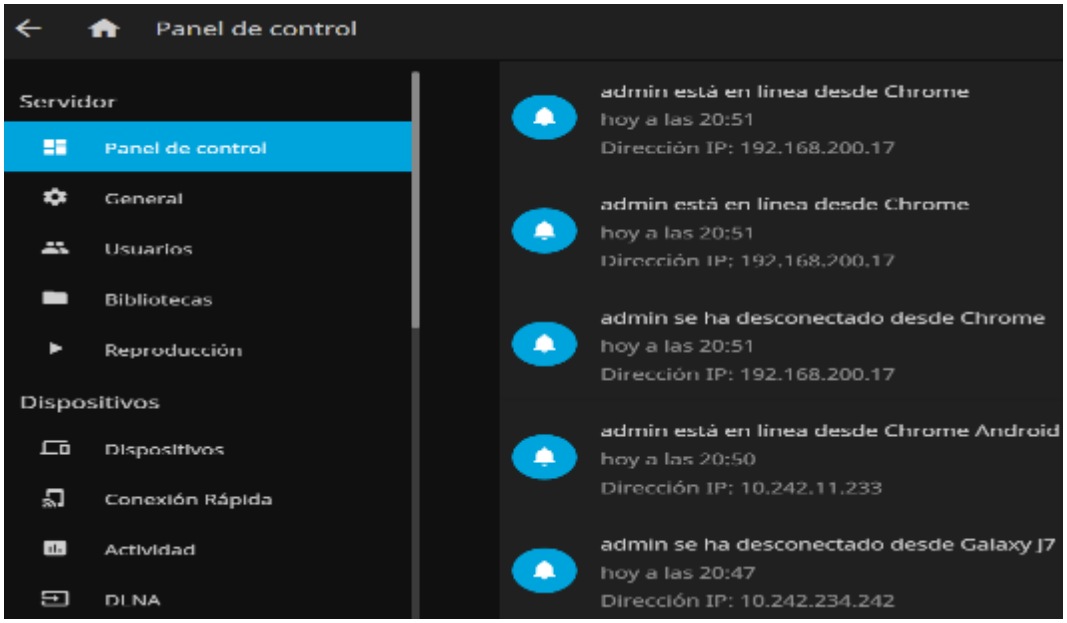


Figura 204. Servidor Jellyfin – Registros del Servidor, (Print Screen).
Fuente: Propia del autor.

36. El Servidor le permite configurar la administración en el escenario cliente- servidor, siendo clientes los usuarios que el administrador habilitó para tener acceso solo al centro de entretenimiento de Jellyfin no a la configuración.

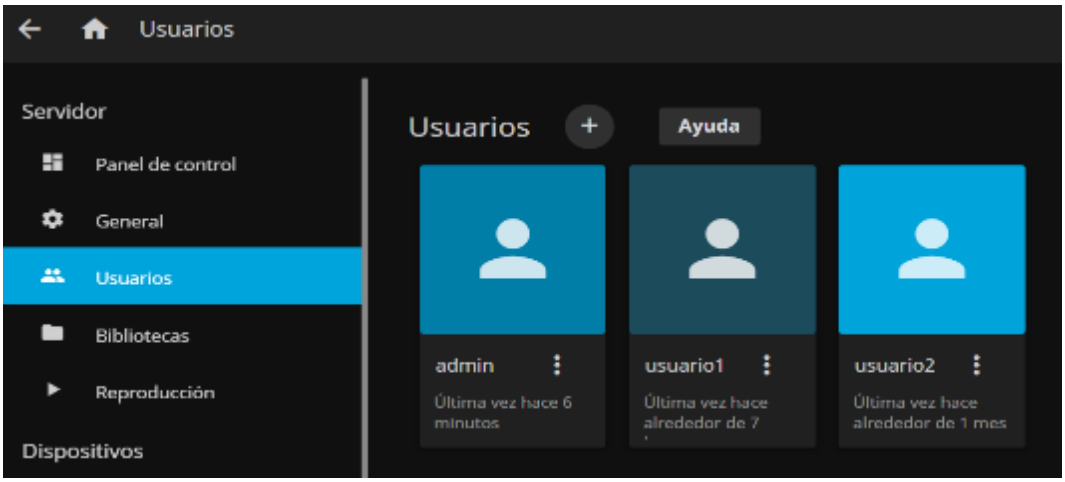


Figura 205. Usuarios del Servidor de Medios, (Print Screen).
Fuente: Propia del autor.

37. Los clientes conectados al servidor Jellyfin gozarán de los contenidos mas no a la configuración de la plataforma. El administrador habilitará el nivel de seguridad para los servicios.

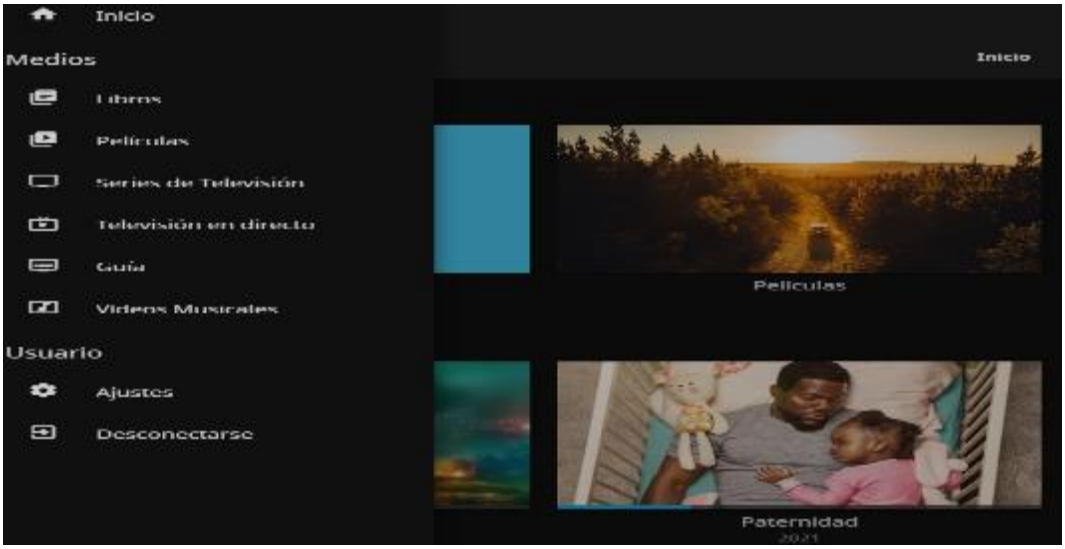


Figura 206. Contenido Multimedia para Clientes, (Print Screen).
Fuente: Propia del autor.

38. El Server de medios estará disponible para transmitir cualquier contenido multimedia entre diferentes tipos de dispositivos y en cualquier ubicación física que se encuentren, siempre que los dispositivos estén conectados a la red Virtual VXLAN.

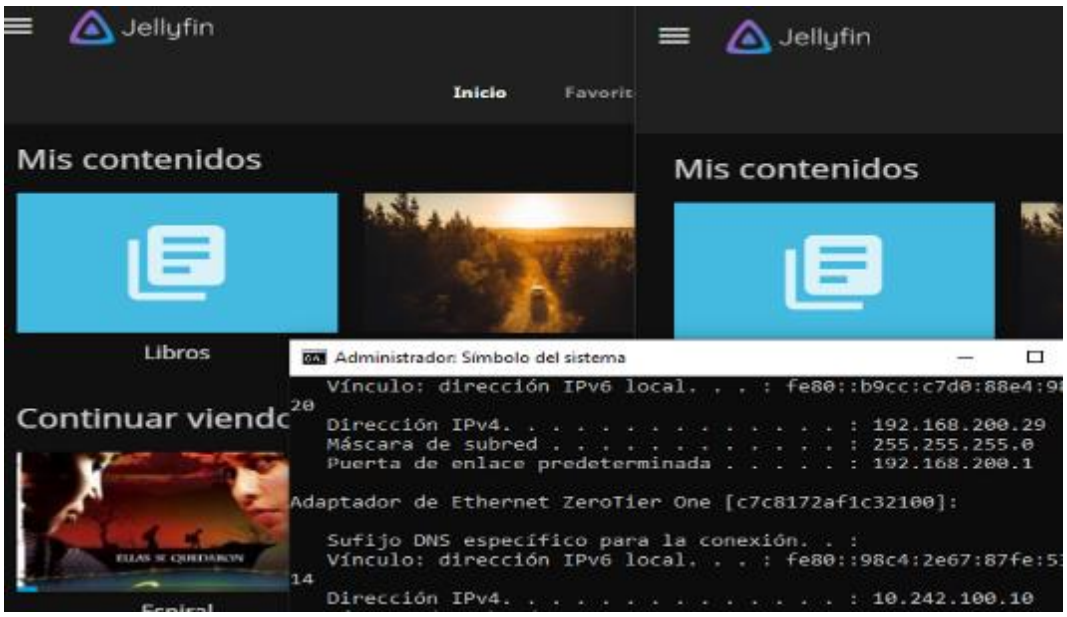


Figura 207. Contenido Disponible para Usuarios, (Print Screen).
Fuente: Propia del autor.

4.5.3. Práctica 3. Documentar los Resultados de la Conectividad y Tráfico de Datos entre Equipos Internos y Remotos de la Red

Marco Metodológico

Al familiarizarnos con la funcionalidad del componente de firewall en el controlador SDN, que filtra el tráfico de acuerdo con restricciones definidas, que pueden ser, la dirección IP de origen y destino, o la dirección MAC (Media Access Control), puerto de origen y destino, protocolo de transporte utilizado o el ID del conmutador para implementar la regla. Estas reglas se almacenan en el controlador y no se distribuyen al conmutador. Por lo tanto, si se inicia la comunicación de red y el conmutador no tiene una entrada para ella en su base de datos, se reenvía una consulta al controlador. Dependiendo del cumplimiento de las reglas definidas en el componente de firewall, la comunicación en el conmutador se habilita o deshabilita.

MARCO PROCEDIMENTAL

1. La tabla de enrutamiento que se muestra en la figura 208 reside en el kernel y especifica cómo se enrutan los paquetes en el host de servicio de la Red Virtual. (hscripts.com, 2016)

Destination: Indica la dirección IP de la red o host de destino.
Gateway: Puerta de enlace desde el cual se alcanza el host o red de destino.
Genmask: Indica el destino de la máscara de subred.
Flags: Indica el estado actual de ruta.

- U - La ruta está activa.
- H - El objetivo es un host.
- G - Utilizar puerta de enlace.

Iface: Indica el nombre de la interfaz.

```
pi@raspberrypi:~ $ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.200.1   0.0.0.0         UG    100    0        0 enxb827eb60f82
0.0.0.0         192.168.200.1   0.0.0.0         UG    100    0        0 enxb827eb60f82
0.0.0.0         192.168.200.1   0.0.0.0         UG    600    0        0 wlan0
10.242.0.0      0.0.0.0         255.255.0.0     U     0      0        0 zt5u45ic7r
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
172.18.0.0      0.0.0.0         255.255.0.0     U     0      0        0 br-11a890d2315
172.30.32.0     0.0.0.0         255.255.254.0   U     0      0        0 hassio
192.168.200.0   0.0.0.0         255.255.255.0   U     0      0        0 wlan0
192.168.200.0   0.0.0.0         255.255.255.0   U     0      0        0 enxb827eb60f82
192.168.200.0   0.0.0.0         255.255.255.0   U     100    0        0 enxb827eb60f82
192.168.200.1   0.0.0.0         255.255.255.255 UH    100    0        0 enxb827eb60f82
192.168.200.1   0.0.0.0         255.255.255.255 UH    600    0        0 wlan0
pi@raspberrypi:~ $
```

Figura 208. Tabla de Enrutamiento en Formato Numérico, (Print Screen).
Fuente: https://www.hscripts.com/es/tutoriales/linux-commands/route.html

2. Interfaces y direcciones de red en docker0 y LAN ethernet del host.

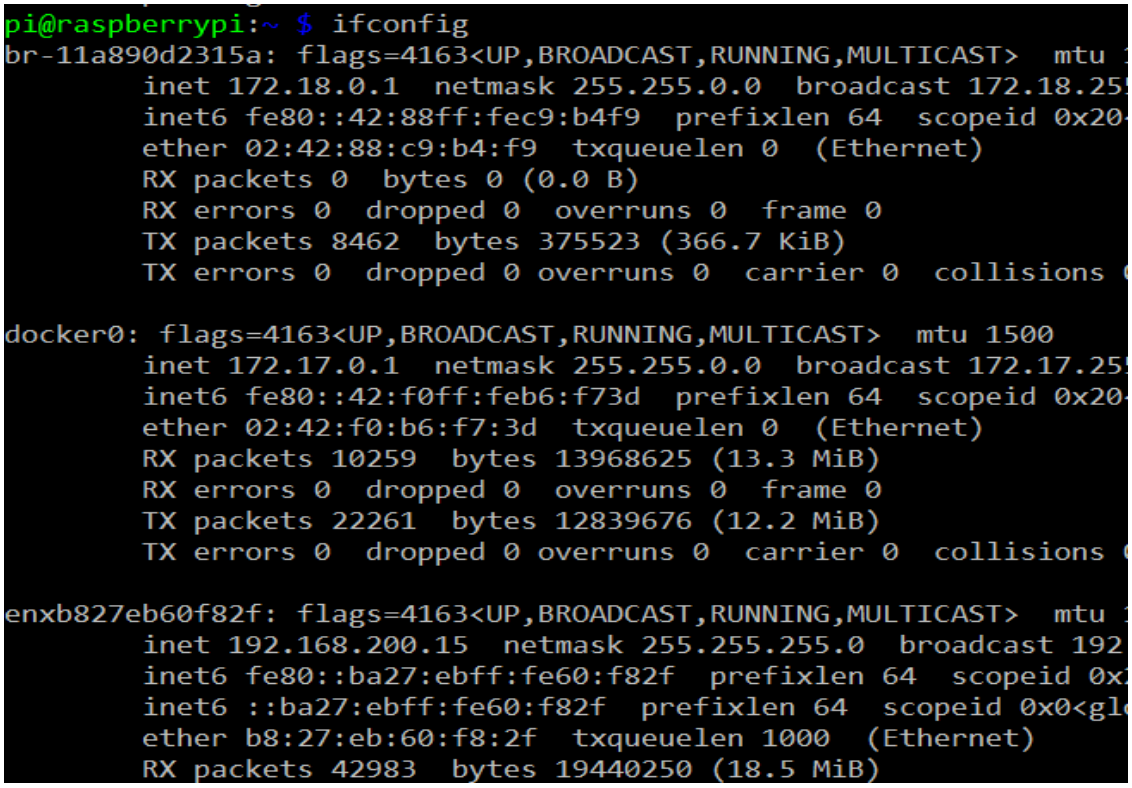


Figura 209. Interfaces y Direcciones de Red de Docker 0 y LAN Ethernet del Host.
Fuente: Propia del autor.

3. Interfaz y dirección de la red LAN wireless y red virtual del Host.

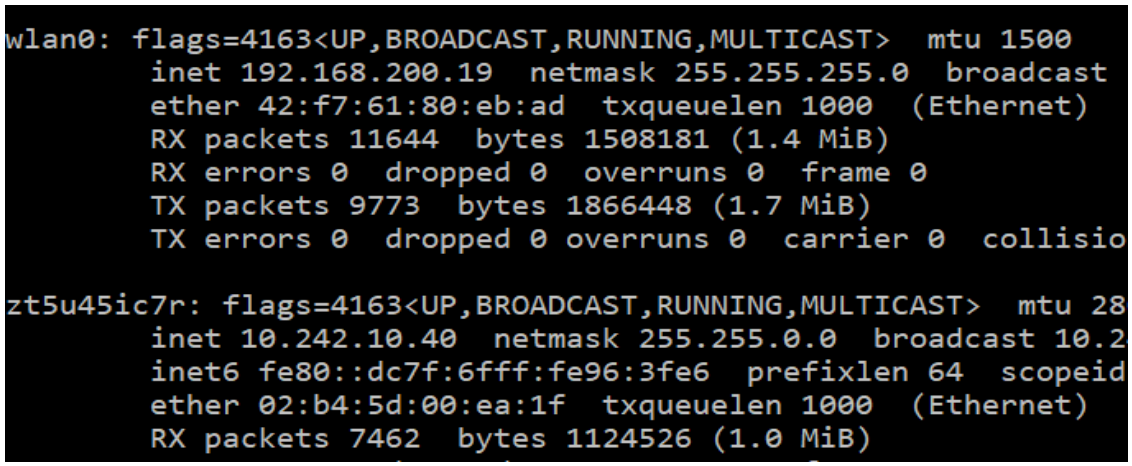


Figura 210. Interfaces y Direcciones de Red de WLAN y ZT_IF del Host.
Fuente: Propia del autor.

4. Información de la instalación del demonio de Docker que se ejecuta en Debian, tenemos los datos relevantes de contenedores, imágenes, directorios y las redes.

```
pi@raspberrypi:~ $ sudo docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Build with BuildKit (Docker Inc., v0.6.1

Server:
 Containers: 12
  Running: 12
  Paused: 0
  Stopped: 0
 Images: 12
 Server Version: 20.10.8
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.run
```

Figura 211. Información del Demonio de Docker. (A), (Print Screen).

Fuente: Propia del autor.

5. Información de la versión de Docker, la arquitectura y kernel del anfitrión.

```
 containerd version: e25210fe30a0a703442421b6
 runc version: v1.0.1-0-g4144b63
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.10.52-v7+
 Operating System: Raspbian GNU/Linux 10 (bus
 OSType: linux
 Architecture: armv7l
 CPUs: 4
 Total Memory: 923.2MiB
 Name: raspberrypi
 ID: Y5CF:XIXP:P67D:OZSR:SE2L:FY32:5EFC:AG4L:
 Docker Root Dir: /home/pi/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
```

Figura 212. Información del Demonio de Docker. (B), (Print Screen).

Fuente: Propia del autor.

6. ID, nombre, puertos, imágenes y estado actual de los contenedores.

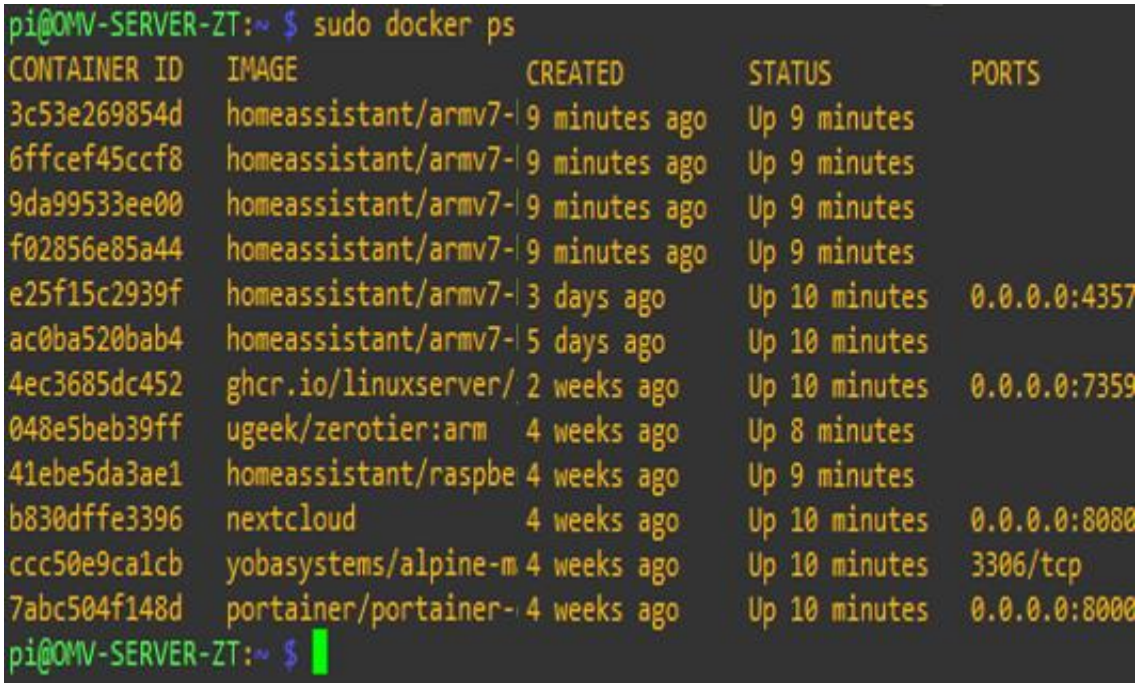


Figura 213. Información Básica de los Contenedores, (Print Screen).
Fuente: Propia del autor.

7. Redes que se ejecutan en el host actual.

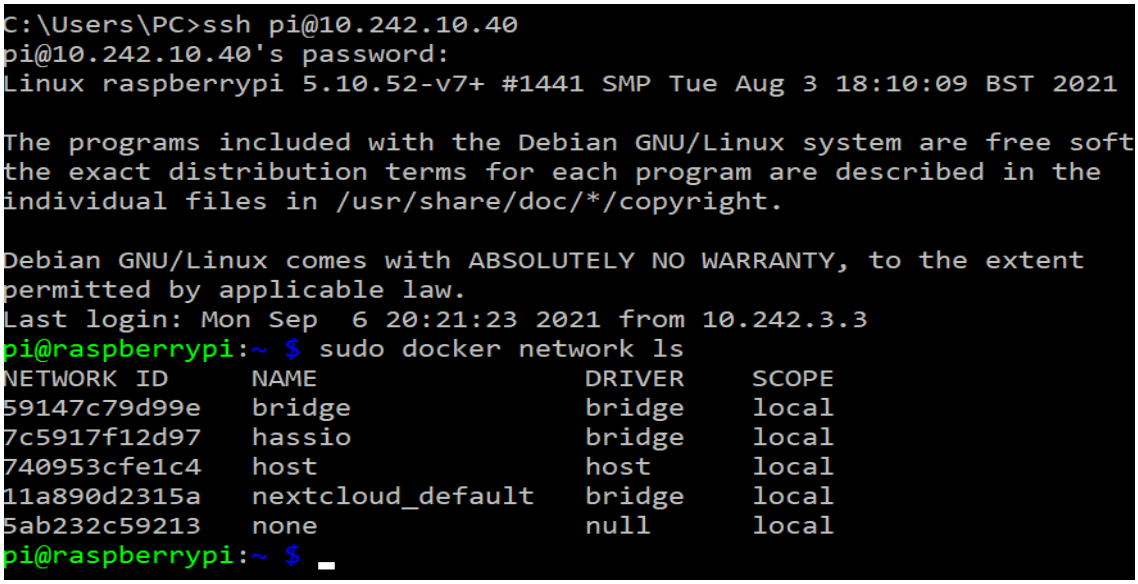


Figura 214. Redes en ejecución del Host del Servicio , (Print Screen).
Fuente: Propia del autor.

8. Reglas de iptables activas como una tabla de reglas ordenadas por cadena.

```
pi@raspberrypi:~ $ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain DOCKER (3 references)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             172.30.32.6          tcp dpt:http
ACCEPT     tcp  --  anywhere             172.17.0.2           tcp dpt:8096
ACCEPT     tcp  --  anywhere             172.17.0.3           tcp dpt:9000
ACCEPT     tcp  --  anywhere             172.17.0.4           tcp dpt:http
ACCEPT     udp  --  anywhere             172.17.0.2           udp dpt:7359
ACCEPT     tcp  --  anywhere             172.17.0.3           tcp dpt:8000

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-ISOLATION-STAGE-2 (3 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-USER (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
pi@raspberrypi:~ $
```

Figura 215. Tabla de Reglas Iptables del Host de Servicio, (Print Screen).
Fuente: Propia del autor.

9. Validación de las rutas de paquetes en los interruptores de las tarjetas de interfaces de red (NIC) entre el Host y clientes de la red.

