ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ACCESO REMOTO A EQUIPOS EN UNA RED A TRAVÉS DEL NAVEGADOR WEB

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR EN REDES Y TELECOMUNICACIONES

KENETH ELIAM BOADA PARRA

DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ

DMQ, marzo 2023

CERTIFICACIONES

Yo, Keneth Eliam Boada Parra declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Keneth Eliam Boada Parra

Keneth.boada@epn.edu.ec

Kenth.boada@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Keneth Eliam Boada Parra, bajo mi supervisión.

Leandro Antonio Pazmiño Ortiz
DIRECTOR

leandro.pazmino@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Keneth Eliam Boada Parra

CI: 1752384345

DEDICATORIA

Este proyecto de titulación se lo dedico a mi familia Lucia Parra, David Boada, Sebastián Boada y Giulianna Boada quienes fueron siempre mi apoyo en todo este recorrido para convertirme en un profesional, también quiero agradecer a mi novia y a mi mejor amigo Kassandra Chamba y Erick Guerrero por sus palabras de aliento, compañía y su cariño. Lo que hoy logre nunca lo hubiera hecho sin ustedes.

AGRADECIMIENTO

Un enorme agradecimiento a mi tutor de tesis que siempre ha sido un ejemplo para mí por ser uno de los mejores maestros en mi carrera y por ser un modelo para seguir dentro de lo que significa ser un profesional. Gracias a mis compañeros de clase y amigos como: Betty, Milena, Myriam y Byron que siempre me apoyaron pese a las dificultades siendo personas responsables y dedicadas hacia tener un mejor futuro.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES I

DECLARACIÓN DE AUTORÍA II

DEDICATORIA III

AGRADECIMIENTO IV

RESUMEN VIII

ABSTRACT IX

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO 10

1.1 Objetivo general 11

1.2 Objetivos específicos 11

1.3 Alcance 11

1.4 Marco Teórico 11

Apache Guacamole 11

Apache Tomcat 12

RDP 12

VNC 12

SSH 12

Daemons TCP/IP 12

HTML5 13

dconf 13

Hipervisor 13

Virtual box 13

2 METODOLOGÍA 13

3 RESULTADOS 15

3.1 Identificación de los requerimientos para el diseño del prototipo 15

Conexión a los protocolos VNC, SSH y RDP 15

Conexión a cualquier tipo de *hardware* o *software* 15

Realizar una conexión sin la necesidad de instalar un aplicativo adicional 15

Que el prototipo tenga la cualidad de ser código abierto 16
Que no genere una gran carga de rendimiento al equipo 16
3.2 Selección del *hardware* y *software* acorde a los requerimientos establecidos .. 16
Selección de *software* para la conexión a los protocolos VNC, SSH y RDP 16
Selección de aplicativo para lograr una conexión a cualquier tipo de *hardware* o *software* 17
Selección del *software* que pueda realizar una conexión sin la necesidad de instalar un aplicativo adicional 18
Selección de prototipo de código abierto 19
Selección según el consumo de *hardware* en el equipo host 19
Selección de *software* en base al cumplimiento de los requerimientos 20
3.3 Diseño del prototipo de acceso remoto mediante un navegador 21
Esquema general de la arquitectura de Guacamole 21
Esquema utilizado para la implementación de Guacamole en el proyecto de titulación 22
3.4 Implementación de Apache Guacamole y los clientes 23
Instalación y configuración de *Virtual box* y las máquinas virtuales 23
Instalación de Guacamole en Ubuntu 22.04 de escritorio 28
Configuración del servidor de Guacamole 31
Configuración del cliente VNC 33
Configuración del cliente RDP 34
3.5 Pruebas de funcionamiento del prototipo 35
Pruebas de funcionamiento desde un navegador web en PC 37
Pruebas de funcionamiento para equipos celulares 40
Video de funcionamiento del prototipo 44
4 CONCLUSIONES 45
5 RECOMENDACIONES 47
6 REFERENCIAS BIBLIOGRÁFICAS 47
7 ANEXOS 50

ANEXO I: Certificado de Originalidad i

RESUMEN

En la primera sección del documento se tiene la introducción del proyecto de titulación la cual es una descripción de este, así también se tienen planteados los objetivos, el alcance del proyecto el cual es cumplido en las siguientes secciones y por último el marco teórico el cual brinda definiciones sobre términos que ayuden al lector a entender el documento.

En la segunda sección del proyecto se tiene la metodología la cual guía al cumplimiento de cada objetivo planteado dentro del proyecto de titulación donde se detallan parámetros importantes para la implementación del *software*.

La tercera sección del documento hace referencia al cumplimiento de los objetivos y el alcance del proyecto donde se presenta los requerimientos de *software* y *hardware*, la selección del prototipo según los requerimientos previos, como está diseñado el *software* y por último la implementación de este con sus respectivas pruebas de funcionamiento.

La cuarta sección hace referencia a las conclusiones obtenidas en relación con los resultados logrados en la tercera parte del documento, teniendo así el cumplimiento de los objetivos, la quinta sección muestra las recomendaciones que se dan para poder solventar cualquier tipo de error y obtener una mejor funcionalidad.

Finalmente, las últimas dos secciones presentan la bibliografía añadida después de una consulta adecuada y los anexos.

PALABRAS CLAVE: Guacamole, VNC, RDP, SSH, cliente, servidor, protocolo.

ABSTRACT

In the first section of the document there is the introduction of the titling project which is a description of it, as well as the objectives, the scope of the project which is fulfilled in the following sections and finally the theoretical framework which Provide definitions of terms that help the reader understand the document.

In the second section of the project there is the methodology which guides us to the fulfillment of each objective set within the titling project where important parameters for the implementation of the software are detailed.

The third section of the document refers to the fulfillment of the objectives and the scope of the project where the software and hardware requirements are presented, the selection of the prototype according to the previous requirements, how the software is designed and finally its implementation with its respective performance tests.

The fourth section refers to the conclusions obtained in relation to the results obtained in the third part of the document, thus meeting the objectives, the fifth section shows the recommendations that are given to solve any type of error and obtain better functionality.

Finally, the last two sections present the bibliography added after proper consultation and the annexes.

KEYWORDS: *Guacamole, VNC, RDP, SSH, client, server, protocol.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En este proyecto se realizará la implementación de un prototipo de sistema de acceso remoto, el cual ayuda a la administración de diversos equipos. Esto sin la necesidad de requerir ningún tipo de cliente adicional, para de esta forma acceder a estos equipos desde cualquier dispositivo tipo PC o celular, teniendo así un control total sobre estos siempre y cuando todos los equipos estén dentro de una misma red de área local.

Esto se realiza con el prototipo de acceso remoto Guacamole, el cual es el que se encarga de centralizar el control de los equipos, donde mediante una página web puede ingresar y controlar a los clientes. Para esto se plantea la conexión por medio de los siguientes protocolos de acceso remoto: *Virtual Network Computing* (VNC) en Linux, *Security Shell* (SSH) en servidores y *Remote Desktop Protocol* (RDP) en Windows.

Esto se lleva a cabo con el *software* Tomcat el cual es un *software* de código abierto, es el encargado de enlazar el servidor de acceso remoto Guacamole con un navegador web, el cual es accesible desde cualquier navegador utilizando HTML5, teniendo como resultado la conexión entre el servidor y el navegador web.

El servidor será levantado en una máquina virtual, se realizarán las configuraciones pertinentes para que el servicio este activo y listo para brindar conexión con las maquinas clientes, en el caso de realizar configuraciones adicionales se reiniciarán los servidores Guacamole y Tomcat para que la conexión no se vea afectada por mantener configuraciones sin registrar o cambios en el sistema.

Se utilizan aplicaciones adicionales las cuales ayudan a un mejor manejo del sistema operativo, como lo es dfconf, el cual permite facilitar ciertas configuraciones dentro en del cliente, como es el caso de la máquina que será controlada con el protocolo VNC, de esta forma teniendo acceso solamente con la contraseña generada dentro del servidor.

Por otra parte, dentro del cliente que realiza la conexión mediante el protocolo RDP, solo se habilita el acceso remoto en las configuraciones de *Windows* y se deshabilita la autenticación por medio de red.

Una vez que los servicios estén levantados y los clientes registrados se procederá a realizar pruebas de funcionamiento donde se tiene como objetivo el control total de los equipos ya registrados dentro de un archivo en el servidor Guacamole, esto con el objetivo de solventar errores de funcionamiento al igual de validar si se cumple el alcance del proyecto ya planteado.

1.1 Objetivo general

Implementación de prototipo de sistema de acceso remoto el cual pueda gestionar equipos mediante cualquier navegador web.

1.2 Objetivos específicos

- Investigar e identificar los requerimientos para el diseño del prototipo de sistema de acceso remoto.
- Elegir el *software* y *hardware* de acuerdo con los requerimientos elegidos.
- Diseñar el prototipo del sistema.
- Ejecutar el prototipo
- Comprobar el funcionamiento del prototipo.

1.3 Alcance

En este proyecto se desea implementar un prototipo el cual permita el acceso remoto a diferentes equipos incluyendo equipos que manejen Linux, Windows y servidores. Debido a esto, el prototipo tendrá la capacidad de realizar lo siguiente:

- Controlar tres dispositivos (equipo con Linux, Windows y un servidor mediante SSH) esto sin la necesidad de la instalación de un cliente en los equipos.
- Los equipos deben ser controlados mediante un navegador web el cual puede ser ejecutado en *smartphone* como en PC los cuales deber estar en la misma LAN.
- Centralizar el control de los equipos desde un solo sistema.

1.4 Marco Teórico

Apache Guacamole

Es una puerta de enlace de escritorio la cual realiza conexiones esta conexiónes empresariales y domésticas. Admite protocolos estándar como, por ejemplo: RDP, SSH y VNC. Además, es conocido como sin cliente, debido a que los clientes no necesitan ningún *software* adicional para su correcto funcionamiento.

Una vez que Apache Guacamole está instalado dentro de un servidor lo único que requiere para que este pueda ingresar a un escritorio por control remoto es un navegador web, todo esto gracias a HTML5 [1].

Apache Tomcat

Apache Tomcat es un *software* de código abierto el cual surge según las especificaciones de *Jakarta Expression Language, Jakarta Servlet, Jakarta Annotations, Jakarta Server Pages, Jakarta WebSocket y Jakarta Authentication*. Es una evolución de la plataforma de Java EE, Apache Tomcat ayuda a las diferentes aplicaciones web de misión crítica a tener una mayor escala dentro de una amplia gama de organizaciones e industrias [2].

RDP

Es un protocolo el cual permite la conexión entre un terminal y un servidor Windows, donde información es convertida en un formato RDP y también se envía a través de la red hacia el servidor, la información está contenida en un paquete de este protocolo para de esta manera reconstruir la imagen que se presenta en la pantalla del terminal. El puerto por defecto para una conexión RDP es el 3389 [3].

VNC

Es un *software* de código libre del tipo cliente-servidor, este permite visualizar el escritorio del equipo servidor o de un conjunto de clientes, esto sin tomar en cuenta el sistema operativo en el que se pueda ejecutar. De esta manera se puede controlar un equipo desde el servidor hasta el cliente. El puerto por defecto para una conexión VNC es el 5900 y el 5900 +n [4].

SSH

Este es un protocolo el cual tiene como finalidad la conexión entre equipos, esto con la característica de no tener una interfaz gráfica, por los que las configuraciones que permite se realizan por líneas de comando, por esto SSH se utiliza para realizar la conexión con uno o varios servidores, esto se logra gracias a Internet.

Una de las características más relevantes que posee este protocolo es que la conexión siempre se logra de una manera segura debido a que gracias a SSH la información que pasa por la red no es legible por una tercera persona, es decir, que la información que genera SSH esta encriptada. El puerto destinado para una conexión con este protocolo es el 22 [5].

Daemons TCP/IP

Son procesos los cuales se están corriendo en segundo plano de manera continua, estos realizan funciones las cuales son de suma importancia para otros procesos. El protocolo TCP/IP es el encargado de implementar ciertos tipos de funciones dentro del

sistema operativo. Los *daemons* siempre están trabajando en segundo plano y se pueden invocar mediante mandatos a nivel de gestión dentro del sistema [6].

HTML5

HTML5 es un estándar el cual define el contenido y la estructura de una página web ya que sirve como una referencia para conectar diferentes páginas web con sus diferentes versiones. El funcionamiento de HTML se lleva a cabo como una petición a un servidor el cual se realiza con el protocolo *HyperText Transfer Protocol* (HTTP) junto con la dirección IP y el respectivo index.html correspondiente al servidor [7].

dconf

Es muy útil para realizar ajustes dentro del sistema operativo, estos pueden ir desde algo muy simple hasta algo más avanzado. Por lo general esta herramienta facilita el encontrar ajustes del sistema o ciertas aplicaciones las cuales no son accesibles usualmente [8].

Hipervisor

Un hipervisor es un proceso el cual ejecuta y crea máquinas virtuales, permite a una máquina el prestar soporte a otras mediante el uso compartido de los recursos de procesamiento y memoria [9].

