

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN SERVIDOR DE ALMACENAMIENTO Y COMPARTICIÓN DE ARCHIVOS EN LA NUBE DEL TIPO OPEN SOURCE

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

KEVIN JOEL CABRERA QUINCHE

DIRECTOR: ANDRÉS FERNANDO REYES CASTRO

DMQ, agosto 2023

CERTIFICACIONES

Yo, KEVIN JOEL CABRERA QUINCHE 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.

KEVIN JOEL CABRERA QUINCHE

kevin.cabrera01@epn.edu.ec

kevincabr.14@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por KEVIN JOEL CABRERA QUINCHE, bajo mi supervisión.

ANDRÉS FERNANDO REYES CASTRO
DIRECTOR

andres.reyes@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.

KEVIN JOEL CABRERA QUINCHE

DEDICATORIA

Mi deseo es asignar este proyecto de titulación en honor de mi familia, quienes han sido una parte fundamental durante mi trayectoria en la universidad, especialmente a mi madre. Maura la cual siempre ha estado pendiente de mi progreso en la universidad, la cual, con su amor incondicional, sabios consejos y apoyo me ha guiado para no rendirme cuando las cosas se ponen mal ya que siempre habrá una salida. A mi padre Cristobal quien a pesar de no estar presente en todo momento ha estado en las situaciones más críticas apoyándome con sus consejos y experiencia, haciendo que de lo mejor de mí para poder ser cada día mejor. Finalmente, se lo dedico a mi hermana Mishell quien ha estado conmigo en situaciones buenas y malas, alegrándose por todos mis logros y ayudándome a seguir adelante en mis momentos más dolorosos, además de ser la persona por la cual me quiera superar y así llegar a ser un ejemplo de superación para ella.

AGRADECIMIENTO

Expreso mi gratitud a Dios por otorgarme salud y fortaleza, permitiéndome perseverar en la búsqueda de mis objetivos y anhelos. Quiero expresar mi profunda gratitud hacia mi familia, por constituir el núcleo de apoyo que me ha impulsado a no claudicar durante mi recorrido universitario. Reconozco y valoro la orientación proporcionada por mi tutor de tesis, el ingeniero Andrés Reyes, cuya experiencia ha sido fundamental para guiarme en la realización de este trabajo de titulación.

Le doy gracias a la Escuela Politécnica Nacional por permitirme formar en su establecimiento y adquirir el conocimiento de un gran grupo de profesionales que forman parte de ella. También por las amistades que he podido formar a lo largo de toda mi vida universitaria las cuales han hecho más amena mi estadía en esta prestigiosa Universidad.

Kevin.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDOS	V
RESUMEN.....	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	1
1.2 Objetivos específicos.....	1
1.3 Alcance.....	2
1.4 Marco Teórico.....	2
Virtualización	2
Computación en la nube	3
Modelos de Servicio en la Nube	3
Máquinas Virtuales de Azure	4
Almacenamiento en la Nube	5
Owncloud.....	5
Ubuntu Server 22.04.02.....	6
PHP (Hypertext Preprocessor).....	6
Apache	7
MySQL	7
OpenSSL.....	8
2 METODOLOGÍA	8
3 RESULTADOS	9
3.1 Requerimientos del servidor.	9
3.2 Selección del <i>hardware</i> y <i>software</i> según los requerimientos identificados...	10

<i>Hardware</i>	10
<i>Software</i>	11
3.3 Implementación del servidor de almacenamiento en un ambiente local.....	12
Creación de máquina virtual	12
Instalación de los repositorios de PHP.....	12
Instalación de los paquetes requeridos e iniciación.	13
Creación de la base de datos	14
Instalación del servidor de Owncloud.....	15
Configuración de IP estática	20
Seguridad	20
Configuración de <i>Memcache</i> , <i>Cron</i> y <i>HTTP Strict Transport Security</i>	28
3.4 Pruebas de funcionamiento del servidor en ambiente local	31
Creación de usuarios.....	31
Carga de archivos y creación de carpetas	31
Compartición de archivos	32
Compartición de carpetas	33
Compartición mediante enlace.....	34
3.5 Implementación del servidor en una plataforma de hosting.	36
3.6 Pruebas de funcionamiento del servidor en plataforma Host.	38
Creación de usuarios.....	38
Carga de archivos y creación de carpetas	39
Compartición de archivos y carpetas	39
Compartición de archivo mediante enlace	40
4 CONCLUSIONES.....	41
5 RECOMENDACIONES.....	42
6 REFERENCIAS BIBLIOGRÁFICAS.....	43
7 ANEXOS.....	47
ANEXO I: Certificado de Originalidad	i
ANEXO II: Enlaces	ii

RESUMEN

En el marco de esta instancia de Integración Curricular, se procede a la configuración de un servidor para el almacenamiento y distribución de archivos en la nube, fundamentado en una solución de código abierto. Este servidor provee la capacidad de resguardar datos de manera segura y simplifica su acceso a través de conexiones remotas, permitiendo a los usuarios acceder y compartir la información almacenada con otros individuos en línea.

El primer capítulo abarca la narrativa sobre la descripción del componente que se pretende desarrollar. Junto a esto, se plantea el objetivo general y los objetivos específicos que constituirán los cimientos del proyecto en cuestión. También se detalla el alcance del proyecto y se presenta un marco teórico en el que se exponen las tecnologías empleadas.

En el segundo capítulo se elabora la metodología que será empleada, partiendo de los objetivos generales y específicos, con el propósito de abordar cada uno de estos de manera integral.

El tercer capítulo detalla los requisitos indispensables para la operación y desempeño adecuado del servidor. Además, se consideran los requisitos tanto de *Hardware* como de *Software*, tomando en consideración la documentación proporcionada por Owncloud. Esto asegura que el sistema operativo y los paquetes requeridos sean compatibles con la versión del servidor. También se proporciona un detalle exhaustivo del procedimiento de implementación del servidor en un entorno local y en la plataforma Microsoft Azure. Adicionalmente, se aborda la mejora de la seguridad y se explican las pruebas de funcionamiento realizadas en ambos entornos.

Los capítulos cuatro y cinco se enfocan en las conclusiones y recomendaciones derivadas al concluir el proyecto, con la finalidad de brindar orientación para el avance de próximas iniciativas.

PALABRAS CLAVE: Nube, *código abierto*, conexión remota, Owncloud, IaaS, seguridad.

ABSTRACT

Within the framework of this instance of Curricular Integration, a server for the storage and distribution of files in the cloud is configured, based on an open source solution. This server provides the ability to securely store data and simplifies its access through remote connections, allowing users to access and share stored information with other individuals online.

The first chapter covers the narrative about the description of the component that is intended to be developed. Along with this, the general objective and the specific objectives that will constitute the foundations of the project in question are stated. The scope of the project is also detailed and a theoretical framework is presented in which the technologies used are exposed.

In the second chapter, the methodology that will be used is elaborated, starting from the general and specific objectives, with the purpose of addressing each of these in an integral way.

The third chapter details the essential requirements for the proper operation and performance of the server. In addition, both Hardware and Software requirements are considered, taking into consideration the documentation provided by Owncloud. This ensures that the operating system and required packages are compatible with the version of the server. A comprehensive detail of the server deployment procedure in an on-premises environment and on the Microsoft Azure platform is also provided. Additionally, the improvement of security is addressed and the functional tests carried out in both environments are explained.

Chapters four and five focus on the conclusions and recommendations derived at the end of the project, in order to provide guidance for the advancement of future initiatives.

KEYWORDS: *Cloud, open source, remote connection, Owncloud, IaaS, security.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El uso de almacenamiento en la nube ha evidenciado ser la solución más eficiente y segura para la administración de datos tanto a nivel individual como empresarial. Permitiendo así acceder, almacenar y gestionar los datos de manera remota mediante el uso de los servicios en la nube, en lugar de solo tener servidores locales; resolviendo así el desafío del almacenamiento de datos.

Basándose en esta premisa, en el proyecto actual se lleva a cabo la instalación de un servidor de almacenamiento y distribución de archivos en la nube, utilizando una solución de código abierto, usando máquinas virtuales además del sistema operativo y paquetes necesarios para la instalación de Owncloud. Owncloud es el servicio de almacenamiento en la nube usado para este proyecto. Mediante el uso de las herramientas mencionadas, el primer servidor instalado permite almacenar y compartir archivos de manera local.

Para poder acceder a la información de manera remota se implementa el servidor en una plataforma de *hosting*, en este caso Microsoft Azure, donde se puede crear una máquina virtual con un costo accesible, pero para el proyecto se utilizan los créditos gratuitos que otorga la plataforma, los cuales son suficientes para implementar el servidor de almacenamiento. Este segundo servidor permite además de almacenar y compartir archivos acceder de manera remota a la información siempre y cuando el usuario tenga conexión a Internet. En ambos casos se usa HTTPS como método de seguridad.

1.1 Objetivo general

Implementar un servidor de almacenamiento y compartición de archivos en la nube del tipo *Open Source*.

1.2 Objetivos específicos

- Establecer los requerimientos del servidor.
- Seleccionar el *hardware* y *software* según los requerimientos identificados.
- Implementar el servidor de almacenamiento en un ambiente local.
- Implementar el servidor en una plataforma de hosting.
- Realizar pruebas de funcionamiento del servidor.

1.3 Alcance

Por medio del presente proyecto se busca realizar la implementación de un servidor de almacenamiento y compartición de archivos en la nube del tipo *Open Source*; para ello se utilizarán máquinas virtuales, servidor Apache, base de datos MySQL o similar y lenguaje de programación PHP. Adicionalmente, el servidor deberá tener seguridad y listas de acceso para garantizar la privacidad de la información de los usuarios.

En primer lugar, se realizará la implementación del servidor en un ambiente local y se realizarán pruebas de funcionamiento del servicio. Una vez realizadas las pruebas locales, se personalizará el servidor para crear un repositorio de archivos y que cada usuario posea su propia carpeta de almacenamiento de archivos.

Finalmente, el servidor será implementado en una plataforma de *hosting* para tener un acceso remoto a los recursos.

1.4 Marco Teórico

Virtualización

La virtualización es una herramienta tecnológica que permite la creación de entornos virtuales aislados y distintos a partir de recursos físicos subyacentes. Mediante esta técnica, es factible operar múltiples sistemas operativos y aplicaciones simultáneamente en un único servidor físico, optimizando la eficiencia de los recursos y simplificando la implementación y gestión de la infraestructura. Algunas de las tecnologías de virtualización más destacadas son [1] [2] [3]:

- Virtualización de servidores: Permite la creación de varios servidores virtuales en un mismo servidor físico mediante la aplicación de un *software*. Esto permite que un solo servidor físico pueda ejecutar múltiples sistemas operativos y aplicaciones de forma independiente, como si cada uno de ellos estuviera en su propio servidor dedicado. Lo que proporciona una mayor eficiencia en el uso de los recursos disponibles.
- Virtualización de redes: Posibilita la creación de redes virtuales separadas y aisladas dentro de un mismo entorno físico compartido. Gracias a esta técnica, se pueden configurar y administrar las redes virtuales de forma independiente, lo que facilita la aplicación de políticas de seguridad y la segmentación de redes para diferentes aplicaciones o clientes.

- Virtualización de escritorios: Permite la ejecución de múltiples escritorios en un mismo hospedaje. Esto se logra gracias a la capacidad de acceso remoto a los escritorios desde cualquier dispositivo o lugar. Esta técnica ofrece una mayor fiabilidad en la disponibilidad de los escritorios. Además, permite una administración centralizada de las mesas de trabajo, lo que facilita la gestión y el mantenimiento de los escritorios virtuales.

Computación en la nube

La computación en la nube es una innovadora herramienta que viabiliza la provisión de servicios informáticos a través de Internet. Para los usuarios, esto implica prescindir de servidores locales, ya que pueden acceder a recursos computacionales como almacenamiento, capacidad de procesamiento y software a través de proveedores de servicios en la nube. Este enfoque ofrece una flexibilidad, escalabilidad y eficiencia significativamente superiores a los métodos convencionales de infraestructura de tecnología de la información. Algunas de las características más importantes de la nube son [4] [5] [6]:

- Elasticidad: Los recursos en la nube pueden ajustarse de manera automática para satisfacer las exigencias y necesidades de los usuarios.
- Pago por uso: Los usuarios solo pagan por los recursos que consumen, lo que mejora la rentabilidad.
- Acceso bajo demanda: Los servicios en la nube están disponibles en cualquier momento y en cualquier lugar según la necesidad del usuario.
- Autoservicio: Los usuarios tienen la capacidad de ajustar y adaptar los servicios en la nube conforme a sus requerimientos, sin necesidad de la intervención del proveedor de servicios.

Modelos de Servicio en la Nube

La computación en la nube presenta diversos modelos de servicio que indican la manera en que se ponen a disposición de los usuarios. Los modelos de servicio más comunes son: [7] [8] [9]:

- Infraestructura como Servicio (IaaS): Los proveedores de servicios en la nube suministran recursos de virtualización de infraestructura, como servidores, redes y almacenamiento. Los usuarios emplean estos recursos para instaurar y gestionar sus propias aplicaciones y sistemas operativos, manteniendo un control absoluto sobre los mismos. Este enfoque en la nube

habilita una flexibilidad y escalabilidad más amplias en la administración de la infraestructura informática. Los ejemplos de proveedores de IaaS incluyen *Amazon Web Services (AWS)* y *Microsoft Azure*.

- **Plataforma como Servicio (PaaS):** El proveedor de servicios en la nube proporciona una plataforma integral destinada al desarrollo y despliegue de aplicaciones. Los usuarios tienen la capacidad de crear, instaurar y gestionar aplicaciones sin la necesidad de lidiar con la infraestructura subyacente. La gestión de la infraestructura, incluyendo servidores y redes, recae en manos de los proveedores de servicios en la nube. Los ejemplos de proveedores de PaaS incluyen *Google Cloud Platform (GCP)* y *Heroku*.
- **Software como Servicio (SaaS):** La provisión integral de una aplicación por parte de un proveedor de servicios en la nube. En esta situación, la aplicación se ejecuta en el entorno de la nube y se encuentra accesible a través de Internet. Los usuarios pueden ingresar a la aplicación mediante un navegador web o una interfaz de usuario, y utilizarla sin la necesidad de atender a la infraestructura subyacente o el mantenimiento de la aplicación. Ejemplos de servicios SaaS son *Salesforce* y *Google Workspace*.