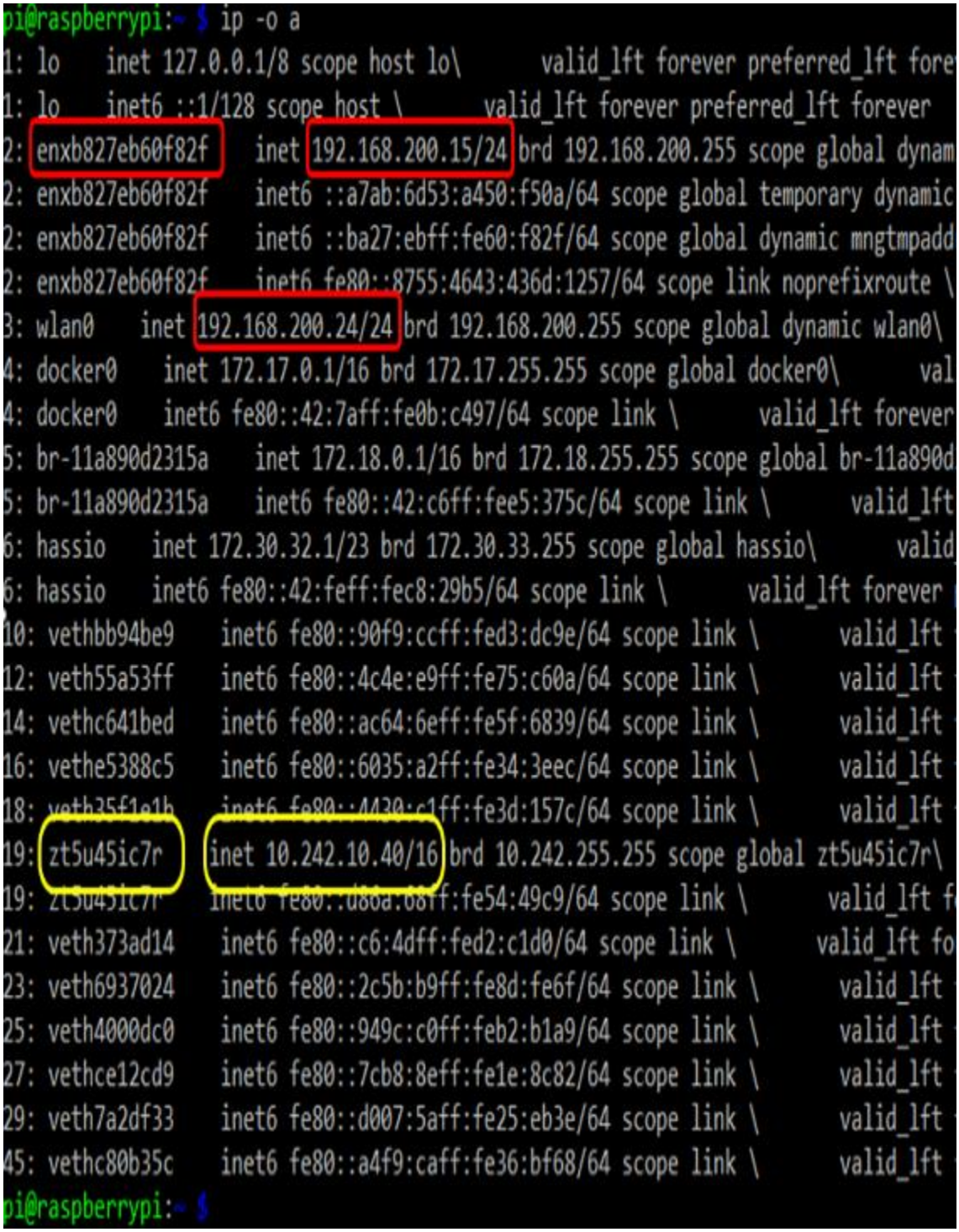


Figura 216. Transmisión de Paquetes entre la Capa Física y Enlace de Datos.
Fuente: Propia del autor.

10. Diagnóstico de conectividad y validez de la información estructurada en la comunicación de la red LAN. La figura 217 nos muestra una radiografía de los metadatos del Host.

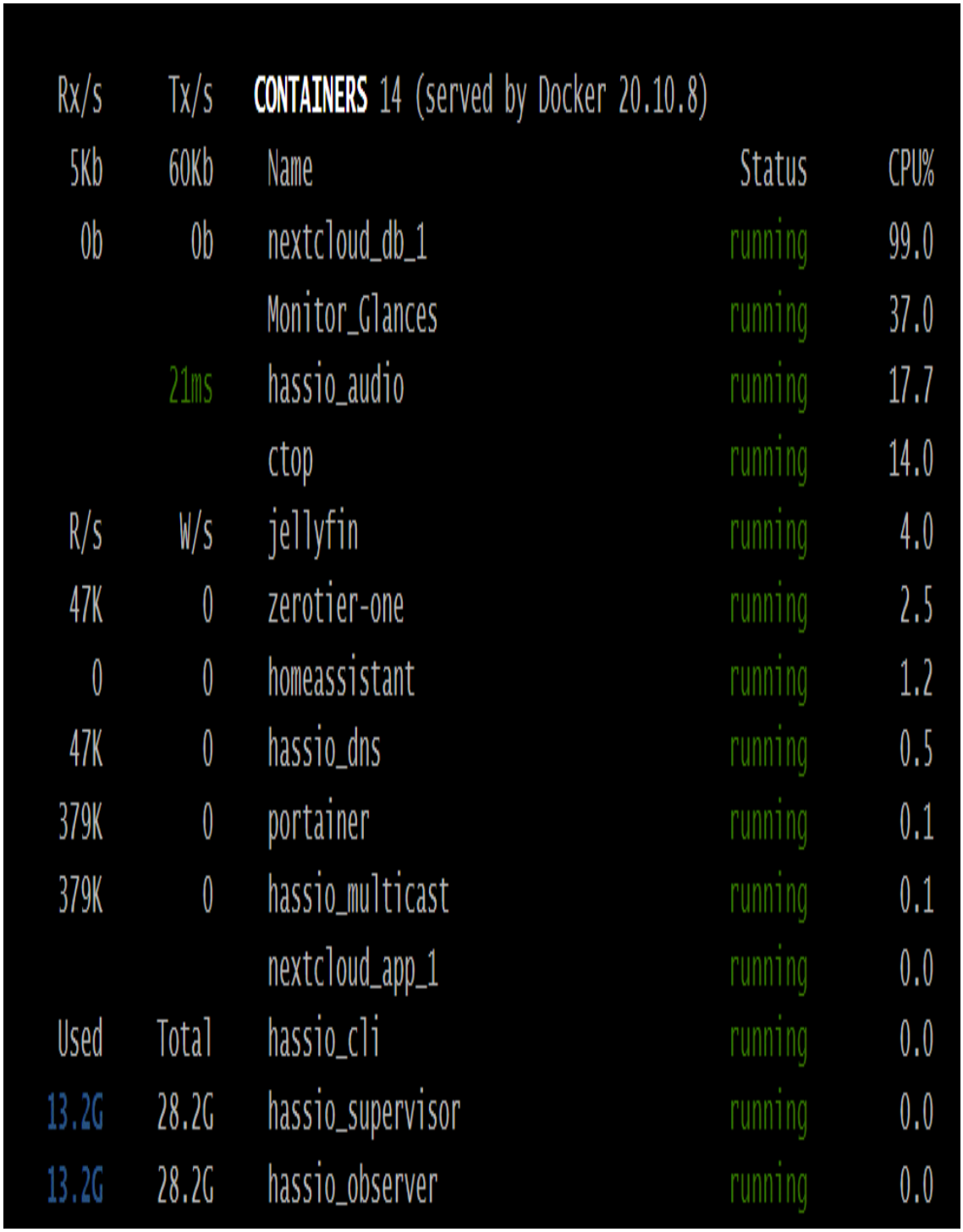


Figura 217. Diagnóstico de Conectividad y Métricas de la Red Virtual.
Fuente: Propia del autor.

11. Análisis de los metadatos en la interfaz lo (bucle local), que usaremos en esta tarea de diagnóstico de conectividad y validez de los protocolos de comunicación de la Red Virtual en funcionamiento.

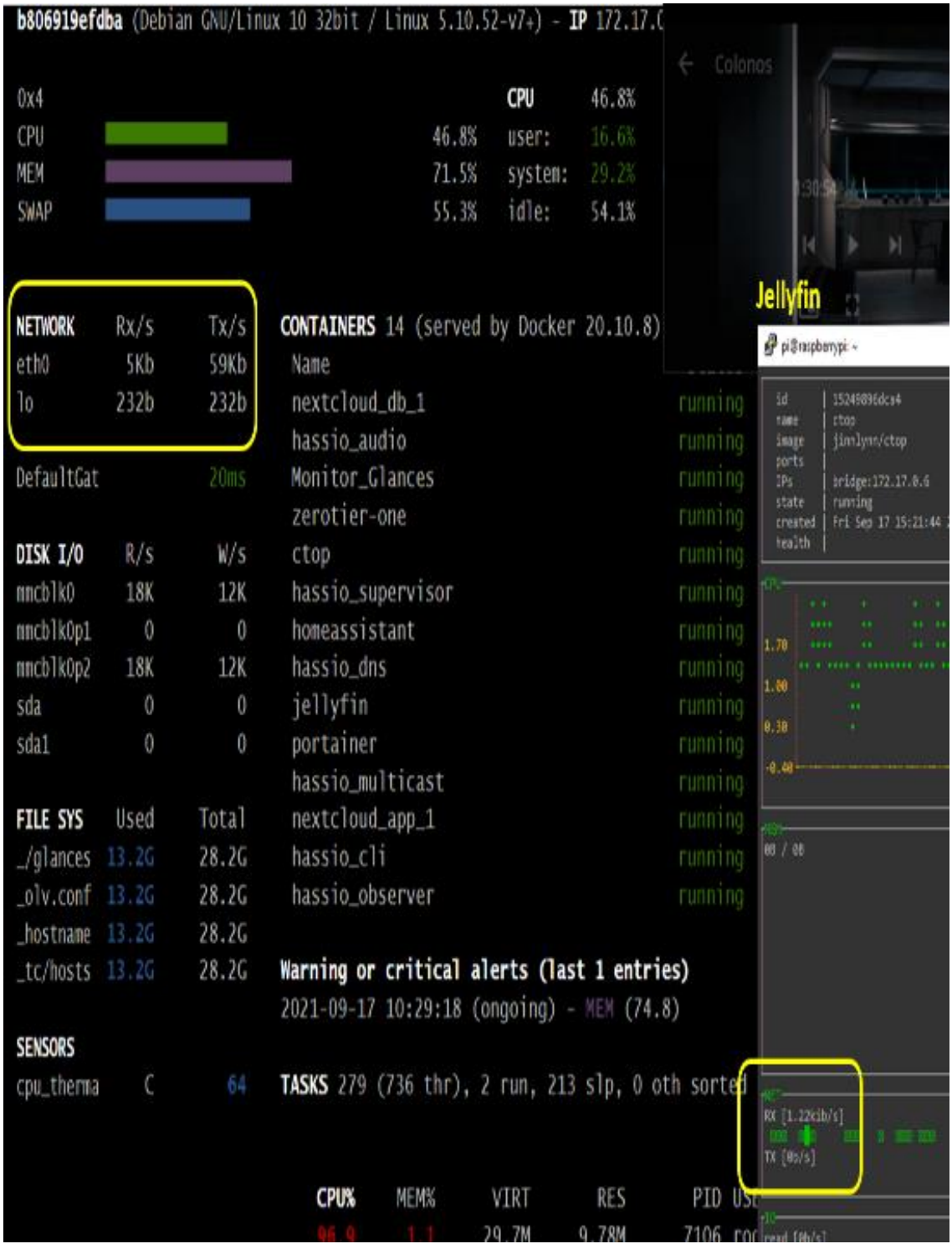


Figura 218. Análisis de los Metadatos en la Interfaz de Bucle Local, (Print Screen).
Fuente: Propia del autor.

12. Estadísticas de transmisión y recepción de datos desde los contenedores.

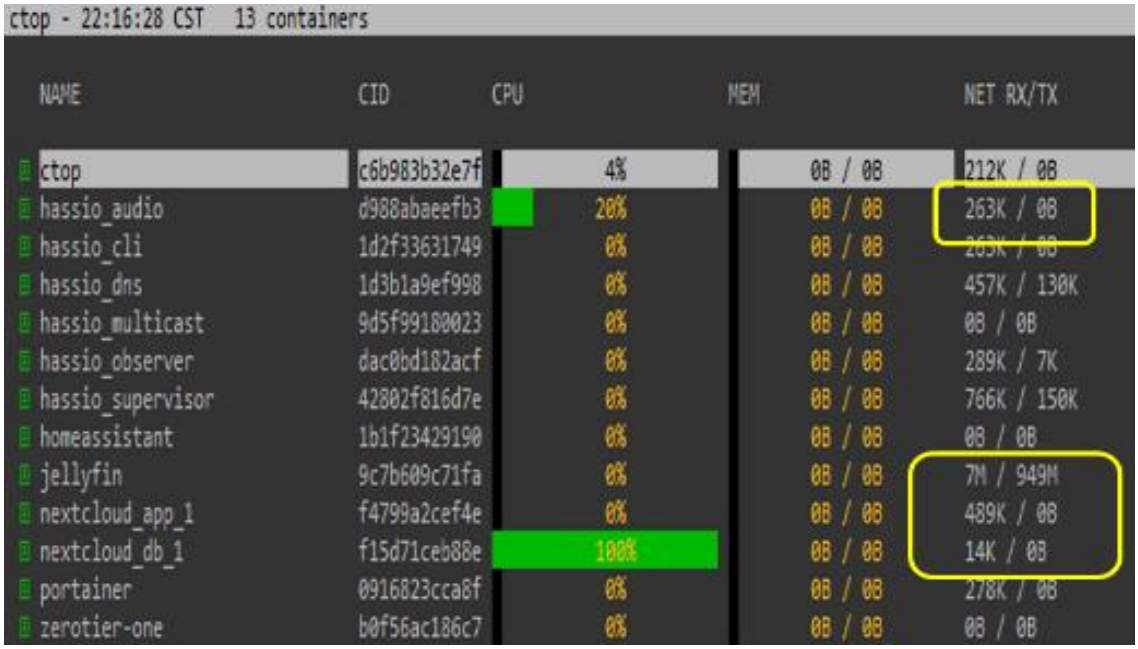


Figura 219. Estadísticas de Transmisión y Recepción de Datos desde el Host.
Fuente: Propia del autor.

13. En esta entrada de red superpuesta, es necesario configurar el componente de firewall para que sea posible invocar tráfico ICMP al servidor http en dos de los invitados presentes. Posteriormente, permita que uno de los invitados se comunique con un servidor https en ejecución en los puertos respectivos.

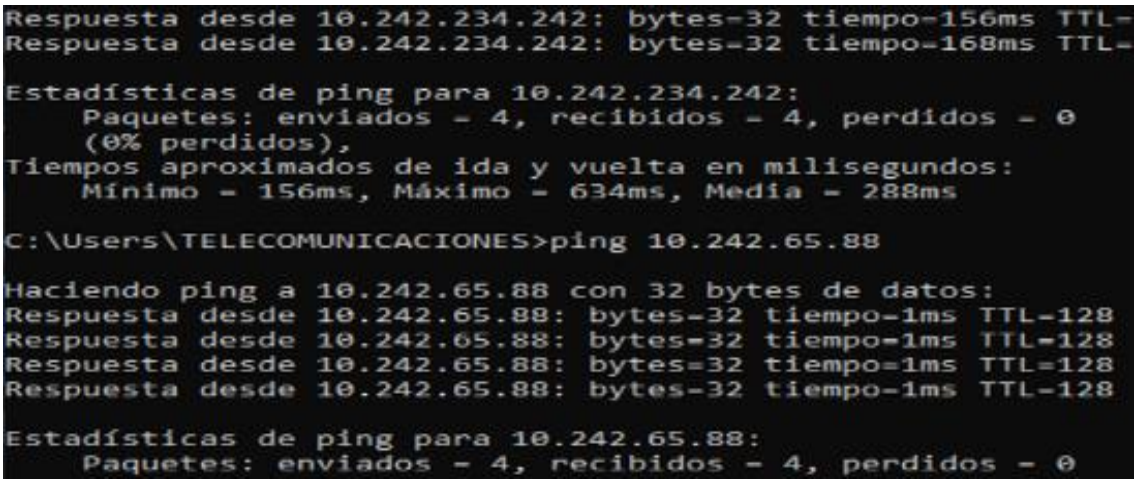


Figura 220. Comunicación del Host con Equipos Remotos de la Red VXLAN.
Fuente: Propia del autor.

14. Análisis de tráfico de paquetes de red con la herramienta Wireshark desde Equipo PORTÁTIL YILDER - IP 10.242.3.3. Interfaz de red Virtual VXLAN.

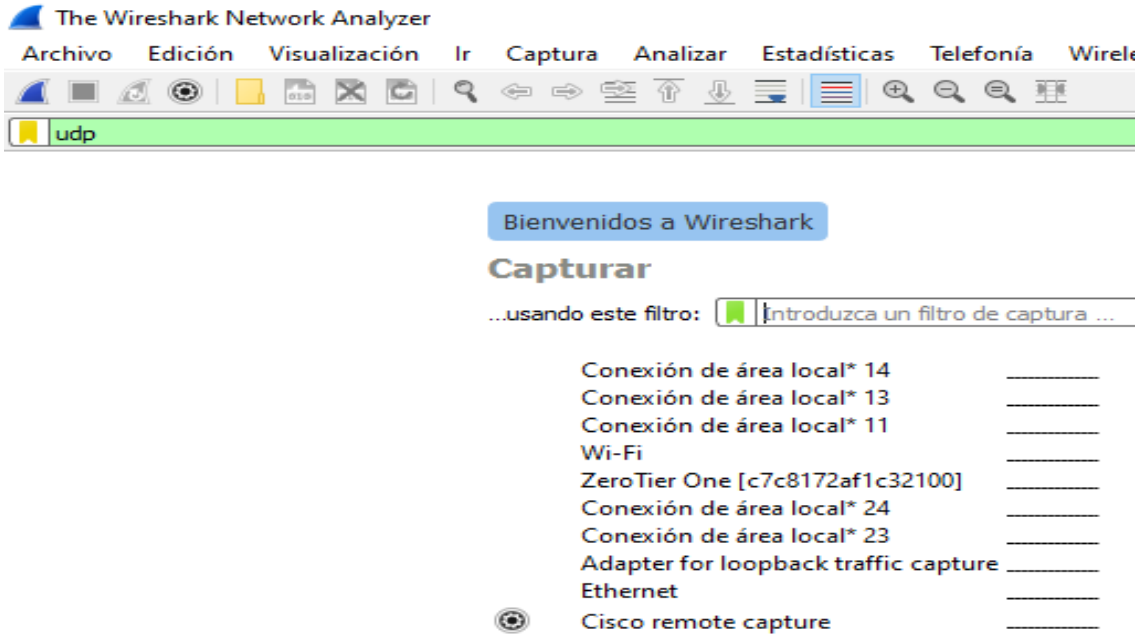


Figura 221. Portal de la Herramienta Wireshark, (Print Screen).
Fuente: Propia del autor.

15. Filtramos la captura de Wireshark para detectar la actividad en la interfaz de la Red Virtual, ID de Red [c7c8172af1c32100].

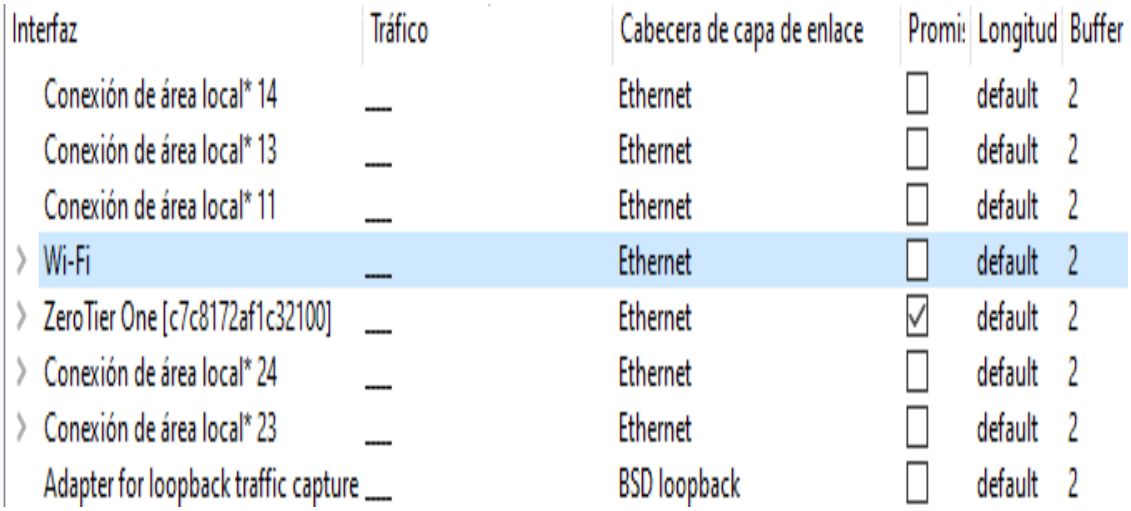


Figura 222. Filtro de Captura para Tráfico de la Red Virtual [c7c8172af1c32100].
Fuente: Propia del autor.

16. Al activar el tráfico de red, se observa el inicio de solicitudes MDNS a todas las interfaces enviando solicitudes de estado NBNS, en búsqueda de dispositivos UPnP. Equipo portátil IP 10.242.3.3

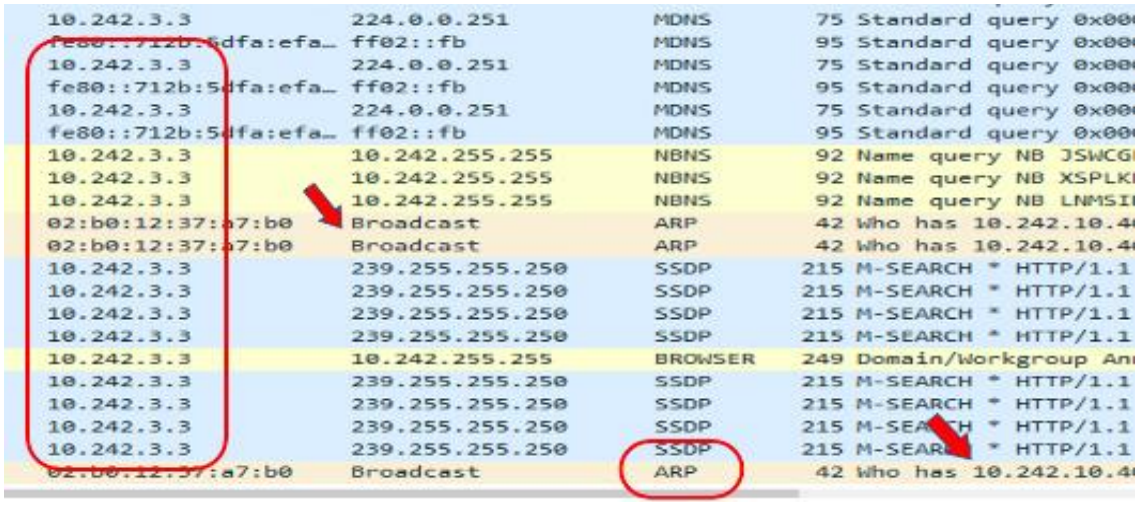


Figura 223. Inicio de Solicitudes MDNS en la Red VXLAN, (Print Screen).
Fuente: Propia del autor.

17. Validación del tráfico de paquetes en la red Virtual. Respuesta de Wireshark en la búsqueda de la dirección UPnP en el broadcast utilizando el protocolo ARP multicast.

Figura 224. Análisis de Tráfico de Datos entre IP 10.242.3.3 y la IP 10.242.10.40.
Fuente: Propia del autor.

18. Generamos tráfico luego de iniciar OpenMediaVault, como podemos observar en el entramado de paquetes de la red, reconocer las peticiones realizadas en toda la secuencia de paquetes como DNS, SSDP y TCP.

Figura 225. Supervisión de Tráfico de Red Asimétrico en OMV, (Print Screen).
Fuente: Propia del autor.

19. Como podemos observar en las peticiones realizadas, el protocolo TCP obliga al emisor a retrasar la emisión de cada nuevo paquete hasta que se recibe el ACK del anterior, esto es así ya que el receptor solo puede generar un ACK como respuesta a la llegada de un paquete.