Virtual box

Es un hipervisor el cual es un potente producto de virtualización para uso doméstico y empresarial, *virtual box* es de gran ayuda para soluciones profesionales debido a que es un *software* gratuito y de código abierto. Este hipervisor se puede ejecutar tanto en masOS, Linux, Solaris, Windows, etc. [10].

2 METODOLOGÍA

Se realizó un análisis en base a una investigación para identificar los requerimientos para realizar el diseño del prototipo y así cumplir con el alcance establecido.

El *software* tuvo que cumplir con la conexión hacia los 3 protocolos: VNC, RDP y SSH, de igual forma debía cumplir con una conexión a cualquier tipo de *hardware* y *software* para obtener un amplio rango de conexiones.

La opción de realizar una conexión sin la necesidad de instalar un aplicativo adicional fue una característica de suma importancia la cual está presente en el alcance del proyecto. Se consideró que el prototipo tenga la cualidad de ser de código abierto para realizar configuraciones necesarias o cambios si se lo requiere.

El conectarse por medio de un navegador web mediante un protocolo cliente-servidor el cual sea accesible para cada equipo también fue un requisito a considerar. El *software* que se implementó debía tener la capacidad de conectarse mediante dispositivos móviles.

Como requerimientos de *hardware* se buscó un aplicativo el cual no discrimine el equipo y que no genere una gran carga de rendimiento al equipo host. Por último, el prototipo no tiene que adquirir algún tipo de *hardware* adicional para su funcionamiento.

Con los requerimientos identificados se procedió a hacer un análisis para determinar las características de *hardware* del equipo en el que se instaló el prototipo.

Después de una comparativa entre distintos *softwares* similares se decidió implementar Guacamole debido a que cumplía con todos los requisitos de *software* ya establecidos, entre los que se destacaron la conexión desde cualquier dispositivo y la amplia gama de conexiones que puede realizar hacia cualquier equipo sin la necesidad de tener un cliente instalado.

Guacamole cumplió con los requerimientos de *hardware* ya que no necesita la implementación de ningún tipo de dispositivo adicional para realizar su conexión, además que, para el enlace con los protocolos requeridos este no genera una gran carga al equipo host.

En base al aplicativo seleccionado se realizó el diseño del prototipo, donde Guacamole fue el encargado de la administración de los clientes. La conexión cliente-servidor se dio mediante un navegador web cualquiera utilizando HTML5.

Se planteó usar máquinas virtuales que funcionen como base tanto del servidor Guacamole como de los clientes, debido a que no se poseen restricciones por parte del *software*, por tanto, las máquinas usan el sistema operativo Linux y Windows.

La implementación del servidor y los clientes se realizó al unir el *hardware* y el *software* seleccionados, para lo cual se realizaron todas las configuraciones necesarias que incluían la instalación del sistema operativo y del aplicativo, esto se llevó a cabo en Ubuntu 22.04 de escritorio.

La configuración del hipervisor y las características que tenía que cumplir cada cliente se realizaron en Virtual box, donde además fue necesario instalar los sistemas operativos y configurar los tres equipos que fueron controlados por Guacamole.

Los clientes que se conectaron al servidor de Guacamole tuvieron que estar en la misma red de área local para que no exista inconvenientes en la conexión, cumpliendo con el alcance del proyecto.

Finalizada la implementación del prototipo se realizaron las respectivas pruebas de funcionamiento, las cuales permitieron validar el correcto desempeño del *software* utilizado. En consecuencia, según los resultados obtenidos en las pruebas, se subsanaron errores que existieron en la implementación, de esta manera se garantizó que el prototipo cumpliera con el alcance establecido en el proyecto.

3 RESULTADOS

En esta sección del documento se va a especificar los requerimientos de *software* y *hardware*, selección del prototipo en base a los requerimientos, diseño del prototipo, implementación y pruebas de funcionamiento, con el objetivo de cumplir con el alcance y los demás puntos pertinentes del plan de titulación.

3.1 Identificación de los requerimientos para el diseño del prototipo

En base al análisis que se realizó se encontraron los siguientes requerimientos de *hardware* y *software*, los cuales satisfacen el alcance del proyecto de titulación.

Conexión a los protocolos VNC, SSH y RDP

Este es el requerimiento más importante debido a que sin este el *software* no tiene validez porque no cumple con el alcance ni con los objetivos planteados, la conexión por medio de estos protocolos es lo que demuestra la extensa variedad de enlaces que se puede obtener dentro de una herramienta de control remoto.

Conexión a cualquier tipo de *hardware* o *software*

La conexión hacia cualquier tipo de sistema operativo es clave para un servicio de acceso remoto, de igual forma el que no se tenga ningún tipo de discriminación hacia el *hardware* abre un amplio tipo de conexiones a diversos equipos lo que ayuda a la administración de cualquier dispositivo que esté conectado a una red.

Realizar una conexión sin la necesidad de instalar un aplicativo adicional

El no poseer un aplicativo adicional es necesario para cumplir con el alcance del presente proyecto debido a que el *software* a implementar debe tener la particularidad de poder conectarse desde cualquier dispositivo sin la necesidad de instalar un cliente previo, lo cual ayuda a que solo se necesite de internet para poder acceder al servidor y así tener un completo control por medio del acceso remoto.

Que el prototipo tenga la cualidad de ser código abierto

Es recomendable que el *software* a implementar sea de código abierto debido a que no se desea pagar por adquirir un aplicativo o suscripción, de igual forma se plantea que este tiene que ser flexible en las posibles configuraciones que se realicen dentro del mismo. Ayuda a que este sea más flexible con otro tipo de sistemas operativos, de igual forma recibe mucha atención y mantenimiento por parte de los usuarios, los cuales ayudan a resolver problemas que pueden surgir en diferentes implementaciones. Una de las características más importantes es que al ser de uso libre este puede ser descargado e instalado en cualquier equipo.

Que no genere una gran carga de rendimiento al equipo

Un requerimiento de *hardware* es que el prototipo no genere un consumo alto de recursos en el sistema ya que uno de los requerimientos para la implementación es que no discrimine ningún tipo de *software* y *hardware*, ya que se busca flexibilidad para que el prototipo pueda ser instalado en cualquier sistema operativo.

3.2 Selección del *hardware* y *software* acorde a los requerimientos establecidos

En base a una investigación se encontraron cuatro tipos de *software* que cumplen con los requerimientos más importantes mencionados anteriormente.

- *Zoho Assist.*
- *Parallels Access.*
- Escritorio remoto de Chrome.
- Apache Guacamole.

Gracias a esto se realiza una comparativa para comprobar si cumplen con los requerimientos de *hardware* y *software* definidos en la sección 3.1.

Selección de *software* para la conexión a los protocolos VNC, SSH y RDP

Los *softwares* nombrados anteriormente pueden conectarse a los 3 protocolos requeridos, siendo así que, *Zoho Assist*, Apache Guacamole, *Parallels Access* y Google Chrome desktop cumplen con una conexión hacia cualquier dispositivo usando un navegador web sin discriminar el medio por el cual se realiza esta.

Software Zoho Assist

Es una herramienta de asistencia remota la cual se brinda a empresas, posee conexiones seguras a través de páginas web utilizando cualquier tipo de protocolo incluidos los necesarios para cumplir con el alcance del proyecto [11].

Software Parallels Access

Es un aplicativo que permite la conexión hacia cualquier equipo Windows, Linux y macOS, permite enlaces por medio de protocolos mediante una aplicación o página web [12].

Software Escritorio remoto de Chrome

Esta herramienta puede crear una conexión por medio del navegador Google Chrome hacia cualquier equipo usando el protocolo a elección, aunque necesita una configuración adicional si se desea utilizar SSH [13].

Software Apache Guacamole

Está basado en HTML5 por lo que proporciona servicios a través de un navegador web utilizando múltiples protocolos como: VNC, SSH, RDP, etc. Gracias a esto es posible la conexión hacia cualquier equipo [1].

Selección de aplicativo para lograr una conexión a cualquier tipo de *hardware* o *software*

Los aplicativos escogidos tienen requisitos de *software* específicos debido a la infraestructura o tipo de sistema operativo que manejan.

Zoho Assist

Zoho es un *software* que brinda servicios de control remoto en casi cualquier sistema operativo, aunque tiene limitantes debido a su infraestructura, las restricciones varían según el tipo de conexión que realiza. El único requerimiento que posee para tener una conexión estable es un ancho de banda de 128 (kbps) como mínimo, aunque es recomendable 256 (kpbs) o más.

El aplicativo es funcional en versiones por encima de Windows *Server* 2003, Macintosh OS 10.6 y superior. Los sistemas operativos tipo Linux son compatibles con todas sus versiones incluyendo Raspberry OS [14].

Parallels Access

Los requisitos de *software* por parte de *Parallels Access* varían debido a que posee una aplicación para dispositivos móviles.

Las versiones de *software* admitidas para la aplicación son las siguientes:

- ¡Pad que ejecute iOS 11 o posterior.
- *iPhone* 5 hacia adelante que ejecute iOS 11.
- *Android* con versiones 5.0 y posteriores.

Para PC's no se tienen muchas restricciones debido a que su conexión es mediante cualquier navegador web [15].

Escritorio remoto de Chrome

Este *software* no tiene preferencias en sistemas operativos ni *hardware*, ya que, solo necesita Google Chrome para su funcionamiento, esto es de gran ayuda, debido a que, no discrimina ningún sistema [16]. Aunque para la conexión de dispositivos móviles necesita de una aplicación externa.

Apache Guacamole

Una de las grandes ventajas de Guacamole es que es muy flexible con cualquier sistema operativo debido a que es de código abierto, lo cual permite que los usuarios puedan proveer asistencia a la implementación de este *software* en diferentes sistemas operativos. Aunque uno de los requerimientos para la implementación de este *software* es descargar las librerías libjpeg, libpng, libuuid en el servidor [17].

Selección del *software* que pueda realizar una conexión sin la necesidad de instalar un aplicativo adicional

Un aspecto importante que tiene que cumplir el *software* es que no necesita de la instalación de un aplicativo adicional, por lo cual los aplicativos seleccionados tienen las siguientes características:

Zoho Assist

Permite la asistencia remota, este puede realizar un enlace mediante una página web por lo que no necesita un aplicativo adicional en los clientes para su funcionamiento y conexión. Gracias a esto puede conectarse desde cualquier dispositivo y soportar cualquier plataforma [18].

Parallels Access

Parallels Access necesita de la instalación de un cliente en el equipo donde se realiza la conexión, es compatible con una conexión vía web, aunque también posee un aplicativo para *smartphones.*

El inconveniente es el utilizar un navegador para la conexión, ya que necesita de una suscripción por pago y la versión gratuita no posee muchas funciones [19].

Escritorio remoto de Chrome

Chrome necesita que su navegador este instalado en el cliente, caso contrario no existirá conexión ya que el enlace se crea entre navegadores [20].

Apache Guacamole

Es óptimo en comparación a los softwares mencionados anteriormente, debido a que su plataforma solo necesita de ciertos requerimientos de librerías en su instalación y por medio de la dirección IP del cliente puede ingresar sin ningún tipo de inconveniente mediante un navegador web [21].

Selección de prototipo de código abierto

La elección de un *software* de código abierto surge por la necesidad de que este sea flexible y escalable, ya que cualquier persona puede utilizar el código para mejorarlo, modificarlo y retribuirlo teniendo como resultado una mayor reutilización y accesibilidad [22].

Zoho Assist

Según la documentación oficial de Zoho este es un *software* cerrado, por lo cual no permite ciertas configuraciones si estas son necesarias, la principal razón por la que es de código cerrado es por el pago por usar este aplicativo [11].

Parallels Access

Parallels Access no es de código abierto debido a que es propiedad de IOS y Windows por lo que las configuraciones son muy limitadas, si hay la necesidad de realizar cambios esto no será posible e iría en contra de las leyes de derechos de autor [23].

Escritorio remoto de Chrome

El escritorio remoto que proporciona Chrome no es un *software* de código abierto, aunque funciona con tecnologías web de código abierto como WebRTC, la cual es la responsable de la conexión de los dispositivos de manera confiable [13].

Apache Guacamole

Guacamole es un aplicativo de código abierto el cual está licenciado desde su versión 2.0. Gracias a su comunidad de desarrolladores se ha mantenido activo y así ha logrado sobresalir y adecuarse a cualquier entorno [1].

Selección según el consumo de *hardware* en el equipo host

Zoho Assist

Debido a que *Zoho Assist* es un servicio que se brinda por medio de un navegador web no consume muchos recursos de un equipo host, solo necesita un ancho de banda definido para su correcto funcionamiento [14].

Parallels Access

El uso este aplicativo no genera un cargo de *hardware* debido a que es un servicio que se brinda para realizar una conexión de acceso remoto, este se puede desplegar en varios dispositivos para ser controlados de manera remota, dependiendo del paquete obtenido puede llegar a realizar varias conexiones desde un solo dispositivo [12].

Escritorio remoto de Chrome

Para llevar a cabo una conexión por acceso remoto se necesita el aplicativo Google Chrome, este consume un excesivo recurso de *hardware* solo con utilizarlo, aunque este puede reducirse si se deshabilitan las aplicaciones en segundo plano, ya que estas son las que llegan a aumentar el nivel de RAM y CPU [24].