Máquinas Virtuales de Azure

Azure Virtual Machines es un servicio IaaS proporcionado por Microsoft Azure. Permite a los usuarios implementar y ejecutar sistemas operativos y aplicaciones completos en la nube de Azure. Los aspectos importantes de *Azure Virtual Machines* son [10] [11] [12]:

- **Configuración y dimensionamiento:** Azure ofrece muchos tamaños de máquinas virtuales para satisfacer diferentes necesidades de rendimiento y capacidad. Los usuarios pueden elegir la cantidad de núcleos de CPU, memoria, almacenamiento y otros recursos según sus necesidades.
- **Administración y escalabilidad:** Las máquinas virtuales de Azure se pueden administrar fácilmente empleando el *Azure Portal*, la CLI (Interfaz de Línea de Comandos) de Azure y más. Los usuarios pueden aumentar o reducir las máquinas virtuales según sea necesario, agregar o eliminar recursos. Además, tiene la capacidad de llevar a cabo copias de respaldo y restablecer máquinas virtuales.

- Redes y conectividad: Azure permite configurar redes virtuales para conectar las VM y proporcionar conectividad segura a través de VPN (Red Privada Virtual) o *ExpressRoute* (conexión de red dedicada).

Almacenamiento en la Nube

El servicio de almacenamiento en la nube permite a los usuarios guardar y recuperar datos de forma segura y escalable a través de Internet. Componentes fundamentales de este servicio comprenden [13] [14] [15]:

- Almacenamiento duradero: Los proveedores de servicios en la nube, como Azure, proporcionan un almacenamiento robusto y confiable para resguardar sus datos a largo plazo. Estos datos pueden ser accedidos y empleados desde cualquier ubicación que tenga acceso a Internet.
- Escalabilidad: El almacenamiento en la nube le permite escalar sus recursos de almacenamiento según sus necesidades. Los proveedores de la nube ofrecen opciones para aumentar o disminuir la capacidad de almacenamiento de manera flexible y en tiempo real.
- Replicación y redundancia: Los servicios de almacenamiento en la nube suelen proporcionar mecanismos de replicación y redundancia para garantizar la accesibilidad y salvaguardia de los datos resguardados. Esto incluye copias de seguridad automatizadas, almacenamiento distribuido y replicación geográfica.
- Acceso seguro: La nube proporciona sofisticadas medidas de seguridad para salvaguardar los datos guardados, abarcando la verificación de identidad de usuarios, el encriptado tanto en el reposo como durante la transferencia, y el manejo de accesos según roles.

Owncloud

Owncloud constituye una plataforma de código abierto para la retención y colaboración en línea de documentos. Concede una vía segura y práctica para el almacenamiento, sincronización y distribución de archivos y directorios a través de diversos dispositivos. Los elementos primordiales de Owncloud abarcan: [16] [17] [18]:

- Almacenamiento y sincronización: Owncloud permite almacenar archivos y carpetas en un servidor central y sincronizarlos automáticamente en varios dispositivos para facilitar el acceso y la colaboración.

- **Compartición de archivos:** Owncloud presenta alternativas adaptables para la distribución de archivos y directorios entre colaboradores internos o externos. Es factible establecer autorizaciones y límites con el propósito de salvaguardar y ejercer dominio sobre los datos compartidos.
- **Seguridad:** La protección de los datos es una principal prioridad para Owncloud, evidenciada por su dedicación a través de sólidas características de autenticación y cifrado. La información se encuentra protegida mediante el cifrado tanto en reposo como durante la transferencia, y se emplean métodos de autenticación para blindar contra accesos indebidos.

Ubuntu Server 22.04.02

La versión 22.04.02 de *Ubuntu Server* representa una variante del sistema operativo Linux arraigada en Ubuntu, concebida especialmente para entornos de servidor. Entre las características fundamentales de *Ubuntu Server 22.04.02* se incluyen [19] [20] [21]:

- **Estabilidad y seguridad:** *Ubuntu Server* se caracteriza por su estabilidad y seguridad. La versión 22.04.02 proporciona correcciones de errores, parches de seguridad y actualizaciones regulares para garantizar un entorno confiable y seguro.
- **Gestión de servicios y paquetes:** En *Ubuntu Server*, se aprovecha el sistema de gestión de paquetes avanzado *APT (Advanced Package Tool)*, simplificando la instalación, actualización y desinstalación de software en su servidor. Además, se integran herramientas de control de servicios que permiten supervisar y administrar los servicios en funcionamiento.
- **Soporte a largo plazo (LTS):** La versión 22.04.02 es una versión LTS. Esto significa que se brinda soporte extendido a largo plazo y actualizaciones de seguridad para brindar estabilidad y confiabilidad a largo plazo.

PHP (Hypertext Preprocessor)

PHP se erige como un lenguaje de programación de código abierto orientado al desarrollo web. Su popularidad se fundamenta en su accesibilidad, versatilidad y el respaldo que recibe de la comunidad. Las principales características de PHP son [22] [23] [24]:

- **Sintaxis amigable para los desarrolladores:** PHP usa una sintaxis similar a otros lenguajes de programación como C y Java, lo que facilita la transición y la comprensión de los desarrolladores.

- Soporte de servidor: La operatividad de PHP tiene lugar en el servidor, implicando que el código PHP es procesado por el servidor web antes de ser enviado al cliente. Esta dinámica posibilita la creación de contenido dinámico y la ejecución de acciones desde la perspectiva del servidor.
- Amplia compatibilidad: PHP es compatible con una amplia gama de servidores web y sistemas operativos, lo que lo hace altamente portátil y accesible para desarrolladores de diferentes entornos.

Apache

Apache se destaca como un servidor *web* de amplio uso en el ámbito de código abierto, habilitando la puesta en marcha de sitios y aplicaciones *web*. Las características clave de Apache son [25] [26] [27]:

- Alto rendimiento: Apache está diseñado para manejar muchas solicitudes web simultáneas y proporcionar un rendimiento óptimo incluso en condiciones de carga pesada.
- Flexibilidad: Apache admite una amplia variedad de módulos y extensiones, lo que le permite personalizar y ampliar su funcionalidad de acuerdo con las necesidades específicas del proyecto.
- Seguridad: Apache dispone de capacidades de seguridad robustas que resguardan tanto sitios web como aplicaciones frente a amenazas comunes, tales como ataques de DDoS (Denegación de Servicio) y ataques de inyección de código.

MySQL

MySQL se presenta como un sistema de administración de bases de datos que se integra de manera perfecta con PHP y Apache. Algunas de las principales características de MySQL son [28] [29] [30]:

- Escalabilidad: MySQL puede manejar grandes cantidades de datos y es altamente escalable. Esto lo convierte en una opción popular para sitios web y aplicaciones que esperan un aumento significativo en el número de usuarios y datos.
- Rendimiento: MySQL está diseñado para proporcionar un rendimiento rápido y eficiente para leer y escribir datos. Mejora el rendimiento de las consultas con índices, consultas optimizadas y almacenamiento en caché en memoria.

- Seguridad: MySQL proporciona funcionalidades de seguridad avanzadas, como la encriptación de datos, la verificación de identidad de usuarios y la gestión de permisos. Estas medidas garantizan la preservación de la confidencialidad y la integridad de los datos almacenados.

OpenSSL

OpenSSL es una librería de *software* de código abierto que brinda funciones criptográficas para garantizar la seguridad de las comunicaciones y transacciones en línea. Entre las principales características de OpenSSL están [31] [32] [33]:

- Protocolos de seguridad: OpenSSL es compatible con diversas normativas de seguridad, tales como SSL (*Secure Sockets Layer*) y TLS (*Transport Layer Security*). Estos protocolos brindan la capacidad de encriptar y autenticar tanto los datos del servidor como los del cliente, asegurando que la información transmitida se mantenga confidencial e intacta.
- Cifrado de datos: OpenSSL proporciona algoritmos criptográficos sólidos y seguros para proteger los datos en reposo y en tránsito. Esto incluye algoritmos simétricos y asimétricos que garantizan la confidencialidad y autenticidad de los datos.
- Certificados digitales: OpenSSL permite generar y administrar certificados digitales utilizados para autenticar las identidades de servidores y clientes. Estos certificados son esenciales para establecer una conexión segura y confiable.

2 METODOLOGÍA

En primer lugar, se realizó un análisis para identificar los requerimientos necesarios para implementar el servidor y cumplir con el alcance establecido.

Mediante la definición de componentes de hardware se determinaron las características del equipo necesario para la implementación del servidor de manera local. Adicionalmente, se determinó el sistema operativo y los paquetes necesarios para levantar el servicio.

En un ambiente local y controlado, y con una máquina virtual se implementaron los paquetes necesarios para el servidor de almacenamiento y compartición de archivos. Se utilizó Apache para levantar el servicio web, MySQL como base de datos y PHP como lenguaje de programación. Una vez instalados, se procedió con la implementación

del servidor. Con respecto a la seguridad del servicio, se aplicó HTTPS (*HyperText Transfer Protocol Secure*).

Una vez culminada la implementación del servidor, se llevaron a cabo pruebas de funcionamiento con el fin de verificar su adecuado rendimiento. Las pruebas consistieron en: creación de usuarios con sus respectivas credenciales de ingreso, compartición de archivos entre usuarios, compartición de carpetas y comprobación de la sincronización de datos compartidos.

Con lo aprendido en el ambiente local, se procedió a implementar el servidor en una plataforma de hosting, en este caso Azure. Si bien esta plataforma no es gratuita, ofrece licencias con créditos para practicar en la misma, lo cual fue suficiente para desarrollar la implementación del servidor.

Asimismo, se ejecutaron pruebas de funcionamiento que posibilitaron la confirmación del desempeño adecuado y la conexión remota exitosa del usuario con el servidor.

3 RESULTADOS

El servidor de Owncloud para el almacenamiento y compartición de archivos, permite que el usuario pueda almacenar archivos, crear carpetas y crear textos con el fin de que este pueda acceder en cualquier momento a la información que necesite revisar. Además, Owncloud permite la compartición de archivos y carpetas, para ello existen dos formas: la primera es mediante la búsqueda de la persona a la que se quiera compartir dicho archivo carpeta en el servidor y la segunda forma es mediante la creación de un enlace público para poder compartir mediante Internet.

Para acceder a su información y poder compartirla el usuario debe tener conexión a Internet, la dirección IP pública del servidor, y las credenciales de acceso al servidor, las cuales son creadas por el administrador del servicio. Una vez obtenidos, el usuario puede iniciar sesión para subir y compartir sus archivos.

3.1 Requerimientos del servidor.

Para el servidor de Owncloud de almacenamiento y compartición de archivos se consideran los siguientes requerimientos para su ejecución y funcionamiento:

- **Iniciar sesión**

- 1) El usuario proporciona sus credenciales de inicio de sesión.
- 2) Owncloud consulta en su base de datos si la información sea correcta y válida.

3) Owncloud permite el ingreso del usuario.

- **Almacenar información**

1) El usuario consulta en las carpetas la información almacenada.

2) Owncloud muestra de manera visual el contenido almacenado.

- **Agregar información/archivos**

1) El usuario agrega información en las carpetas.

2) Owncloud permite que se agregue la información pertinente en función del espacio de almacenamiento disponible.

- **Modificar información**

1) El usuario intenta realizar modificaciones a las carpetas o archivos almacenados.

2) Owncloud valida que el usuario tenga los permisos correspondientes.

3) Owncloud permite la modificación de las carpetas y guarda los cambios del usuario autorizado.

- **Compartir información**

1) El usuario comparte archivos o carpetas a otro usuario.

2) Owncloud permite compartir archivos a otro usuario.

- **Eliminar información**

1) El usuario intenta eliminar carpetas o archivos seleccionados.

2) Owncloud valida que el usuario tenga los permisos correspondientes.

3) Owncloud permite la eliminación de los archivos y carpetas seleccionados.

3.2 Selección del *hardware* y *software* según los requerimientos identificados.

Hardware

Para el proyecto se escogió el Hardware con las siguientes características:

- Laptop: Dell inspiron 15 3000
- Memoria RAM :12GB

- Sistema operativo: Windows 10 PRO
- Procesador: Core Intel i7 de 11 va generación
- Núcleos: 8
- Almacenamiento: 458 GB

Con respecto al espacio de almacenamiento, la documentación de Owncloud recomienda un mínimo de 100 GB para grupos o empresas pequeñas [34].

Software

De acuerdo al manual de Owncloud, para que este servidor sea estable y funcional se debe tener en cuenta los siguientes requerimientos [35]:

- Sistema Operativo (64bits): Ubuntu server 20.04 o 22.04
- Memoria RAM: Mínimo 512 MB
- Hipervisor: *VirtualBox, Hyper-VMWare ESX o Xen, KVM*
- Servidor web: Apache 2.4
- PHP: 7.4
- Base de datos: MySQL

Teniendo en cuenta los requerimientos para la virtualización del servidor se procedió a usar la herramienta de VirtualBox, el cual es un hipervisor de tipo 2 debido a que se ejecuta sobre un sistema operativo host y es adecuado para entornos de desarrollo y pruebas. En este se instaló Ubuntu Server con las siguientes especificaciones:

- Sistema operativo(64bits): Ubuntu server 22.04.02
- Número de procesadores: 2
- Memoria RAM: 6680MB
- Almacenamiento: 127GB
- Conexión de red: Se emplea el adaptador puente para efectuar actualizaciones e instalaciones de paquetes necesarios desde una fuente externa.

3.3 Implementación del servidor de almacenamiento en un ambiente local.

Creación de máquina virtual

Una vez seleccionados los parámetros del servidor de Linux se completó la información colocando el nombre del servidor y las credenciales de usuario con su contraseña; lo cual permitirá el acceso al servidor. La Figura 3.1 presenta los parámetros de la creación del servidor de Linux.