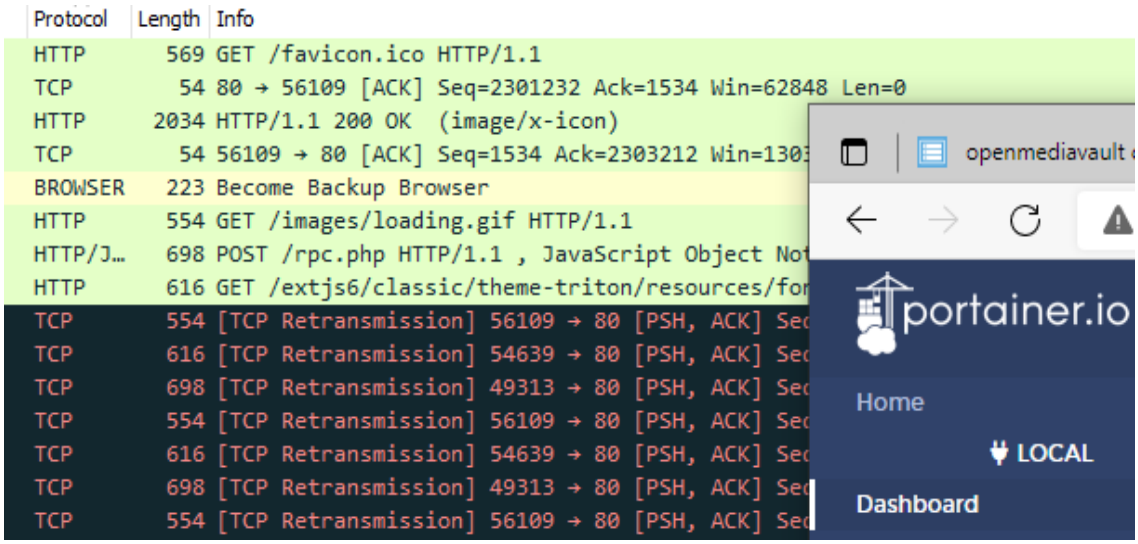


Figura 226. Tráfico de Red en la Gestión de Contenedores, (Print Screen).
Fuente: Propia del autor.

20. Entramado de la red en marcos y paquetes con el servicio Multimedia Jellyfin, se observa el envío de paquetes broadcast desde Servidor NAS ip 10.242.10.40, protocolo (TCP), puerto 8096.

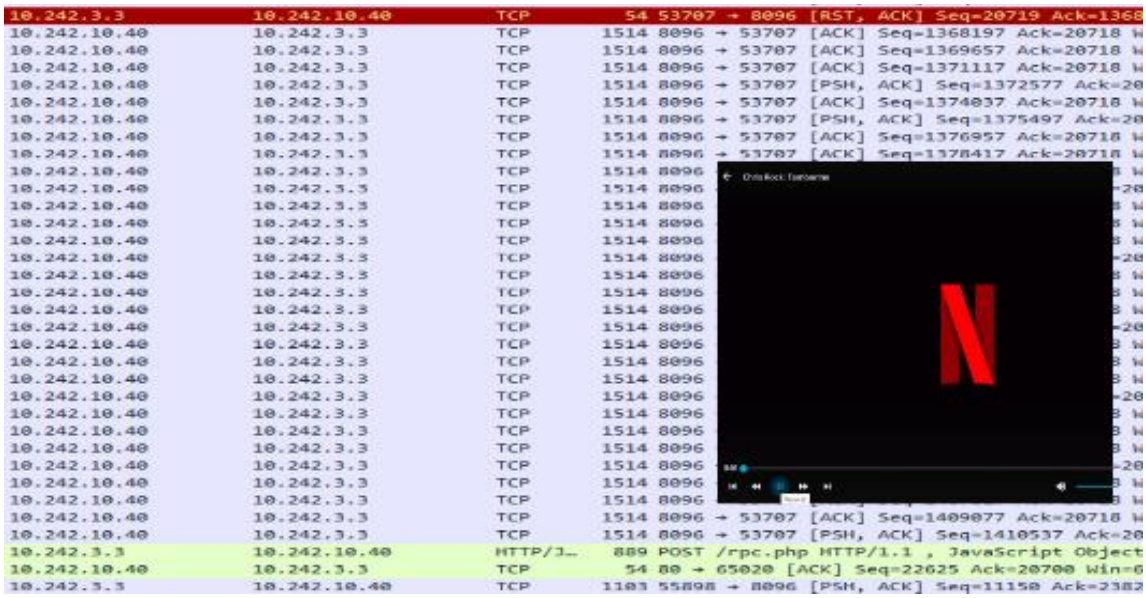


Figura 227. Reenvío de Paquetes Broadcast desde Servidor NAS ip 10.242.10.40.
Fuente: Propia del autor.

21. Una vez que se completa el enlace del Servidor NAS hasta el equipo cliente portátil Windows. Vemos como se dispara el tráfico de paquetes de 31347 a 69274 en cuestión de segundos.

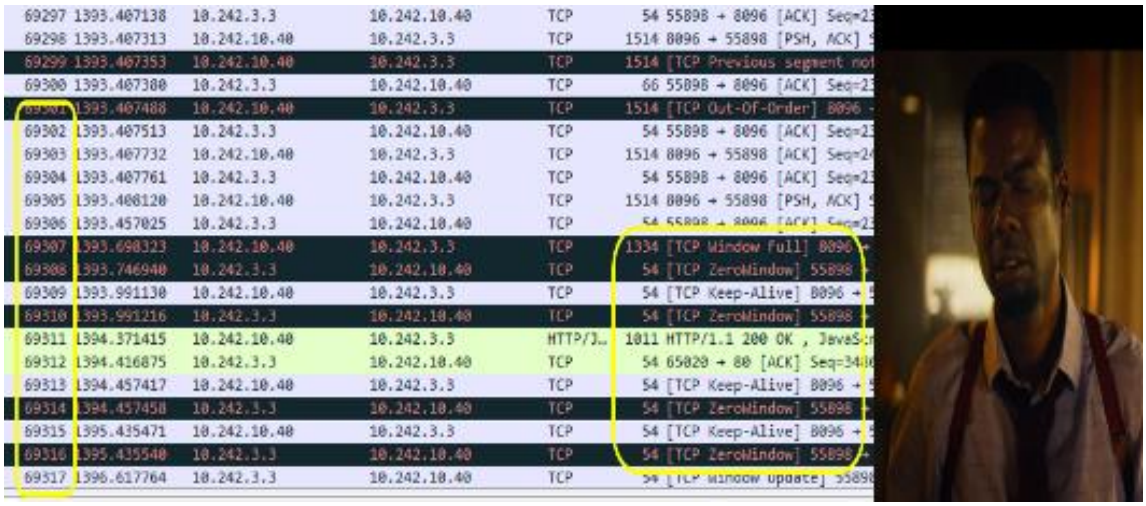


Figura 228. Disparo del Tráfico de Paquetes entre el Servidor y Clientes Portátil.
Fuente: Propia del autor.

22. Entramado de la red en marcos y paquetes en la comunicación de la nube personal, se observa el envío de paquetes asimétricos broadcast en la red del Host del Servidor NAS, utilizando el puerto 8080. Una vez que se establece la conexión, todos los paquetes deben tener configurado ACK y coincidir con el número de secuencia de los paquetes recibidos para un transporte / seguridad confiable. RST sin ACK no será aceptado. Cuando un lado envía RST, el socket se cierra inmediatamente y el lado receptor también cierra el socket inmediatamente después de recibir un RST válido.

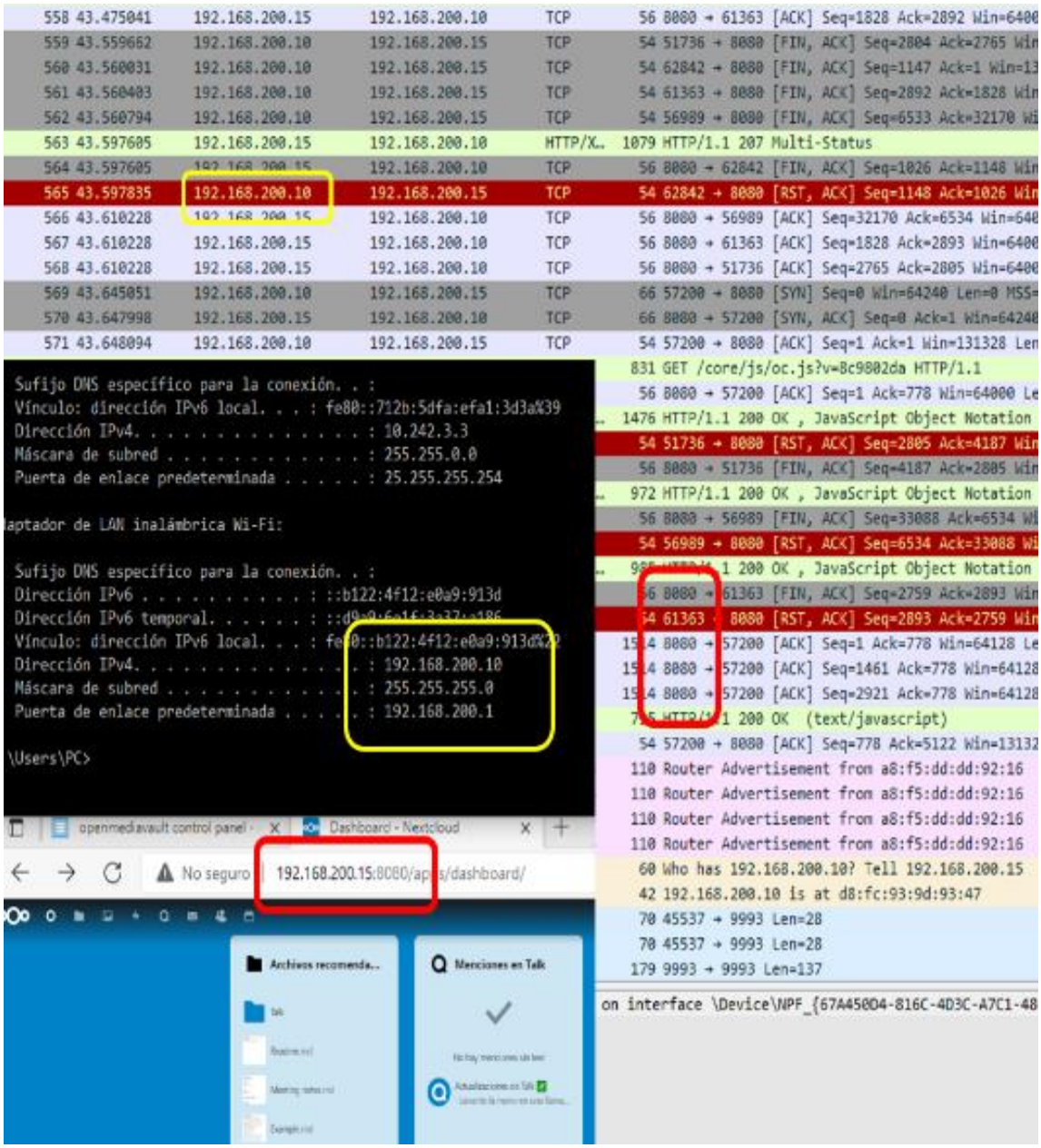


Figura 229. Supervisión de Tráfico de Red Asimétrico de la Nube Personal.
Fuente: Propia del autor.

CAPÍTULO V

5. ANÁLISIS DE RESULTADOS

5.1. Análisis de la Red Virtual

La idea principal de este trabajo fue crear una herramienta que facilite la enseñanza, la práctica y el aprendizaje de los principios fundamentales de las redes definidas por software, que con la sencilles y calidad de la infraestructura propuesta permita la ejecución automatizada de asignaciones predefinidas proporcionando al usuario una retroalimentación directa sobre la configuración. Se realizó la respectiva verificación para que el proyecto cumpla con los objetivos propuestos.

5.2. Comunicación de Equipos en la Red Virtual

Con base en los resultados de la práctica realizada en el subcapítulo 4.5.1 la red superpuesta funciona, pues hemos podido comunicar todos aquellos clientes conectados al mismo Network ID de la Red Virtual. El conmutador central de ZeroTier, localiza las interfaces de todos los dispositivos que están conectados a la red virtual, gracias a la app de ZeroTier One y los conecta, creando una red de equipos en una misma VXLAN sin importar la ubicación física que se encuentren, garantizando así que tenemos comunicación entre equipos y en apartados remotos de la red LAN a la cual estamos conectados.

5.3. Funcionamiento del Servidor NAS con la Tecnología SDN de ZeroTier

Con base en los resultados de la práctica realizada en el subcapítulo 4.5.2. Esta aplicación es una propuesta de solución NAS que seguro se debe enriquecer en el futuro. Es competitivo, ya que funciona de una forma ágil, pues tenemos un servidor con ciertos servicios en la red disponibles en internet, y sin necesidad de montar una VPN, saltándonos firewall, switch y demás seguridades de sistemas informáticos en un ambiente de administración controlada. El tráfico va totalmente cifrado de extremo a extremo y esta disponibles para todos los dispositivos de redes como Windows, Linux, Mac, Android, IOS, Raspberry y demás SBC.

5.4. Conectividad y Tráfico de Datos en Equipos Internos y Remotos de la Red

Para analizar la conectividad de los equipos, se realizó un mapeo de los registros desde los contenedores del equipo ya que la comunicación de redes de docker tienen ciertas reglas de tráfico entre contenedores y con respecto de cara hacia el exterior, es necesario configurar el componente de iptables para que sea posible invocar tráfico ICMP al servidor https en dos de los invitados presentes. Para analizar el tráfico P2P utilizamos el sniffer Wireshark en nuestra red. El tráfico generado por la comunicación de VXLAN deja una serie de huellas que filtramos solo para la red virtual y en los puertos que ya se han mencionado, trabajamos para el respectivo análisis del tráfico de paquetes en un ambiente controlado.

CAPÍTULO VI

6. CONCLUSIÓN

Con este tipo de soluciones basadas en redes definidas por software se destaca netamente que no nos encontramos atados a un solo escenario de comunicación, ya que las SDN se están abriendo paso a las redes de datos modernos en un mundo donde las capacidades de las redes son cada vez más exigentes.

SDN representa una revolución en el mundo de las redes de comunicaciones, como vemos en esta investigación las empresas como Cisco, Huawei, Hp, empiezan a confiar cada día más en estas nuevas tecnologías porque son el futuro de las redes al permitir aprovechar mejor los recursos de las tecnologías actuales.

Las prácticas efectuadas en este proyecto con el conmutador ZeroTier de tecnología SDN permitirá que los dispositivos conectados mediante la red controlen remotamente las unidades de control y de administración, permitiendo así automatizar, controlar los dispositivos y redes de una forma altamente escalables y flexibles, que se adaptan rápidamente a los requerimientos de los usuarios y administradores de redes.

Cualquier dispositivo puede, a través de una dirección de red de ZeroTier, ponerse en contacto con cualquier otro dispositivo conectado en el mundo en cualquier momento. La inicialización de la conectividad no debería tardar más de un segundo en promedio, idealmente lo más cerca posible de la latencia de la red subyacente.

Se deja una plataforma con prestaciones de comunicación SDN - SDWAN apropiada para ejecutar soluciones prometedoras como plataforma de control para redes, abre posibilidades más amplias para la prestación de servicios, Hardening de servidores, podría facilitar el mantenimiento de sistemas remotamente, la resolución de problemas de comunicación y ahorrar enormes costos en la administración de la red.

Por todo lo expuesto con respecto a los aspectos positivos que brinda el proyecto debería dar como resultado la prestación de servicios de mejor calidad para el consumidor final.

CAPÍTULO VII

7. RECOMENDACIONES

Verificar que todas las conexiones en los adaptadores y puertos de la raspberry pi estén correctamente conectadas, ya que en varias ocasiones se perdió la comunicación y surgieron problemas en la configuración del almacén de la red.

Realizar backup del sistema a medida que vaya avanzando el proyecto, la pérdida de comunicación y errores de código será el aliado a la hora de realizar las configuraciones, de lo contrario será necesario volver a configurar todo desde el principio.

Se debe tener cuidado que la alimentación brindada a la tarjeta de la raspberry pi sea la adecuada según las especificaciones 5V y asegurarse que la capacidad de corriente sea de 2000 mA, esto asegura el buen funcionamiento de la tarjeta electrónica.

Durante el tiempo de operación del equipo, se recomienda mantenerlo en un lugar fresco y ventilado a fin de que sus disipadores de calor actúen eficientemente.

Se recomienda revisar las alternativas que brindan los distintos fabricantes en materia de SDN para valorar las posibles potencialidades que pueden ofrecer otros protocolos o tecnologías.

Es importante que, si estás publicando puertos instalando servicios, cambies el puerto 80 y si lo usas, también el 443, por otros puertos.

Es recomendable que tanto las universidades, los profesionales como estudiantes, docentes e ingenieros desarrollen el campo de la investigación en materia de SDN , para optimizar y mejorar las migraciones en los centros de datos, con el fin de lograr soluciones rápidas y eficaces.

Interpretar, colaborar y desarrollar mejoras en el Código fuente del Proyecto ZeroTier sería extremadamente beneficioso para la comunidad, pues se tendría una estimación más precisa de los parámetros de la red y, principalmente, para la predicción de eventos dentro de la red, la misma que brinda las posibilidades de ser un referente de comunicación para las redes futuras y la oportunidad de integrar varias técnicas de aprendizaje automático sobre redes SDN.

CAPÍTULO VIII

8. REFERENCIAS BIBLIOGRÁFICAS

ebizLatam. (26 de 01 de 2020). *El Perímetro Definido por Software es la arquitectura de Seguridad Moderna que reemplaza a las VPN*. Obtenido de Perímetro Definido por Software: la tendencia que incorporan: http://www.ebizlatam.com/perimetro-definido-por-software-la-tendencia-que-incorporan-las-empresas/

A. Farrel, Ed. (04 de 09 de 2017). *An Architecture for Use of PCE and PCEP in a Network*. Obtenido de Una arquitectura para el uso de PCE y PCEP en una red con control central: https://www.rfc-editor.org/rfc/rfc8283.txt

Amazon Web Services, Inc. (2021). *¿Qué es Docker?* Obtenido de Contenedores de Docker | ¿Qué es Docker? | AWS: https://aws.amazon.com/es/docker/

atlassian.net. (9 de 2 de 2021). *Bridge your ZeroTier and local network with a RaspberryPi*. Obtenido de Bridge your ZeroTier and local network with a RaspberryPi: https://zerotier.atlassian.net/wiki/spaces/SD/pages/193134593/Bridge+your+ZeroTier+and+local+network+with+a+RaspberryPi

Bill Kleyman. (18 de 12 de 2015). *Understanding the Rise and Impact of SDN and NFV - Data*. Obtenido de Understanding the Rise and Impact of SDN and NFV: https://www.datacenterknowledge.com/archives/2015/12/18/understanding-the-rise-and-impact-of-sdn-and-nfv

Casierra Cavada, J., & Quiñónez Ku, X. (03 de 2018). Virtualización de Redes y Servidores Emulando Infraestructuras Tecnológicas. *Revista Científica Hallazgos21, 3 (Suplemento Especial).*, https://revistas.pucese.edu.ec/hallazgos21/article/view/236. Obtenido de VIRTUALIZACIÓN DE REDES Y SERVIDORES EMULANDO INFRAESTRUCTURAS TECNOLÓGICAS: https://www.researchgate.net/publication/334000398_VIRTUALIZACION_DE_REDES_Y_SERVIDORES_EMULANDO_INFRAESTRUCTURAS_TECNOLOGICAS

Casierra, J., Quiñónez, X., Herrera, L., & Egas, C. (2018). Virtualización de Redes y Servidores Emulando Infraestructuras Tecnológicas. *Revista Científica Hallazgos21, 3(Suplemento Especial).*, 2-6.

community.home-assistant.io. (06 de 2020). *Instalación de Home Assistant supervisado en Raspberry Pi OS*. Obtenido de Installing Home Assistant Supervised on a Raspberry Pi with: https://community.home-assistant.io/t/installing-home-assistant-supervised-on-raspberry-pi-os/201836

Docker Inc. (31 de 08 de 2021). *Descripción general de Docker*. Obtenido de Docker Documentation | Docker Documentation-Docker Inc: https://docs.docker.com/get-started/overview/

dockerhub. (act.2021). *linuxserver / jellyfin*. Obtenido de linuxserver/jellyfin - Docker Image: https://hub.docker.com/r/linuxserver/jellyfin

Dorantes Alonso, G. M. (2017). *Characterization of the NFV networks*. Obtenido de Caracterización de las redes NFV - DSpace@UCLV: https://dspace.uclv.edu.cu/handle/123456789/10917?show=full

Ediiones. (05 de 08 de 2021). *Concepto {Version PDF}*. Obtenido de Aatamia: https://concepto.de/anatomia/

Espressobin. (s.f.). *Marvell ESPRESSObin | An SBC Unlike Any Other*. Obtenido de http://wiki.espressobin.net/tiki-index.php?page=Board+Distributors: https://wiki.espressobin.net/tiki-index.php?page=Ports+and+Interfaces

ETSI Group (ISG). (01 de 12 de 2014). *Network Functions Virtualisation (NFV): Architectural Framework*. Obtenido de https://www.etsi.org/: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

Fernádez, P. (10 de 2015). *https://www.etsit.upm.es/*. Obtenido de Programación de redes SDN mediante el controlador POX: https://repositorio.upct.es/xmlui/bitstream/handle/10317/5254/tfg729.pdf?sequence=1&isAllowed=y

Flachs, F. (22 de 05 de 2020). *Automatizovaný systém konfigurace SDN sí´ových za°ízeních*. Obtenido de Ceské vysoké ucení technické v Praze: https://dspace.cvut.cz/bitstream/handle/10467/88059/F3-DP-2020-Flachs-Frantisek-Automatizovany_system_konfigurace_SDN_sitovych_zarizenich.pdf?sequence=-1&isAllowed=y

ganacell.com. (act.2021). *Tablet EET1, 10 pulg*. Obtenido de Tablet EET1, 10 pulg, 16gb, 1gb ram, quad core, hasta 32gb: https://www.ganacell.com/tablet-eet1--10-pulg--16gb--1gb-ram--quad-core--hasta-32gb-107/p

Geekflare. (9 de 09 de 2019). *Docker Networking*. Obtenido de Redes Docker 101 - Geekflare: https://geekflare.com/es/docker-networking/

get.docker. (s.f.). *Script de instalación de Docker CE para Linux*. Obtenido de https://get.docker.com/: https://get.docker.com/

GitHub. (13 de 03 de 2017). *Armada 3700 · MarvellEmbeddedProcessors / Wiki principal · GitHub*. Obtenido de Armada 3720 - GitHub: https://github.com/MarvellEmbeddedProcessors/main/wiki/Armada-3700

github.com. (act.2021). *portainer/portainer*. Obtenido de GitHub - portainer/portainer: Making Docker and Kubernetes ...: https://github.com/portainer/portainer

github.com. (act.2021). *zerotier/ZeroTierOne*. Obtenido de zerotier/ZeroTierOne: A Smart Ethernet Switch for Earth - GitHub: https://github.com/zerotier/ZeroTierOne

González, I. (2015). *Diseño e implementación de sistema interactivo de información de Docentes, con Raspberry PI.* Obtenido de Diseño e implementación de sistema interactivo de información de Docentes, con Raspberry PI.: https://dspace.ups.edu.ec/handle/123456789/10408

Google Sites. (s.f.). *TELECOMUNICACIONES - Google Sites*. Obtenido de Virtualización: https://sites.google.com/site/wwwvirtualizacioneducom/

hackaday.io. (act.2020). *Redes Raspberry Pi de la manera systemd*. Obtenido de Instructions | Raspberry Pi networking the systemd way: https://hackaday.io/project/162164/instructions

Hashemian, R. (2021). *Escáner de puerto TCP / IP*. Obtenido de TCP/IP Port Scanner - Hashemian.com: https://www.hashemian.com/tools/port-scanner.php

hscripts.com. (2016). *Comandos Linux route*. Obtenido de Comandos Linux route - Hscripts.com: https://www.hscripts.com/es/tutoriales/linux-commands/route.html

HUAWEI TECHNOLOGIES CO., L. (2015). *Huawei Agile Campus Network Solution [versión PDF]*. Obtenido de Huawei: https://www.karma-group.ru/upload/iblock/496/Huawei%20Agile%20Campus%20Network%20Solution%20Brochure%20(Detailed%20Version)%20.pdf

James , L. (16 de 05 de 2020). *Configure una VPN usando Zerotier y una Raspberry Pi*. Obtenido de Configurar una VPN usando Zerotier y una Raspberry Pi (Cómo): https://www.jamesleighton.com/2020/05/16/vpn-using-zerotier-and-a-raspberry-pi/

Khondoker, R., Zaalouk, A., Marx, R., & Bayarou, K. (2014). "Comparación basada en funciones y selección de controladores de redes definidas por software (SDN)". *Congreso Mundial de Aplicaciones Informáticas y Sistemas de Información 2014 (WCCAIS)* (pág. págs. 1 a 7). Hammamet, Túnez: IEEE.