Apache Guacamole

Gracias a la flexibilidad de Apache Guacamole este *software* puede ser levantado e instalado en servicios en nube, por lo que no se genera una gran carga a equipos host. Es posible implementar el servicio en cualquier sistema operativo con características mínimas de hardware y aun así funcionar sin inconvenientes ya que todas sus conexiones se realizan mediante HTML5 [1].

Selección de *software* en base al cumplimiento de los requerimientos

Después de una extensa investigación y comparación entre los posibles candidatos para llevar a cabo una conexión remota mediante un navegador web, Apache Guacamole es el *software* escogido, debido a que cumple con todos los requerimientos establecidos en la sección 3.1. En la Tabla 3.1 se realiza la comparativa de los aplicativos considerados con los requerimientos que llegan a cumplir.

Tabla 3.1 Comparación de *softwares*

Requerimientos	Conexión a los protocolos VNC, SSH y RDP	Conexión a cualquier tipo de *hardware* o *software*	Conexión sin la necesidad de instalar un aplicativo adicional	*Software* de código abierto	Generar una carga baja de rendimiento al equipo
Zoho Assist	✓	X	✓	X	✓
Access Parallels	✓	X	X	X	✓

Requerimientos	Conexión a los protocolos VNC, SSH y RDP	Conexión a cualquier tipo de *hardware* o *software*	Conexión sin la necesidad de instalar un aplicativo adicional	*Software* de código abierto	Generar una carga baja de rendimiento al equipo
Escritorio Remoto de Chrome	✓	✓	X	X	✓
Apache Guacamole	✓	✓	✓	✓	✓

3.3 Diseño del prototipo de acceso remoto mediante un navegador

Esquema general de la arquitectura de Guacamole

El esquema representa el funcionamiento y estructura de Apache Guacamole, este es el encargado de administrar la conexión de los clientes, estos se conectan mediante un navegador web hacia el servidor utilizando el protocolo HTTP y Guacamole. Un archivo es el responsable de la gestión de los equipos, el cual tiene como objetivo el registrar la dirección IP y el protocolo que se está utilizando para realizar el enlace.

Dentro del servidor se encuentra el protocolo Guacamole y el *daemon* guacd, la combinación de estos genera agnosticismo de protocolo, esto quiere decir que ni el cliente ni la aplicación necesitan conocer el tipo de protocolo de control remoto que se utiliza para la conexión [25].

El protocolo Guacamole ayuda a la visualización y transferencia de eventos remotos, los principios de este son diferentes a cualquier otro protocolo específico como VNC o RDP, ya que este no está diseñado para desarrollar funciones en un único entorno de escritorio remoto.

Guacd es el corazón de este *software* ya que realiza una conexión dinámica según se especifique en la página web o en el archivo de registro dentro del servidor, se ejecuta en segundo plano y escucha conexiones *Transmission Control Protocol* (TCP). Estos 2 agentes realizan la comunicación entre el navegador web y los clientes, como se muestra en la Figura 3.1 [25].

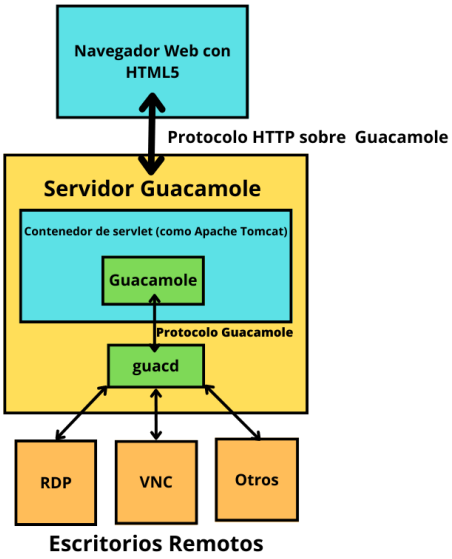

Figura 3.1 Arquitectura de Apache Guacamole [1]

Esquema utilizado para la implementación de Guacamole en el proyecto de titulación

Se plantea utilizar hipervisores para el levantamiento de Guacamole y los clientes con sistemas operativos tipo Linux y servidor. Para la instalación del servidor se siguieron las recomendaciones de instalación de páginas web, donde se pide el sistema operativo Ubuntu versión 22.04 como base para la implementación de este *software* [26].

Por parte de los clientes debido a los protocolos que se utilizan para las conexiones, se utilizan sistemas operativos tipo Linux de escritorio y *server* para el cliente con VNC y SSH respectivamente. La elección de Ubuntu 22.04 para realizar los enlaces con sus protocolos se basa en conocimiento previo, así como en el arreglo de errores por parte de esta versión [27]. Por falta de una imagen de sistema operativo tipo Windows, este no se pudo implementar en una máquina virtual, por lo que se realiza la conexión RDP hacia un equipo que funcione con Windows 10, esto debido a que es el sistema operativo más común en equipos PC en la actualidad.

Las conexiones entre los clientes y el servidor de Guacamole se realizan mediante un navegador web desde cualquier dispositivo móvil o PC, siempre y cuando estos estén dentro de la misma red de área local, como se ve en la Figura 3.2.

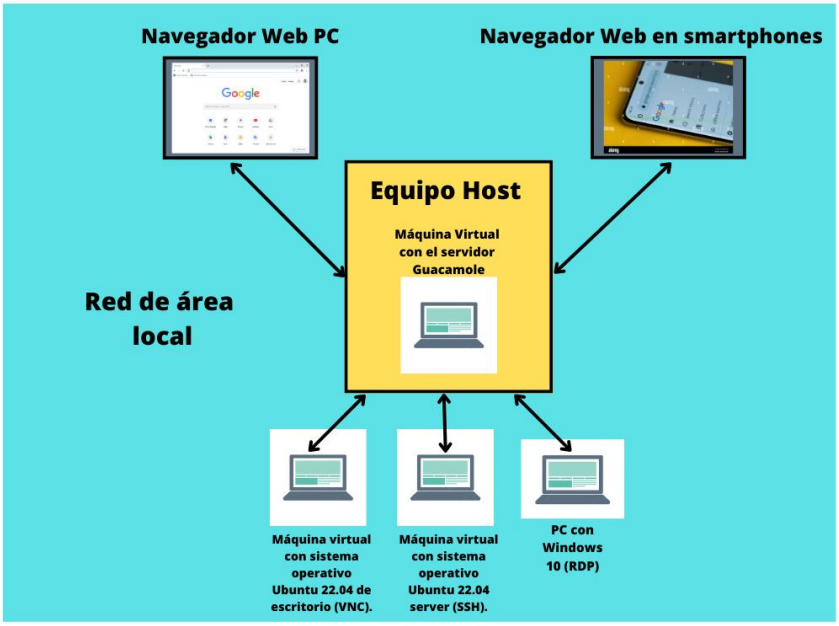

Figura 3.2 Esquema de implementación del prototipo

3.4 Implementación de Apache Guacamole y los clientes

Se selecciona el hipervisor *virtual box* como el encargado de administrar, contener y levantar el sistema operativo Ubuntu 22.04 tanto para los clientes como para el servidor.

Instalación y configuración de *Virtual box* y las máquinas virtuales

La instalación de este *software* se realizó desde su página oficial como se muestra en la Figura 3.3. La versión 7.0 es la más actual de este hipervisor, esta es escogida para levantar el servicio de Guacamole, así como los clientes con sistema operativo Ubuntu.

Figura 3.3 Descarga de *virtual box*

Una vez el hipervisor está instalado se descargaron las imágenes de Ubuntu 22.04 de escritorio y *server*, como se puede ver en la Figura 3.4. La descarga de los sistemas operativos se realiza desde su página oficial ya que ahí se encuentra la información y requerimientos de *hardware* para la instalación de estos sistemas operativos [28].

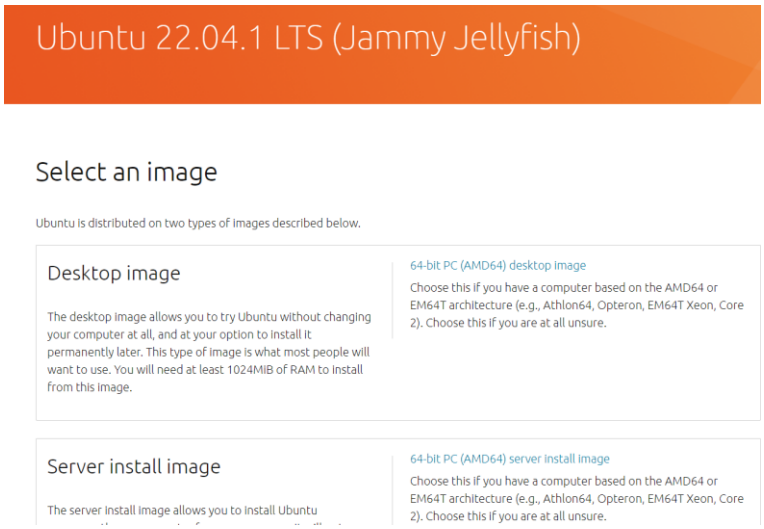

Figura 3.4 Página oficial de Ubuntu 22.04

Una vez descargados los sistemas operativos y *Virtual box*, comenzó la configuración de las imágenes, donde la versión de escritorio se implementó para el servidor y el cliente VNC, para SSH se configuró la misma versión, pero tipo *server*.

Se siguieron las sugerencias por parte de Ubuntu para instalar el sistema operativo que se usó para levantar el servidor Guacamole [29].

Las recomendaciones por seguir, así como la configuración se pueden ver en la Figura 3.5. Para este sistema operativo las sugerencias fueron:

- 4 (GB) de memoria RAM.
- 25 (GB) de disco duro.
- Acceso a Internet.

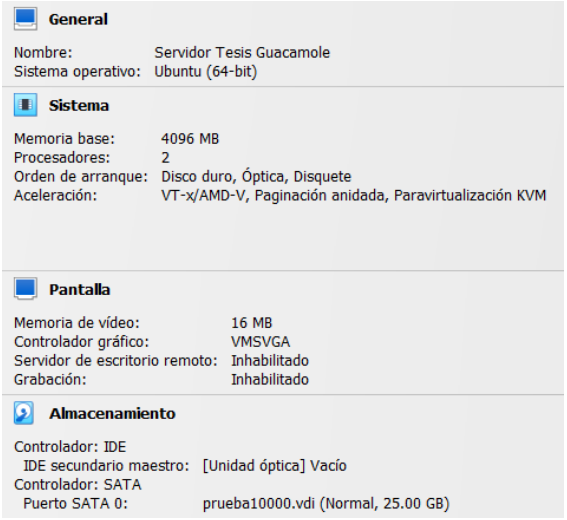

Figura 3.5 Configuración de Virtual box para servidor Guacamole

La configuración de los clientes SSH y VNC en *virtual box* fue similar a la del servidor, la única diferencia son los 2 (GB) de RAM que poseen como se puede ver en la Figura 3.6, esta disminución de RAM es debido a que el equipo se controla de manera remota, así que no consume muchos recursos para su conexión gracias a Guacamole.

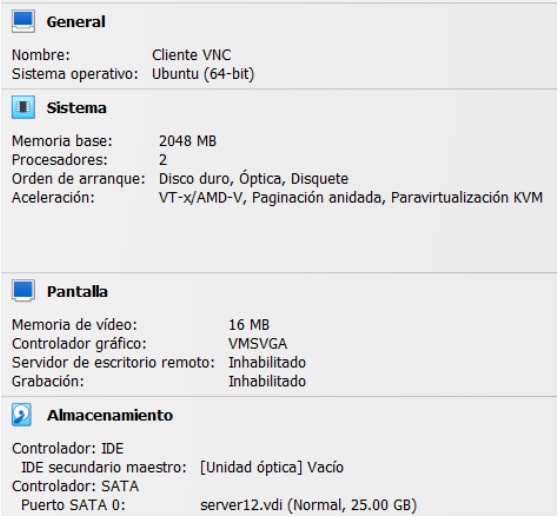

Figura 3.6 Configuración de los clientes VNC y SSH en Virtual box

Para las 3 máquinas virtuales se configuró un adaptador tipo puente para que obtenga una dirección IP dentro de la red de área local, como se ve en la Figura 3.7.

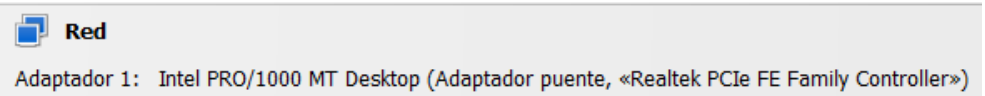

Figura 3.7 Configuración de adaptador de red

Una vez hechas todas las configuraciones de *hardware* para las máquinas virtuales, se realizó una instalación estándar para los sistemas operativos, se creó el usuario: kenethb y la contraseña: chester146. Este proceso se repitió en cada uno de los equipos, como se puede visualizar en las Figura 3.8,Figura **3.9** y Figura **3.10**.