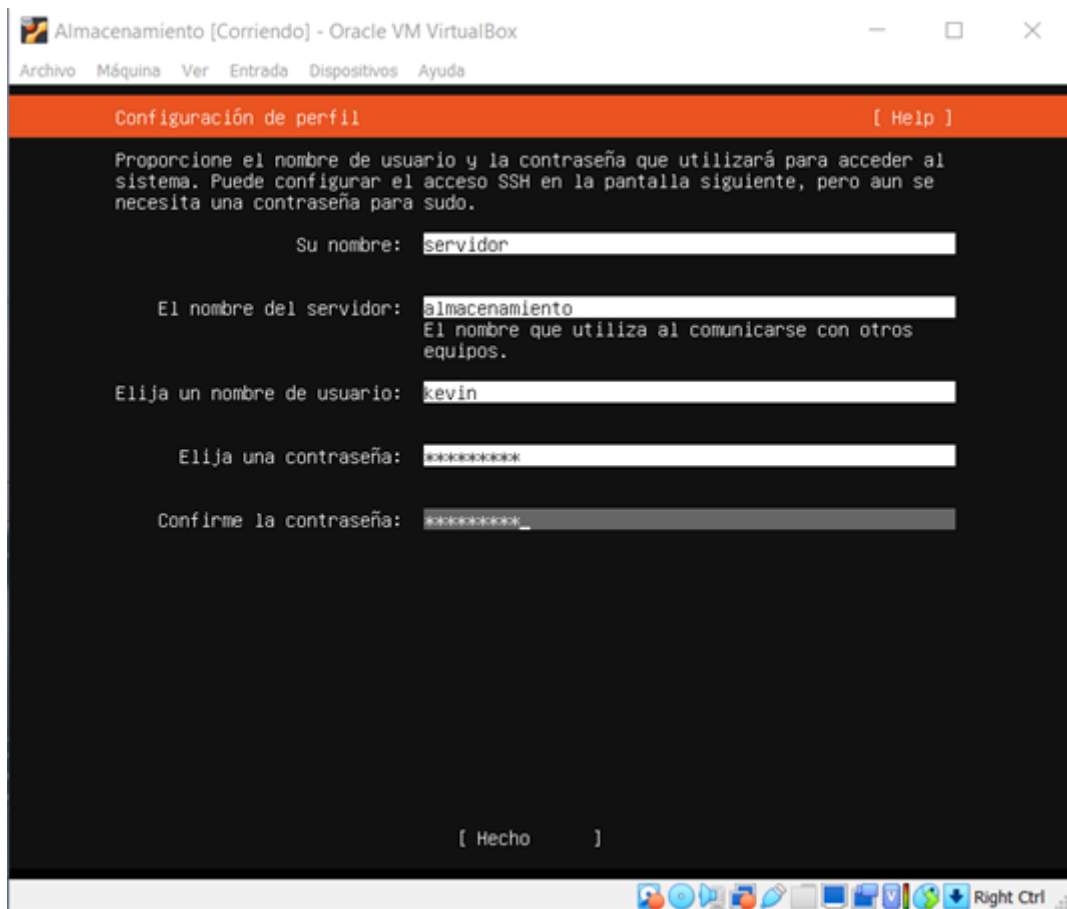


Figura 3.1 Creación del servidor

Instalación de los repositorios de PHP

Tras finalizar el proceso anterior, se actualizó el servidor mediante los comandos "apt update" y "apt upgrade" para instalar las dependencias necesarias y así poder instalar versiones anteriores de PHP (7.0, 7.1, 7.2, 7.3, 7.4) usando el comando "apt-get install software-properties-common gnupg2 -y".

Ala finalizar la instalación de las dependencias, se procedió a agregar el repositorio de PHP utilizando el comando "add-apt-repository ppa:ondrej/php". Después de realizar

esta acción, se actualizó nuevamente el repositorio. La inclusión del repositorio de PHP puede ser examinada en la Figura 3.2.

```
root@almacenamiento:/# apt-get install software-properties-common gnupg2 -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
software-properties-common ya está en su versión más reciente (0.99.22.6).
gnupg2 ya está en su versión más reciente (2.2.27-3ubuntu2.1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios
.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@almacenamiento:/#
```

Figura 3.2 Instalación del repositorio de PHP

Instalación de los paquetes requeridos e iniciación.

A continuación, se instalaron las herramientas que se utilizarán para el levantamiento del servidor de Owncloud. Para ello, se utilizó el comando "apt install bind9 apache2 php7.4 mysql-server mysql-client -y". Este comando permite la instalación del servicio de página web Apache2, la base de datos MySQL y el lenguaje de programación PHP. En este caso, se utilizó la versión 7.4 de PHP, ya que Owncloud aún no es compatible con la última versión hasta la fecha (versión 8.0). En la Figura 3.3 muestra la instalación de los paquetes requeridos.

```
root@almacenamiento:/# apt install bind9 apache2 php7.4 mysql-server mysql-client -y
```

Figura 3.3 Instalación de herramientas

Tras instalarse las herramientas, se iniciaron los servicios correspondientes.

Como primer paso se levantó el servicio de Apache. Para ello se utilizó el comando "systemctl start apache2". Para verificar que el servicio esté activo, se empleó el comando "systemctl status apache2". También se pudo comprobar la disponibilidad del servicio ingresando la dirección IP del servidor (en este caso, 192.168.200.25) en el navegador. La Figura 3.4 ilustra la activación del servicio de Apache.

```
root@almacenamiento:/# systemctl start apache2
root@almacenamiento:/# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-16 10:15:40 UTC; 35s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 11466 (apache2)
     Tasks: 6 (limit: 7615)
  Memory: 10.2M
     CPU: 118ms
   CGroup: /system.slice/apache2.service
           └─11466 /usr/sbin/apache2 -k start
             └─11468 /usr/sbin/apache2 -k start
               └─11469 /usr/sbin/apache2 -k start
                 └─11470 /usr/sbin/apache2 -k start
                   └─11471 /usr/sbin/apache2 -k start
                     └─11472 /usr/sbin/apache2 -k start

may 16 10:15:40 almacenamiento systemd[1]: Starting The Apache HTTP Server...
may 16 10:15:40 almacenamiento apachectl[11465]: AH00558: apache2: Could not reliably dete
may 16 10:15:40 almacenamiento systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Figura 3.4 Activación de Apache

Como segundo paso se levantó el servicio de MySQL usando el comando "systemctl start mysql". Luego, se verificó el estado de la activación del servicio con el comando "systemctl status mysql". En la Figura 3.5 se visualiza la activación del servicio de MySQL.

```
root@almacenamiento:/# root@almacenamiento:/#
root@almacenamiento:/# systemctl start mysql
root@almacenamiento:/# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-16 10:15:30 UTC; 1min 27s ago
     Main PID: 10230 (mysqld)
   Status: "Server is operational"
     Tasks: 37 (limit: 7615)
  Memory: 365.1M
     CPU: 2.570s
   CGroup: /system.slice/mysql.service
           └─10230 /usr/sbin/mysqld

may 16 10:15:28 almacenamiento systemd[1]: Starting MySQL Community Server...
may 16 10:15:30 almacenamiento systemd[1]: Started MySQL Community Server.
root@almacenamiento:/#
```

Figura 3.5 Activación del servicio de MySQL

Creación de la base de datos

Para generar la base de datos, se empleó el comando "mysql -u root -p" para abrir una sesión en la consola de MySQL. Una vez en la consola, se creó la base de datos con el comando "CREATE DATABASE owncloud;". Posteriormente, se estableció un nuevo usuario administrador de la base de datos para Owncloud utilizando el comando

"CREATE USER 'owncloudusr'@'localhost' IDENTIFIED WITH mysql_native_password BY 'tesis2023';".

A continuación, se otorgaron los privilegios correspondientes al nuevo usuario administrador con el comando "GRANT ALL PRIVILEGES ON owncloud. * TO 'owncloudusr'@'localhost';". Para asegurar que los cambios surtan efecto, se vaciaron los privilegios de usuario que se otorgan por defecto con el comando "FLUSH PRIVILEGES;". En la Figura 3.6 se presenta la creación de la base de datos.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> CREATE DATABASE owncloud;
Query OK, 1 row affected (0,01 sec)

mysql> GRANT ALL ON owncloud.* TO 'owncloudusr'@'localhost' IDENTIFIED BY 'tesis2023';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MySQL server version for the right syntax to use near 'IDENTIFIED BY 'tesis2023''
at line 1
mysql> CREATE USER 'owncloudusr'@'localhost' IDENTIFIED WITH mysql_native_password BY 'tesi
s2023';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT ALL PRIVILEGES ON owncloud.* TO 'owncloudusr'@'localhost';
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,01 sec)

mysql>
```

Figura 3.6 Creación de la base de datos

Instalación del servidor de Owncloud

Tras la creación de la base de datos, se empleó el comando "echo" en la terminal para añadir la URL pertinente al repositorio de Owncloud. El comando utilizado se encuentra detallado en la Figura 3.7.

Este comando agrega la URL del repositorio a un archivo en la ruta "/etc/apt/sources.list.d/isv: owncloud: server:10.list" para que pueda ser utilizado por el sistema de gestión de paquetes.

```
root@almacenamiento:/# echo 'deb http://download.opensuse.org/repositories/isv:/ownCloud:/s
erver:/10/Ubuntu_21.10/ /' | sudo tee /etc/apt/sources.list.d/isv:ownCloud:server:10.list
```

Figura 3.7 Repositorio de Owncloud

Luego, se verificó el estado de los repositorios mediante el comando "curl" para descargar la clave del repositorio de Owncloud.

Este comando descarga la clave del repositorio, la descripta con "gpg" y la guarda en el archivo "/etc/apt/trusted.gpg.d/isv_ownCloud_server_10.gpg". Estos pasos permiten

agregar y validar el repositorio de Owncloud en el sistema. La descarga del repositorio de Owncloud se muestra en la Figura 3.8.

```
root@almacenamiento:/# curl -fsSL https://download.opensuse.org/repositories/isv:ownCloud:server:10/Ubuntu_21.10/Release.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/isv_ownCloud_server_10.gpg > /dev/null
root@almacenamiento:/# apt update
Obj:1 http://ec.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://ec.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu jammy-security InRelease
Obj:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu jammy InRelease
Des:6 http://download.opensuse.org/repositories/isv:/ownCloud:/server:/10/Ubuntu_21.10 InRelease [1.539 B]
Des:7 http://download.opensuse.org/repositories/isv:/ownCloud:/server:/10/Ubuntu_21.10 Packages [889 B]
Descargados 2.428 B en 10s (254 B/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 2 paquetes. Ejecute «apt list --upgradable» para verlos.
root@almacenamiento:/#
```

Figura 3.8 Descarga de los repositorios de Owncloud

Una vez finalizada la descarga de los repositorios, se llevó a cabo la actualización del sistema para que los cambios se apliquen. Esto se realizó utilizando el comando "apt update". Luego, para instalar los complementos de Owncloud, se utilizó el siguiente comando "apt install owncloud-complete-files". Este comando instala los archivos completos de Owncloud junto con los complementos necesarios para el funcionamiento adecuado del servicio. En la Figura 3.9 se contempla la instalación de los complementos de Owncloud.

```
root@almacenamiento:/home/kevin# apt install owncloud-complete-files
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Paquetes sugeridos:
 owncloud-deps
Se instalarán los siguientes paquetes NUEVOS:
 owncloud-complete-files
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 65,1 MB de archivos.
Se utilizarán 242 MB de espacio de disco adicional después de esta operación.
* [Trabajando] *
```

Figura 3.9 Instalación de complementos de Owncloud

Se ingresó al directorio "/var/www" utilizando el comando "cd /var/www", debido a que dentro de este directorio se encuentran las carpetas "owncloud" y "html". Luego, se movió la carpeta "owncloud" dentro de la carpeta "html" utilizando el comando "mv owncloud/ html/owncloud", esto se hizo con el objetivo de desplegar el contenido de

Owncld en una página web. Con esto, se ha realizado el movimiento de la carpeta "owncloud" al directorio "html", manteniendo la estructura de directorios deseada (var/www/html/owncloud). El traslado de la carpeta necesaria se ilustra en la Figura 3.10.

```
root@almacenamiento:/home/kevin# root@almacenamiento:/home/kevin#
root@almacenamiento:/home/kevin# cd ../../
root@almacenamiento:/# ls
bin dev home lib32 libx32 media opt root sbin srv sys usr
boot etc lib lib64 lost+found mnt proc run snap swap.img tmp var
root@almacenamiento:/# cd /var/www
root@almacenamiento:/var/www# ls
html owncloud
root@almacenamiento:/var/www# mv owncloud/ html/owncloud
root@almacenamiento:/var/www#
```

Figura 3.10 Movimiento de la carpeta owncloud a html

Tras llevar a cabo los pasos previos, será posible acceder a la página de Owncloud ingresando la siguiente URL en el navegador: "http://192.168.200.25/owncloud". Al ingresar a la página, se observó que Owncloud está operativo y en funcionamiento. Esto se confirma en la Figura 3.11.

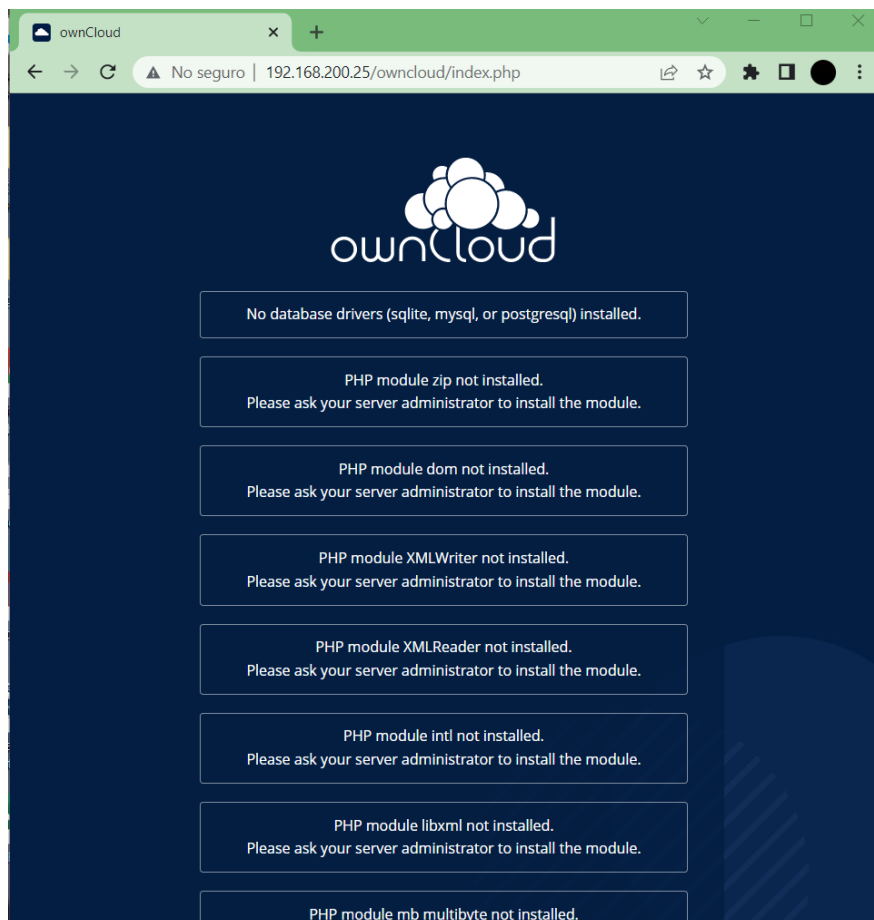


Figura 3.11 Levantamiento del servicio de Owncloud

Para instalar los servicios pendientes de PHP, se utilizó el siguiente comando “apt-get install php7.4-zip php7.4-gd php7.4-curl php7.4-mbstring php7.4-mysql php7.4-pgsql php7.4-dom php7.4-xmlreader php7.4-intl php7.4-xmlwriter -y” Este comando instala los paquetes necesarios para los servicios adicionales de PHP en su versión 7.4. Se incluyen módulos como [36] [37]:

- Zip: Permite la manipulación de archivos comprimidos en formato ZIP.
- GD: Proporciona funcionalidades para la manipulación de imágenes, como redimensionamiento, recorte y aplicaciones de filtros.
- Curl: Permite realizar solicitudes HTTP y transferencias de datos utilizando diferentes protocolos, como HTTP, FTP y SMTP.
- Mbstring: Ofrece capacidades para el tratamiento de cadenas de caracteres *multibyte*, como la manipulación de caracteres no ASCII.
- MySQL: Permite la conexión y manipulación de bases de datos MySQL utilizando funciones específicas de PHP.
- Pgsq: Proporciona soporte para la conexión y manipulación de bases de datos PostgreSQL utilizando funciones específicas de PHP.
- Dom: Permite la manipulación de documentos XML utilizando el estándar DOM (*Document Object Model*).
- Xmlreader: Proporciona funcionalidades para leer y analizar documentos XML de manera eficiente.
- Intl: Proporciona funciones relacionadas con la internacionalización y localización, como la manipulación de caracteres y formatos de fecha y hora en diferentes idiomas.
- Xmlwriter: Permite la generación y escritura de documentos XML de manera estructurada.