LaDuke, T. (09 de 02 de 2021). *Conecte su ZeroTier y su red local con una RaspberryPi*. Obtenido de Overriding Default Route / Full Tunnel Mode: https://zerotier.atlassian.net/wiki/spaces/SD/pages/193134593/Bridge+your+ZeroTier+and+local+network+with+a+RaspberryPi

Lanner. (21 de 11 de 2019). *6 ejemplos de casos de uso de NFV - Lanner America*. Obtenido de 6 ejemplos de casos de uso de NFV: https://www.lanner-america.com/es/blog-es/6-ejemplos-de-casos-de-uso-de-nfv/

Lugo C, N. (2014). *TECNOLOGÍAS DE VIRTUALIZACIÓN EN LOS SISTEMAS*. Obtenido de Tecnologías de virtualización en los sistemas informáticos de las organizaciones empresariales del Estado Zulia: https://dialnet.unirioja.es/servlet/articulo?codigo=5157975

Mediaprice. (2021). *Convertidor Ide Sata Externo Delta*. Obtenido de Convertidor Ide Sata Externo Delta - MediaPrice Ecuador: https://www.mediaprice.com.ec/producto/convertidor-ide-sata-externo-delta-2/

Mikéska, M. (30 de 09 de 2019). *SPRÁVA A ŘÍZENÍ DATOVÝCH SÍTÍ POMOCÍ SOFTWAROVĚ ŘÍZENÉ SÍTĚ [Versión PDF]*. Obtenido de ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE: https://dspace.cvut.cz/bitstream/handle/10467/83339/F3-DP-2019-Mikeska-Martin-Sprava-a-rizeni-datovych-siti-pomoci-softwarove-rizene-site.pdf?sequence=-1&isAllowed=y

MORENO, J., & GARCIA, H. (2019). *Rediseñar la red LAN y WAN basado en el protocolo VXLAN para mejorar la comunicación de todas las sedes de la compañía CRC, en la ciudad de Bogotá*. Obtenido de Rediseñar la red LAN y WAN basado en el protocolo VXLAN: https://repository.ucc.edu.co/bitstream/20.500.12494/15833/2/2019_Redise%C3%B1o_de_la_Red_LAN_y_WAN_-_CRC_SAS.pdf

opengroup.org. (2019). *Foro de Jericó*. Obtenido de Jericho Forum - The Open Group Library: https://publications.opengroup.org/white-papers/security/jericho-forum

OpenMediaVault.org. (julio de 2021). *OpenMediaVault*. Obtenido de openmediavault - The open network attached storage solution: https://www.openmediavault.org/

Oracle. (08 de 2012). *https://www.oracle.com/index.html*. Obtenido de Introducción a Java® SE Embedded en Raspberry Pi: https://www.oracle.com/technical-resources/articles/java/raspberrypi.html

Oscar, E. (27 de 07 de 2021). *RZ Redes Zone*. Obtenido de Qué es el servicio SD-WAN en las operadoras: https://www.redeszone.net/tutoriales/redes-cable/que-es-sd-wan/

playonlinew. (act.2021). *Instale NEXTCLOUD en una Raspberry Pi 4 con OpenMediaVault y Docker*. Obtenido de Instale NEXTCLOUD en una Raspberry Pi 4 ... - PlayOnlineW: http://www.playonlinew.com/nextcloud

Portainer.io. (act.2020). *documentation.portainer.io/*. Obtenido de Portainer: Container Management GUI for Kubernetes, Docker: https://www.portainer.io/features/platform-management

Profesionalreview. (2021). *VPN: Todo lo que necesitas saber*. Obtenido de VPN: Todo lo que necesitas saber - Profesional Review: https://www.profesionalreview.com/redes/vpn/

Raspberry Pi, Foundation. (s.f.). *Raspberry Pi 3 Modelo B +*. Obtenido de Raspberry Pi, Foundation: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

SANTIAGO, C. C. (02 de 2021). *UPS - TTS270.pdf - UNIVERSIDAD POLITÉCNICA[version pdf]*. Obtenido de ANÁLISIS, DISEÑO, Y PROPUESTA DE UNA SOLUCIÓN DE ESCRITORIOS: https://dspace.ups.edu.ec/bitstream/123456789/19857/1/UPS%20-%20TTS270.pdf

Shahriar , S. (2018). *Instalar Docker en Raspberry Pi*. Obtenido de How to Install Docker on Raspbian OS - Raspberry Pi - Linux ...: https://linuxhint.com/install_docker_raspberry_pi/#

Stallings, W. (2017). *Foundations of Modern Networking: SDN, NFV, QoE, IoT and Cloud.* New Jersey: Pearson Education/Addison-Wesley Professional.

Standards, E. T. (24 de 10 de 2012). *Network Functions Virtualisation-An Introduction, Benefits, Enablers, Challenges & Call for Action*. Obtenido de Network Functions Virtualisation – Introductory White Paper: http://portal.etsi.org/NFV/NFV_White_Paper.pdf

systemd.network. (act.2021). *systemd.network - Configuración de red*. Obtenido de systemd-networkd - Freedesktop.org: https://systemd.network/systemd.network.html

Udemy.com. (03 de 2021). *Curso Completo de Linux Debian*. Obtenido de Curso Completo de Linux Debian | Udemy: https://www.udemy.com/course/debian-desktop-server/learn/lecture/18656040?components=buy_button%2Cdiscount_expiration%2Cgift_this_course%2Cpurchase%2Cdeal_badge%2Credeem_coupon#overview

ugeek.github.io. (09 de 04 de 2020). *Zerotier. Funcionamiento e instalación por docker o repositorios*. Obtenido de https://ugeek.github.io/blog/post/2020-04-09-zerotier-funcionamiento-e-instalacion-por-docker-o-repositorios.html: https://github.com/docker-projects/docker-zerotier

Volker , T. (09 de 08 de 2021). *Openmediavault Documentation - Read the Docs*. Obtenido de Openmediavault Documentation: https://buildmedia.readthedocs.org/media/pdf/openmediavault/latest/openmediavault.pdf

Voruganti, S., & Subramanian, S. (2016). *Software-Defined Networking (SDN) with OpenStack.* Birmingham-Mumbai: Packt Publishing Ltd.

westerndigital.com. (s.f.). *WD Green™ SATA SSD 2,5 pulgadas/7 mm con carcasa*. Obtenido de WD Green ™ SATA SSD con carcasa de 2,5 ”/ 7 mm: https://shop.westerndigital.com/products/internal-drives/wd-green-sata-2-5-ssd#WDS120G2G0A

Zafra CPR. (Abril de 2021). *Introducción a los contenedores y a Docker*. Obtenido de Curso “Introducción a Docker” desarrollado para el CPR de Zafra: https://iesgn.github.io/curso_docker_2021/sesion2/

Zafra CPR. (06 de mayo de 2021). *Redes en docker*. Obtenido de Curso “Introducción a Docker” desarrollado para el CPR de Zafra: https://iesgn.github.io/curso_docker_2021/sesion4/

ZeroTier Inc. (act 2021). *Topología de red y descubrimiento de pares*. Obtenido de Documentación ZeroTier- website: https://docs.zerotier.com/zerotier/manual

ZeroTier.org. (2021). *ZeroTier-One*. Obtenido de zerotier.com: https://www.zerotier.com/download/

CAPÍTULO IX

9. ANEXOS

9.1. ANEXO A. Cronograma de Duración del Proyecto Técnico

DURACION	5 MESES																			
ACTIVIDAD	MAYO				JUNIO				JULIO				AGOSTO				SEPTIEMBRE			
	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4
PRUEBAS Y ARMADO DEL HIPERVISOR																				
FINALIZACION DEL BORRADOR DE LA MEMORIA Y REVISION DEL TUTOR																				
REVISION TECNICA																				
REVISION CURRICULAR																				
REVISION DE ANTIPLAGIO																				
ENTREGA DEL LIBRO Y SUSTENTACIÓN																				

Tabla 11. Cronograma del Proyecto Técnico

9.2. ANEXO B. Disco de Estado Sólido (SSD) – WESTER DIGITAL 120GB

Número de modelo	WDS120G2G0A
Interfaz	SATA III de 6 Gb/s
Capacidad	120GB
Rendimiento4 [4 KB QD32]	Lectura de hasta 545 MB/s
Potencia	80/10 mW activo/suspensión/ 0,03 W Hibernación / 2,2 W (máx) lectura / 2,2 W (máx) escritura
Confiabilidad MTTF	Hasta 1 millón de horas
Temperaturas operativas Vibración operativa	Desde 0 °C hasta 70 °C 5,0 g RMS, 10-2000 Hz
Dimensiones	100,5 mm x 69,85 mm x 7,0 mm (2,5”)
Peso	40g
Certificaciones	FCC, UL, TUV, KC, BSMI,VCCI, CE, Marruecos, RCM,CAN ICES-3(B)/NMB-3(B)
Bytes escritos en total (TBW)[4]	120GB

Tabla 12: Detalles Técnicos del SSD Wester Digital SATA de 120GB – 2,5"

9.3. ANEXO C. Presupuesto para la Construcción del Proyecto de Red Virtual

EQUIPO	VALOR
RASPBERRY PI	$ 88
DISCO DE ESTADO SOLIDO - SSD	$ 65
ADAPTADOR SATA A USB - SSD	$ 22
TABLET ANDROID DRAGON K10’	$ 145
PROTECCIONES / INTERRUPTORES / TOMA DE CORRIENTE	$ 06
MicroSD 32GB	$ 11
BASES DE VIDRIO VICELADO CAPAS DE 35x50/25,5x31,3cm / SOPORTES DE ALUMINIO	$ 58
TRABAJOS EN LA ESTRUCTURA METÁLICA SOPORTE DE LOS ELEMENTOS ELECTRÓNICOS / PINTURA Y ACABADO	$ 15
TOTAL	$ 410

Tabla 13: Materiales del Proyecto Técnico

9.4. ANEXO D. Disponibilidad de ZeroTier One

A continuación, encontrará una guía rápida de la disponibilidad de Zerotier One

- **Microsoft Windows** - Instalador de MSI ZeroTier One (x86 / x64), apk para-Windows 7 /10= ZeroTier One.msi
- **MAC OS** - Instalador de PKG Se admite MacOS 10.12 o posterior, apk MAX OS = ZeroTier One.pkg
- **APPLE IOS** - Obtener desde la App Store, admite iOS 10. app iOS = ZeroTier One
- **ANDROID** - Obtener en Google Play Store, app android = ZeroTier One
- **LINUX** – Ejecutable para DEB / RPM, las distribuciones basadas en Debian y RPM, incluidas Debian, Ubuntu, CentOS, RHEL, Fedora y otras, son compatibles a través de un script que agrega el repositorio correcto e instala el paquete.
- **BSD** – Ejecutable y disponible en paquetes FreeBSD
- **CONTENEDORES** – La versión de biblioteca de ZeroTier ARM-UNIX, El repositorio contiene un Docker file que se puede usar para crear un ZeroTier en contenedor para usar con distribuciones de Linux de solo contenedor puro.

Otras distribuciones de Linux pueden tener sus propios paquetes. Si no, intente compilar e instalar desde la fuente, para unirse a la red desde la CLI debería funcionar en la mayoría de las plataformas Linux.

Esto es así:

$ docker run -it --rm --cap-add = NET_ADMIN --cap-add = SYS_ADMIN --device = / dev / net / tun [].

(SYS_ADMIN es necesario porque NET_ADMIN no incluye el ioctl () requerido para poner / dev / net / tun en modo tap.

Aquí hay una transcripción de una sesión en la que iniciamos un símbolo del sistema en un contenedor, instalamos ZeroTier One, lo iniciamos (debe hacerse manualmente aquí porque el contenedor no ejecuta init o systemd),

Esto es así:

$ docker run -it --rm --cap-add = NET_ADMIN --cap-add = SYS_ADMIN --device = / dev / net / tun [distro Linux] / bin / bash

Detectando distribución de Linux. Destinos admitidos para este script:

**MacOS (10.7+) en x86_64 (solo instala ZeroTier One pkg).
**Linux / Debian (wheezy +) i386, x86_64 y armhf (Raspbian / Jessie).
**Linux / Ubuntu (confiable o más reciente) en i386 y x86_64.
**Linux / SuSE (12+) en i386 y x86_64.
**Linux / CentOS (6+) en i386 y x86_64.
**Linux / Fedora (22+) en i386 y x86_64.
**Linux / Amazon (2016.03+) en x86_64

Encontrado RHEL / CentOS, se crea /etc/yum.repos.d/zerotier. Repos

Luego: Instalando el paquete ZeroTier One ... tenemos que unirnos a una red.

$ / usr / sbin / zerotier-cli unirse a [ID de ZeroTier].

Luego listamos la red con:

$ / usr / sbin / zerotier-cli listnetworks = 200 ok % red lista

ping earth.zerotier.net

PING earth.zerotier.net (29.209.112.93) 56 (84)

% 64 bytes de 29.209.112.93: icmp_seq= 2 ttl= 64 tiempo = 13,1 ms

Fuente: Adaptado de https://zerotier.atlassian.net/

9.5. ANEXO E. Servicios Cargados en la Red Virtual

Guía de Instalación de la Nube Personal con Next Cloud

DATOS INFORMATIVOS

Recursos Utilizados

- RaspberryPi.
- Disco solido SSD.
- Tarjeta microSD.
- Conexión a red de internet (ethernet o inalámbrica).
- Imagen OpenMediaVault (OMV).
- Imagen de Docker.
- Imagen de Portainer.
- Imagen de Next Cloud

Preparación

- Raspberry Pi 3B + con sistema operativo instalado y servidor SSH en ejecución.
- Disco solido configurado como sistema de Almacenamiento en la Red.
- Sistema FreeNAS OpenMediaVault cargado en la red del Host, habilitado protocolo ssh.
- Administrar el almacen de directorios en OpenMediaVault.
- Instalar Docker Host desde OMV.
- Instalar Portainer desde OMV.
- Instalar y Configurar el programa Open Source Next Cloud en Docker.

Procedimiento

1. Habilitar la administración de los contenedores, Portainer.
2. Gestionar la capacidad de servicio de FreeNAS OpenMediaVault como almacenamiento en Red de Next Cloud.
3. Gestión de los directorios de NextCloud.
4. Gestión del servidor de Next Cloud como nube personal.
5. Pruebas de Comunicación de equipos

Prerrequisitos y co-requisitos

- Trabajo básico en la terminal de linux.
- Funciones de componentes individuales (Administración de contenedores, OpenMediaVault, Portainer, NextCloud).
- Instalación y configuración de software en la CLI.

DATOS DE LA PRÁCTICA

Objetivo general

Realizar la configuración básica de elementos de red para lograr obtener un servidor de almacenamiento en la nube llamada Next Cloud adaptado al hardware de minicomputadoras como la Raspberry Pi en contenedores de Docker.

Objetivos específicos

- Armar un escenario de Nube personal con el software Next Cloud y OpenMediaVault en una Raspberry Pi.
- Realizar la configuración básica de los directorios del software NextCloud**.**
- Administrar y distribuir archivos en línea.

Glosario

LAN: Red de Área Local o LAN (Local Área Network). Es una red que conecta equipos en un área relativamente limitada.

NAS: Almacenamiento Conectado en Red o NAS (Network Attached Storage).

ARM: (Acorn RISC Machine – CPU de Arquitectura basada en RISC).

RISC: (Reduced Instruction Set Computer –Ordenador con un Conjunto de Instrucciones Reducidas). Es un tipo de diseño de microprocesadores generalmente utilizados en computadoras que está a favor de conjuntos de instrucciones pequeñas y simples que toman menor tiempo en ejecutarse.

Marco Teórico

NAS: (Network Attached Storage – Almacenamiento Conectado en Red). Dispositivo de almacenamiento de software libre o comercial con sistema operativo y prestaciones según sea su diseño, accesible desde la red, por ejemplo, para almacenamientos masivos de información, música, backups, etc.

Raspberry Pi: Es una serie de ordenador de placa reducida RISC (arquitectura ARM), desarrollado en Reino Unido por la Raspberry Pi Fundation, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital, abre las puertas de la experimentación en proyectos de electrónica.

Nube personal con NextCloud Gracias a la tecnología del software de código abierto NextCloud permitiremos que los usuarios creen un vínculo de

Cloud en redes seguras y muy fáciles de configurar, proporciona una interfaz web para configurar en detalle esta Nube privada, los clientes que se pueden conectar a la nube podrá compartir y colaborar en documentos, enviar y recibir correos electrónicos, administrar su calendario y tener video conferencias sin filtraciones de datos, (playonlinew, act.2021)

Figura 1. NextCloud en Raspberry Pi, (Print Screen).
Fuente: https://edgars.eizvertins.com/install-nextcloud-on-raspberry-pi/

MARCO PROCEDIMENTAL:

1. Script de Instalación de OpenMediaVault

```
# Conceder privilegios de administrador con sudo su

$ wget -O - https://github.com/OpenMediaVault-Plugin- Developers/install Script/raw/master/install | sudo bash

# Si tiene inconsistencia de privilegios ejecute.

$ sudo curl -sSL https://github.com/OpenMediaVault-Plugin-Developers /installScript/raw/master/install | sudo bash

#instalar OMV

$ sudo ./install -n

#Reiniciar el Host

$ sudo reboot
```

Fuente: https://github.com/OpenMediaVault-Plugin-Developers/installScript

2. Gestión de almacenamiento OpenMediaVault. El programa escaneará el disco SSD que esté conectado a la raspberry pi. El almacén se

gestionará directamente en el disco luego de formatearlo y montarlo, crear un repositorio de carpetas para los registros.

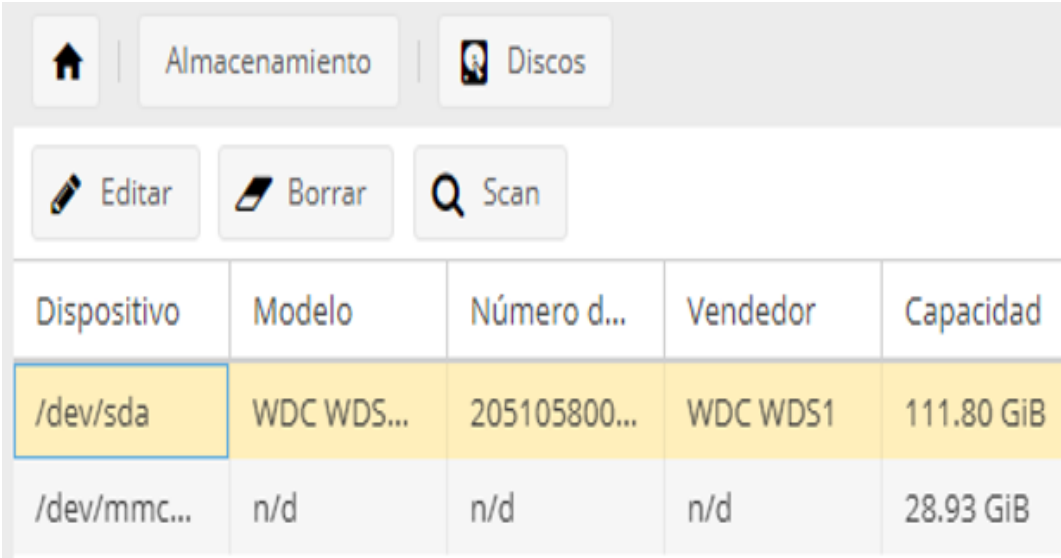


Figura 2. Scanner del Disco Sólido Conectado, (Print Screen).
Fuente: Propia del autor.

3. Instalar el contenedor de Docker y Portainer desde los plugin extras de OMV. Debe direccionar la ruta del almacén de docker a /home/pi/docker.

Figura 3. Ruta de Almacenamiento de Plugin de Docker y Portainer, (Print Screen).
Fuente: Propia del autor.

4. Escenario de instalación de Docker y Portainer.

Figura 4. Instalación de Docker y Portainer desde OpenMediaVault, (Print Screen).
Fuente: Propia del autor.

5. Crear carpeta con el nombre Cloud en Gestión de Archivos. Añadir carpeta.

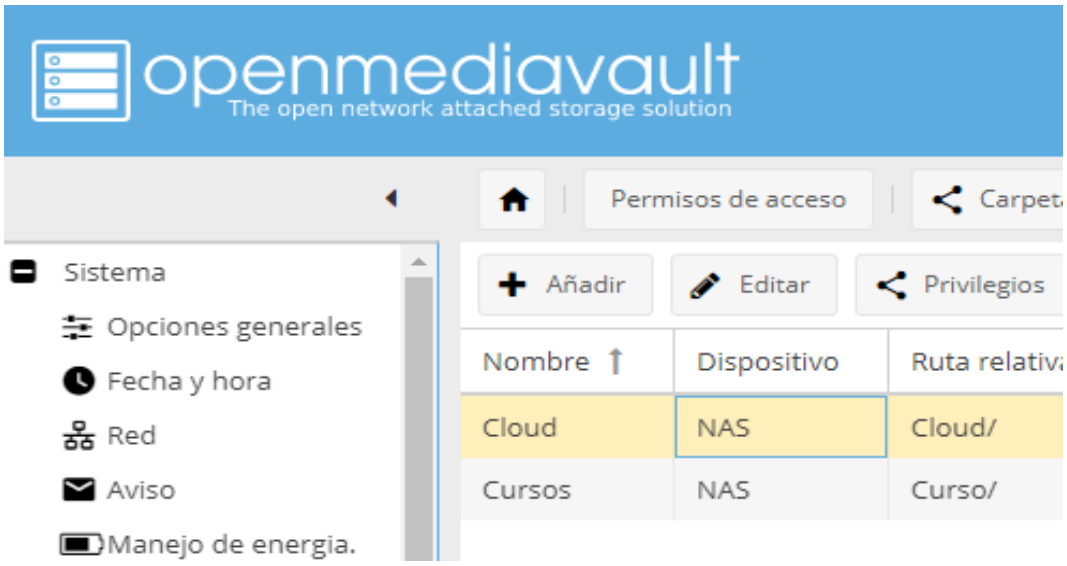


Figura 5. Crear carpeta para Repositorios de los Directorios, (Print Screen).
Fuente: Propia del autor.

6. Abrir ruta absoluta de carpeta Cloud.

◙ Click en Ruta Relativa>>click derecho.

- ✓ Elija visualizar ruta absoluta de la carpeta.
- ✓ Copie la ruta absoluta, esta deberá ser la ruta donde se cargarán los directorios al momento de crear el contenedor de NextCloud en la pila de Portainer.
- ✓ Luego de copiar la ruta absoluta de la carpeta, regrese la visualización a ruta relativa de Cloud en OMV.

◙ Actualizar para guardar los cambios.

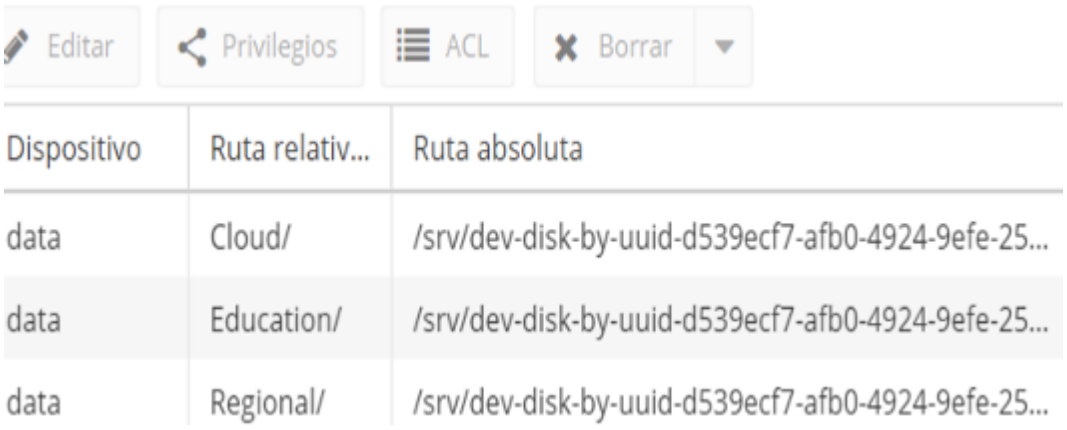


Figura 6. Gestionar Ruta Absoluta en el Repositorio de Archivos, (Print Screen).
Fuente: Propia del autor.

7. Gestionar el administrador de Contenedores Portainer.

Figura 7. Administrador de Contenedores Portainer, (Print Screen).
Fuente: Propia del autor.