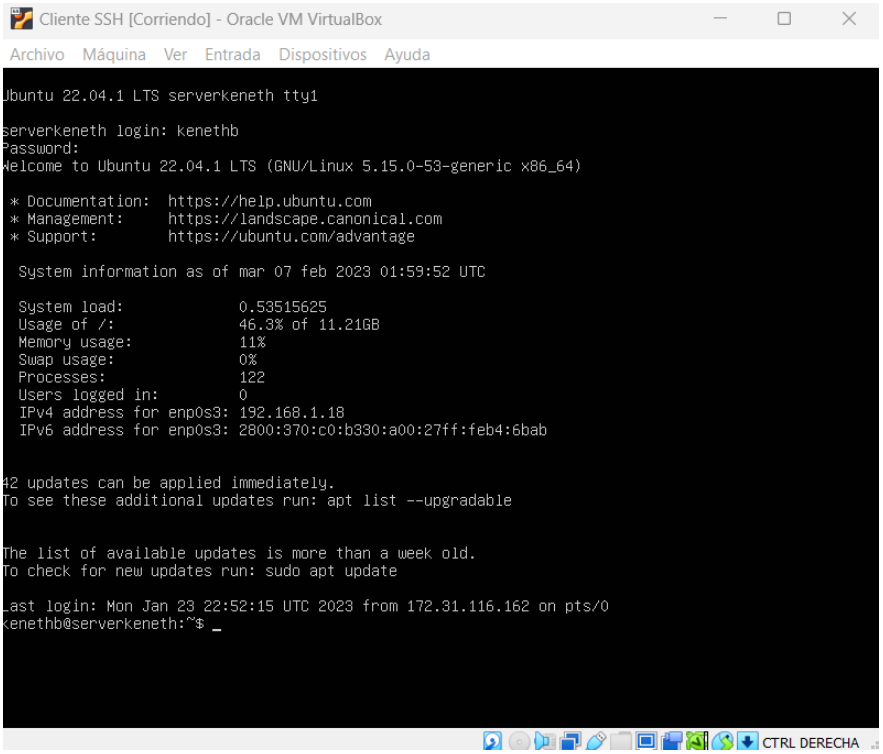

Figura 3.8 Cliente SSH en Ubuntu 22.04 server

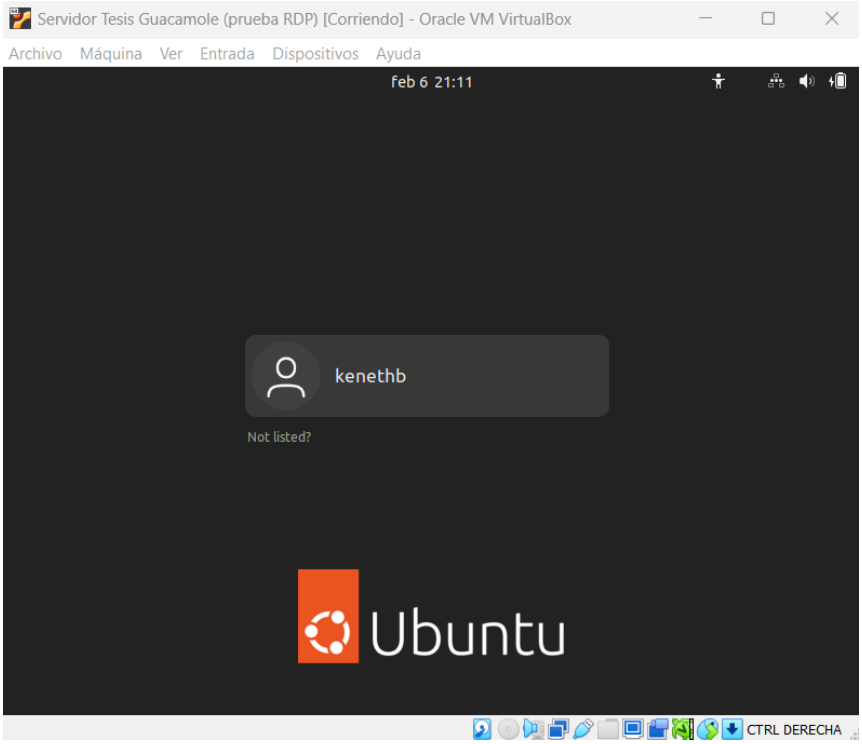

Figura 3.9 Servidor Guacamole en Ubuntu 22.04 de escritorio

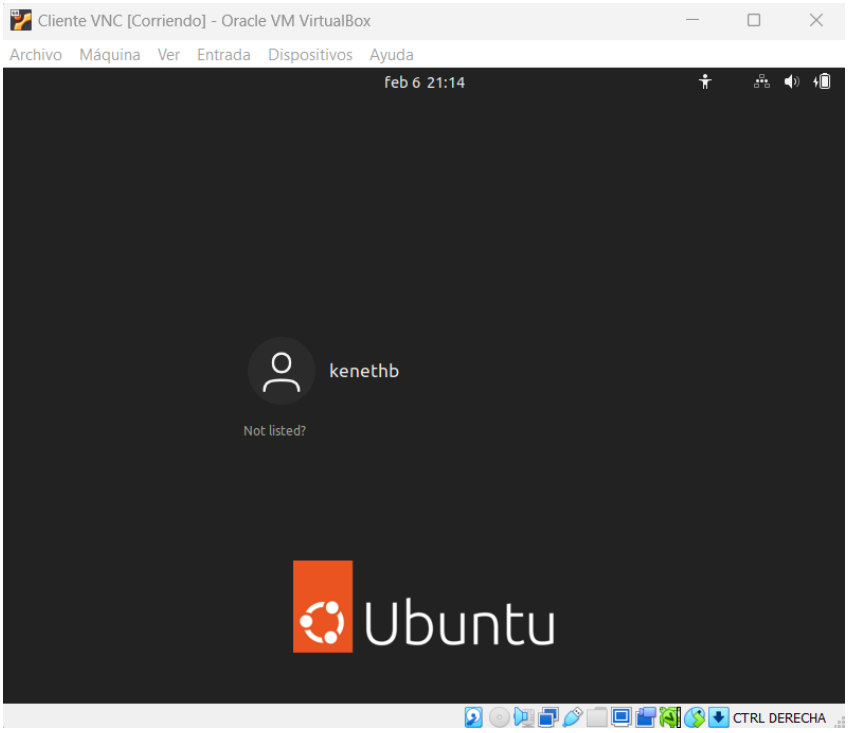

Figura 3.10 Cliente VNC en Ubuntu 22.04 de escritorio

En la máquina virtual tipo *server* se levantó Apache para tener un servidor web dentro de este equipo, debido a que es sencillo de implementar ya que solo se necesita descargar e instalar el *software* para su correcto funcionamiento. El estado del servicio, así como la conexión a este se puede visualizar en las Figura 3.11 yFigura **3.12**.

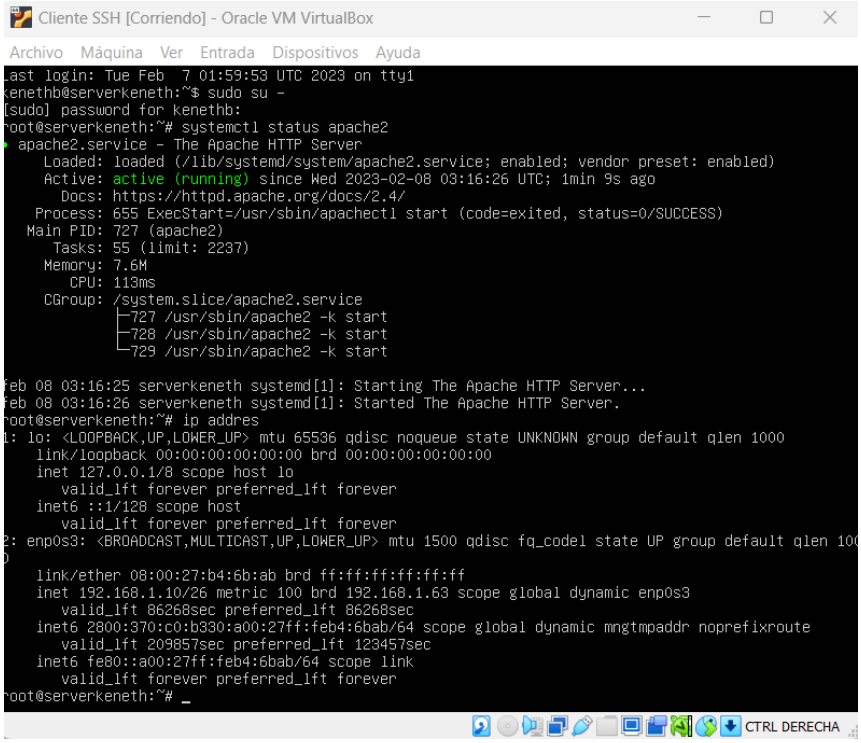

Figura 3.11 Estado del servicio web Apache

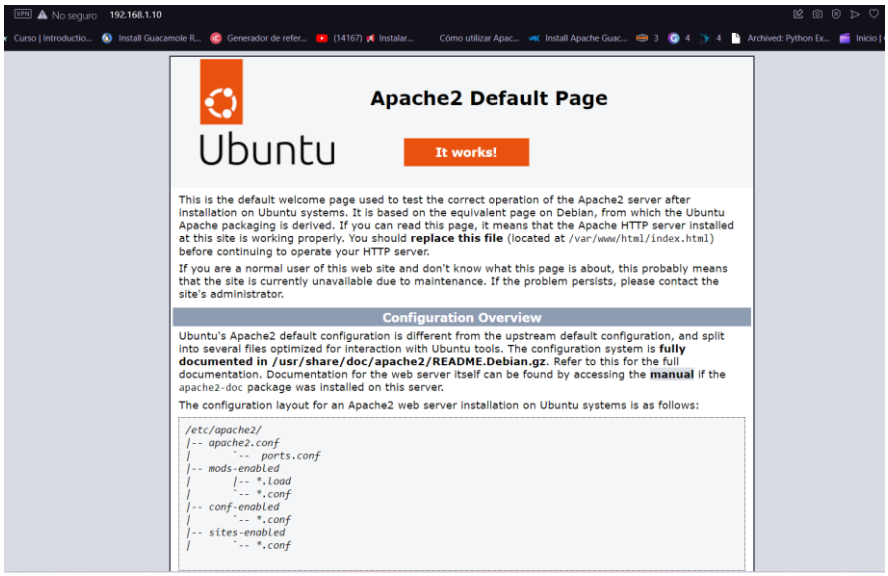

Figura 3.12 Ingreso al servicio web mediante un navegador

Instalación de Guacamole en Ubuntu 22.04 de escritorio

Para la instalación de Guacamole se siguieron se utilizaron páginas web las cuales ayudaron en la instalación y correcto funcionamiento de este *software* [30].

Primero hay que descargar e instalar todas las dependencias de Guacamole como se ve en la Figura 3.13.

```
kenethb@prueba10000:~$ sudo apt install -y gcc nano vim curl wget g++ libcairo2-dev l
ibjpeg-turbo8-dev libpng-dev libtool-bin libossp-uuid-dev libavcodec-dev  libavformat
-dev libavutil-dev libswscale-dev build-essential libpango1.0-dev libssh2-1-dev libvn
cserver-dev libtelnet-dev libpulse-dev libvorbis-dev libwebp-dev
```

Figura 3.13 Descarga e instalación de dependencias de Guacamole

Se ingresa en el súper usuario de Ubuntu ya que este es el que permite todas las configuraciones del servidor, tal y como se puede ver en la Figura 3.14. Una vez hecho esto hay que descargar la versión 1.4.0 desde su página de lanzamientos, esta es la última versión que ha sacado este *software*, por último, se descomprime el *software* como se puede ver en las Figura 3.15 y Figura **3.16**.

```
kenethb@prueba10000:~$ su
Password:
root@prueba10000:/home/kenethb#
```

Figura 3.14 Ingreso al súper usuario de Ubuntu

```
root@prueba10000:/home/kenethb# wget https://downloads.apache.org/guacamole/1.4.0/sou
rce/guacamole-server-1.4.0.tar.gz
```

Figura 3.15 Descarga de la última versión de Guacamole

```
root@prueba10000:/home/kenethb# tar xzf guacamole-server-1.4.0.tar.gz
```

Figura 3.16 Descompresión del archivo guacamole-server

Una vez que Guacamole es descargado en la máquina virtual se procedió a ingresar en la carpeta guacamole-server-1.4.0 donde comenzó la instalación de las dependencias y se verifica que no falta alguna. Debido a que este *software* funciona con la versión *OpenSSL* 1 y Ubuntu 22.04 viene con la versión 3 de forma predeterminada, se escribió el comando que se puede visualizar en la Figura 3.17 para evitar que exista errores en la compilación y esta fallara [31].

```
root@prueba10000:/home/kenethb# cd guacamole-server-1.4.0
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# CFLAGS=-Wno-error ./configure
--with-systemd-dir=/etc/systemd/system/
```

Figura 3.17 Ingreso a la carpeta del servidor y ejecución de comando

Aplicados estos comandos se abre el servidor de Guacamole indicando el estado de las dependencias y así verificar si todo está correctamente instalado, como se puede ver en la Figura 3.18.

```
----------------------------------------------
guacamole-server version 1.4.0
----------------------------------------------

   Library status:

     freerdp2 ............ yes
     pango ............... yes
     libavcodec .......... yes
     libavformat.......... yes
     libavutil ........... yes
     libssh2 ............. yes
     libssl .............. yes
     libswscale .......... yes
     libtelnet ........... yes
     libVNCServer ........ yes
     libvorbis ........... yes
     libpulse ............ yes
     libwebsockets ....... no
     libwebp ............. yes
     wsock32 ............. no

   Protocol support:

      Kubernetes .... no
      RDP ........... yes
      SSH ........... yes
      Telnet ........ yes
      VNC ........... yes
```