Una vez instalados los servicios de PHP pendientes, es crucial reiniciar el servicio de Apache para que las modificaciones tengan efecto. Tras actualizar la página de Owncloud, apareció una nueva pantalla que requiere la creación de una cuenta de administrador y las credenciales de acceso a la base de datos previamente generada. La configuración inicial de Owncloud se encuentra representada en la Figura 3.12.

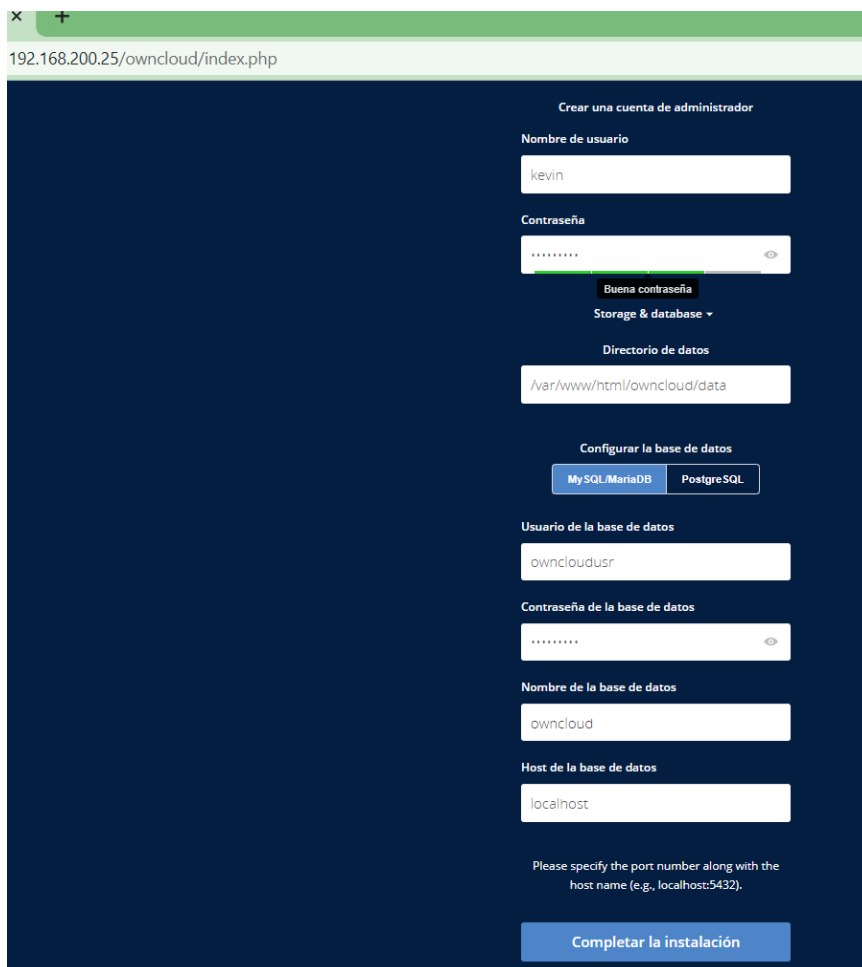


Figura 3.12 Configuración inicial de Owncloud

Una vez completada la instalación y configuración inicial de Owncloud, se procedió a iniciar sesión con la cuenta de administrador para acceder a la página principal. En la Figura 3.13 se presenta la ventana de acceso al servidor, en la que se deben ingresar las credenciales de administración. En la Figura 3.14 se exhibe la página principal de Owncloud.



Figura 3.13 Inicio de sesión

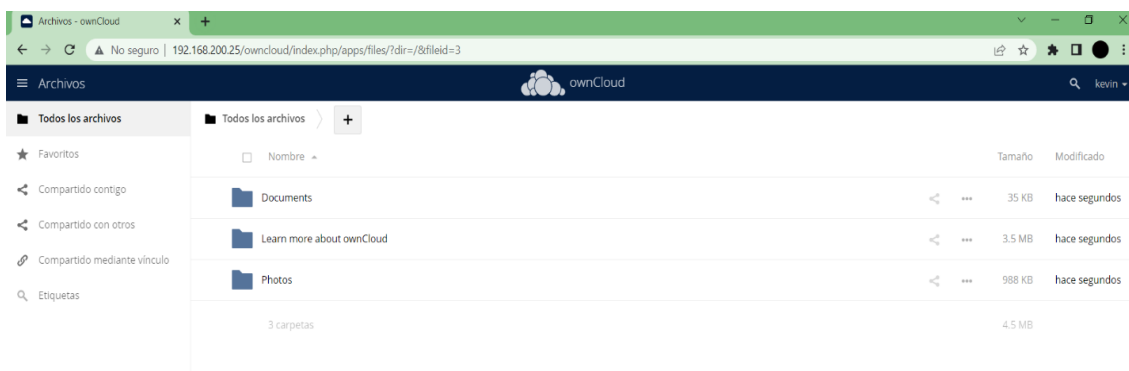


Figura 3.14 Página principal de Owncloud

Configuración de IP estática

Con el objetivo de evitar cambios en la dirección IP del servidor debido al uso de DHCP, se configuró una dirección IP estática. Para llevarlo a cabo, se accedió al directorio de configuración de redes mediante el comando `cd /etc/netplan`, y luego se editó el archivo de configuración de redes (`00-installer-config.yaml`) utilizando un editor de texto. Una vez dentro del archivo, se estableció a `dhcp4` como `false` y se ingresó la dirección IP estática, la puerta de enlace (*gateway*) y las direcciones de los servidores DNS. Para aplicar los cambios, se utilizó el comando `"netplan apply"`. La configuración de la dirección IP estática se aprecia en la Figura 3.15.

```
GNU nano 6.2                                00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.200.25/24]
      gateway4: 192.168.200.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Figura 3.15 Configuración de IP estática

Seguridad

Con el propósito de reforzar la seguridad de Owncloud, se implementó HTTPS, ya que en la configuración actual el servidor opera exclusivamente con HTTP (*HyperText Transfer Protocol*). Esto significa que la comunicación entre el cliente y el servidor no está encriptada. Para abordar esta situación, se llevó a cabo la instalación de OpenSSL. La Figura 3.16 presenta el proceso de instalación de OpenSSL.

```

root@almacenamiento:/home/kevin# apt install openssl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
openssl ya está en su versión más reciente (3.0.2-0ubuntu1.9).
Fijado openssl como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios

 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
root@almacenamiento:/home/kevin#

```

Figura 3.16 Instalación de OpenSSL

También es importante activar el módulo SSL de Apache. Para hacerlo, se debe ejecutar el comando "a2enmod ssl" en el terminal el cual se muestra en la Figura 3.17.

```

root@almacenamiento:/home/kevin# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@almacenamiento:/home/kevin# systemctl restart apache2
root@almacenamiento:/home/kevin#

```

Figura 3.17 Activación del módulo ssl

Es requerido modificar el archivo de configuración de OpenSSL para extender la validez del certificado a ser generado y alterar la ubicación de almacenamiento del mismo. Para ello, se empleó el comando "nano /etc/ssl/openssl.cnf". Una vez dentro del archivo, se busca la presencia de las siguientes líneas.

Tabla 3.1 Configuración de OpenSSL

Líneas a cambiar	Cambios realizados
dir = ./dmoCA	dir = /root/sslcert
default_days = 365	default_days = 3650

Estas líneas se alteraron siguiendo las indicaciones presentes en la Tabla 3.1. Una vez implementadas las modificaciones, se procedió a guardar y cerrar el editor de texto. En función de la practicidad, el certificado recién creado se almacenará en el directorio /root/sslcert y poseerá una validez de 10 años.

Se generó la carpeta destinada a contener el certificado mediante el comando "mkdir /root/sslcert". Posteriormente, se asignaron permisos de lectura, escritura y ejecución al

usuario root para esta carpeta, ya que es el único propietario. Para realizar esto, se utilizó el comando "chmod 700 /root/sslcert".

Posteriormente, se ingresó a la carpeta "sslcert" utilizando el comando "cd /root/sslcert". Dentro de esta carpeta, se generaron varias subcarpetas esenciales (certs, private y newcerts) para que OpenSSL pudiera crear nuevas claves y certificados. Esto se llevó a cabo mediante el comando "mkdir certs private newcerts".

Una vez creadas estas carpetas, se crearon los archivos "index.txt" y "serial". Se asignó el valor "1000" al archivo "serial" utilizando el comando "echo 1000 > serial", ya que "1000" sería el número de serie del certificado. Además, se generó el archivo "index.txt" con el comando "touch index.txt". Todos estos pasos se encuentran detallados en la Figura 3.18.

```
root@almacenamiento:/home/kevin# root@almacenamiento:/home/kevin#
root@almacenamiento:/home/kevin# mkdir /root/sslcert
root@almacenamiento:/home/kevin# chmod 700 /root/sslcert
root@almacenamiento:/home/kevin# cd /root/sslcert
root@almacenamiento:~/sslcert# mkdir certs private newcerts
root@almacenamiento:~/sslcert# echo 1000 > serial
root@almacenamiento:~/sslcert# touch index.txt
root@almacenamiento:~/sslcert#
```

Figura 3.18 Creación de carpetas y archivos para generar el certificado ssl

En este caso, al no tener una entidad que proporcione un certificado, se creó una autoridad propia que generará el certificado. Para ello en la terminal se colocó el siguiente comando `openssl req -new -x509 -days 3650 -extensions v3_ca \-keyout private/cakey.pem -out cacert.pem \-config /etc/ssl/openssl.cnf`. Al ejecutar el comando se colocó una clave la cual se debe recordar ya que se la usará más adelante y se completan los diferentes apartados que pide es importante el campo de *common name*. La Figura 3.19 se muestra la creación de la Autoridad de Certificación.

Por último, se llevó a cabo la generación del certificado autofirmado, siendo exclusivo. Para esta tarea, se introdujo el siguiente comando en la terminal: "openssl ca \-config /etc/ssl/openssl.cnf \-out apache-cert.pem \-infiles apache-req.pem". Al ejecutar este comando, se solicitó la contraseña definida durante la creación de la Autoridad de Certificación.

Luego de ingresar la contraseña y confirmar con la tecla *enter*, se consultará si se desea firmar el certificado y añadirlo a la base de datos. En ambos casos, se optó por seleccionar "sí". Una vez culminada esta secuencia, se presentó el certificado generado junto con la fecha de emisión, la fecha de vencimiento y los datos relativos al certificado. Esto es evidente mediante la Figura 3.21.

```
root@almacenamiento:~/sslcert# openssl ca \-config /etc/ssl/openssl.cnf \-out apache-cert.pem \-infiles apache-req.pem
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for /root/sslcert/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 4096 (0x1000)
  Validity
    Not Before: May 17 10:02:40 2023 GMT
    Not After : May 14 10:02:40 2033 GMT
  Subject:
    countryName           = EC
    stateOrProvinceName   = PICHINCHA
    organizationName      = EPN
    organizationalUnitName = EPN
    commonName            = geekland.sytes.net
    emailAddress          = kevincabr.14@gmail.com
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Subject Key Identifier:
      76:FC:99:5C:0E:25:33:BE:85:4A:10:5F:3A:0A:30:7A:02:05:00:69
    X509v3 Authority Key Identifier:
      48:DC:51:18:7F:38:29:88:B6:2E:ED:09:70:02:28:CA:68:52:BC:51
Certificate is to be certified until May 14 10:02:40 2033 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@almacenamiento:~/sslcert#
```

Figura 3.21 Creación y firma del certificado SSL

La etapa siguiente implicó la generación de las carpetas designadas para alojar las copias de la clave y el certificado producidos. De esta manera, se dispusieron adecuadamente en sus respectivas ubicaciones, como se ilustra mediante la Figura 3.22.

```
root@almacenamiento:~/sslcert# mkdir /etc/ssl/crt
root@almacenamiento:~/sslcert# mkdir /etc/ssl/key
root@almacenamiento:~/sslcert# cp /root/sslcert/apache-cert.pem /etc/ssl/crt
root@almacenamiento:~/sslcert# cp /root/sslcert/private/apache-key.pem /etc/ssl/key
root@almacenamiento:~/sslcert#
```

Figura 3.22 Copia de claves y certificados en ubicaciones apropiadas

A continuación, se creó la carpeta "owncloudat", la cual almacenará la información correspondiente de los usuarios. Se le otorgaron permisos utilizando el comando "chmod -755 /home/owncloudat -R" para asegurar que los archivos y subdirectorios dentro de ella tengan los permisos adecuados.

Además, se agregó la carpeta "owncloudat" a la base de datos ejecutando el comando "chown -R www-data:www-data /home/owncloudat". Esto asegura que el servidor web tenga los permisos necesarios para acceder y modificar los archivos dentro de la carpeta. Por último, se reinició el servidor Apache para aplicar las actualizaciones realizadas como se observa mediante la Figura 3.23.