8. Código fuente para generar el contenedor de NextCloud desde una Pila de Portainer. Se realiza un pull de la imagen desde la fuente de GitHub.

```
version: '2'

volumes:
  NextCloud:
  db:

services:
  db:
    image: yobasystems/alpine-mariadb:latest
    command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=Open*****
      - MYSQL_PASSWORD=Open*****
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud

  app:
    image: nextcloud
    ports:
      - 8080:80
    links:
      - db
    volumes:
      -/srv/dev-disk-by-uuid-874940c4-e38a-44b3-b935-
06a9a93731b2/Cloud/config/nextcloud:/var/www/html
    restart: always
```

Fuente: https://www.playonlinew.com/nextcloud

9. Crear el contenedor de NextCloud con una Pila de Portainer
   - ◙ Click en Pilas.
     - ✓ Nombre “nextcloud”.
   - ◙ Click en Editor Web.
     - ✓ Copie el código del script de instalación. y péguelo en el Editor Web.

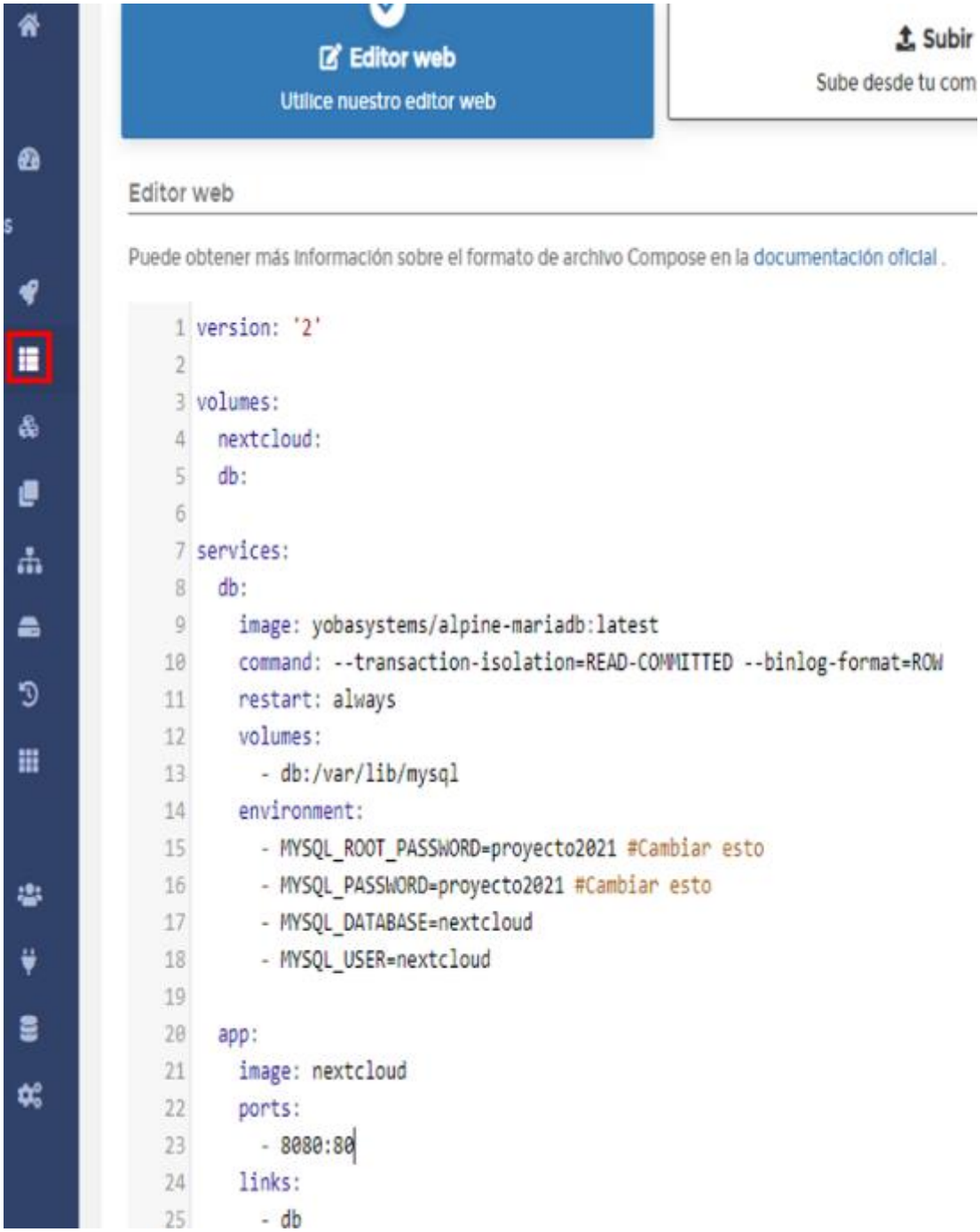


Figura 8. Código de Instalación de Nextcloud, (Print Screen).
Fuente: Propia del autor.

10. Habilitar la implementación del contenedor – Habilitar el control de acceso.
    - ◙ Click en Habilitar el control de acceso.
    - ◙ Click en implementación de contenedor.

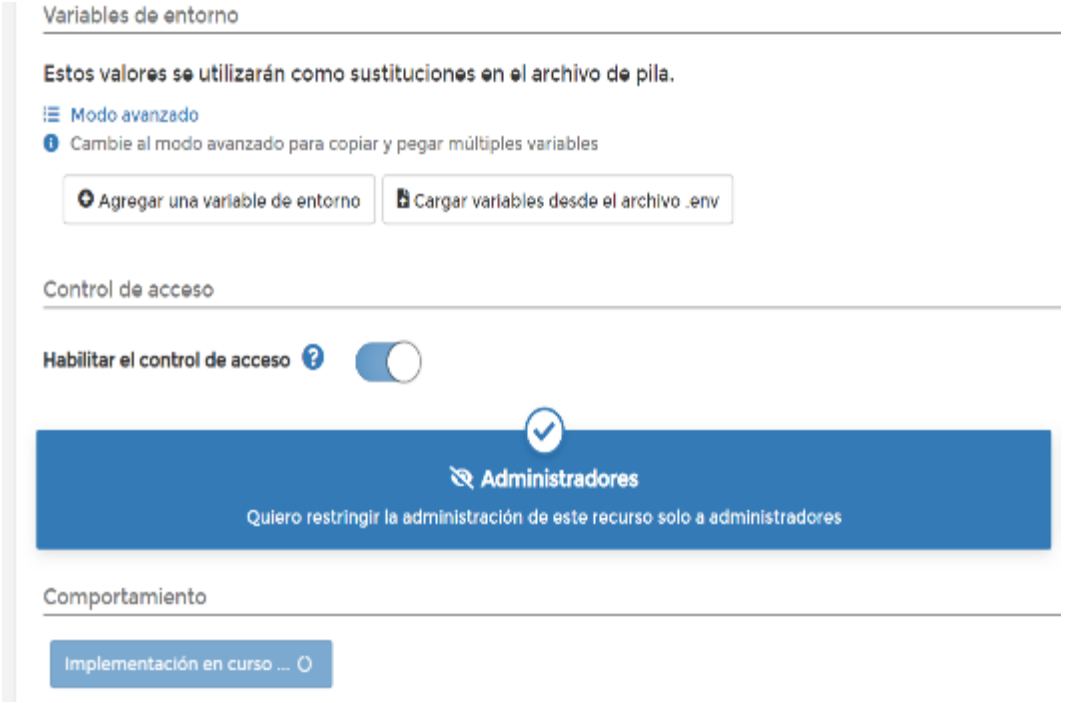


Figura 9. Generando Credenciales del contenedor de Nextcloud, (Print Screen).
Fuente: Propia del autor.

11. Podemos verificar el nuevo contenedor creado, por línea de código o mediante el administrador de contenedores Portainer.

Figura 10. Administración del contenedor de Nextcloud, (Print Screen).
Fuente: Propia del autor.

12. Ingresar a NextCloud desde un navegador apuntando a la dirección localhost y el puerto (8080) correspondiente de la ip que le brindo el Router LAN a la cual está conectada la Raspberry Pi.

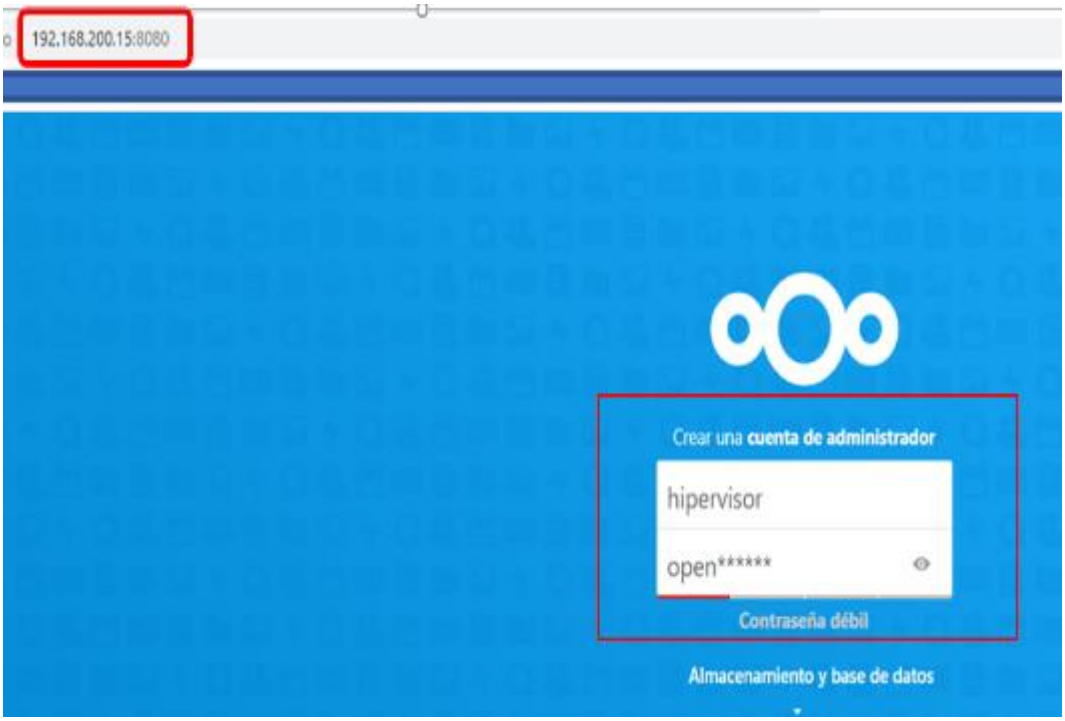


Figura 11. Acceso al Portal de Next Cloud, (Print Screen).
Fuente: Propia del autor.

13. Cargamos la plantilla de servicios - click en instalar las aplicaciones.

Figura 12. Cargando los Servicios al Vault de Almacenamiento en la Red, (Print Screen).
Fuente: Propia del autor.

14. Se detallan los servicios a disposición de los usuarios en la Nube.

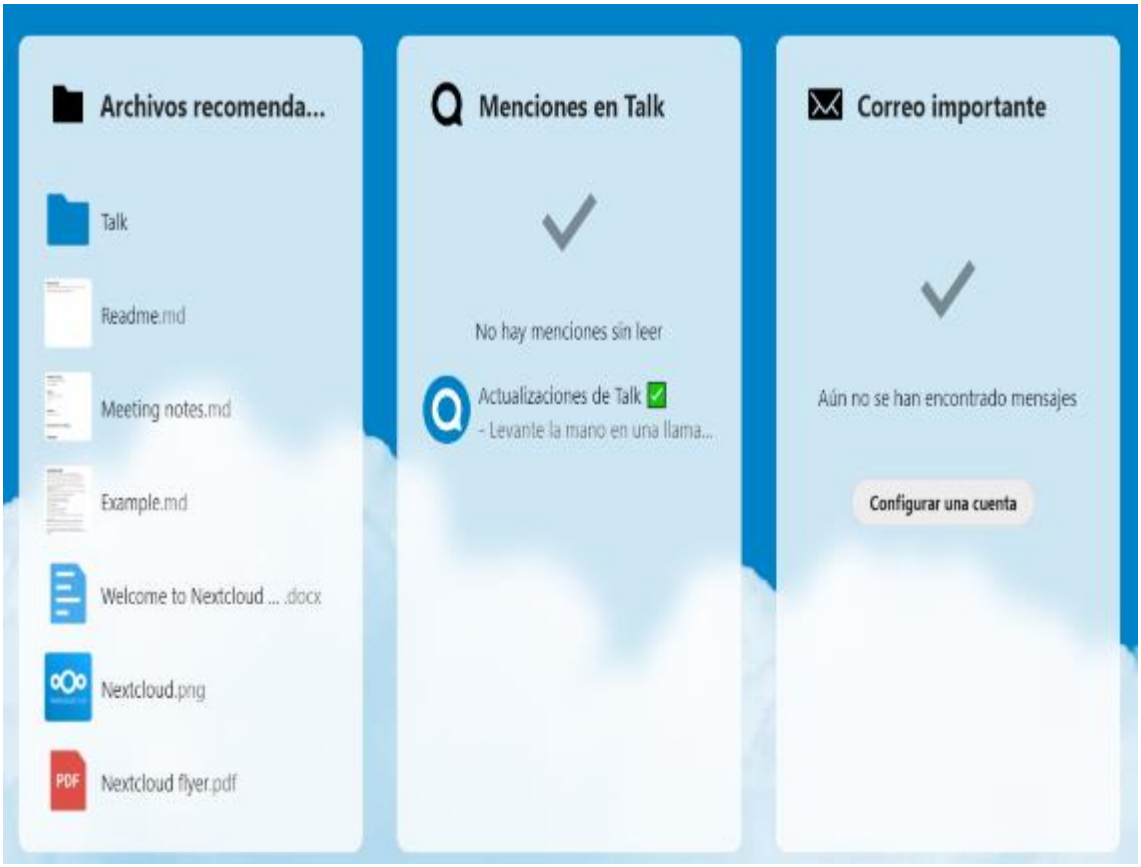


Figura 13. Servicios Disponibles en la Nube Personal, (Print Screen).
Fuente: Propia del autor.

Proyectos de investigación vinculados

Docker Inc. (31 de 08 de 2021). Descripción general de Docker. Obtenido de Docker Documentation | Docker Documentation-Docker Inc: https://docs.docker.com/get-started/overview/

Portainer.io. (s.f.). documentation.portainer.io/. Obtenido de Portainer: Container Management GUI for Kubernetes, Docker: https://www.portainer.io/features/platform-management

playonlinew. (act.2021). Instale NEXTCLOUD en una Raspberry Pi 4 con OpenMediaVault y Docker. Obtenido de Instale NEXTCLOUD en una Raspberry Pi 4 ... - PlayOnlineW: http://www.playonlinew.com/nextcloud

Guía de Instalación de Plataforma para IoT con Home Assistant

DATOS INFORMATIVOS

Recursos Utilizados

- Raspberry Pi.
- Disco solido SSD.
- Tarjeta microSD.
- Conexión a red de internet (ethernet o inalámbrica).
- Imagen FreeNAS OpenMediaVault (OMV).
- Imagen de Docker Host.
- Imagen de Portainer.
- Imagen de Home Assistant.

Preparación

- Raspberry Pi 3B + con sistema operativo instalado y servidor SSH en ejecución.
- Disco solido configurado como sistema de Almacenamiento en la Red.
- Sistema FreeNAS OpenMediaVault cargado en la red del Host, habilitado protocolo ssh.
- Administrar el almacen de directorios en OpenMediaVault.
- Instalar Docker Host desde OMV.
- Instalar Portainer desde OMV.
- Instalar y Configurar el programa Open Source Home Assistant en Docker.

Procedimiento

1. Habilitar la administración de los contenedores, Portainer.
2. Gestionar la capacidad de servicio de FreeNAS OpenMediaVault como almacenamiento en Red de OpenMediaVault.
3. Gestión de los directorios de Home Assistant.
4. Gestión del software Home Assistant como plataformas de comunicaciones para IoT.
5. Pruebas de Comunicación de equipos

Prerrequisitos y co-requisitos

- Trabajo básico en la terminal de linux.
- Funciones de componentes individuales (Administración de contenedores, OpenMediaVault, Portainer, Home Assistant).
- Instalación y configuración de software en la CLI.

DATOS DE LA PRÁCTICA

Objetivo general

Instalar Home Assistant para la comunicación de plataformas IoT, adaptado al hardware de minicomputadoras como la Raspberry Pi en contenedores de Docker.

Objetivos específicos

- Instalar el software open source Home Assistant en docker para RaspBerry Pi
- Realizar la configuración básica de los directorios del software Home Assistant.

Glosario

NAS: Almacenamiento Conectado en Red o NAS (Network Attached Storage).

ARM: (Acorn RISC Machine – CPU de Arquitectura basada en RISC).

RISC: (Reduced Instruction Set Computer – Ordenador con un Conjunto de Instrucciones Reducidas). Es un tipo de diseño de microprocesadores generalmente utilizados en computadoras que está a favor de conjuntos de instrucciones pequeñas y simples que toman menor tiempo en ejecutarse.

IoT: Del inglés "Internet Of Things", es decir, "Internet de las cosas". La definición de IoT podría ser la agrupación e interconexión de dispositivos y objetos a través de una red (bien sea privada o Internet, la red de redes).

Marco Teórico

NAS: (Network Attached Storage – Almacenamiento Conectado en Red). Dispositivo de almacenamiento de software libre o comercial con sistema operativo y prestaciones según sea su diseño, accesible desde la red, por ejemplo, para almacenamientos masivos de información, música, backups, etc.

Raspberry Pi: Es una serie de ordenador de placa reducida RISC (arquitectura ARM), desarrollado en Reino Unido por la Raspberry Pi Fundation, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital, abre las puertas de la experimentación en proyectos de electrónica.

IoT con Home Assistant Gracias a la tecnología del software de código abierto Home Assistant permitiremos que los usuarios creen un vínculo de

control para plataformas de IoT en redes seguras y muy fáciles de configurar, home assistant proporciona una interfaz web para configurar en detalle la plataforma de comunicación para sistemas automatizados, los clientes que se pueden conectar a Home Assistant podrán automatizar sus sistemas gestionando datos en la red. (community.home-assistant.io, 2020)

Figura 1. Home Assistant en Raspberry Pi, (Print Screen).
Fuente: https://www.reichelt.com/magazin/en/home-automation-getting-started-with-the-home-assistant-software/

MARCO PROCEDIMENTAL:

1. Script de Instalación de OpenMediaVault

```
# Conceder privilegios de administrador con sudo su

$ wget -O - https://github.com/OpenMediaVault-Plugin- Developers/install Script/raw/master/install | sudo bash


# Si tiene inconsistencia de privilegios ejecute.

$ sudo curl -sSL https://github.com/OpenMediaVault-Plugin-Developers /installScript/raw/master/install | sudo bash

#instalar OMV

$ sudo ./install -n

#Reiniciar el Host

$ sudo reboot
```

Fuente: https://github.com/OpenMediaVault-Plugin-Developers/installScript

2. Gestión de almacenamiento OpenMediaVault. El programa escaneara el disco SSD que esté conectado a la raspberry pi. El almacén se

gestionará directamente en el disco luego de limpiarlo y montarlo, crear un repositorio de carpetas para los registros.

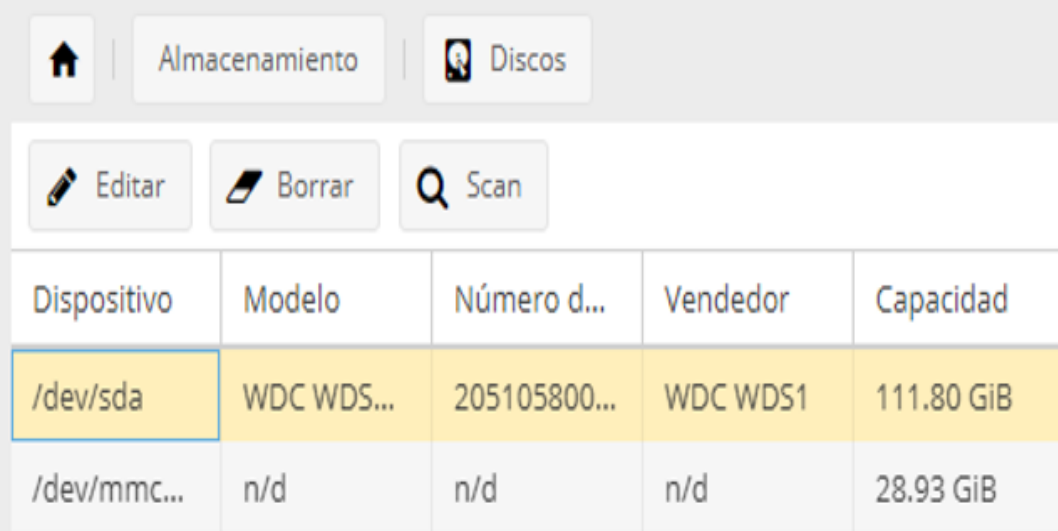


Figura 2. Scanner del Disco Sólido Conectado, (Print Screen).
Fuente: Propia del autor.

3. Instalar el contenedor de Docker y Portainer desde los plugin extras de OMV. Debe direccionar la ruta del almacén de docker a /home/pi/docker.

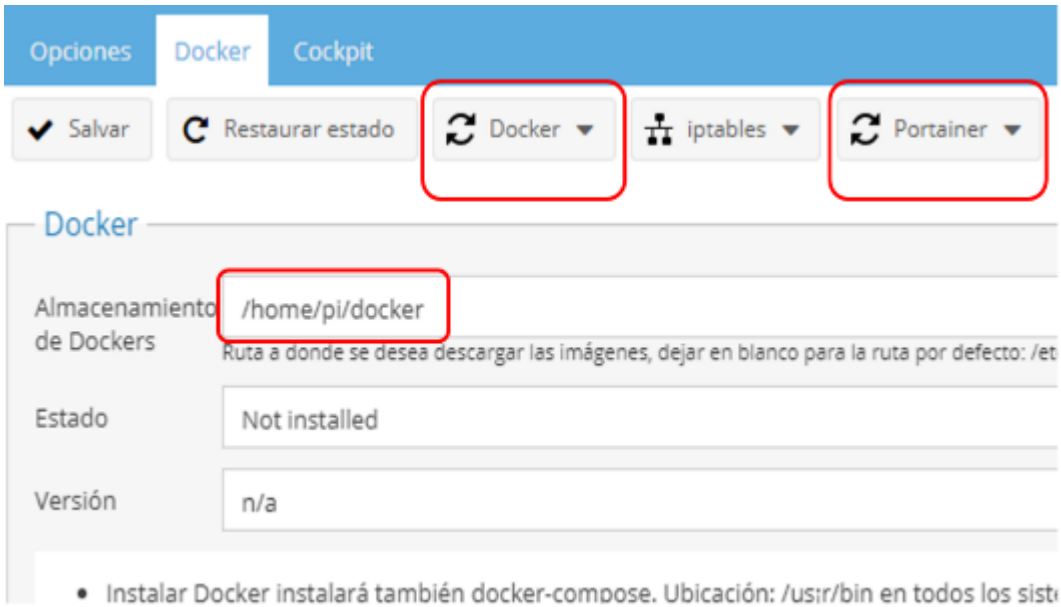


Figura 3. Ruta de Almacenamiento de Plugin de Docker y Portainer, (Print Screen).
Fuente: Propia del autor.