Figura 3.18 Estado de las dependencias de Guacamole

Mediante el comando *make* comienza la compilación de las dependencias con el sistema operativo y gracias al comando *ldconfig* se crea la memoria cache, así como los enlaces indispensables para las bibliotecas que se encuentran en el servidor de Guacamole. El uso de estos 2 comandos se puede ver en la Figura 3.19 yFigura **3.20**.

```
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# make
```

Figura 3.19 Uso del comando *make* para la compilación de Guacamole

```
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# ldconfig
```

Figura 3.20 Uso del comando ldconfig

Con las dependencias instaladas y listas para el funcionamiento del servidor, se habilita y verifica el estado del servicio, donde se puede observar que está funcionando correctamente, como se puede apreciar en la Figura 3.21.

```
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# systemctl enable --now guacd
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# systemctl status guacd
* guacd.service - Guacamole Server
     Loaded: loaded (/etc/systemd/system/guacd.service; enabled; vendor preset: enab>
     Active: active (running) since Tue 2023-02-07 23:01:15 -05; 27min ago
       Docs: man:guacd(8)
   Main PID: 727 (guacd)
      Tasks: 1 (limit: 4625)
     Memory: 12.0M
        CPU: 42ms
     CGroup: /system.slice/guacd.service
             `-727 /usr/local/sbin/guacd -f

feb 07 23:01:15 prueba10000 systemd[1]: Started Guacamole Server.
feb 07 23:01:16 prueba10000 guacd[727]: Guacamole proxy daemon (guacd) version 1.4.0>
feb 07 23:01:16 prueba10000 guacd[727]: guacd[727]: INFO:        Guacamole proxy dae>
feb 07 23:01:16 prueba10000 guacd[727]: guacd[727]: INFO:        Listening on host 1>
feb 07 23:01:16 prueba10000 guacd[727]: Listening on host 127.0.0.1, port 4822
lines 1-16/16 (END)
```

Figura 3.21 Estado del servidor Guacamole

Un aspecto importante para el funcionamiento del servidor por medio de un navegador web es el uso de Tomcat, este es compatible con Guacamole, es el que ayuda a la conexión entre el servidor y un navegador web [32]. La versión 9 de Tomcat es la más actual hasta la fecha y es compatible con Guacamole 1.4.0, debido a esto se descarga e instala este *software*, también se habilita el puerto 8080 o también conocido como el puerto alternativo para HTTP, tal y como se presenta en la Figura 3.22.

```
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# apt install tomcat9 tomcat9-ad
min tomcat9-common tomcat9-user -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tomcat9 is already the newest version (9.0.58-1ubuntu0.1).
tomcat9-admin is already the newest version (9.0.58-1ubuntu0.1).
tomcat9-common is already the newest version (9.0.58-1ubuntu0.1).
tomcat9-user is already the newest version (9.0.58-1ubuntu0.1).
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# ufw allow 8080/tcp
```

Figura 3.22 Instalación de Tomcat 9

La instalación del cliente de Guacamole en el servidor es necesario ya que este posee una aplicación web el cual contiene el estándar HTML5, esto sirve para que los usuarios se puedan conectar al servidor y así brindar acceso remoto.

Para obtener el cliente primero hay que crear una carpeta donde se instala esta versión del *software*, como se puede ver en la Figura 3.23.

```
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# mkdir /etc/guacamole
mkdir: cannot create directory '/etc/guacamole': File exists
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# wget https://downloads.apache.
org/guacamole/1.4.0/binary/guacamole-1.4.0.war -O /etc/guacamole/guacamole.war
```

Figura 3.23 Instalación del cliente de Guacamole

Se crea un enlace simbólico desde el cliente Guacamole hacia el directorio de aplicaciones de Tomcat, una vez hecho esto se reinicia Tomcat y guacd, como se puede ver en la Figura 3.24.

```
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# ln -s /etc/guacamole/guacamole
.war /var/lib/tomcat9/webapps/
ln: failed to create symbolic link '/var/lib/tomcat9/webapps/guacamole.war': File exi
sts
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# systemctl restart tomcat9
root@prueba10000:/home/kenethb/guacamole-server-1.4.0# systemctl restart guacd
```

Figura 3.24 Creación de enlace simbólico

Como siguiente se debe crear una carpeta para las extensiones del servidor, como se muestra en la Figura 3.25, se establece una variable de entorno la cual se agrega al directorio de Tomcat 9, esto se puede visualizar en la Figura 3.26 [31].

```
root@prueba10000:/home/kenethb# mkdir /etc/guacamole/{extensions,lib}
```

Figura 3.25 Creación de carpeta para las extensiones de Guacamole

```
root@prueba10000:/home/kenethb# echo "GUACAMOLE_HOME=/etc/guacamole" >> /etc/default/
tomcat9
```

Figura 3.26 Variable de entorno establecida en Tomcat 9

Configuración del servidor de Guacamole

En la Figura 3.27Figura 3.28 se visualiza como se crea el archivo *guacamole.properties*, el cual define como se conecta el servidor guacd.

```
root@prueba10000:/home/kenethb# vim /etc/guacamole/guacamole.properties
```

Figura 3.27 Creación de archivo *guacamole.properties*

```
root@prueba10000: /home/kenethb

guacd-hostname: localhost
guacd-port:     4822
user-mapping:   /etc/guacamole/user-mapping.xml
auth-provider:  net.sourceforge.guacamole.net.basic.BasicFileAuthenticationProvide
~
~
```

Figura 3.28 Configuración de archivo *guacamole.properties*

Después de crear el archivo y guardarlo se vincula el directorio del servidor de Guacamole con el de Tomcat, como se puede apreciar en la Figura 3.29.

```
root@prueba10000:/home/kenethb# ln -s /etc/guacamole /usr/share/tomcat9/.guacamole
```

Figura 3.29 Vinculación de Guacamole y Tomcat

Como método de autenticación, se debe crear el archivo *user-mapping.xml*, donde se encuentran los usuarios que pueden acceder a Guacamole desde su interfaz web. En este caso se tiene el usuario: *kenethadmin,* el cual es el único que va a poder ingresar a Guacamole para realizar una conexión, también se define una contraseña la cual es: chester146.

En este archivo también se ingresan los clientes junto a sus direcciones IP y el protocolo a utilizar para la conexión. Estas configuraciones se pueden visualizar en la Figura 3.30Figura 3.31.

```
root@prueba10000:/home/kenethb# vim /etc/guacamole/user-mapping.xml
```

Figura 3.30 Creación de archivo de registro

```
<user-mapping>

    <!-- Per-user authentication and config information -->

    <!-- A user using md5 to hash the password
         guacadmin user and its md5 hashed password below is used to
             login to Guacamole Web UI-->
    <authorize
            username="kenethadmin"
            password="chester146">

        <!-- First authorized Remote connection -->
        <connection name="prueba VNC">
            <protocol>vnc</protocol>
            <param name="hostname">172.31.116.163</param>
            <param name="port">5900</param>
        </connection>

        <!-- Second authorized remote connection -->
        <connection name="prueba RDP">
            <protocol>rdp</protocol>
            <param name="hostname">172.31.116.140</param>
            <param name="port">3389</param>
            <param name="username">tech</param>
            <param name="ignore-cert">true</param>
         </connection>

        <!-- Third authorized remote connection -->
        <connection name="SSH">
            <protocol>ssh</protocol>
            <param name="hostname">172.31.116.166</param>
                    <param name="port">22</param>
        </connection>

    </authorize>

</user-mapping>
```

Figura 3.31 Registro de clientes y usuarios

Con toda la implementación de este *software* terminada ya es posible realizar una conexión a este desde un navegador web, para realizar el enlace es necesario ingresar la dirección IP del equipo como también el puerto 8080 y la palabra Guacamole, como, por ejemplo: 192.168.1.10:8080/guacamole. El funcionamiento del servidor se puede visualizar en la Figura 3.32.

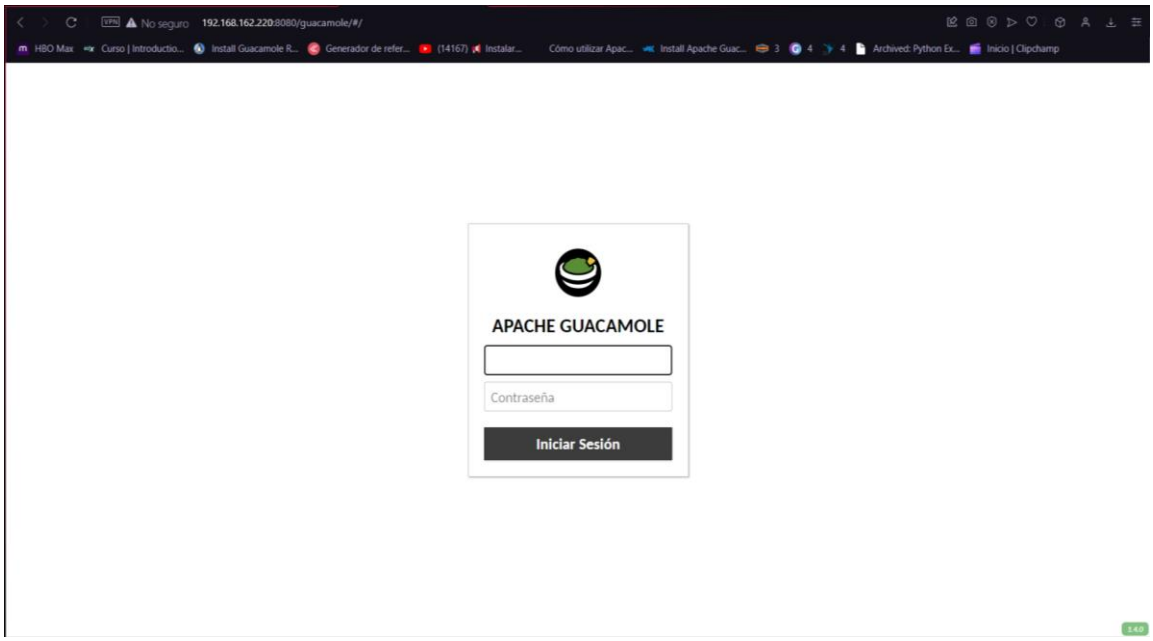

Figura 3.32 Conexión a Guacamole desde navegador web

Configuración del cliente VNC

Para realizar una conexión por este protocolo es necesario activar esta opción manualmente, gracias a que se utiliza una versión de escritorio el habilitar esto es sencillo, aunque existen parámetros que se tienen que cambiar dentro de Ubuntu para permitir un enlace adecuado. Gracias a la herramienta dconf se pueden realizar configuraciones dentro de Ubuntu con una interfaz gráfica, por lo cual hay que ingresar a la parte de autenticación de VNC y cambiarla a la opción contraseña. El uso de esta herramienta y el cambio de método de autenticación se visualiza en la Figura 3.33.

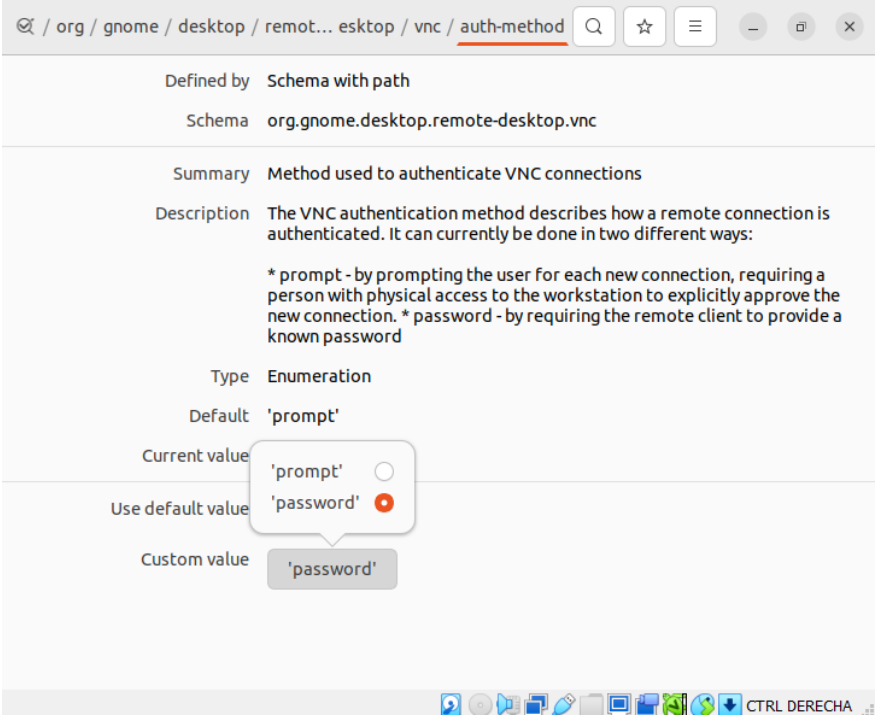

Figura 3.33 Uso de dconf y cambio de autenticación

Una vez realizado este cambio es necesario añadir un usuario y una contraseña en la parte de *remote desktop* que se encuentra en la opción de *sharing* en la configuración de Ubuntu. El usuario y la contraseña con visibles en la Figura 3.34.