```
root@almacenamiento:/var/www/html# mkdir /home/owncloudat
root@almacenamiento:/var/www/html# chmod -755 /home/owncloudat -R
root@almacenamiento:/var/www/html# chown -R www-data:www-data /home/owncloudat
root@almacenamiento:/var/www/html# systemctl restart apache2
root@almacenamiento:/var/www/html#
```

Figura 3.23 Creación de carpeta de datos

Para permitir que Apache utilice HTTPS, se procedió a editar el archivo "default-ssl.conf" utilizando el comando "nano /etc/apache2/sites-available/default-ssl.conf". Al abrir el archivo, se realizaron los siguientes cambios:

1. Se modificó el campo "ServerAdmin" para especificar la dirección de correo electrónico del administrador del servidor.
2. Se actualizó el campo "ServerName" para indicar el nombre de dominio o la dirección IP del servidor.
3. Se agregó la configuración del directorio de datos donde se almacenarán los archivos de OwnCloud.
4. Se estableció la ubicación de los certificados y las claves mediante la configuración adecuada que se muestra mediante la Figura 3.24.

```
GNU nano 6.2 /etc/apache2/sites-available/default-ssl.conf *
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin kevincabr.14@gmail.com
    ServerName geekland.sytes.net
    DocumentRoot /var/www/html
    <Directory /var/www/html/owncloud>
      AllowOverride All
    </Directory>
    <Directory /home/owncloudat>
      AllowOverride All
    </Directory>
    SSLEngine on
    SSLCertificateFile /etc/ssl/crt/apache-cert.pem
    SSLCertificateKeyFile /etc/ssl/key/apache-key.pem
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
  </VirtualHost>
</IfModule>
```

Figura 3.24 Configuración de apache para HTTPS

Después de aplicar las modificaciones en el archivo "default-ssl.conf" y guardar los ajustes realizados, se avanzó a generar un enlace simbólico para dicho archivo utilizando el comando "ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/default-ssl.conf". Este enlace simbólico permitirá la activación del archivo de configuración. La acción de crear el enlace simbólico se representa mediante la Figura 3.25.

```
root@almacenamiento:~/sslcert# ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/default-ssl.conf
root@almacenamiento:~/sslcert# service apache2 restart
root@almacenamiento:~/sslcert#
```

Figura 3.25 Activación de https

Para verificar la activación del servicio HTTPS, se introdujo en un navegador la dirección "https://192.168.200.25/owncloud". Al tratar de acceder al servidor, emergió un mensaje de advertencia que indicaba que la conexión no era privada, dado que el certificado producido es auto-firmado y no está emitido por una autoridad confiable. La confirmación de la conexión al servidor Owncloud a través de HTTPS se constata mediante la Figura 3.26.



La conexión no es privada

Es posible que algunos atacantes intenten robar tu información de **192.168.200.25** (p. ej., contraseñas, mensajes o tarjetas de crédito). [Más información](#)

NET::ERR_CERT_AUTHORITY_INVALID



Para acceder al nivel más alto de seguridad de Chrome, [activa la protección mejorada](#).

Ocultar detalles avanzados

Volver a un sitio seguro

Este servidor no pudo probar que su dominio es **192.168.200.25**; el sistema operativo de la computadora no confía en el certificado de seguridad. Es posible que esto se deba a una configuración incorrecta o a que un atacante interceptó la conexión.

[Continuar a 192.168.200.25 \(no seguro\)](#)

Figura 3.26 Comprobación de activación de https

Para acceder al servidor, bastó con hacer clic en la opción "Continuar" que se presenta en la parte inferior junto a la dirección IP del servidor. Al realizar este paso, se desplegó la página de inicio de sesión del servidor, tal como se exhibe mediante la Figura 3.27.

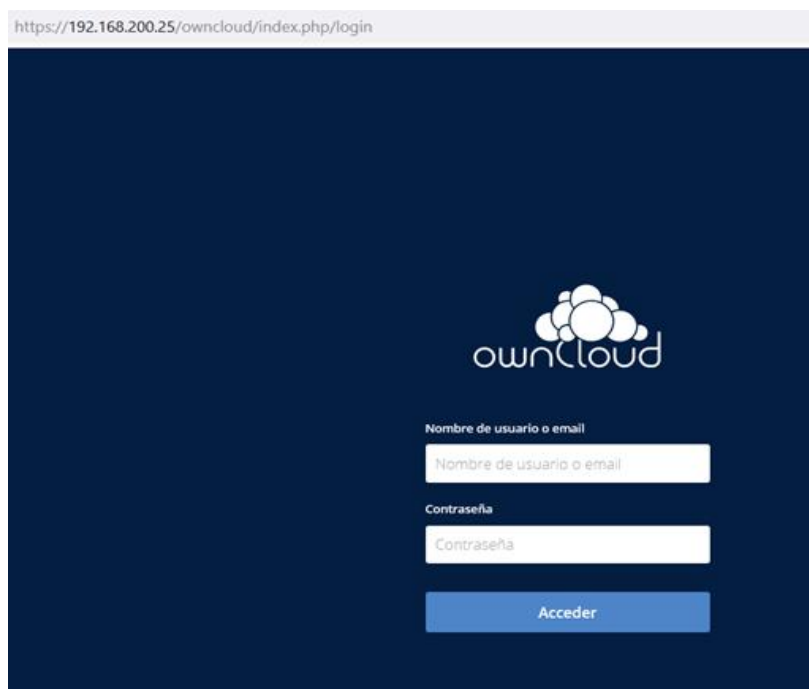


Figura 3.27 Ingreso a Owncloud usando HTTPS

Para redireccionar todas las solicitudes de HTTP a HTTPS en el servidor, se editó el archivo "000-default.conf" empleando el comando "nano /etc/apache2/sites-available/000-default.conf". Se añadió la siguiente línea: "Redirect permanent / https://192.168.0.25/". La configuración mencionada se expone mediante la Figura 3.28.

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin kevincabr.14@gmail.com
DocumentRoot /var/www/html
Redirect permanent / https://192.168.200.25/
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

Figura 3.28 Redirección permanente a https

Configuración de Memcache, Cron y HTTP Strict Transport Security

Al ingresar al panel de administración de OwnCloud con el usuario "administrador", se pudo observar en la sección "General" avisos en color rojo que indican correcciones necesarias, así como avisos en color negro que son recomendaciones. Los avisos mencionados se presentan mediante la Figura 3.29.

El primer aviso (*Memcache*) se refiere a la memoria caché, la cual puede mejorar la carga de las páginas si se gestiona correctamente. También menciona el archivo transaccional de bloqueo (*Transactional File Locking*), el cual evita la edición simultánea de un archivo por parte de múltiples usuarios y los posibles problemas que esto puede causar [38].

El *system cron* se encarga de automatizar tareas de mantenimiento y generación de informes que son esenciales para el funcionamiento eficiente y seguro de la plataforma, lo que permite que OwnCloud funcione correctamente y realice las tareas necesarias para mantener la aplicación en funcionamiento [38].

El servicio de HSTS (*HTTP Strict Transport Security*) permite prevenir que los usuarios ignoren certificados inválidos al acceder a la nube y también indica a los navegadores que las conexiones a Owncloud no deben realizarse mediante HTTP [38].

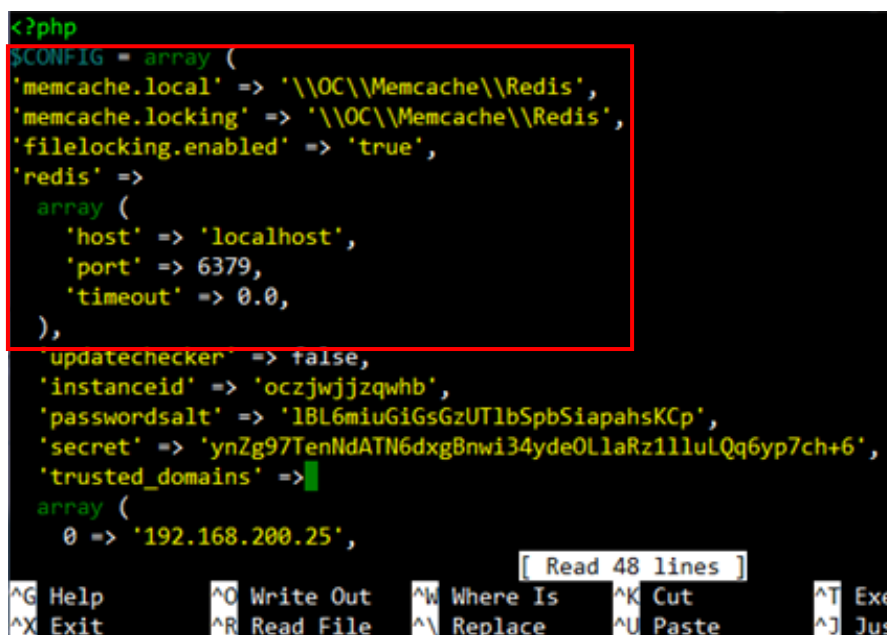
Seguridad & alertas de configuración

- Transactional file locking should be configured to use memory-based locking, not the default slow database-based locking. See the documentation ↗ for more information.
- We recommend to enable system cron as any other cron method has possible performance and reliability implications.
- The "Strict-Transport-Security" HTTP header is not configured to at least "15552000" seconds. For enhanced security we recommend enabling HSTS as described in our security tips.
- No memory cache has been configured. To enhance your performance please configure a memcache if available. Further information can be found in our documentation.

Please double check the installation guides ↗, and check for any errors or warnings in the log.

Figura 3.29 Avisos a corregirse

Para solucionar el error de *memcache*, se instalaron los paquetes necesarios con el comando "apt install redis-server php7.4-redis". Luego, se procedió a editar el archivo "config.php" utilizando el comando "nano /var/www/html/owncloud/config/config.php", abriendo así el editor de texto. En el editor, se agregaron las primeras líneas mencionadas en la Figura 3.30 relacionadas con la memoria caché. Una vez agregadas, se guardaron los cambios y se reinició el servidor.



```
<?php
$CONFIG = array (
  'memcache.local' => '\\OC\\Memcache\\Redis',
  'memcache.locking' => '\\OC\\Memcache\\Redis',
  'filelocking.enabled' => 'true',
  'redis' =>
    array (
      'host' => 'localhost',
      'port' => 6379,
      'timeout' => 0.0,
    ),
  'updatechecker' => false,
  'instanceid' => 'oczjwjzqwhb',
  'passwordsalt' => '1BL6miuGiGsGzUT1bSpbS1apahsKCp',
  'secret' => 'ynZg97TenNdATN6dxgBnwi34yde0LlaRz11luLQq6yp7ch+6',
  'trusted_domains' =>
    array (
      0 => '192.168.200.25',
    ),
);
```

Figura 3.30 Corrección de memoria caché

Para abordar la advertencia de cron, se procede a generar una nueva configuración para cron. Esto se realiza ejecutando el comando "crontab -u www-data -e" y añadiendo una línea adicional al final del archivo. Los detalles de la configuración corregida para cron se aprecian mediante la Figura 3.31.

Posteriormente, se accedió a la página de administración de Owncloud y, en la sección de Cron, se modificó la tarea de AJAX a Cron. La modificación necesaria se muestra mediante la Figura 3.32.

```
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command*
*/15 * * * * /usr/bin/php -f /var/www/html/owncloud/occ system:cron
```

Figura 3.31 Configuración de Cron

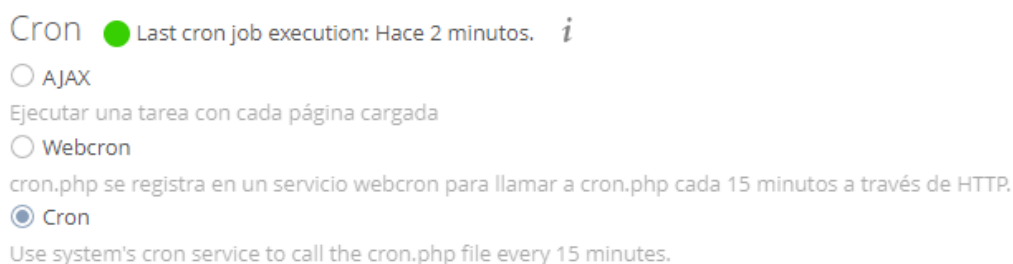


Figura 3.32 Cambio a Cron en Owncloud

Para habilitar el servicio de HSTS, es necesario activar el módulo "mod_headers" mediante el comando "a2enmod headers". A continuación, se utilizó el comando "nano /etc/apache2/sites-available/default-ssl.conf" para editar el archivo mencionado, donde se agregaron las líneas resaltadas en el cuadro rojo. Después de guardar los cambios, se reinició el servidor. La configuración para rectificar HSTS se aprecia mediante la Figura 3.33.