4. Instalación en curso de Docker y Portainer en OpenMediaVault.

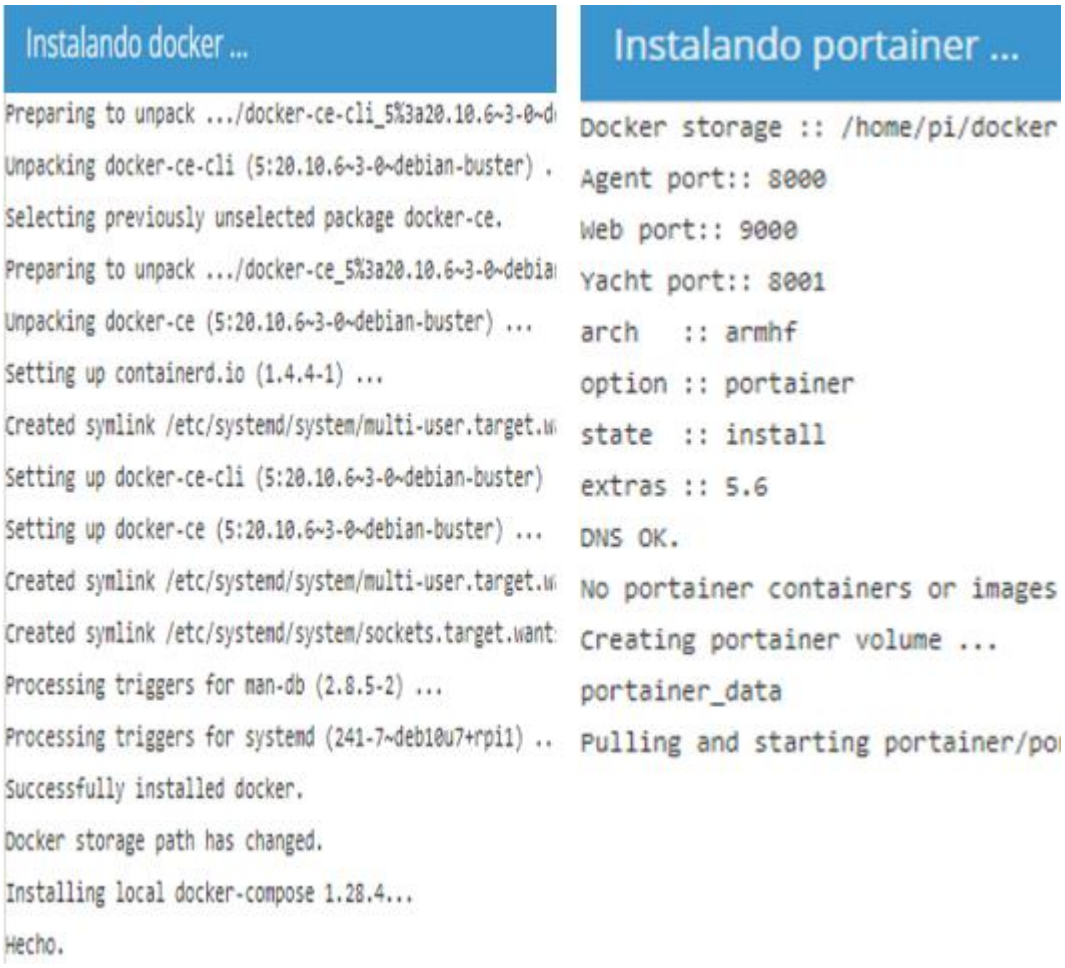


Figura 4. Instalación de Docker y Portainer desde OpenMediaVault, (Print Screen).
Fuente: Propia del autor.

5. Escenario de Administración de contenedores Portainer.

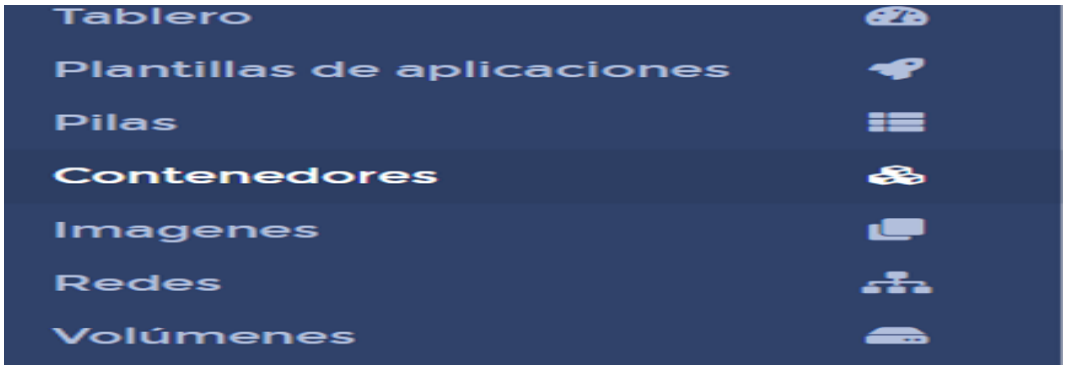


Figura 5. Administración del Contenedor Portainer, (Print Screen).
Fuente: Propia del autor.

6. Crear carpeta con el nombre Cloud en Gestión de Archivos. Añadir carpeta.

Figura 6. Crear Carpeta para Repositorios de los Directorios, (Print Screen).
Fuente: Propia del autor.

7. Script para crear carpeta hassistant en directorio home/pi/docker.

```
# En home/pi/docker crear carpeta nueva

# Conceder privilegios de administrador con sudo su

        $ sudo su

        $ ls

        $ cd docker

        $ mkdir -m 777 /home/pi/docker/hassistant


# ACTUALIZAR LOS REPOSITORIOS

        $ sudo apt update && sudo apt dist-upgrade -y && sudo apt autoremove -y


        $ exit
```

Fuente: https://github.com/Kanga-Who/home-assistant

8. Script de Instalación Home Assistant.

GESTIONAMOS ALGUNAS PROPIEDADES DE RED

$ sudo apt-get install -y software-properties-common apparmor-utils apt-transport-https avahi-daemon ca-certificates curl dbus jq network-manager socat

$ sudo mkdir -m 777 /home/pi/docker/hassistant

VERIFICAR QUE ESTE CREADA LA CARPETA DE HASSISTANT

$ ls
$ cd docker
$ sudo ls

siguientes comandos ejecutará un script de Shell como root en %%pi@raspberrypi:home/pi/docker#

$ sudo curl -sL "https://raw.githubusercontent.com/Kanga-Who/home-assistant/master/supervised-installer.sh" | bash -s -- -m raspberrypi3

Reiniciamos

sudo reboot

Fuente: https://community.home-assistant.io/

9. Podemos administrar el nuevo contenedor creado desde la CLI de Raspberry Pi o desde la GUI de Portainer.

Figura 7. Administración de Contenedores Home Assistant desde Portainer, (Print Screen).
Fuente: Propia del autor.

10. Ingresar a Home Assistant desde un navegador apuntando a la dirección localhost y el puerto (8123) correspondiente de la ip que le brindo el Router LAN a la cual está conectada la Raspberry Pi.

Figura 8. Completando la Instalación de Home Assistant, (Print Screen).
Fuente: Propia del autor.

11. Crear cuenta de usuario en Home Assistant.

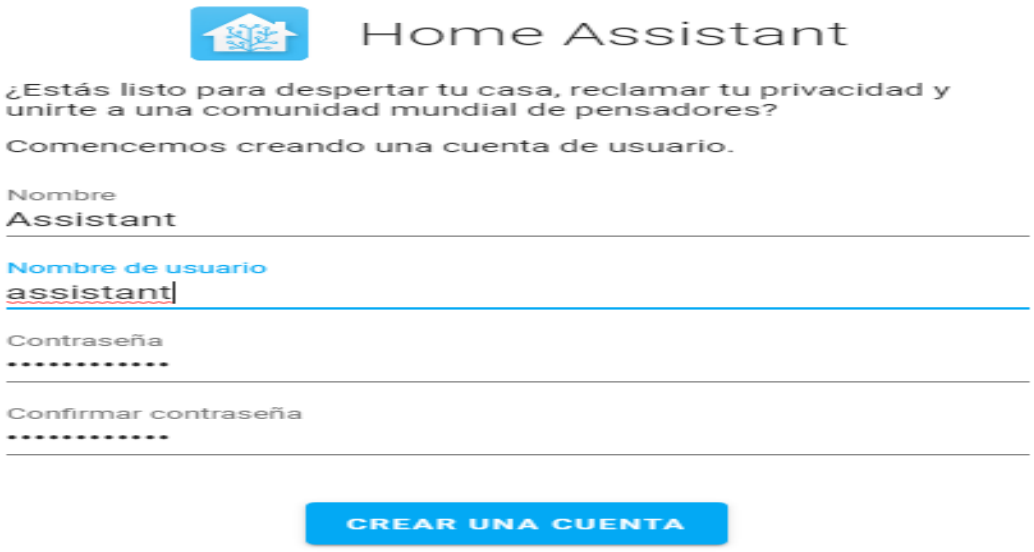


Figura 9. Creando Usuario de la Plantilla Home Assistant, (Print Screen).
Fuente: Propia del autor.

12. Seleccionamos las instancias activas de análisis del sistema Home Assistant.

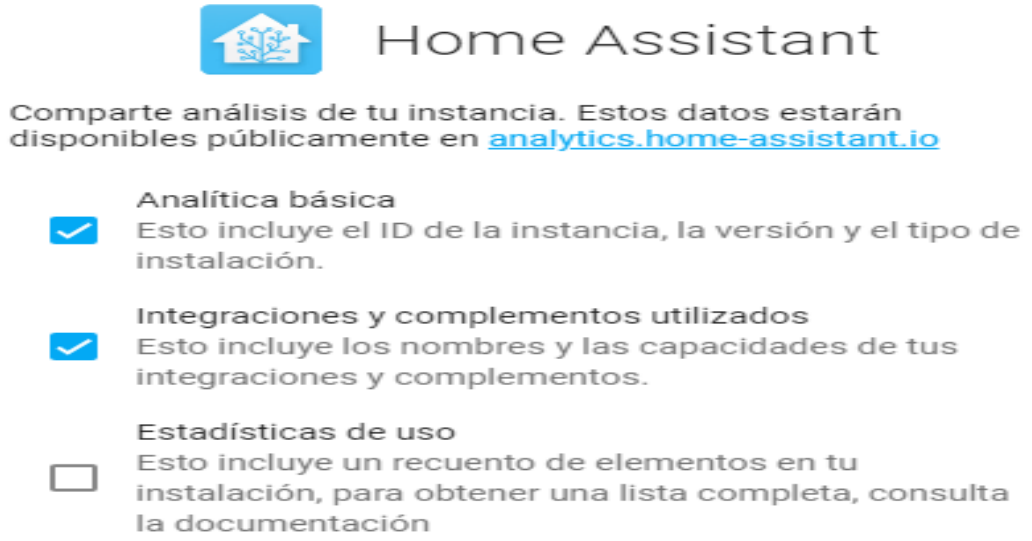


Figura 10. Instancias de Análisis Activas en Home Assistant, (Print Screen).
Fuente: Propia del autor.

13. Panel de Control Supervisor del Asistente de Home Assistant.

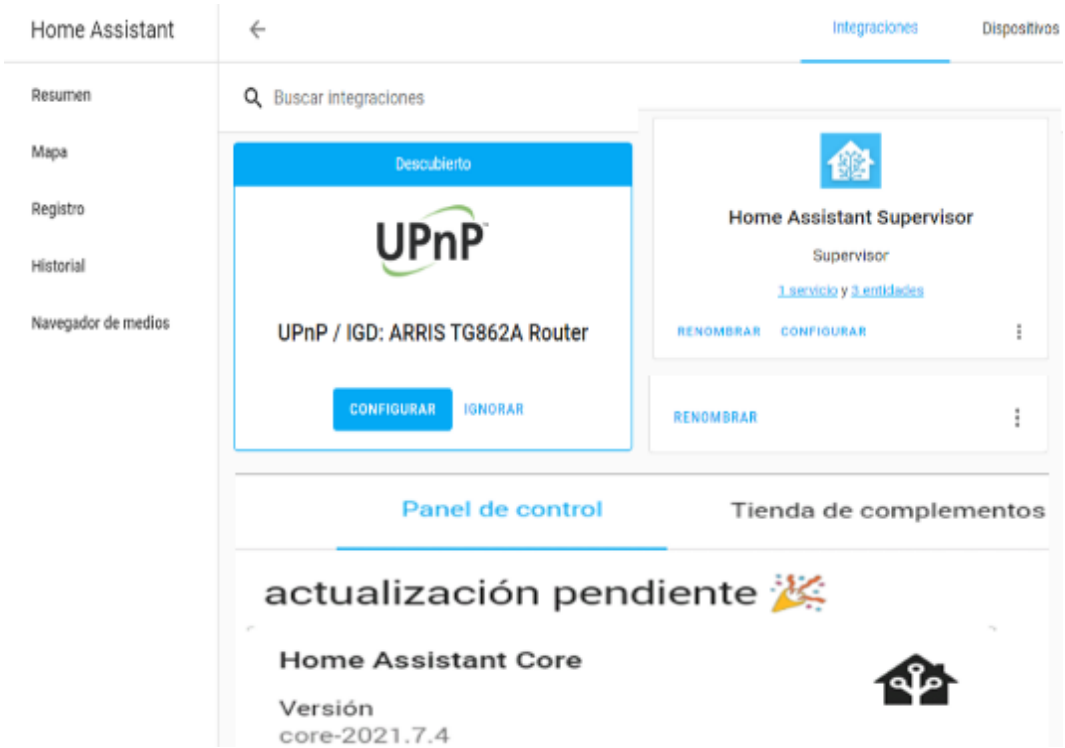


Figura 11. Supervisor Controlador de la Plataforma, (Print Screen).
Fuente: Propia del autor.

14. Plantilla de IoT Operacional – Pruebe sus Integraciones de control.

Figura 12. Plataforma de Comunicación IoT Home Assistant Operativa, (Print Screen).
Fuente: Propia del autor.

Proyectos de investigación vinculados

Docker Inc. (31 de 08 de 2021). Descripción general de Docker. Obtenido de Docker Documentation | Docker Documentation-Docker Inc: https://docs.docker.com/get-started/overview/

Portainer.io. (s.f.). documentation.portainer.io/. Obtenido de Portainer: Container Management GUI for Kubernetes, Docker: https://www.portainer.io/features/platform-management

Community.home-assistant.io. (06 de 2017). Instalación de Home Assistant supervisado en Raspberry Pi OS. Obtenido de Installing Home Assistant Supervised on a Raspberry Pi with: https://github.com/Kanga-Who/home-asistant

Guía de Instalación del Centro Multimedia Jellyfin

DATOS INFORMATIVOS

Recursos Utilizados

- RaspberryPi.
- Disco solido SSD.
- Tarjeta microSD.
- Conexión a red de internet (ethernet o inalámbrica).
- Imagen FreeNAS OpenMediaVault (OMV).
- Imagen de Docker Host.
- Imagen de Portainer.
- Imagen de Jellyfin

Preparación

- Raspberry Pi 3B + con sistema operativo instalado y servidor SSH en ejecución.
- Disco solido configurado como sistema de Almacenamiento en la Red.
- Sistema FreeNAS OpenMediaVault cargado en la red del Host, habilitado protocolo ssh.
- Administrar el almacen de directorios en OpenMediaVault.
- Instalar Docker Host desde OMV.
- Instalar Portainer desde OMV.
- Instalar y Configurar el programa Open Source Jellyfin en Docker.

Procedimiento

1. Habilitar la administración de los contenedores, Portainer.
2. Gestionar la capacidad de servicio de FreeNAS OpenMediaVault como almacenamiento en Red de Jellyfin.
3. Gestión de los directorios de Jellyfin.
4. Gestión del servidor de Jellyfin como servidor Multimedia.
5. Pruebas de Comunicación de equipos

Prerrequisitos y co-requisitos

- Trabajo básico en la terminal de linux.
- Funciones de componentes individuales (Administración de contenedores, OpenMediaVault, Portainer, Jellyfin).
- Instalación y configuración de software en la CLI.

DATOS DE LA PRÁCTICA

Objetivo general

Realizar la configuración básica de los elementos de red y el almacenamiento en la nube para aprovisionar un servicio de entretenimiento Multimedia llamado Jellyfin adaptado al hardware de mini computadoras como la Raspberry Pi en contenedores de Docker.

Objetivos específicos

- Armar un escenario de entretenimiento Multimedia con el software Jellyfin y OpenMediaVault en una Raspberry Pi.
- Realizar la configuración básica de los directorios del software Jellyfin.

Glosario

LAN: Red de Área Local o LAN (Local Área Network). Es una red que conecta equipos en un área relativamente limitada.

NAS: Almacenamiento Conectado en Red o NAS (Network Attached Storage).

ARM: (Acorn RISC Machine – CPU de Arquitectura basada en RISC).

RISC: (Reduced Instruction Set Computer –Ordenador con un Conjunto de Instrucciones Reducidas). Es un tipo de diseño de microprocesadores generalmente utilizados en computadoras que está a favor de conjuntos de instrucciones pequeñas y simples que toman menor tiempo en ejecutarse.

Marco Teórico

NAS: (Network Attached Storage – Almacenamiento Conectado en Red). Dispositivo de almacenamiento de software libre o comercial con sistema operativo y prestaciones según sea su diseño, accesible desde la red, por ejemplo, para almacenamientos masivos de información, música, backups, etc.

Raspberry Pi: Es una serie de ordenador de placa reducida RISC (arquitectura ARM), desarrollado en Reino Unido por la Raspberry Pi Fundation, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital, abre las puertas de la experimentación en proyectos de electrónica.

Centro Multimedia con Jellyfin Gracias a la tecnología del software de código abierto Jellyfin permitiremos que los usuarios creen un vínculo de

servicio Multimedia en redes seguras y muy fáciles de configurar, proporciona una interfaz web para configurar en detalle este servidor privado, los clientes que se conecten al centro multimedia podrá compartir y colaborar en el contendido sin filtraciones de datos, (dockerhub, act.2021)

Figura 1. Jellyfin en Raspberry Pi, (Print Screen).
Fuente: https://pimylifeup.com/raspberry-pi-jellyfin/

MARCO PROCEDIMENTAL:

1. Script de Instalación de OpenMediaVault.

```
# Conceder privilegios de administrador con sudo su

$ wget -O - https://github.com/OpenMediaVault-Plugin- Developers/installScript/raw/master/install | sudo bash


# Si tiene inconsistencia de privilegios ejecute.

$ sudo curl -sSL https://github.com/OpenMediaVault-Plugin-Developers/installScript/raw/master/install | sudo bash

#instalar OMV

$ sudo ./install -n

#Reiniciar el Host

$ sudo reboot
```

Fuente: https://github.com/OpenMediaVault-Plugin-Developers/installScript

2. Gestión de almacenamiento OpenMediaVault. El programa escaneara el disco SSD que esté conectado a la raspberry pi. El almacén se

gestionará directamente en el disco luego de limpiarlo y montarlo, crear un repositorio de carpetas para los registros.

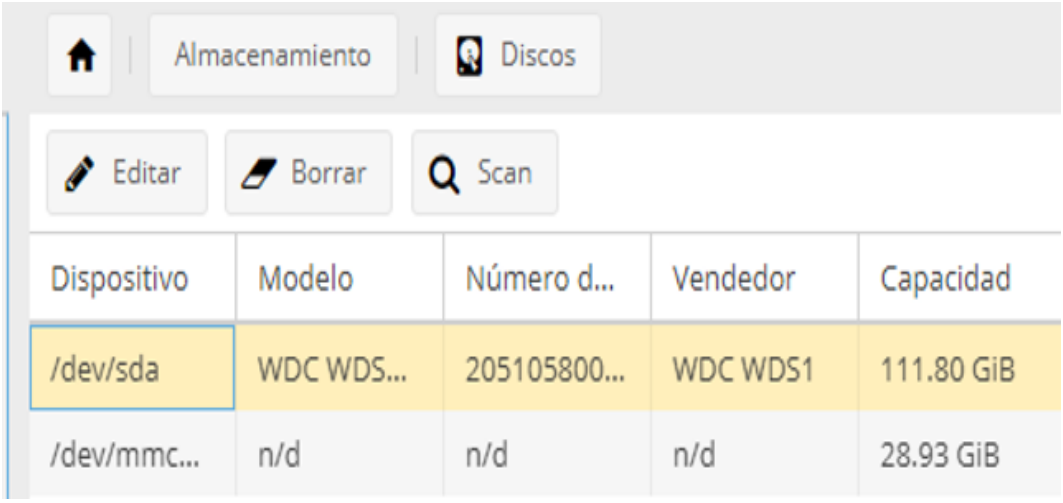


Figura 2. Scanner del Disco Sólido Conectado, (Print Screen).
Fuente: Propia del autor.

3. Instalar el contenedor de Docker y Portainer desde los plugin extras de OMV. Debe direccionar la ruta del almacén de docker a /home/pi/docker.

Figura 3. Ruta de Almacenamiento de Plugin de Docker y Portainer, (Print Screen).
Fuente: Propia del autor.

4. Escenario de instalación de docker y portainer.

Figura 4. Instalación de Docker y Portainer desde OpenMediaVault, (Print Screen).
Fuente: Propia del autor.

5. Crear carpeta con el nombre Cloud en Gestión de Archivos. Añadir carpeta.

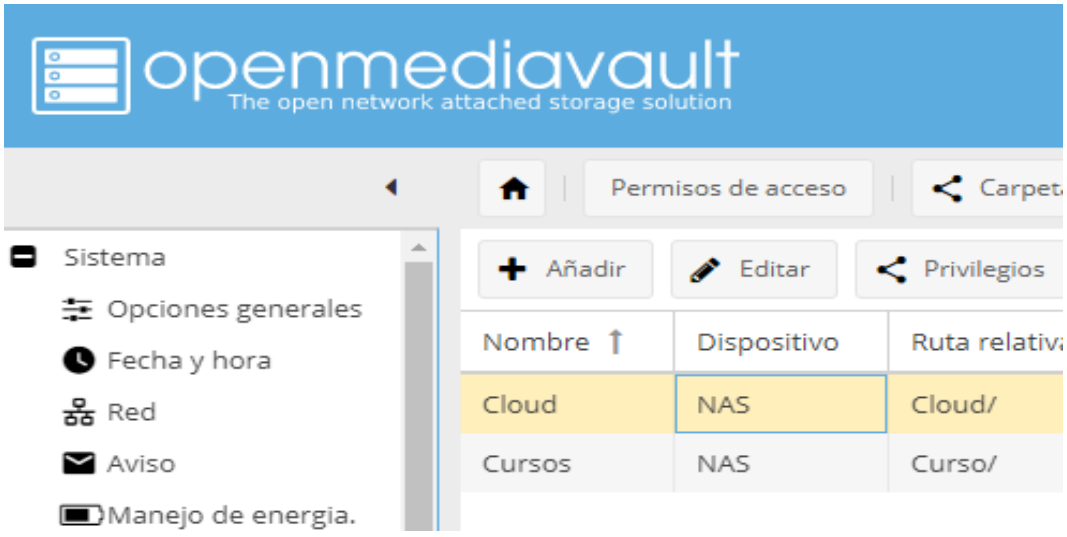


Figura 5. Crear Carpeta para Repositorios de los Directorios, (Print Screen).
Fuente: Propia del autor.

6. Crear carpetas para el servidor de medios Emby. Con el nombre multimedia, media, movies, series, tv, etc. Gestión de Archivos de

OMV. Siga los pasos descritos en OpenMediaVault para crear carpetas compartidas.

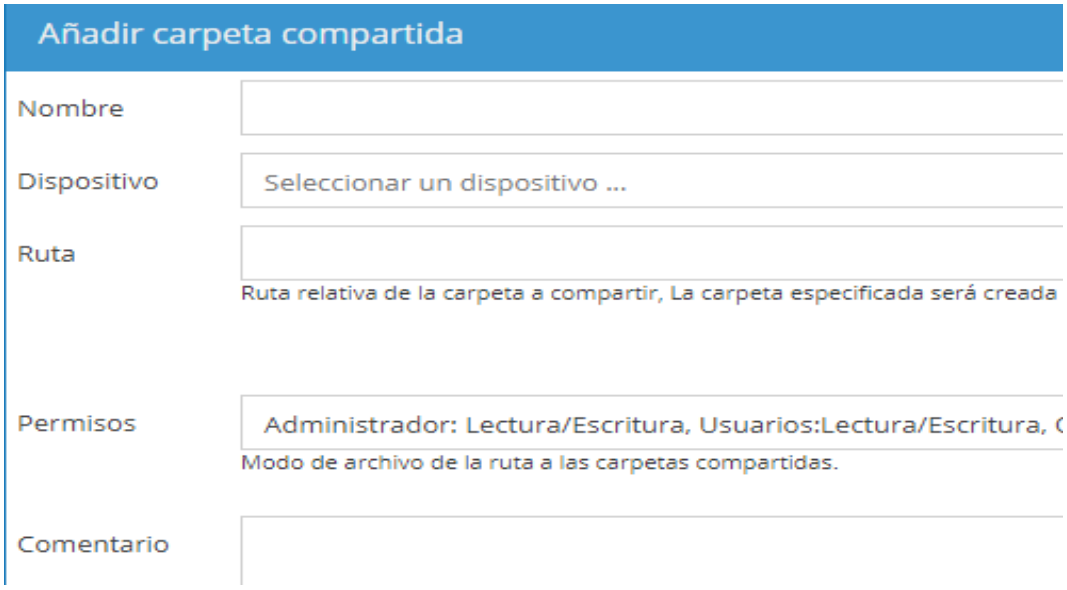


Figura 6. Gestión de Carpetas para el Servidor de Medios Emby, (Print Screen).
Fuente: Propia del autor.