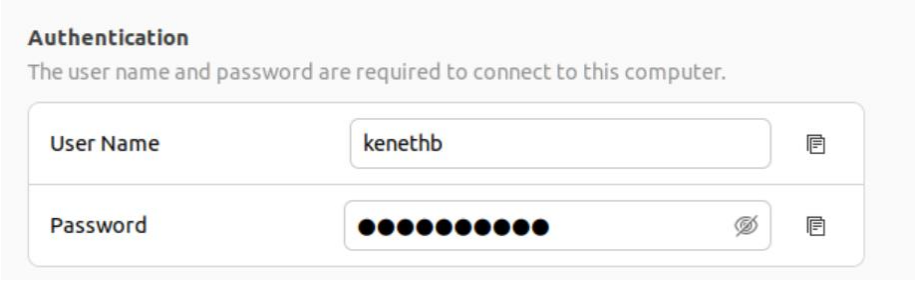

Figura 3.34 Creación de usuario y contraseña para conexión VNC

Configuración del cliente RDP

Dentro del equipo cliente Windows hay que habilitar la opción de escritorio remoto, también es necesario deshabilitar la autenticación para que Guacamole pueda acceder al equipo por medio del protocolo RDP. Estas configuraciones se pueden evidenciar en la Figura 3.35 Figura 3.36.

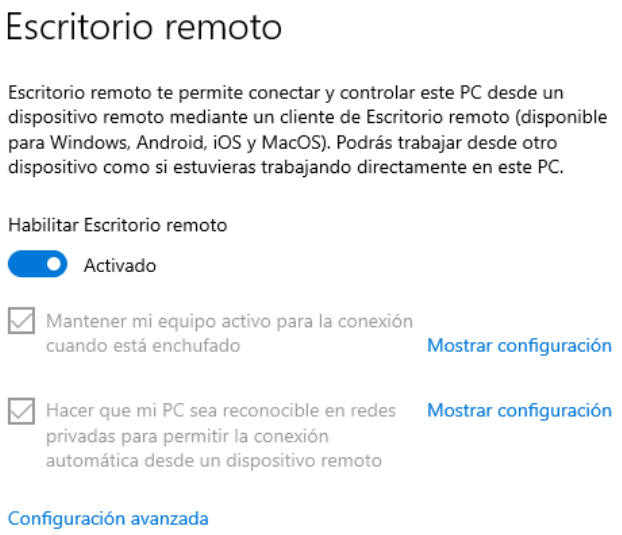

Figura 3.35 Habilitar escritorio remoto en Windows

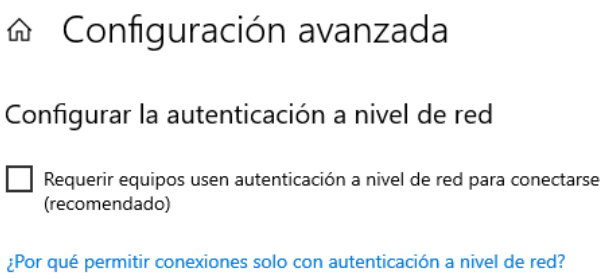

Figura 3.36 Deshabilitar autenticación en escritorio remoto

3.5 Pruebas de funcionamiento del prototipo

Una vez los clientes, así como el servidor son configurados y levantados correctamente se comienza con las pruebas de funcionamiento, donde estas tienen que cumplir con todo el alcance del proyecto.

Como se mencionó en la sección 3.4 en la configuración de Guacamole, los clientes son registrados en un archivo el cual contiene el protocolo que se utiliza, así como el direccionamiento IP de estos. El registro de los clientes para las pruebas de funcionamiento se puede verificar en la Figura 3.37.

```
<user-mapping>

    <!-- Per-user authentication and config information -->

    <!-- A user using md5 to hash the password
         guacadmin user and its md5 hashed password below is used to
          login to Guacamole Web UI-->
    <authorize
            username="kenethadmin"
            password="chester146">

        <!-- First authorized Remote connection -->
        <connection name="orueba VNC">
            <protocol>vnc</protocol>
            <param name="hostname">172.31.116.99</param>
            <param name="port">5900</param>
        </connection>

        <!-- Second authorized remote connection -->
        <connection name="orueba RDP">
            <protocol>rdp</protocol>
            <param name="hostname">172.31.116.78</param>
            <param name="port">3389</param>
            <param name="username">tech</param>
            <param name="ignore-cert">true</param>
         </connection>

        <!-- Third authorized remote connection -->
        <connection name="SSH">
            <protocol>ssh</protocol>
            <param name="hostname">172.31.116.100</param>
                    <param name="port">22</param>
        </connection>

    </authorize>

</user-mapping>
```

Figura 3.37 Registro de los clientes en Guacamole

Con los clientes registrados ya es posible comenzar con las pruebas de funcionamiento. Para esto se ingresa a Guacamole mediante un navegador web, con la dirección IP 172.31.116.97 tal y como se muestra en la Figura 3.38.

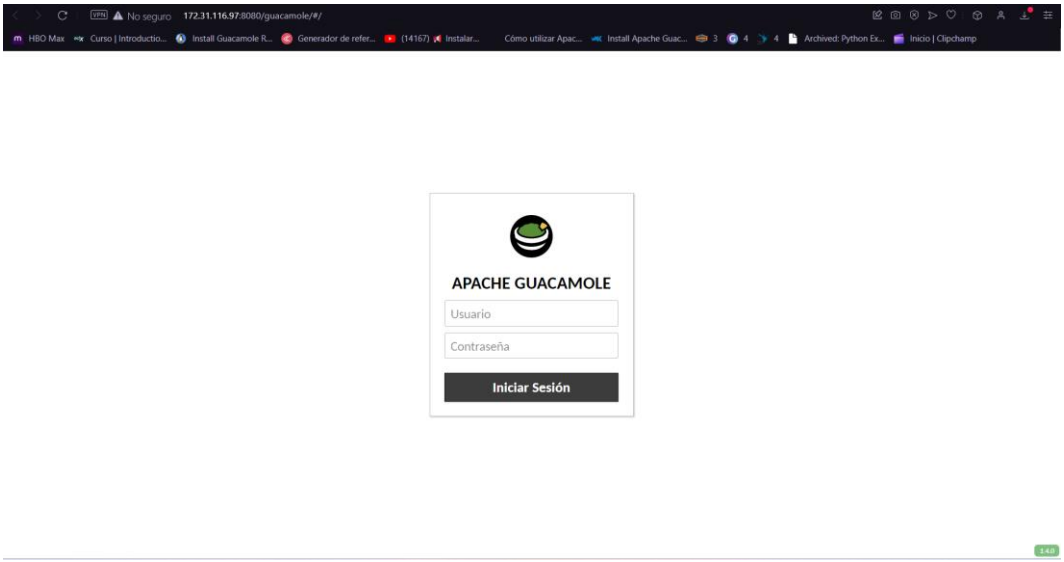

Figura 3.38 Ingreso a Guacamole

Como siguiente se ingresan las credenciales creadas en el archivo donde se encuentran los clientes. Esto se puede visualizar en la Figura 3.39.

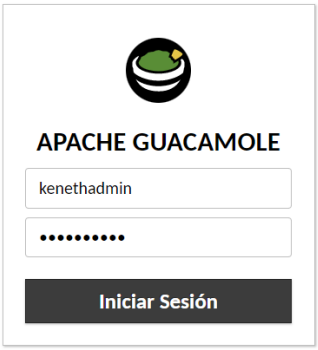

Figura 3.39 Ingreso de credenciales

Una vez ingresados en el servidor se pueden visualizar todas las conexiones que se crearon en el archivo de autenticación. Se pueden ver los clientes a conectar en la Figura 3.40.

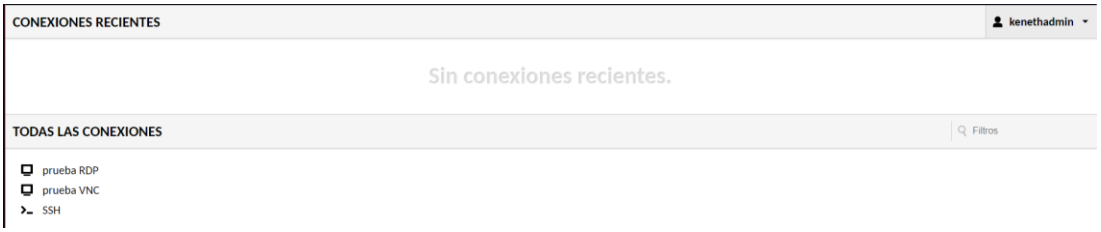

Figura 3.40 Posibles conexiones según el archivo de autenticación

Pruebas de funcionamiento desde un navegador web en PC

La primera prueba de funcionamiento se realiza al equipo con el protocolo RDP, donde se selecciona esta conexión y Guacamole indicara una interfaz para ingresar el usuario al que se desee ingresar con su respectiva contraseña. Esta prueba de conexión se puede evidenciar en la Figura 3.41.

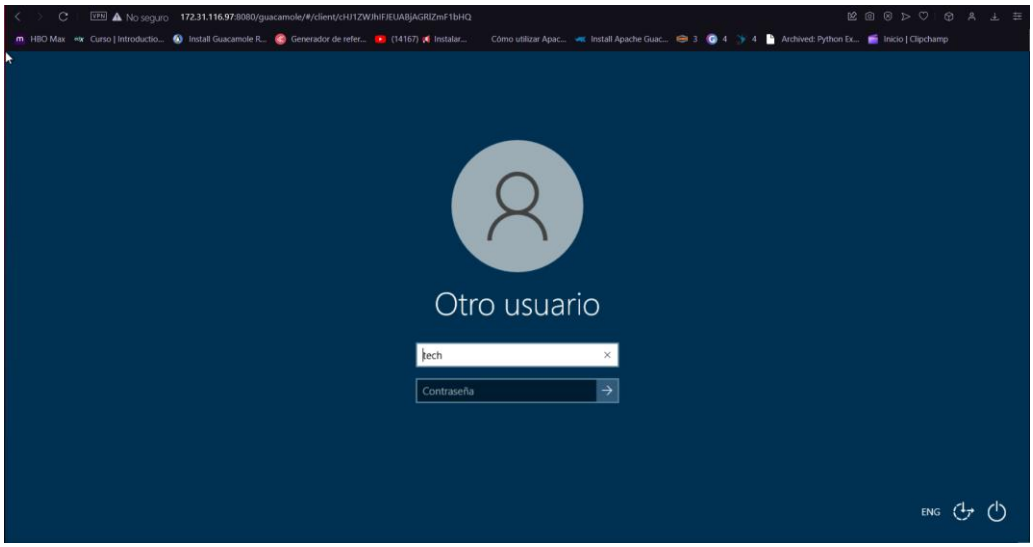

Figura 3.41 Prueba de conexión con el protocolo RDP

Una vez ingresadas las credenciales correspondientes al usuario del equipo Windows, este permite el ingreso y las configuraciones dentro del mismo, como ejemplo sobre el control que se posee del equipo se realizan configuraciones dentro del símbolo del sistema. Estas configuraciones se pueden visualizar en la Figura 3.42.

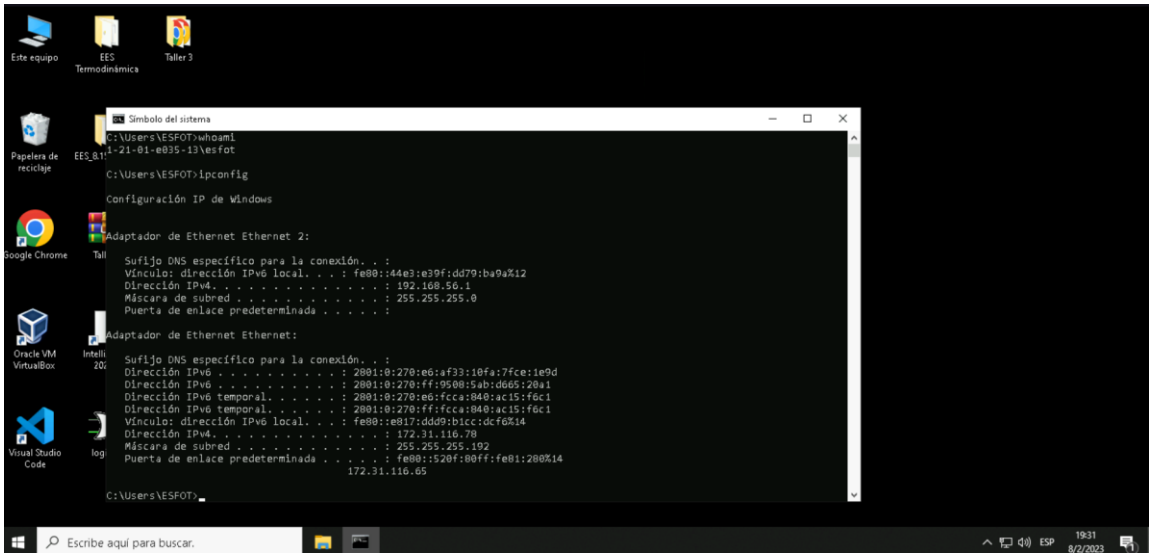

Figura 3.42 Prueba de funcionamiento en equipo Windows con RDP

La siguiente conexión es hacia el equipo con sistema operativo Ubuntu mediante el protocolo VNC. Para realizar esto se seleccionó el equipo correspondiente y se ingresaron las credenciales creadas para la conexión. Esta conexión se puede observar en la Figura 3.43.