```
GNU nano 6.2 /etc/apache2/sites-available/default-ssl.conf *
# their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
# "force-response-1.0" for this.
# BrowserMatch "MSIE [2-6]" \
#     nokeepalive ssl-unclean-shutdown \
#     downgrade-1.0 force-response-1.0

IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=15552000; includeSubDomains"
/Ifmodule>
</Ifmodule>
```

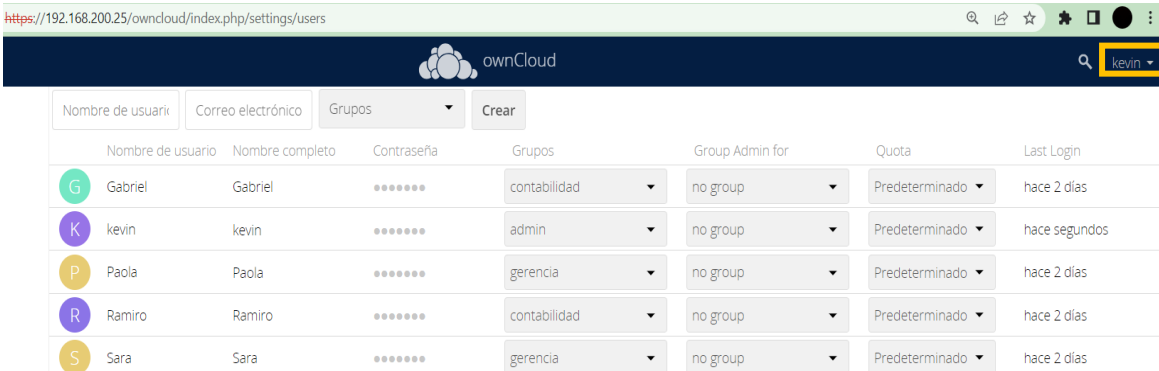
Figura 3.33 Habilitación de HSTS

3.4 Pruebas de funcionamiento del servidor en ambiente local

Creación de usuarios

Para comprobar el funcionamiento del servidor se crearon varios usuarios mediante la interfaz gráfica que proporciona Owncloud y desde la cuenta de administrador antes creada. A cada usuario se le asignó un grupo y su contraseña, además de la cantidad de almacenamiento que tendrán en el servidor la cual es la predeterminada por este (1GB). Los usuarios creados y la cantidad de almacenamiento se visualizan en la Figura 3.34.

Cabe mencionar que los archivos que se compartan en un grupo solo podrán ser visualizados por los usuarios pertenecientes a él, creando así una lista de acceso para los recursos compartidos.



The screenshot shows the 'users' management page in Owncloud. At the top, there is a search bar with 'kevin' entered. Below the search bar, there are tabs for 'Nombre de usuario', 'Correo electrónico', 'Grupos', and 'Crear'. The main content is a table with the following columns: 'Nombre de usuario', 'Nombre completo', 'Contraseña', 'Grupos', 'Group Admin for', 'Quota', and 'Last Login'. The table lists five users: Gabriel, kevin, Paola, Ramiro, and Sara. Each user has a unique icon, a group assigned (contabilidad, admin, gerencia), and a quota of 'Predeterminado'. The 'Last Login' column shows the time since the user last logged in, such as 'hace 2 días' or 'hace segundos'.

Nombre de usuario	Nombre completo	Contraseña	Grupos	Group Admin for	Quota	Last Login
G Gabriel	Gabriel	contabilidad	no group	Predeterminado	hace 2 días
K kevin	kevin	admin	no group	Predeterminado	hace segundos
P Paola	Paola	gerencia	no group	Predeterminado	hace 2 días
R Ramiro	Ramiro	contabilidad	no group	Predeterminado	hace 2 días
S Sara	Sara	gerencia	no group	Predeterminado	hace 2 días

Figura 3.34 Creación de usuarios

Carga de archivos y creación de carpetas

Una vez creados los usuarios se procedió a acceder a uno de ellos, en la página principal del usuario se mostraron los archivos por defecto que proporciona Owncloud, tales como la guía de Owncloud y carpetas de pruebas.

Para que el usuario pueda subir un archivo, crear una carpeta o un archivo de texto, se hizo clic en el ícono (+), al hacerlo se desplegaron las opciones mencionadas y al finalizar la carga del archivo o carpetas se almacenaron en el servidor. El proceso de cargar archivos al servidor de Owncloud es visible mediante la Figura 3.35.

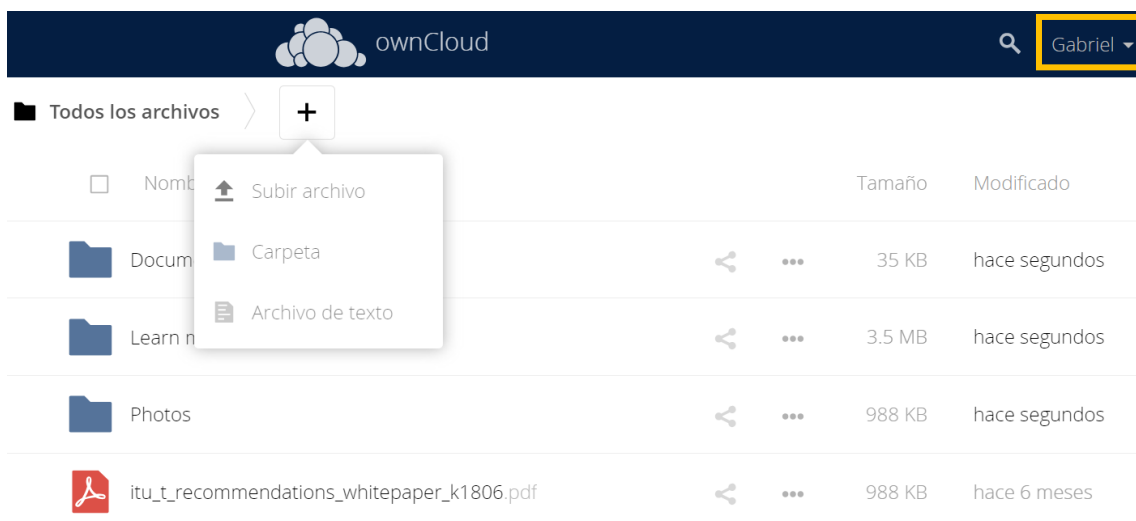


Figura 3.35 Carga de archivos al servidor

Compartición de archivos

Para compartir un archivo que el usuario haya almacenado en el servidor, se hizo clic en el símbolo de compartir el cual apareció al pasar el cursor sobre el archivo o carpeta que quiera compartir, así se desplegó el apartado de compartir, en este se colocó el nombre de la persona o grupo al que quiera compartir, además de otorgar permisos de lectura, edición o eliminación. La Figura 3.36 exhibe el proceso de compartición de archivos.

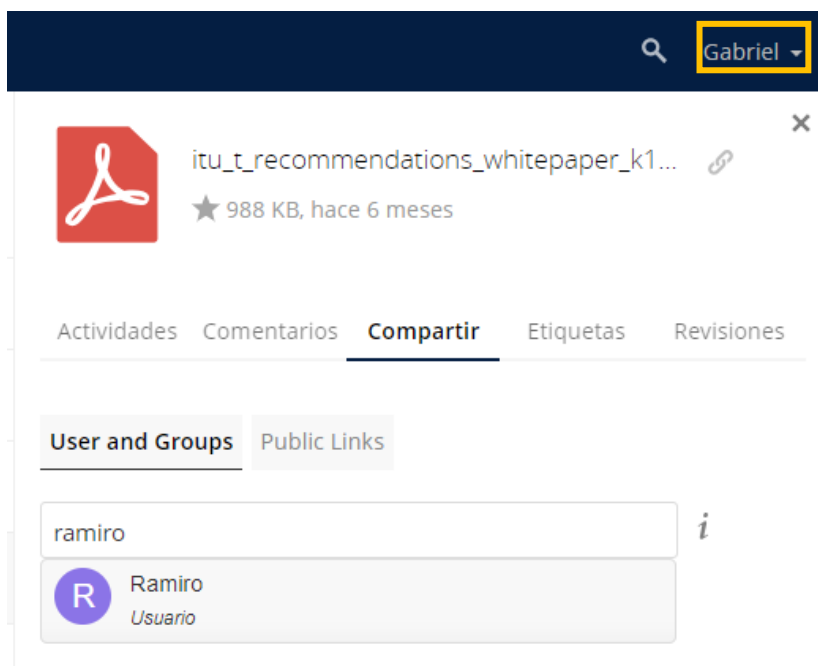


Figura 3.36 Compartición de archivos

Para comprobar que el archivo se compartió correctamente se ingresó a la cuenta del usuario destinatario. Tal como se ilustra en la Figura 3.37, el archivo ha sido compartido y además exhibe el nombre del remitente. Este proceso confirma la sincronización de la información.

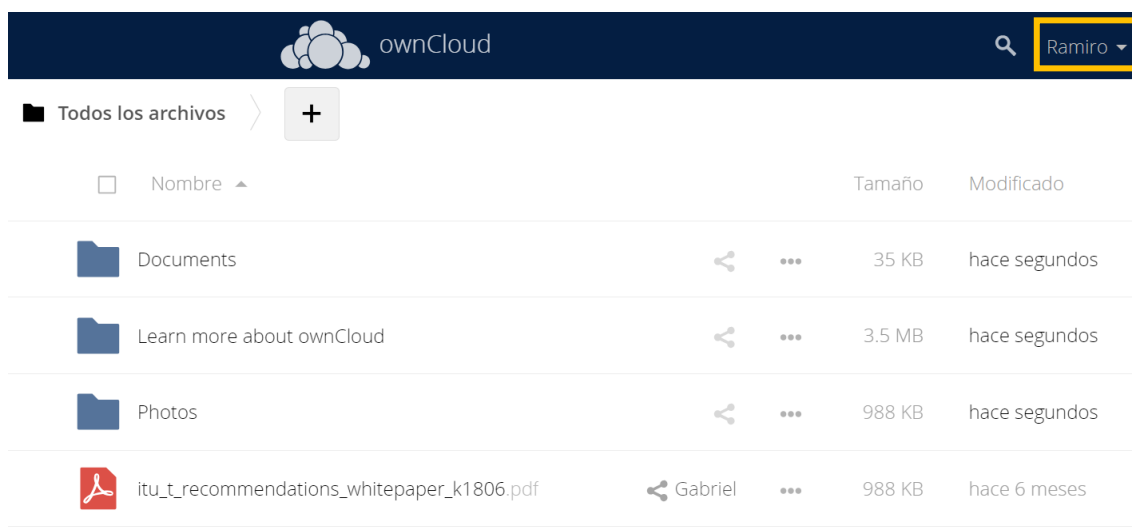


Figura 3.37 Comprobación del archivo compartido

Compartición de carpetas

Para compartir una carpeta se siguieron los pasos ya antes mencionados, es importante recalcar que los archivos que se suban a dicha carpeta después de haberla compartido automáticamente se compartirán con los usuarios o grupos añadidos. La compartición de carpetas se verifica en la Figura 3.38, mientras que en la Figura 3.39 se confirma la compartición de archivos que carguen en la carpeta.

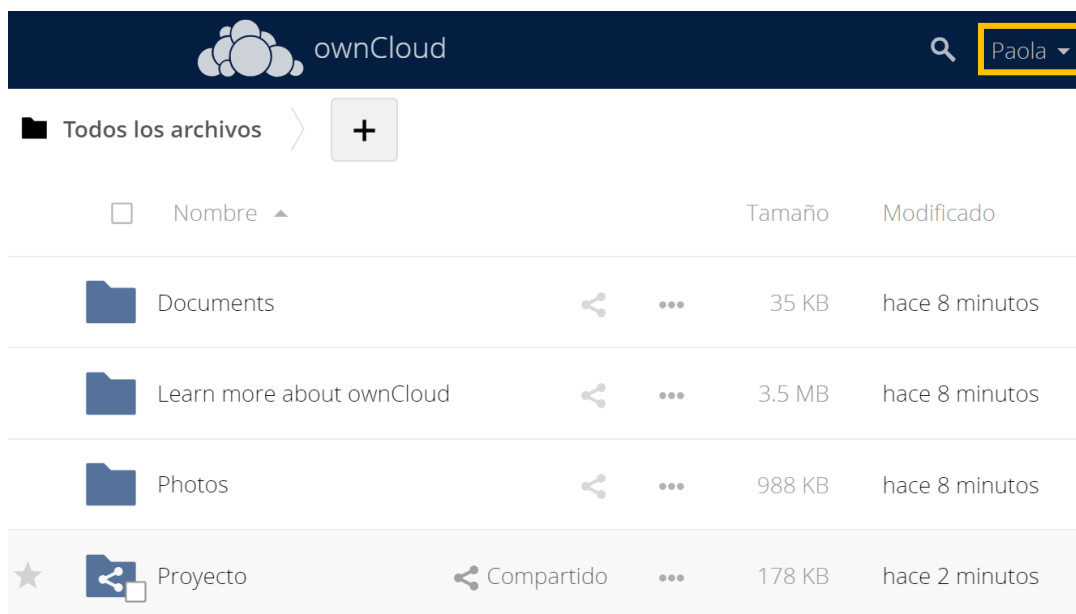


Figura 3.38 Compartición de carpetas

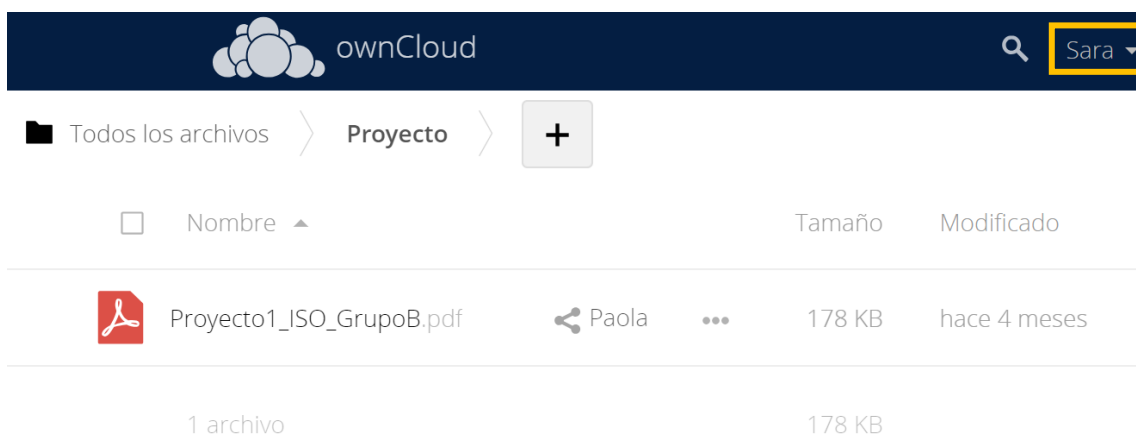


Figura 3.39 Archivos compartidos dentro la carpeta

Compartición mediante enlace

Otra forma de compartir archivos a usuarios conectados a la misma red del servidor fue mediante la creación de un enlace público. Para ello se hizo clic en esta opción, al hacerlo se desplegó una ventana en la que se escogió si el destinatario puede ver y descargar el archivo o descargar, ver y editar el archivo. La Figura 3.40 muestra proceso de compartición, y la Figura 3.41 exhibe la ventana del documento generada por el enlace.

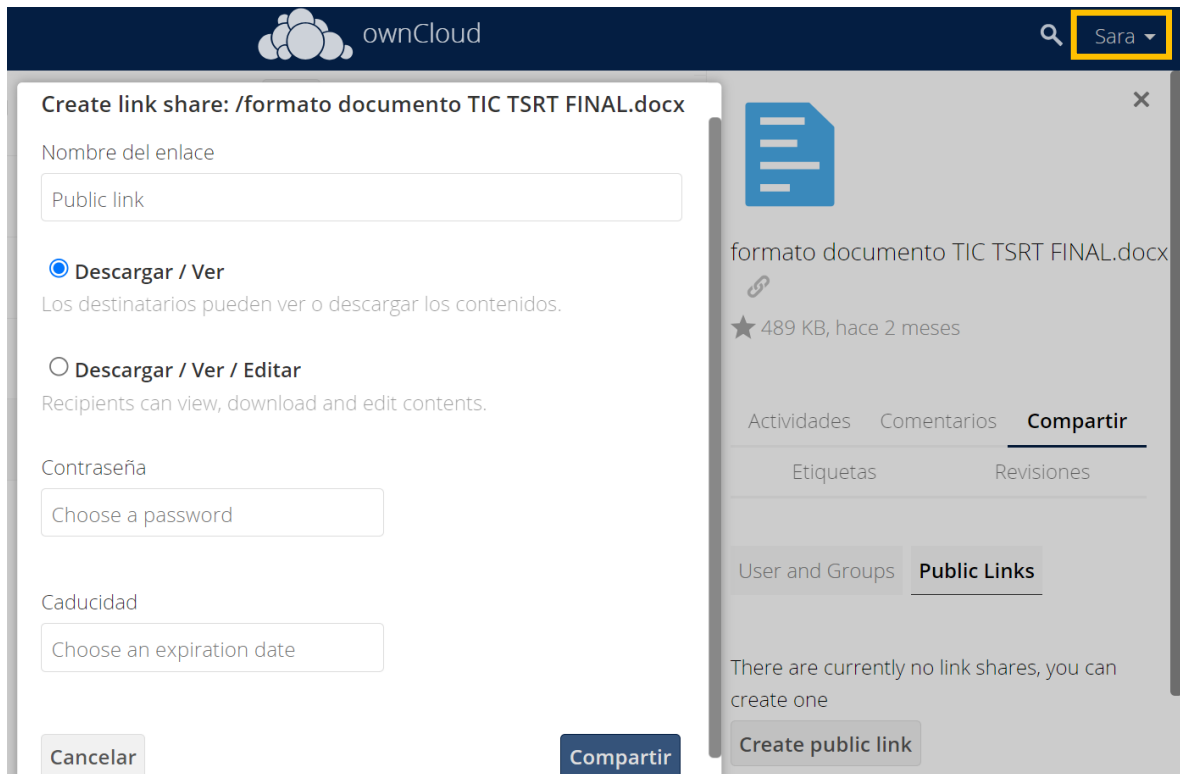
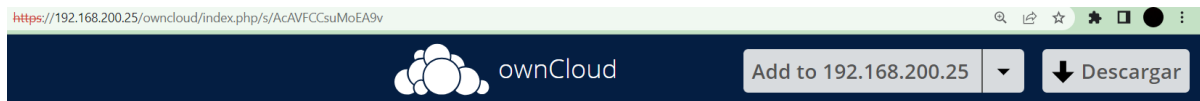


Figura 3.40 Creación del enlace público



↓ Descargar formato documento TIC TSRT FINAL.docx (489 KB)

Enlace directo

ownCloud – Un lugar seguro para todos tus datos

Figura 3.41 Acceso al archivo mediante enlace

3.5 Implementación del servidor en una plataforma de hosting.

En la plataforma de Azure, se procedió a crear la máquina virtual, siguiendo los parámetros predefinidos para el servidor Linux y la implementación del servicio Owncloud. También se establecieron las credenciales de acceso al servidor. La Figura 3.42 presenta la configuración de la MV, mientras que la Figura 3.43 verifica la creación exitosa de la MV.

Crear una máquina virtual ...

Suscripción *	<input type="text" value="Azure for Students"/>
Grupo de recursos *	<input type="text" value="(Nuevo) Owncloud"/> Crear nuevo
Detalles de instancia	
Nombre de máquina virtual *	<input type="text" value="Owncloud"/>
Región *	<input type="text" value="(US) South Central US"/>
Opciones de disponibilidad	<input type="text" value="No se requiere redundancia de la infraestructura"/>
Tipo de seguridad	<input type="text" value="Máquinas virtuales de inicio seguro"/> Configurar características de seguridad
Imagen *	<input type="text" value="Ubuntu Server 22.04 LTS - x64 gen. 2"/> Ver todas las imágenes Configurar la generación de máquinas virtuales
Arquitectura de VM	<input type="radio"/> Arm64 <input checked="" type="radio"/> x64
Ejecución de Azure Spot con descuento	<input type="checkbox"/>
Tamaño *	<input type="text" value="Standard_D2s_v3 - 2 vcpu, 8 GiB de memoria (80,30 US\$/mes)"/> Ver todos los tamaños
Cuenta de administrador	
Tipo de autenticación	<input type="radio"/> Clave pública SSH <input checked="" type="radio"/> Contraseña
Nombre de usuario *	<input type="text" value="kevin"/>
Contraseña *	<input type="password" value="....."/>
Confirmar contraseña *	<input type="password" value="....."/>