7. Escenario de carpetas compartidas para el servidor de medios.

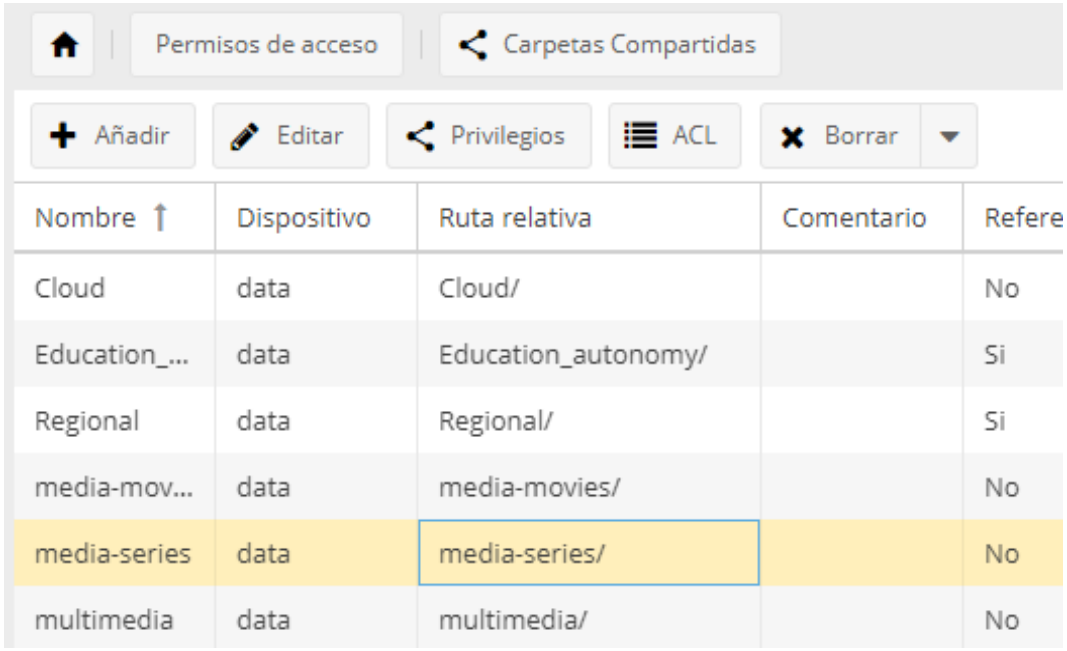


Figura 7. Carpetas Compartidas para el Servidor de Medios Emby, (Print Screen).
Fuente: Propia del autor.

8. Gestionar ruta absoluta para los directorios emby.

◙ Click en Ruta Relativa >> click derecho.
- ✓ Elija visualizar ruta absoluta de la carpeta.
- ✓ Copie la ruta absoluta, esta deberá ser la ruta donde se cargarán los medios del contenedor de jellyfin.
- ✓ Regrese la visualización a ruta relativa de carpetas.

◙ Actualizar para guardar los cambios.

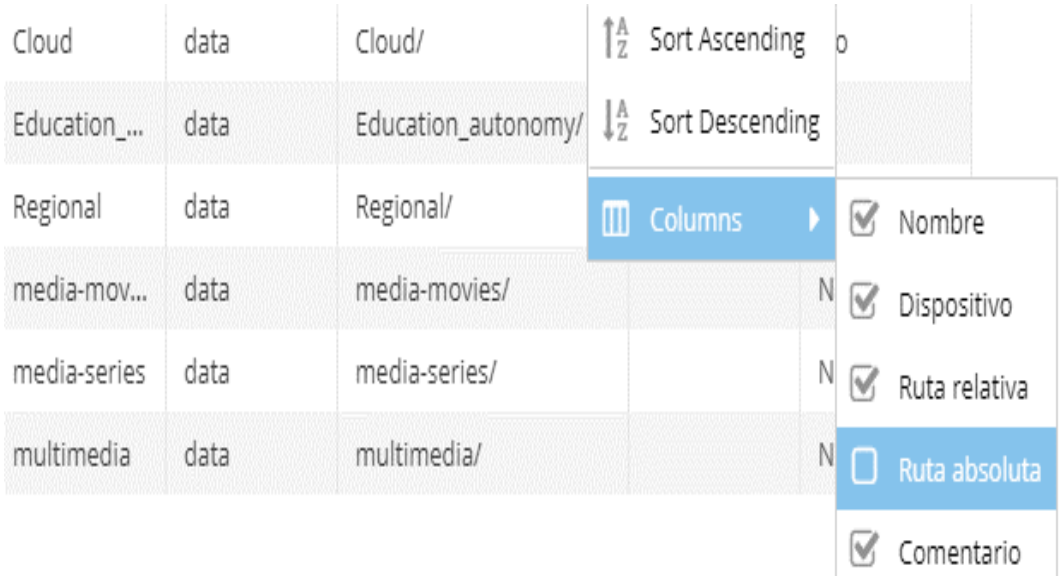


Figura 8. Gestionar Ruta Absoluta en el Repositorio de Medios Multimedia, (Print Screen).
Fuente: Propia del autor.

9. Visualización de rutas absolutas de cada repositorio.

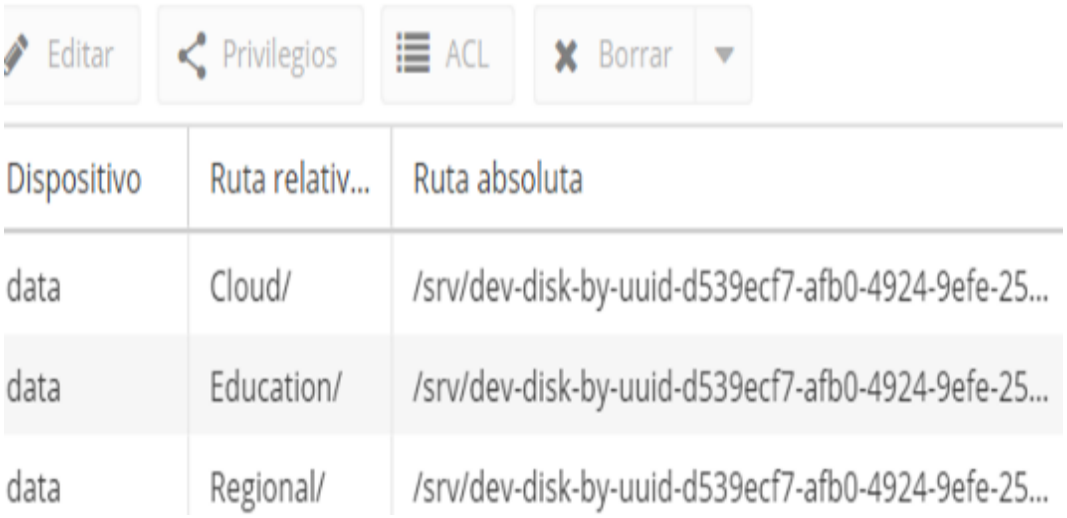


Figura 9. Rutas Absolutas de las Carpetas de Medios, (Print Screen).
Fuente: Propia del autor.

10. Código fuente para implementar la pila del contenedor Multimedia Jellyfin.

```
---
version: "2.1"
services:
  jellyfin:
    image: ghcr.io/linuxserver/jellyfin
    container_name: jellyfin
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Ecuador
      - JELLYFIN_PublishedServerUrl=10.242.10.40
  volumes:
    -/srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-f070af1eebb/multimedia/jellyfin:/config
    -/srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/media-series:/data/series
    -/srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/media-movies:/data/movies
    - /srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/media-tv:/data/tv

  ports:
    - 8096:8096
    - 7359:7359/udp #optional
  restart: unless-stopped
```

Fuente: Adaptado de https://hub.docker.com/r/linuxserver/jellyfin

11. Gestionar la Pila de Jellyfin desde el administrador Portainer.

- ◙ Click en Pila
  - ✓ Escriba el nombre del nuevo contenedor “jellyfin”.
- ◙ Click en Editor Web.
  - ✓ Copie el código del literal 10 y péguelo en el Editor Web.
  - ✓ Cambie los valores de la ruta personal.

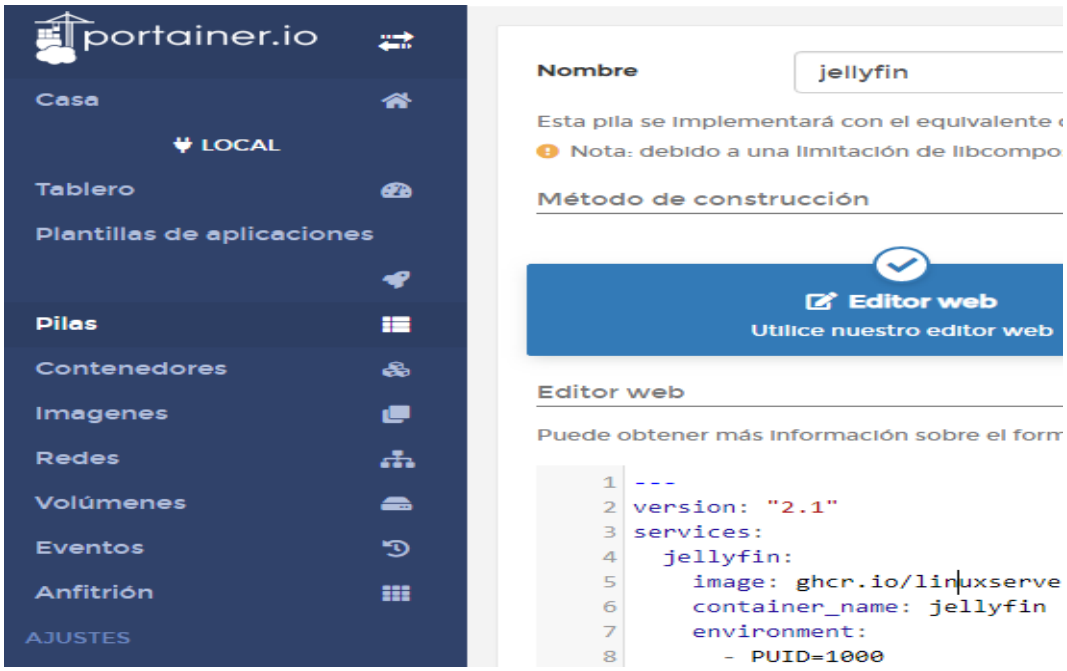


Figura 10. Pila de Jellyfin desde el Administrador Portainer, (Print Screen).
Fuente: Propia del autor.

12. Implementación del contenedor - Habilitar el control de acceso.

```
jellyfin:
  image: ghcr.io/linuxserver/jellyfin
  container_name: jellyfin
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=America/New_York
    - JELLYFIN_PublishedServerUrl=10.242.10.40 #optional
  volumes:
    - /srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/
    - /srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/
    - /srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/
    - /srv/dev-disk-by-uuid-cbd8cf2d-75e3-4ea1-8dac-6f070af1eebb/
  ports:
    - 8096:8096
    - 7359:7359/udp #optional
  restart: unless-stopped
```

Figura 11. Implementación del Contenedor de Jellyfin, (Print Screen).
Fuente: Propia del autor.

13. Podemos verificar el nuevo contenedor creado, por línea de código en la terminal o mediante el administrador de contenedores Portainer

Figura 12. Contenedor Jellyfin Creado Satisfactoriamente, (Print Screen).
Fuente: Propia del autor.

14. Ingresar a Jellyfin desde un navegador apuntando a la dirección localhost y el puerto (8096) correspondiente de la ip que le brindo el Router LAN a la cual está conectada la Raspberry Pi.

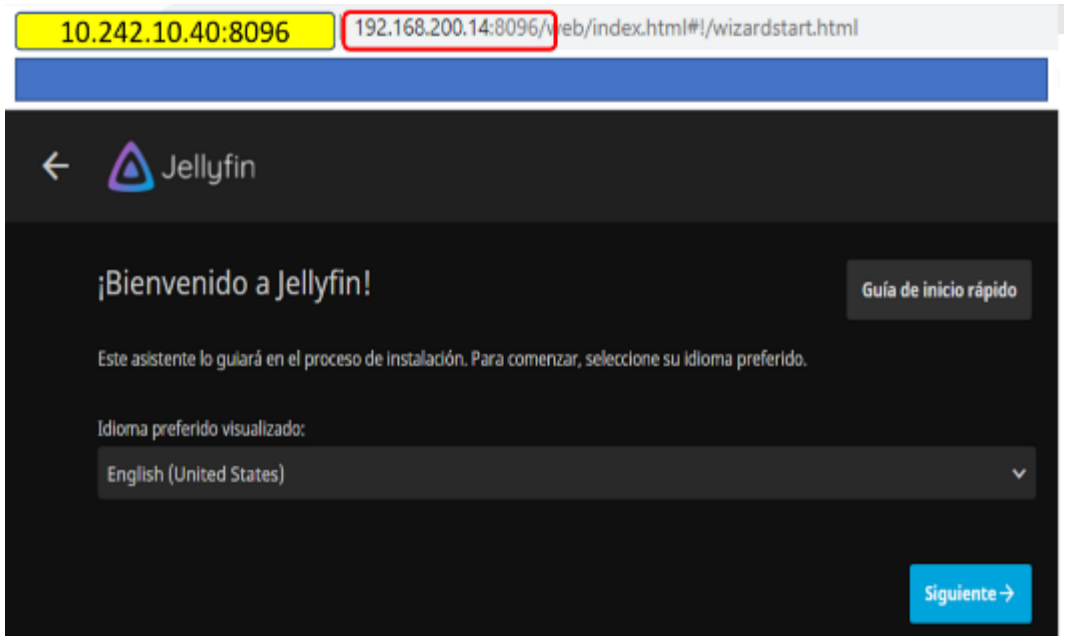


Figura 13. Acceso al Portal de Jellyfin, (Print Screen).
Fuente: Propia del autor.

15. Pantalla de bienvenida configuración del Administrador de Jellyfin.

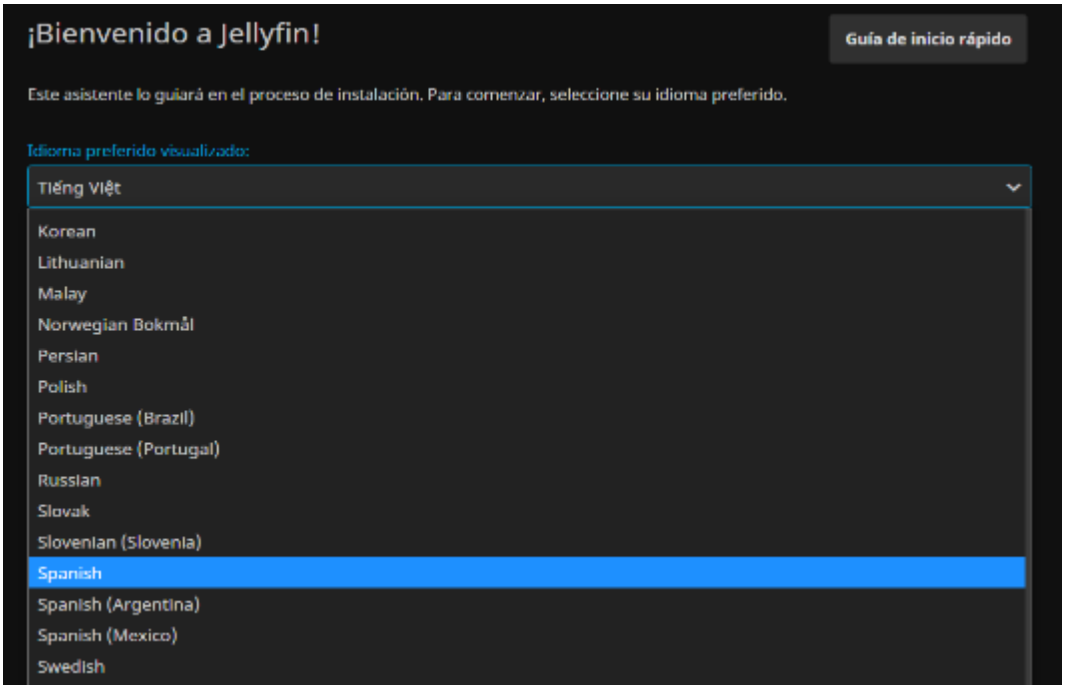


Figura 14. Configurar la Interfaz Administrador de Jellyfin, (Print Screen).
Fuente: Propia del autor.

16. Registrar cuenta de administrador.

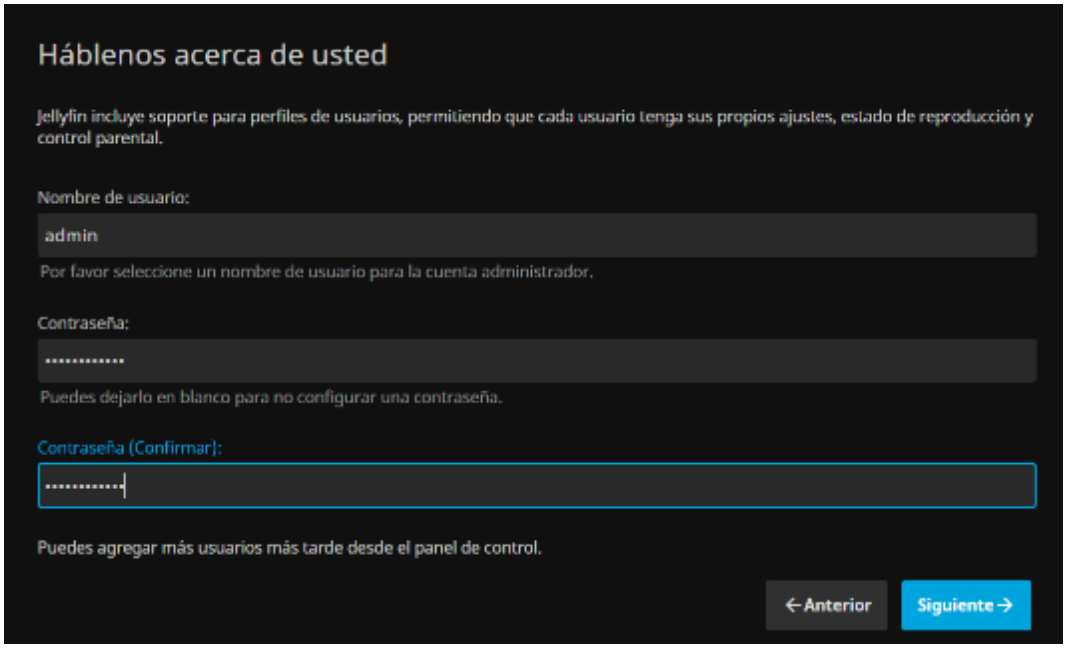


Figura 15. Cuenta de Administrador de Jellyfin, (Print Screen).
Fuente: Propia del autor.

17. Inicio de sesión de Administrador.

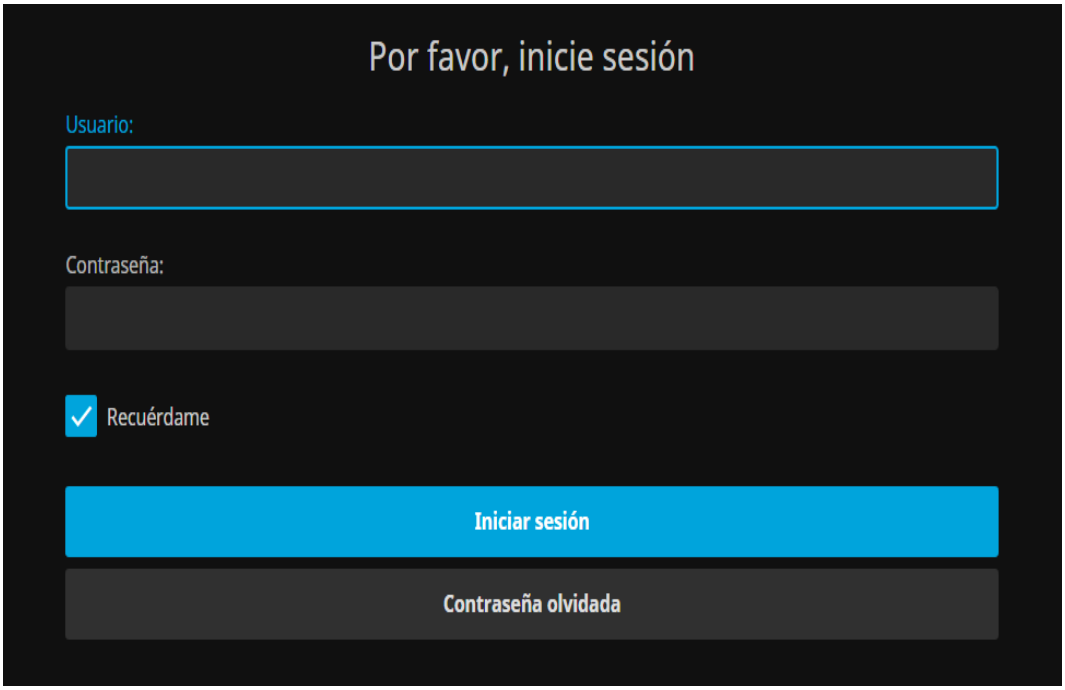


Figura 16. Inicio de Sesión Administrador, (Print Screen).
Fuente: Propia del autor.

18. Configurar la biblioteca de medios multimedia.

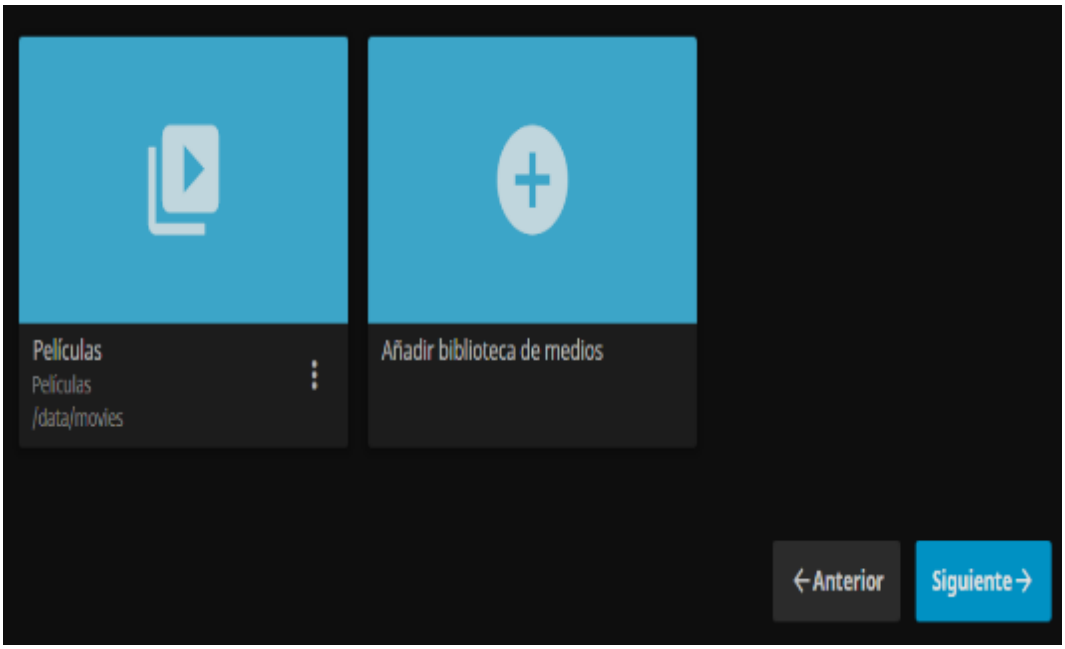


Figura 17. Biblioteca de Medios Multimedia, (Print Screen).
Fuente: Propia del autor.

19. Conectar carpeta de medios Emby. Para generar contenido, deberá cargar sus archivos en las carpetas de medio que creo en el OMV.