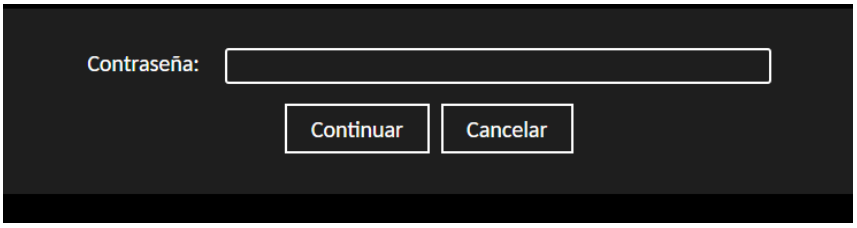

Figura 3.43 Conexión hacia equipo mediante el protocolo VNC

Una vez ingresadas las credenciales como se puede visualizar en la Figura 3.44 se valida el control que se tiene en este. Se utiliza la herramienta Nmap para visualizar los puertos abiertos dentro del sistema operativo y así se comprueba que funciona mediante el protocolo seleccionado.

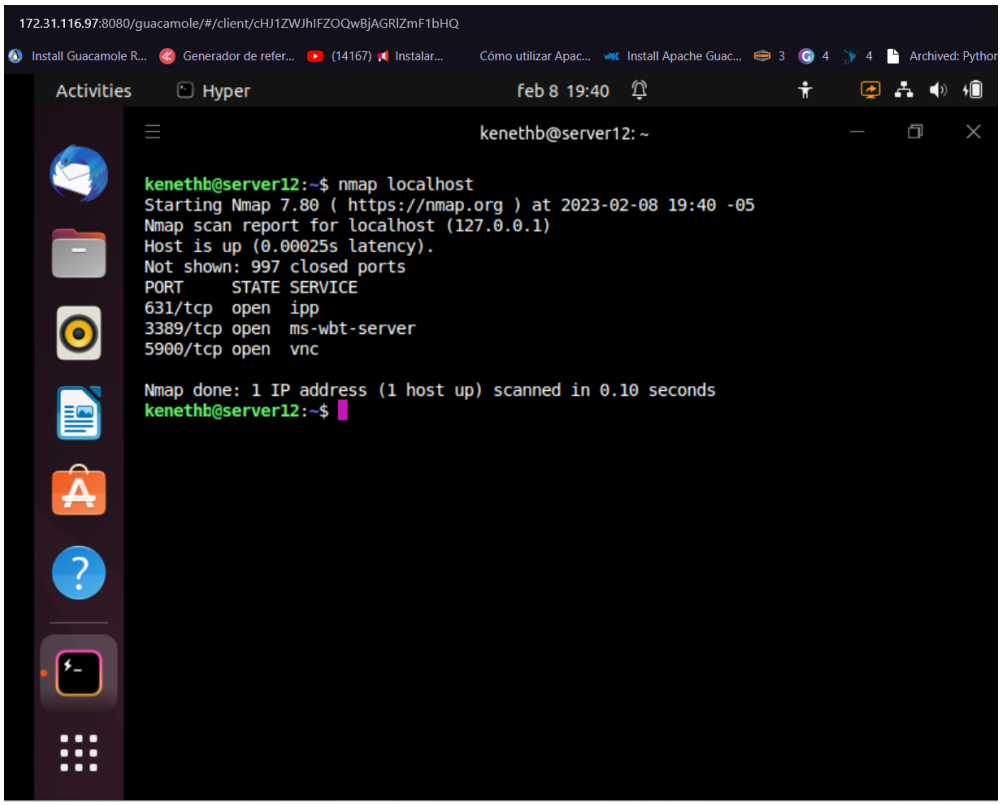

Figura 3.44 Prueba de conexión hacia equipo mediante el protocolo VNC

Para la conexión al servidor con el cliente SSH se repitió el procedimiento anterior, una vez se ingresa a la interfaz se deben escribir las credenciales del usuario con el que se desea conectar. Este procedimiento se puede ver en la Figura 3.45.

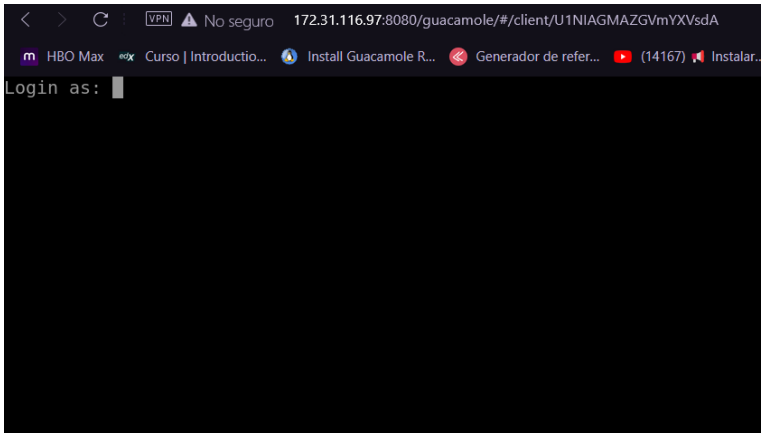

Figura 3.45 Conexión hacia el equipo tipo *server* con SSH

Una vez realizada la conexión, para ver el nivel de control que se tiene por medio de acceso remoto se realizó un escaneo con Nmap, el cual muestra los puertos habilitados en el equipo. Esto se puede visualizar en la Figura 3.46.

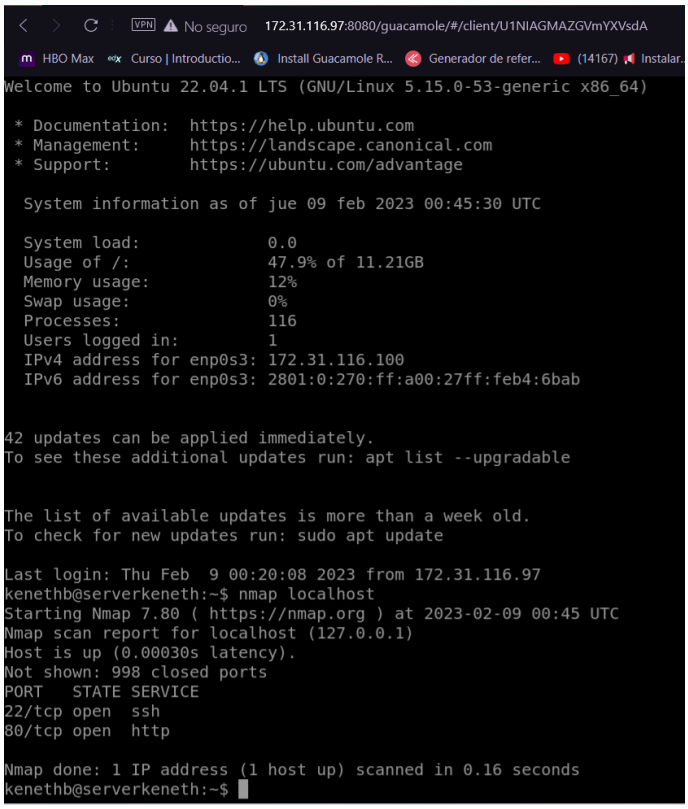

Figura 3.46 Prueba de conexión hacia el equipo tipo *server* con SSH

Pruebas de funcionamiento para equipos celulares

El ingreso hacia Guacamole desde un dispositivo móvil es idéntico al que se tiene en una PC, una vez se ingresa al servidor se escriben las credenciales del usuario autorizado creado en el archivo del servidor. Esto se puede visualizar en la Figura 3.47Figura 3.48.

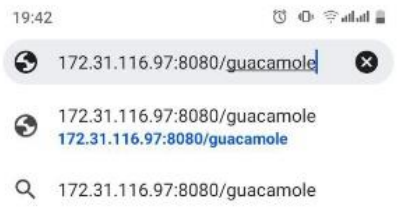

Figura 3.47 Ingreso a Guacamole desde el navegador de un dispositivo celular

Figura 3.48 Ingreso de credenciales

Una vez se ingresa a la página principal de Guacamole se pueden visualizar todas las conexiones. Esto se observa en la Figura 3.49.

Figura 3.49 Página principal de Guacamole en navegador web de teléfono móvil

Igual que con el navegador de PC la primera prueba que se realizó fue con el equipo Windows utilizando el protocolo RDP, por tanto, se realizó la conexión donde se ingresaron las credenciales del usuario y se hicieron configuraciones dentro del equipo para validar el nivel de control que se tiene sobre este. Uno de los grandes beneficios de Guacamole es que posee periféricos digitales como un teclado para controlar la conexión, este se puede activar deslizando la pantalla con el dedo de izquierda a derecha. Para esta demostración se utilizó el teclado de pantalla, el proceso para realizar esto se muestra en la Figura 3.50Figura 3.51Figura 3.52.

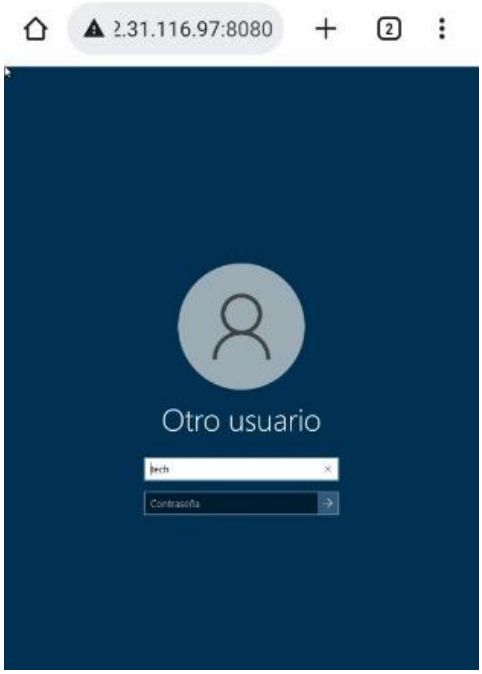

Figura 3.50 Ingreso por medio del protocolo RDP desde un navegador de teléfono celular

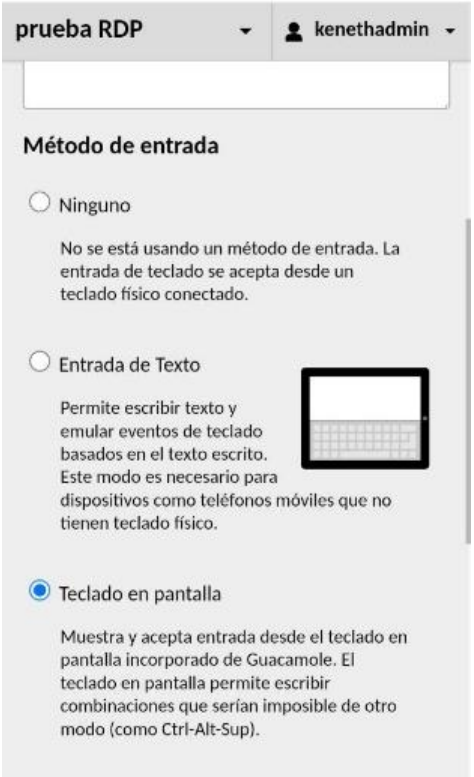

Figura 3.51 Uso de los periféricos digitales de Guacamole

Figura 3.52 Prueba de funcionamiento mediante el protocolo RDP desde el navegador de un teléfono celular

La conexión con los demás clientes es la misma que con el navegador en la PC, aunque el teclado virtual debe ser habilitado cada vez que se ingresa a un nuevo equipo por medio del servidor Guacamole. Las conexiones con los protocolos VNC y SSH se pueden ver en la Figura 3.53 respectivamente.

Figura 3.53 Prueba de funcionamiento mediante el protocolo VNC desde un navegador de teléfono celular

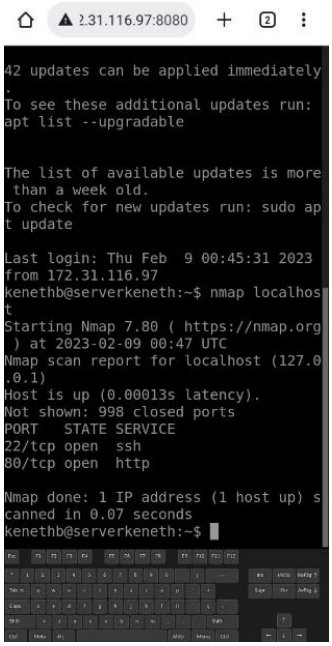

Figura 3.54 Prueba de funcionamiento mediante el protocolo SSH desde un navegador de teléfono celular

Video de funcionamiento del prototipo

En esta sección se encuentra el video donde se evidencia el funcionamiento de Guacamole visto en la sección 3.5. Como se puede ver en la Figura 3.55.