Figura 3.42 Configuración de la MV

Información esencial	
Grupo de recursos (mover)	: Owncloud
Estado	: En ejecución
Ubicación	: South Central US
Suscripción (mover)	: Azure for Students
Id. de suscripción	: f2c7536e-40cb-4d70-b123-a4d5f571d7bf
Etiquetas (editar)	: Haga clic aquí para agregar etiquetas.
Sistema operativo	: Linux
Tamaño	: Standard D2s v3 (2 vcpu, 8 GiB de memoria)
Dirección IP pública	: 20.225.140.165
Red virtual/subred	: Owncloud-vnet/default
Nombre DNS	: Sin configurar
Estado de mantenimiento	: -

Figura 3.43 Máquina virtual creada

Una vez que la máquina virtual fue creada en Azure, se inició con la instalación de Owncloud siguiendo los pasos descritos anteriormente en la instalación local (Sección 3.3). Posteriormente, se accedió a la página de Owncloud utilizando la dirección IP pública suministrada por Azure (20.225.140.165) y se procedió a crear el usuario administrador, así como a ingresar las credenciales respectivas a la base de datos. La Figura 3.44 ilustra los resultados obtenidos.

Figura 3.44 Creación de usuario administrador

De la misma manera que en la instalación local (Sección Seguridad), se aplicó HTTPS al servidor de Owncloud y se realizó la configuración necesaria para que al ingresar la dirección IP en el navegador, este redirija automáticamente a HTTPS en lugar de HTTP. Esto se hizo para mejorar y complementar la seguridad del servidor. La activación de HTTPS se puede observar en la Figura 3.45.

También se resolvieron los avisos correspondientes al *Memcache*, *Cron* y *HTTP Strict Transport Security* con el fin de mejorar la seguridad y funcionalidad de Owncloud (Sección Configuración de *Memcache*, *Cron* y *HTTP Strict Transport Security*).

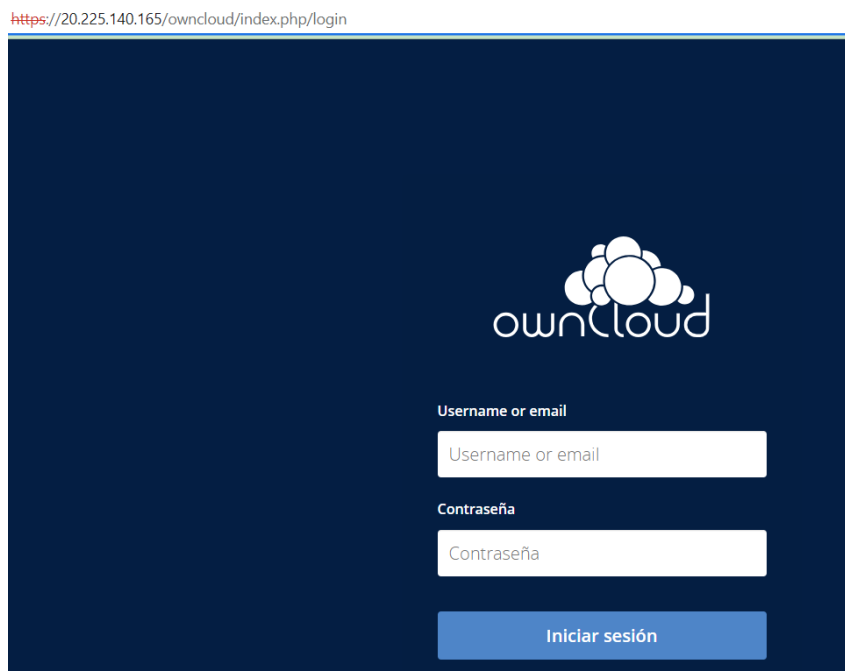


Figura 3.18 Acceso al servidor con HTTPS

3.6 Pruebas de funcionamiento del servidor en plataforma Host.

Creación de usuarios

Del mismo modo que en el entorno local, se procedió a crear usuarios con su respectivo grupo y credenciales mediante la interfaz gráfica proporcionada por Owncloud. También se asignó la cantidad de almacenamiento correspondiente a cada usuario. La Figura 3.46 exhibe los usuarios establecidos junto con su correspondiente espacio de almacenamiento.

Nombre de usuario	Correo electrónico	Grupos	Crear	Nombre de usuario	Nombre completo	Contraseña	Grupos	Group Admin for	Quota	Last Login
A	admin	admin	admin	admin	admin	*****	admin	no group	Predeterminado	hace segundos
J	Javier	Javier	contabilidad	Javier	Javier	*****	contabilidad	no group	Predeterminado	hace 2 días
J	Joel	Joel	TI	Joel	Joel	*****	TI	no group	Predeterminado	hace 22 días
M	Maria	Maria	contabilidad	Maria	Maria	*****	contabilidad	no group	Predeterminado	hace 2 días
M	Marlon	Marlon	TI	Marlon	Marlon	*****	TI	no group	Predeterminado	hace 23 días

Figura 3.46 Creación de usuarios

Carga de archivos y creación de carpetas

Una vez creados los usuarios, se inició sesión en cada uno de ellos y se procedió a cargar un archivo de la misma manera que en el servidor local. El proceso de subida del archivo es visible mediante la Figura 3.47.

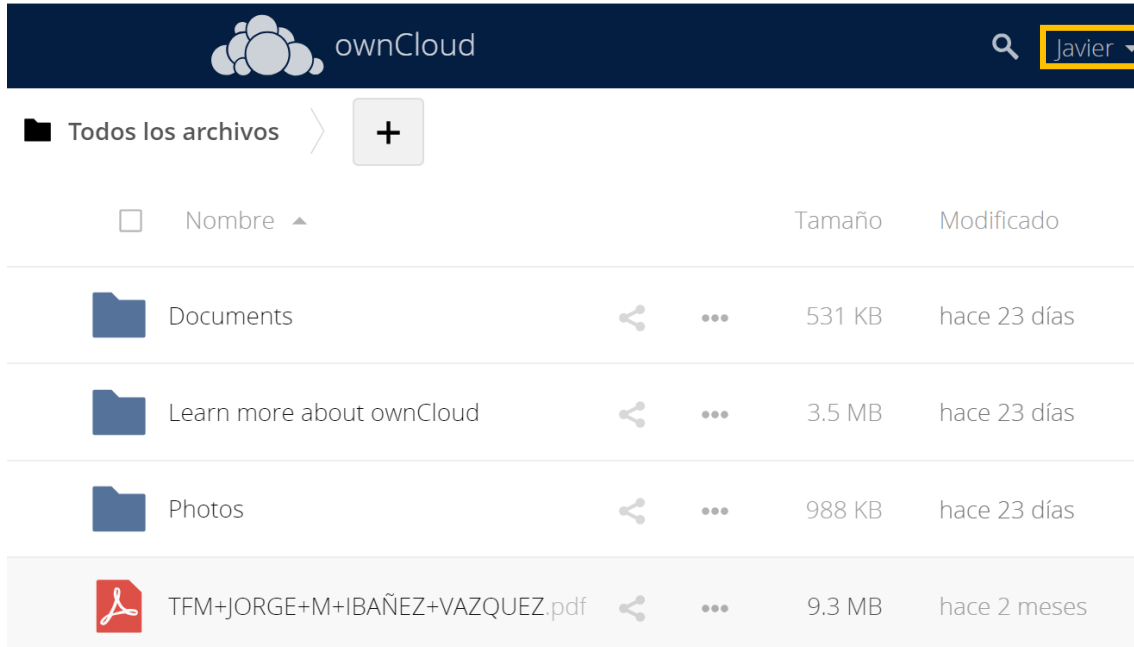


Figura 3.47 Carga de archivos

Compartición de archivos y carpetas

Al igual que en el servidor local, se realizaron las pruebas de compartición de archivos y carpetas que los usuarios hayan almacenado en el servidor. En la Figura 3.48 se puede verificar la compartición de un archivo, por otro lado, la Figura 3.49 confirma que la carpeta se ha compartido.

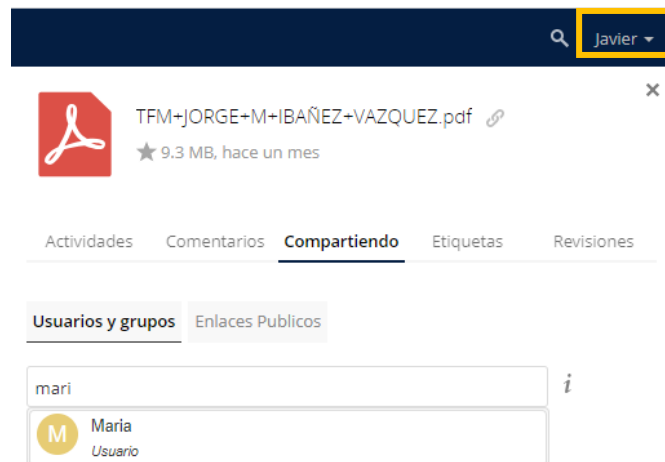


Figura 3.48 Compartición de archivo

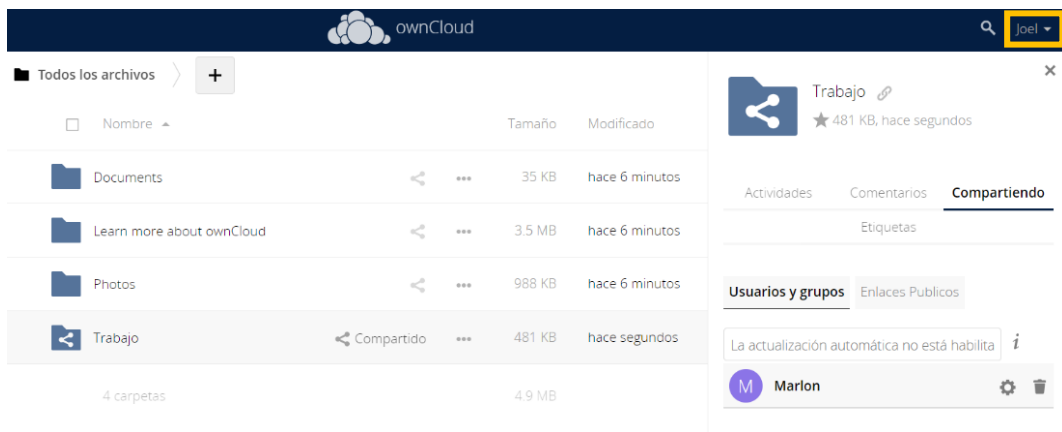


Figura 3.49 compartición de carpeta

Compartición de archivo mediante enlace

La compartición de archivos a través del enlace público difiere de la compartición en el servidor local, debido a que permite compartir archivos con personas a través de Internet. Además, esto permite comprobar el acceso remoto. Es importante destacar que se deben seguir los mismos pasos que en el servidor local para crear este enlace. La creación del enlace se ilustra en la Figura 3.50, mientras que en la Figura 3.51 se muestra el acceso al enlace generado.

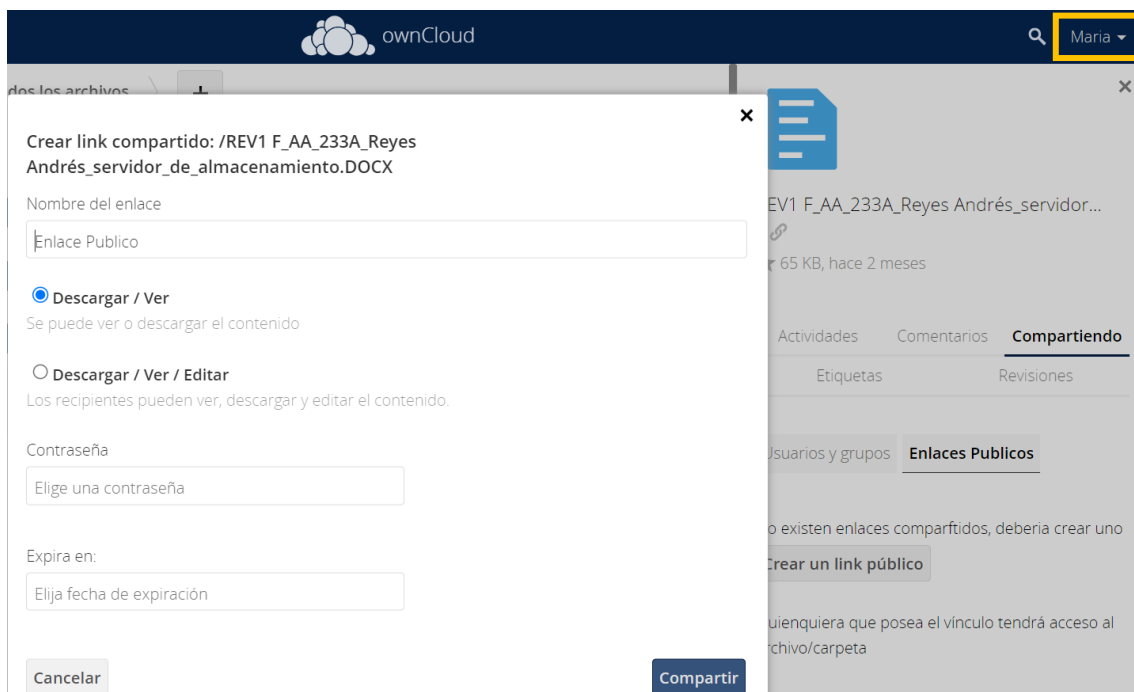


Figura 3.50 Creación del enlace público

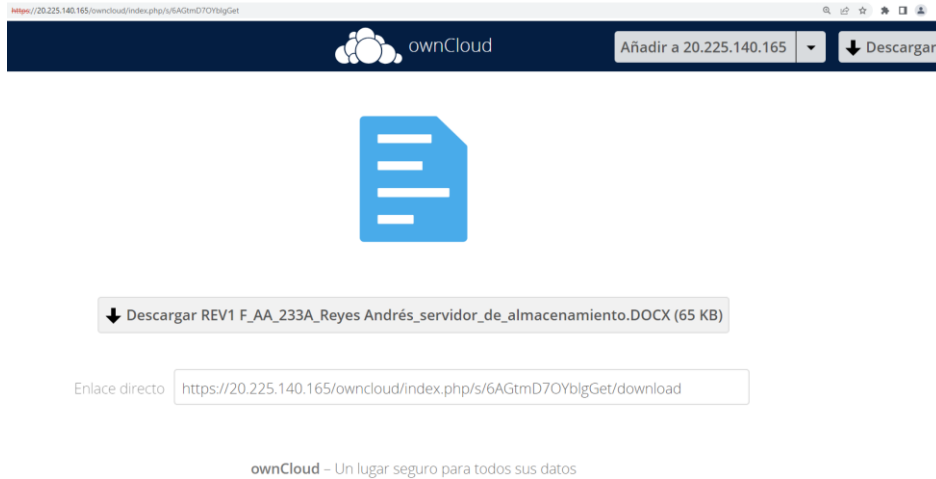


Figura 3.51 Acceso al enlace público

4 CONCLUSIONES

- Mediante la integración de tecnologías como *software* de código abierto, virtualización y la nube se puede implementar un servidor de almacenamiento y compartición de archivos en un ambiente local totalmente gratuito, pero teniendo siempre en cuenta que la información almacenada no podrá ser compartida con redes externas a la del servidor.
- La plataforma de almacenamiento (Owncloud) usada en este proyecto, permite almacenar y compartir archivos de una manera fácil e intuitiva gracias a su interfaz gráfica, haciendo así que sea fácil para las personas hacer uso de esta, por ende, esta puede ser usada en pequeñas empresas u organizaciones que quieran migrar su información a la nube.
- Haciendo uso de una plataforma host y software de código abierto, se puede implementar un servidor de almacenamiento de bajo costo, el cual a diferencia del entorno local se puede compartir la información almacenada a través de Internet.
- Mediante las pruebas de funcionamiento del servidor implementado en una plataforma host, se pudo comprobar que el acceso a este es completamente remoto, además de que cuenta con la sincronización de información haciendo posible que la información que se cargue al servidor esté siempre actualizada.
- Owncloud cuenta con la función de que cada usuario puede pertenecer a grupos y la información que sea compartido en los grupos solo pueda ser consultada

por los usuarios pertenecientes a él, creando así una lista de acceso que permite que la información sea exclusiva para cada grupo.

- Al implementar HTTPS en el servidor, se asegura la transferencia de archivos entre el cliente y el servidor web, lo que protege la información que los usuarios desean almacenar en el servidor contra posibles ataques cibernéticos.

5 RECOMENDACIONES

- Se recomienda asegurar que el servidor cumpla con los requisitos mínimos para desplegar Owncloud. Para ello se debe verificar la compatibilidad del sistema operativo que se vaya a usar, la versión de PHP y los paquetes requeridos, así como también el espacio en disco y la memoria RAM necesaria para el almacenamiento de archivos.
- La seguridad es esencial para el servidor de almacenamiento, por lo cual se debe optar por las mejores prácticas de seguridad, como mantener actualizado la versión de Owncloud y todos los componentes del sistema, usar contraseñas fuertes y habilitar el cifrado SSL. También se podría implementar un cortafuego y considerar implementar la autenticación en dos pasos (2FA).