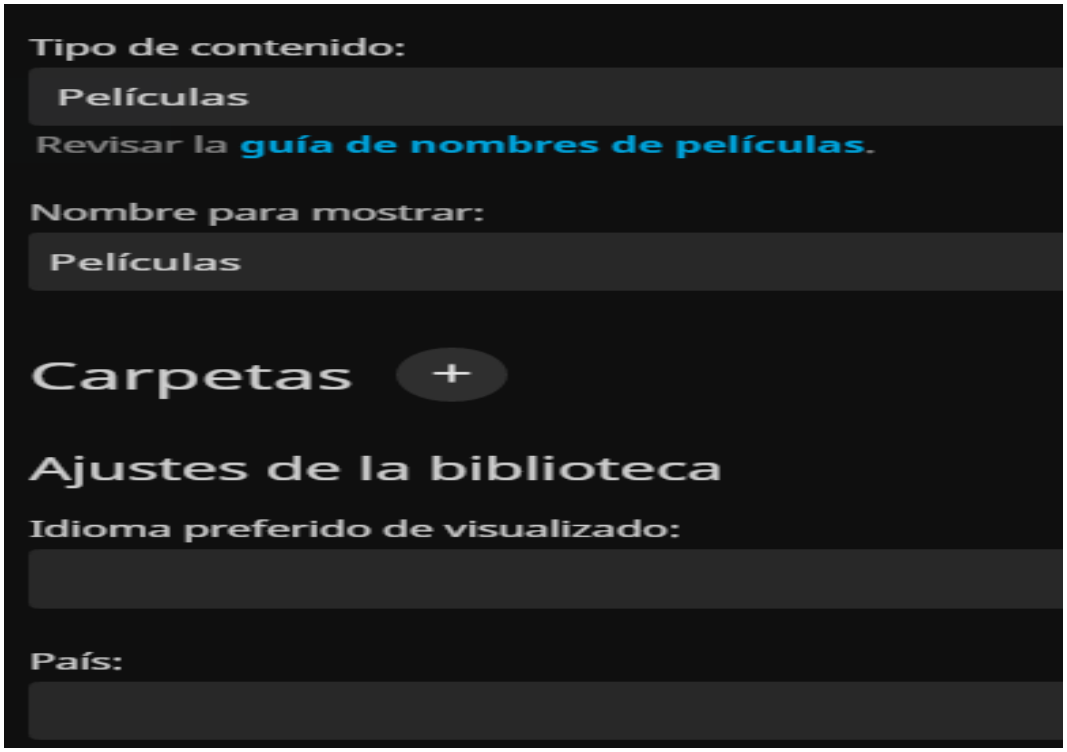


Figura 18. Carpeta de Medios Emby, (Print Screen).
Fuente: Propia del autor.

20. Direccionar la ruta de data/medios hacia los repositorios de OMV.

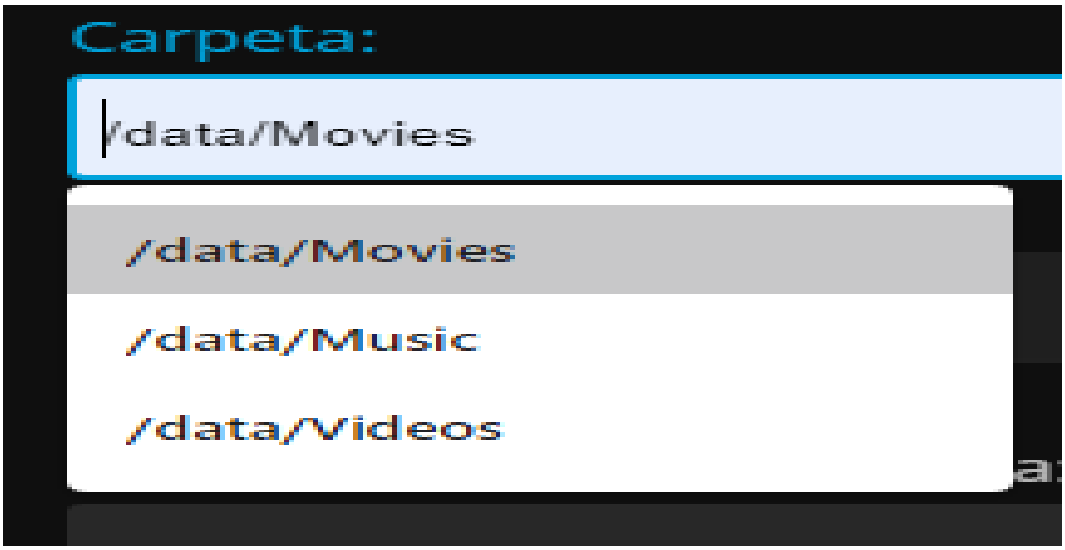


Figura 19. Ruta de data/medios, (Print Screen).
Fuente: Propia del autor.

21. Seleccionar el idioma de etiquetas para el contenido.

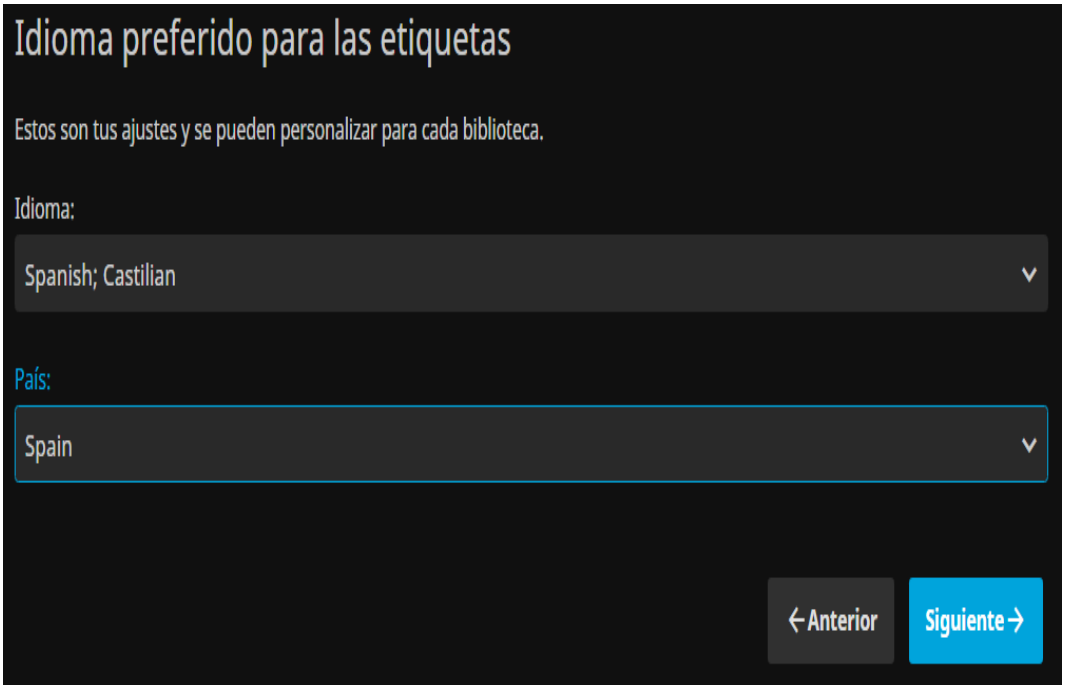


Figura 20. Idioma de Etiquetas, (Print Screen).
Fuente: Propia del autor.

22. Configurar medio de Televisión en directo.

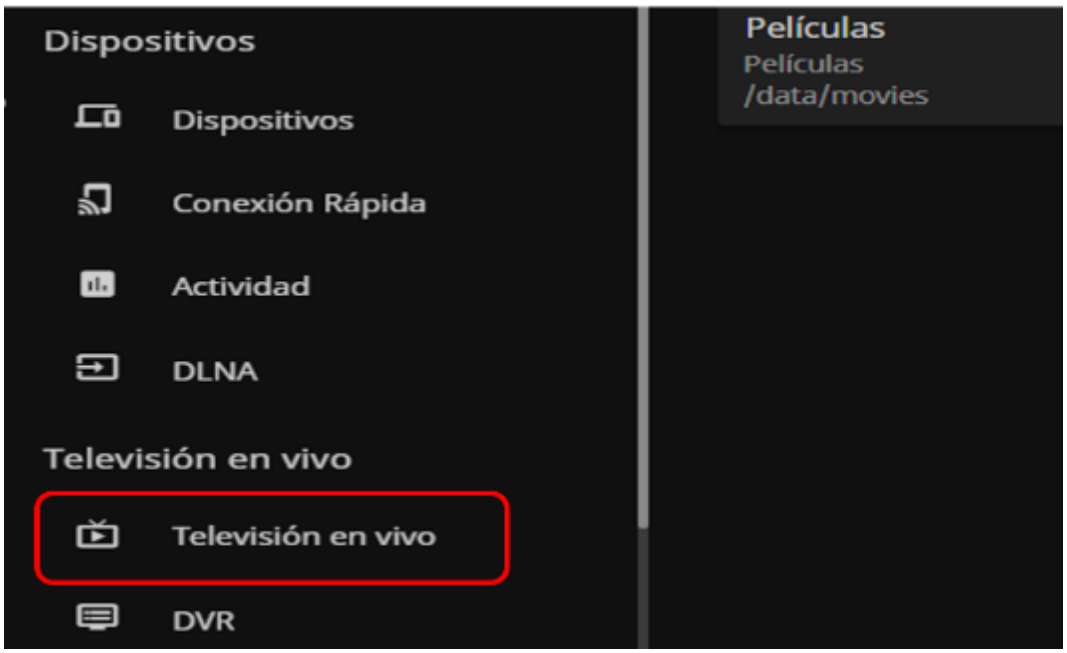


Figura 21. Configurar la Televisión en Directo, (Print Screen).
Fuente: Propia del autor.

23. Seleccionar el tipo lista del sintonizador – M3U Tuner.

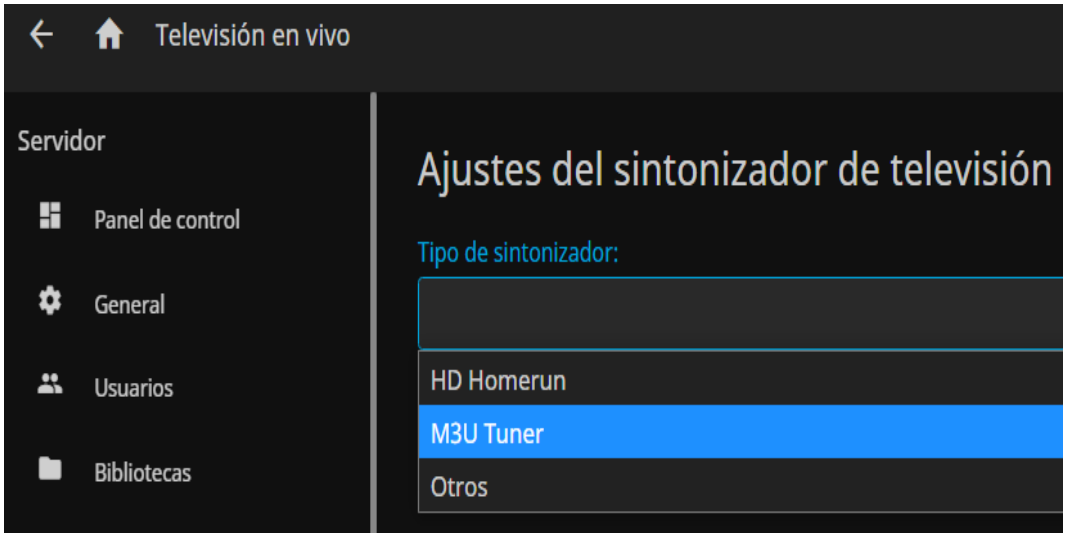


Figura 22. Lista del Sintonizador – M3U Tuner, (Print Screen).
Fuente: Propia del autor.

24. Seleccionamos la url de la lista de canales de su preferencia - grabar

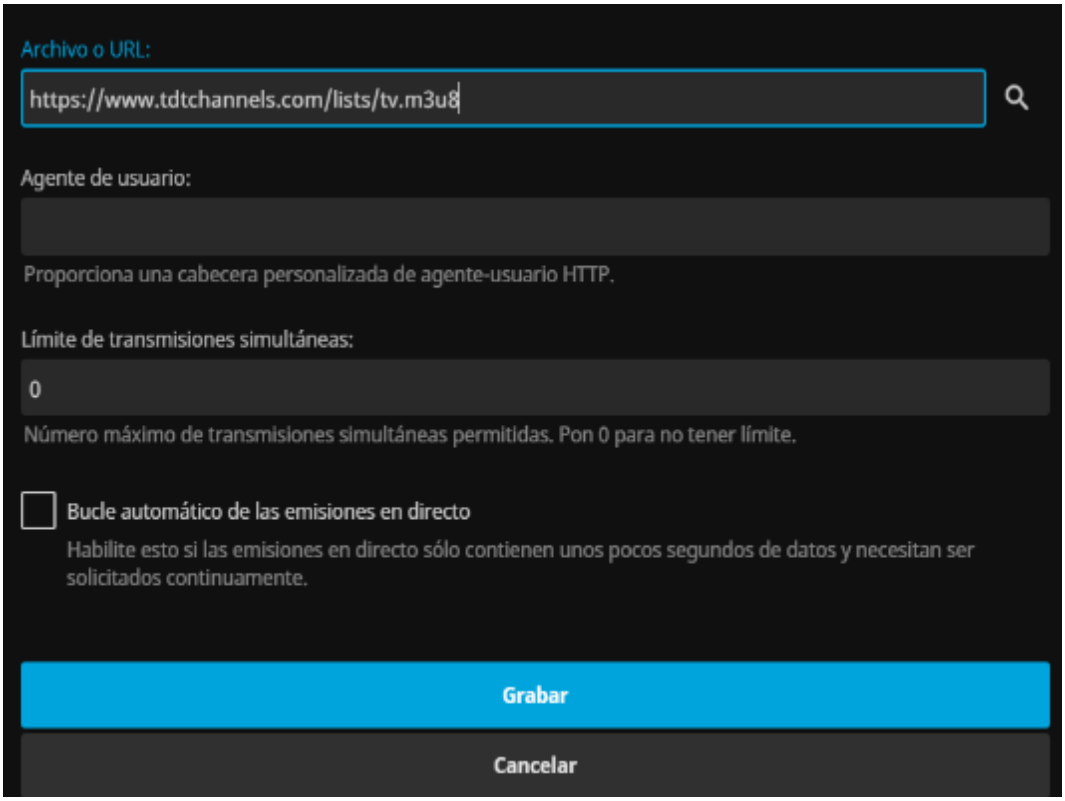


Figura 23. URL de la Lista de Canales, (Print Screen).
Fuente: Propia del autor.

25. Actualizar la instalación – URL Data de medios.

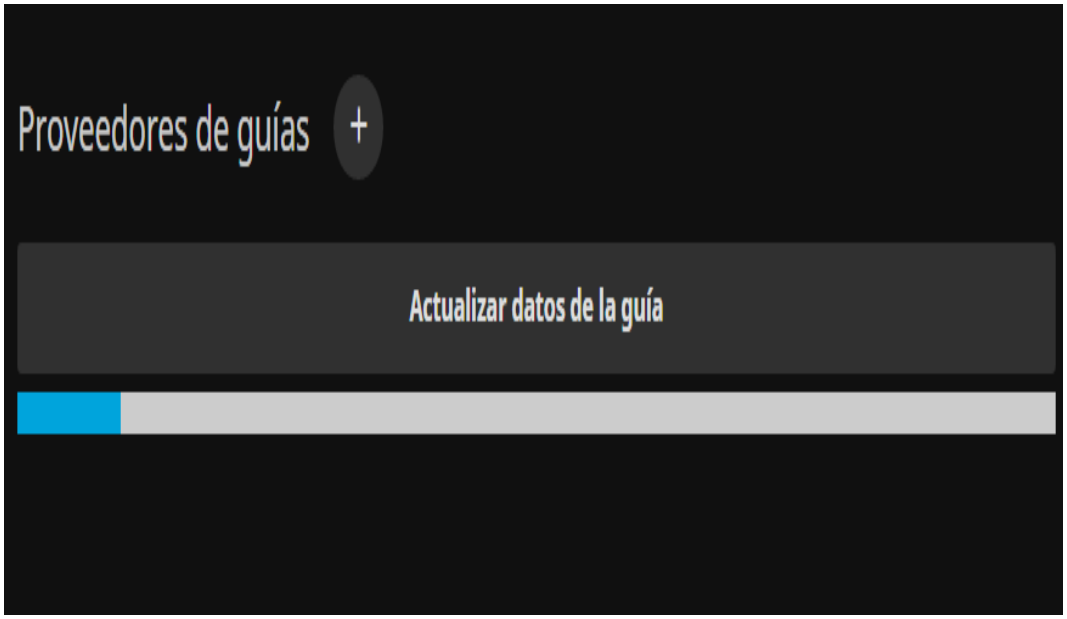


Figura 24. Actualizar la Instalación, (Print Screen).
Fuente: Propia del autor.

26. Operación del Centro Multimedia.

Figura 25. Operación del Centro Multimedia, (Print Screen).
Fuente: Propia del autor.

27. Medio Multimedia Operativo. Ahora puede disfrutar de su centro Multimedia.

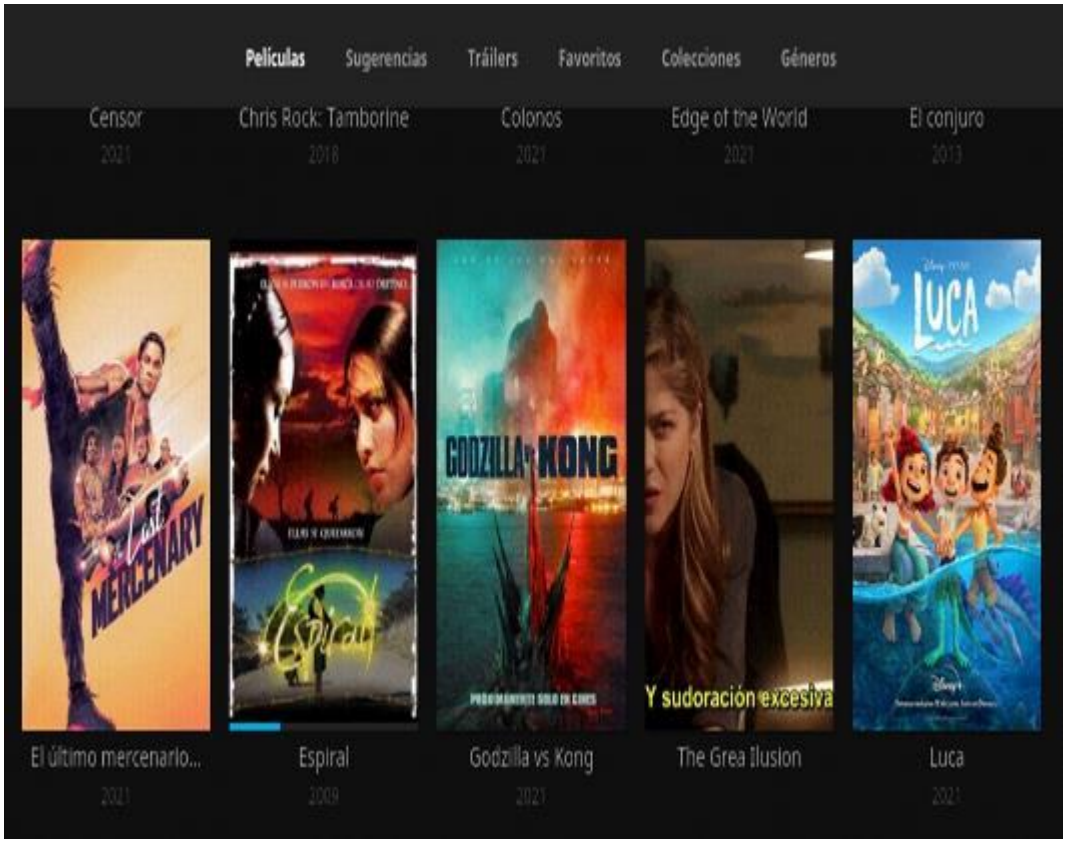


Figura 26. Medio Multimedia Operativo, (Print Screen).
Fuente: Propia del autor.

Proyectos de investigación vinculados

Docker Inc. (31 de 08 de 2021). Descripción general de Docker. Obtenido de Docker Documentation | Docker Documentation-Docker Inc: https://docs.docker.com/get-started/overview/

Portainer.io. (s.f.). documentation.portainer.io/. Obtenido de Portainer: Container Management GUI for Kubernetes, Docker: https://www.portainer.io/features/platform-management

dockerhub. (act.2021). linuxserver / jellyfin. Obtenido de linuxserver/jellyfin - Docker Image: https://hub.docker.com/r/linuxserver/jellyfin

9.6. ANEXO F. Acrónimos

ACI: Infraestructura Centrada de Aplicaciones (Application Centric Infrastructure).
AMD: Micro Dispositivos Avanzado (Advanced Micro Devices).
API: Interface de Programación de Aplicaciones (Application Programming Interface).
AP: Punto de Acceso (Access Point).
ARP. Protocolo de Resolución de Direcciones (Address Resolution Protocol).
ATS: Interruptor de Transferencia Automática (Automatic Transfer Switch).
BGP: Protocolo de Puerta de Enlace (Border Gateway Protocolo).
CERT: Equipo de Respuesta Emergencias Informáticas. (Computer Emergency Response Team)
CLI: Interfaz de Línea de Comando (Command Line Interface).
CPU: Unidad de Procesamiento Central (Central Processing Unit).
DCI: Centro de Datos de Interconexión. (Data Center Interconnection).
DHCP: Protocolo de Configuración Dinámica de Host (Dynamic Host Configuration Protocol).
FTP: Protocolo de Transferencia de Archivos (File Transfer Protocol).
GUI: Interfaz Gráfica de Usuario (Graphical User Interface).
HTTPS: Protocolo de transferencia de hipertexto Seguro (Hyper Text Transfer Protocol Secure).
IBN: Redes Basadas en Intención (Intent - Based Network).
ICT: Tecnologías de la Información y Comunicación(Information and Communication Technology)
IEEE: Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical-Electronics Engineers
IP: Protocolo de Internet (Internet Protocol).
ISP: Proveedor de Servicios de Internet (Internet Services Provider) .
IT: Tecnología de la Información (Information Technology).
ITU: Unión Internacional de Telecomunicaciones (International Telecommunication Union).
LSP: Rutas Conmutadas Etiquetadas (Labeled Switched Paths).
JSON: Notación de Objetos JavaScript (JavaScript Object Notation).
MANO: Gestión y Orquestación (Management and Orchestration).
MPLS: Conmutación de Etiquetas Multiprotocolo (MultiProtocol Label Switching).
NAT: Traducción de Direcciones de Red (Network Address Translation).
NAS: Almacenamiento Enlazado a Red (Network Attached Storage).
NE: Elementos de Red (Network Elements).
NETCONF: Protocolo de Configuración de Red (Network Configuration Protocol).
NFV : Virtualización de las Funciones de Red (Network Functions Virtualization).
NMS: Estación de Administración de Red (Network Management Station).
ONF: Fundación de Red Abierta (Open Networking Fundation).
OSGi: Iniciativa de Puerta de Enlace de Servicios Abiertos (Open Services Gateway Initiative).
OSI: Interconexión de Sistemas Abiertos (Open System Interconnection).
OSS: Sistema de Soporte Operativo (Operational Support System).
OVSDB: Base de Datos Abierta de vSwitch (Open vSwitch Database).
QoS: Calidad de Servicio (Quality of Service).
REST: Transferencia de Estado Representacional (Representational State Transfer).
SDN: Redes Definidas por Software (Software Defined Networking).
SD-WAN: Red de Área Extensa Definida por Software (Software-Defined Wide Área Network).
SNMP: Protocolo Simple de Gestión de Redes (Simple Network Management).
SQL: Lenguaje de Consulta Estructurada (Structured Query Language).
SSL : Capa de Conectores Seguros (Secure Sockets Layer).
SSH: Cápsula Segura (Secure Socket Shell).
STP: Protocolo de Árbol Extendido (Spanning Tree Protocol).
TED: Base de Datos de Ingeniería de Tráfico (Traffic Engineering Database).
TCP: Protocolo de Control de Transmisión (Transmission Control Protocol).
TLS: Seguridad de la Capa de Transporte (Transport Layer Security).
URL: Localizador Uniforme de Recursos (Uniform Resource Locator).
VIM: Administrador de Infraestructura Virtual (Virtualized Infrastructure Manager).
VLAN: Red de Área Local Virtual (Virtual Local Área Network).
VNF: Función de Red Virtual (Virtualized Network Function).
VPN: Red Privada Virtual (Virtual Private Network).
VRRP: Protocolo Virtual de Redundancia (Virtual Redundancy Protocol).
VXLAN: Red de Área Local Extensible Virtual (Virtual Extensible Local Área Network).
WAN: Red de Área Amplia (Wide Área Network).
WiFi: Fidelidad Inalámbrica (Wireless Fidelity).
XML: Lenguaje de Marcado Extensible (Extensible Markup Language).

9.7. ANEXO G. Proyecto de Red Virtual en Laboratorio de Telecomunicaciones

Figura 1. Pruebas de la Red Virtual en el Laboratorio de Telecomunicaciones.

Figura 2. Equipo Entregable - Proyecto de Red Virtual Operativo.