Figura 3.55 Video de Funcionamiento

4 CONCLUSIONES

- El considerar requerimientos de *software* se basa en las necesidades del proyecto de titulación, debido a esto definir estos logra reducir el número de posibles candidatos a considerar como aplicativos de acceso remoto, donde las características más importantes a considerar son que los *softwares* posean la capacidad de realizar un enlace sin la necesidad de un cliente instalado en cada equipo, la forma de conexión sea mediante un navegador web. Estos parámetros objetivos y el alcance del proyecto.

- Una vez realizada una adecuada investigación en diferentes sitios web los candidatos a escoger fueron: *Zoho Assit*, *Access Parallels*, escritorio remoto de Chrome y Apache Guacamole. Estos fueron escogidos debido a que en su mayoría cumple con los 2 requerimientos más importantes mencionados anteriormente, además de otras características que también se plantearon como requerimiento.

- Después de una comparativa entre *softwares* se evidencia que Guacamole es la herramienta que cumple con todos los requerimientos ya establecidos, la característica más importante por la que se eligió este aplicativo es por ser de código abierto y la gran flexibilidad que posee al implementarse como servidor. La versión implementada fue la 4.0 debido a que corrige múltiples errores de conexión con RDP además de tener una comunidad activa la cual ayuda con problemas o errores que pueda tener el usuario como tal.

- El conocer como está diseñado Guacamole ayuda a ver los diferentes agentes que trabajan para el correcto funcionamiento de este, además que permite conocer las limitaciones que puede llegar a tener como *software*, teniendo así un mayor entendimiento hacia este. Entre los diferentes componentes de Guacamole se puede evidenciar que guacd es el *daemon* encargado de toda la arquitectura ya que es el que logra la conexión hacia los dispositivos, produciendo agnosticismo de protocolo, lo cual permite concluir porque este aplicativo puede realizar conexiones hacia cualquier equipo sin discriminación de *hardware* o *software.*

- Para una adecuada implementación los diferentes equipos fueron levantados en máquinas virtuales donde se utilizó *virtual box* debido a las ventajas con respecto a *vm-ware*, la elección de Ubuntu como sistema operativo base se debe a recomendaciones dadas por diferentes sitios web, la cual fue acertada ya que

muchas dependencias de Guacamole son compatibles con este sistema operativo teniendo una instalación sin inconvenientes.

- Por parte de los clientes fue necesario verificar si los puertos correspondientes a el protocolo para la conexión estaban abiertos, una vez comprobado eso fue necesario el habilitar el acceso remoto en cada equipo. El cliente que maneja el protocolo VNC necesito utilizar una herramienta la cual permita el método de autenticación para realizar una conexión por medio de una contraseña, en cambio los clientes con los protocolos SSH y RDP no necesitaron descargar ningún tipo de *software* adicional que permita el habilitar la opción de escritorio remoto.

- Debido a que muchas de las dependencias de Guacamole son compatibles con Ubuntu la instalación de este no tuvo muchos inconvenientes, aunque la versión de *OpenSSL* que maneja la última versión de Ubuntu no es compatible con la versión que necesita el aplicativo, aunque esto es fácilmente reparado al indicar al sistema operativo que no tome los mensajes de advertencia como errores ya que esto impide una correcta complicación en el arranque de las dependencias.

- Una vez el servidor esta levantado y los clientes son correctamente instalados se realizaron pruebas de funcionamiento donde se valida que exista conexión hacia los 3 equipos, para esto fue necesario conocer la dirección IP de cada cliente, además del protocolo correspondiente a cada enlace, con esta información se edita un archivo dentro del servidor de Guacamole para que este pueda realizar las conexiones correspondientes.

- Existieron inconvenientes al realizar las pruebas de funcionamiento con el protocolo RDP ya que la configuración de la dependencia guacd se realizó con un usuario local con privilegios, mas no con el súper usuario de Ubuntu, debido a esto te tuvieron problemas al realizar la conexión al equipo Windows. Aunque esto fue resuelto volviendo a configurar guacd con el usuario correspondiente.

- La conexión hacia dispositivos celulares se realizó de la misma forma que con los equipos tipo PC, aunque se tuvieron inconvenientes por el desconocimiento de periféricos virtuales que brinda Guacamole, este problema fue solventado investigando el problema en la documentación oficial, donde indica el procedimiento para poder acceder a estos periféricos.

5 RECOMENDACIONES

- Se recomienda utilizar un programa adicional que funcione como terminal debido a un error que existe con la última versión de Ubuntu con *Virtual box.*
- Es necesario el realizar todas las configuraciones del servidor como súper usuario, debido a que muchas carpetas y funciones se encuentran restringidas para un usuario local.
- Se recomienda utilizar herramientas como Putty para realizar cambios en el sistema operativo, así como en el aplicativo, también se utiliza para los cambios en el archivo de registro dentro del servidor de Guacamole.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] Apache Guacamole, «Apache Guacamole,» 01 01 2022. [En línea]. Available: https://guacamole.apache.org/. [Último acceso: 19 12 2022].

[2] Apache Software Fundation, «Apache Tomcat,» Apache Software Fundation, 2022. [En línea]. Available: https://tomcat.apache.org/. [Último acceso: 21 12 2022].

[3] F. M. Rojas, «Sistema de Clientes Livianos para entornos Universitarios,» Universidad Cooperativa de Colombia , Colombia , 2011.

[4] J. A. Soto, «¿Que es VNC y para que sirve?,» GEEKNETIC, 16 08 2020. [En línea]. Available: https://www.geeknetic.es/VNC/que-es-y-para-que-sirve. [Último acceso: 21 12 2022].

[5] F. G. d. Zúñiga, «SSH: qué es y cómo funciona este protocolo,» arsys, 17 08 2022. [En línea]. Available: https://www.arsys.es/blog/ssh#Que_es_SSH. [Último acceso: 21 12 2022].

[6] IBM, «Daemons TCP/IP,» 12 04 2021. [En línea]. Available: https://www.ibm.com/docs/es/aix/7.2?topic=protocol-tcpip-daemons. [Último acceso: 21 12 2022].

[7] J. D. P. Jimenez, «Que es HTML5,» OpenWebinars, 20 01 2019. [En línea]. Available: https://openwebinars.net/blog/que-es-html5/. [Último acceso: 21 12 2022].

[8] deepin, «dconf,» 21 01 2021. [En línea]. Available: https://xn--deepinenespaol-1nb.org/wiki/dconf/. [Último acceso: 21 12 2022].

[9] VMware, «Hipervisor,» VMware, 2023. [En línea]. Available: https://www.vmware.com/latam/topics/glossary/content/hypervisor.html#:~:text=Un%20hipervisor%2C%20conocido%20tambi%C3%A9n%20como,la%20memoria%20y%20el%20procesamiento.. [Último acceso: 23 01 2023].

[10] VirtualBox, «Bienvenido a VirtualBox.org,» 2023. [En línea]. Available: https://www.virtualbox.org/. [Último acceso: 23 01 2023].

[11] Zoho , «Software de asistencia remota de nivel empresarial. Software seguro de acceso remoto a dispositivos desatendidos.,» 2023. [En línea]. Available: https://www.zoho.com/es-xl/assist/. [Último acceso: 25 01 2023].

[12] Parallels, «Parallels Access,» 2023. [En línea]. Available: https://www.parallels.com/es/products/access/. [Último acceso: 25 01 2023].

[13] Google Chrome, «Conéctate fácilmente de forma remota a la computadora del hogar o del trabajo, o comparte tu pantalla con otras personas.,» 2023. [En línea]. Available: https://remotedesktop.google.com/?hl=es-419&pli=1. [Último acceso: 25 01 2023].

[14] Zoho Assit, «Requisitos del sistema,» 2023. [En línea]. Available: https://www.zoho.com/es-xl/assist/system-requirements.html. [Último acceso: 25 01 2023].

[15] Parallels, «Requisitos del sistema para Parallels Access,» 25 01 2023. [En línea]. Available: https://kb.parallels.com/es/116670. [Último acceso: 25 01 2023].

[16] Google Chrome, «Requisitos del sistema del navegador Chrome,» 2023. [En línea]. Available: https://support.google.com/chrome/a/answer/7100626?hl=es. [Último acceso: 25 01 2023].

[17] Apache Guacamole, «Instalando Guacamole de forma nativa,» 2023. [En línea]. Available: https://guacamole.apache.org/doc/gug/installing-guacamole.html#:~:text=Required%20dependencies,cannot%20be%20built%20without%20them.&text=Cairo%20is%20used%20by%20libguac%20for%20graphics%20rendering.. [Último acceso: 25 01 2023].

[18] Zoho , «Conexión a escritorio remoto,» 2023. [En línea]. Available: https://www.zoho.com/es-xl/assist/remote-desktop-connection.html. [Último acceso: 25 01 2023].

[19] Parallels, «Seleccione el plan que se adapte mejor a sus necesidades,» 2023. [En línea]. Available: https://www.parallels.com/es/products/access/compare-plans/. [Último acceso: 25 01 2023].

[20] Google Chrome, «Cómo acceder a otra computadora con el Escritorio remoto de Chrome,» 2023. [En línea]. Available: https://support.google.com/chrome/answer/1649523?hl=es-419&co=GENIE.Platform%3DDesktop#:~:text=de%20forma%20remota.-,En%20la%20computadora%2C%20abre%20Chrome.,luego%2C%20haz%20clic%20en%20Conectar.. [Último acceso: 25 01 2023].

[21] Apache Guacamole, «Using Guacamole,» 2023. [En línea]. Available: https://guacamole.apache.org/doc/gug/using-guacamole.html. [Último acceso: 25 01 2023].

[22] IBM, «Software de código abierto frente a software de código cerrado,» 2022. [En línea]. Available: https://www.ibm.com/ar-es/topics/open-source#:~:text=Este%20tipo%20de%20software%20escalable,una%20mayor%20accesibilidad%20y%20reutilizaci%C3%B3n.. [Último acceso: 25 01 2023].

[23] P. I. GmbH, «Parallels Access Guia de usuario,» Suiza- Schaffhausen, 2018.

[24] D. Oniera, «¿Chrome consume mucha CPU, incluso cerrado? Así puedes evitarlo,» 22 09 2021. [En línea]. Available: https://www.softzone.es/programas-top/chrome/evitar-chrome-consuma-cpu-apagado/. [Último acceso: 02 02 2023].

[25] Apache Guacamole, «Manual Apache Guacamole,» 2023. [En línea]. Available: https://guacamole.apache.org/doc/gug/installing-guacamole.html#:~:text=Required%20dependencies,cannot%20be%20built%20without%20them.&text=Cairo%20is%20used%20by%20libguac%20for%20graphics%20rendering.. [Último acceso: 23 01 2023].

[26] idroot, «How To Install Apache Guacamole on Ubuntu 22.04 LTS,» 17 10 2022. [En línea]. Available: https://idroot.us/install-apache-guacamole-ubuntu-22-04/. [Último acceso: 24 01 2023].

[27] Ubuntu , «Notas de la versión de Jammy Jellyfish,» 17 10 2022. [En línea]. Available: https://discourse.ubuntu.com/t/jammy-jellyfish-release-notes/24668?_ga=2.43795501.922155397.1673976792-220368905.1673976792. [Último acceso: 24 01 2023].

[28] Ubuntu, «Download Ubuntu Desktop,» 2023. [En línea]. Available: https://ubuntu.com/download/desktop. [Último acceso: 24 01 2023].

[29] Ubuntu, «Download Ubuntu Desktop,» 2023. [En línea]. Available: https://ubuntu.com/download/desktop. [Último acceso: 06 02 2023].

[30] K. John, «Install Guacamole Remote Desktop on Ubuntu 22.04 (Jammy Jellyfish),» 06 06 2022. [En línea]. Available: https://computingforgeeks.com/install-guacamole-remote-desktop-on-ubuntu-jammy-jellyfish/. [Último acceso: 06 02 2023].

[31] koromicha, «Instale Apache Guacamole en Ubuntu 22.04,» 02 05 2022. [En línea]. Available: https://kifarunix.com/install-apache-guacamole-on-ubuntu/. [Último acceso: 07 02 2023].

[32] Apache Guacamole, «Chapter 2. Installing Guacamole,» 2022. [En línea]. Available: https://guacamole.apache.org/doc/0.9.1/gug/installing-guacamole.html. [Último acceso: 24 01 2023].

7 ANEXOS

ANEXO I. Certificado de originalidad

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 28 de febrero de 2023

De mi consideración:

Yo, LEANDRO ANTONIO PAZMIÑO ORTIZ, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ACCESO REMOTO A EQUIPOS EN UNA RED A TRAVÉS DEL NAVEGADOR WEB elaborado por KENETH ELIAM BOADA PARRA estudiante de la carrera en TECNÓLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11 %.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

Reporte de similitud completo - KBoada.pdf

Atentamente,

Leandro Antonio Pazmiño Ortiz

Docente

Escuela de Formación de Tecnólogos