- Para futuros proyectos se podría considerar crear un plan de respaldo que permita asegurar que la información almacenada en Owncloud estén protegidos ante posibles fallos del servidor o si se eliminan por error alguna información almacenada.
- Se debe definir una estructura concisa de como estarán formados los grupos y los usuarios para facilitar la gestión de permisos además de la compartición de archivos. Por ende, se deben otorgar los roles adecuados para cada usuario para limitar el acceso a la información privada.
- La configuración de Owncloud es esencial para obtener el mejor rendimiento posible. Para ello es necesario resolver los avisos que se presentan en la sección general, que aparece en la cuenta de administrador y tiene que ver con la memoria cache y otros parámetros según lo requiera el servidor.
- Es importante educar a los usuarios sobre las mejores prácticas para compartir archivos de manera segura. Si se usa el enlace público como método de compartición se debe fomentar el uso de contraseñas fuertes para compartir información de manera segura además de explicar los riesgos asociados.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] VMware, «Introducción a la virtualización,» 2023. [En línea]. Available: <https://www.vmware.com/es/solutions/virtualization.html>. [Último acceso: 02 Junio 2023].
- [2] AWS Amazon, «¿Qué es la virtualización?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/virtualization/>. [Último acceso: 02 Junios 2023].
- [3] RedHat, «¿Qué es y para qué sirve la virtualización?,» 25 Enero 2023. [En línea]. Available: <https://www.redhat.com/es/topics/virtualization/what-is-virtualization>. [Último acceso: 02 Junio 2023].
- [4] Lucidchart, «Principios básicos del almacenamiento en la nube,» [En línea]. Available: <https://www.lucidchart.com/blog/es/aspectos-basicos-computacion-en-la-nube>. [Último acceso: 03 Junio 2023].
- [5] Next U, «Características de la computación en la nube,» [En línea]. Available: <https://www.nextu.com/blog/computacion-en-la-nube-rc22/>. [Último acceso: 03 Junio 2023].
- [6] Atlassian, «Visión general de la nube,» [En línea]. Available: <https://www.atlassian.com/es/microservices/cloud-computing>. [Último acceso: 03 Junio 2023].
- [7] StackScale, «Principales modelos de servicio cloud: IaaS, PaaS y SaaS,» 01 Febrero 2023. [En línea]. Available: <https://www.stackscale.com/es/blog/modelos-de-servicio-cloud/#:~:text=Los%20tres%20principales%20modelos%20de,de%20control%2C%20seguridad%20y%20escalabilidad>. [Último acceso: 04 Junio 2023].
- [8] AWS Amazon, «Tipos de computación en la nube,» [En línea]. Available: <https://aws.amazon.com/es/types-of-cloud-computing/>. [Último acceso: 04 Junio 2023].
- [9] Openintl, «Tres modelos de servicio de la computación en la nube,» 24 Agosto 2018. [En línea]. Available: <https://www.openintl.com/es/tres-modelos-de-servicio-de-la-computacion-en-la-nube/>. [Último acceso: 04 Junio 2023].

- [10] Azure, «¿Qué es una máquina virtual (VM)?», [En línea]. Available: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-virtual-machine>. [Último acceso: 05 Junio 2023].
- [11] Microsoft, «Máquinas virtuales en Azure,» 08 Marzo 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/virtual-machines/overview>. [Último acceso: 05 Junio 2023].
- [12] Microsoft, «Características de las máquinas virtuales de Azure Stack Hub,» 22 Mayo 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure-stack/user/azure-stack-vm-considerations?view=azs-2301&tabs=az1%2Caz2>. [Último acceso: 05 Junio 2023].
- [13] Vegagestion, «Almacenamiento en la nube,» [En línea]. Available: <https://vegagestion.es/almacenamiento-la-nube-caracteristicas-ventajas-desventajas/>. [Último acceso: 06 Junio 2023].
- [14] Mcmtelecom, «5 características de la nube, pilares de su funcionamiento,» 21 febrero 2022. [En línea]. Available: <https://www.mcmtelecom.com/blog/transformacion-digital/5-caracteristicas-de-la-nube-pilares-de-su-funcionamiento>. [Último acceso: 06 Junio 2023].
- [15] Ntxpro, «Principales características de la Nube,» [En línea]. Available: <https://www.ntxpro.net/sistemas/caracteristicas-de-la-nube/>. [Último acceso: 06 Junio 2023].
- [16] Neolo, «¿Para qué sirve OwnCloud?,» 19 Febrero 2021. [En línea]. Available: <https://www.neolo.com/blog/que-es-owncloud-y-para-que-sirve.php>. [Último acceso: 07 Junio 2023].
- [17] IDgroup, «Principales Funciones de Owncloud,» [En línea]. Available: <https://idgrup.com/owncloud-tu-plataforma-de-comparticion-de-archivos/>. [Último acceso: 07 Junio 2023].
- [18] G. Martinez, «¿Qué es Owncloud y para qué sirve?,» Webempresa, 19 Octubre 2020. [En línea]. Available: <https://www.webempresa.com/blog/que-es-owncloud-y-para-que-sirve.html>. [Último acceso: 07 Junio 2023].

- [19] Stacksacle, «Ubuntu 22.04 LTS (Jammy Jellyfish),» 16 Junio 2022. [En línea]. Available: <https://www.stackscale.com/es/blog/ubuntu-22-04-lts/>. [Último acceso: 08 Junio 2023].
- [20] J. Izquierdo, «Ubuntu Server 22.04 LTS. Novedades y características,» Weblinus, 07 Mayo 2022. [En línea]. Available: <https://weblinus.com/ubuntu-server-22-04-lts-novedades-y-caracteristicas-instalacion/>. [Último acceso: 08 Junio 2023].
- [21] J. Pomeyrol, «Disponible Ubuntu 22.04.2 LTS con Linux 5.19 y más novedades,» MuyLinux, 27 Febrero 2023. [En línea]. Available: <https://www.muylinux.com/2023/02/27/ubuntu-22-04-2-lts/>. [Último acceso: 08 Junio 2023].
- [22] Assamblar Institute, «¿Qué es PHP y para qué sirve?,» [En línea]. Available: <https://assamblarinstitute.com/blog/que-es-php/>. [Último acceso: 09 Junio 2023].
- [23] PHP, «¿Qué es PHP?,» [En línea]. Available: <https://www.php.net/manual/es/intro-what-is.php>. [Último acceso: 09 Junio 2023].
- [24] I. Souza, «¿Qué es PHP y cómo funciona?,» Rockcontent, 09 Marzo 2020. [En línea]. Available: <https://rockcontent.com/es/blog/php/>. [Último acceso: 09 Junio 2023].
- [25] Dinahosting, «¿Qué es Apache y para qué sirve?,» [En línea]. Available: <https://dinahosting.com/ayuda/que-es-apache-y-para-que-sirve/>. [Último acceso: 10 Junio 2023].
- [26] Next U, «Servidor Apache: qué es y cuál es su importancia,» [En línea]. Available: <https://www.nextu.com/blog/que-es-el-servidor-apache-rc22/>. [Último acceso: 10 Junio 2023].
- [27] Webempresa, «¿Qué es Apache y cómo funciona?,» [En línea]. Available: <https://www.webempresa.com/hosting/que-es-servidor-apache.html>. [Último acceso: 10 Junio 2023].
- [28] G. B, «¿Qué es MySQL?,» Hostinger, 08 Febrero 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-mysql>. [Último acceso: 10 Junio 2023].

- [29] ComputerWeekly, «MySQL,» [En línea]. Available: <https://www.computerweekly.com/es/definicion/MySQL>. [Último acceso: 11 Junio 2023].
- [30] A. Robledano, «Qué es MySQL: Características y ventajas,» OperrWebinars, 24 Septiembre 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-mysql/>. [Último acceso: 10 Junio 2023].
- [31] Keepcoding, «¿Qué es OpenSSL?,» [En línea]. Available: <https://keepcoding.io/blog/que-es-openssl/>. [Último acceso: 11 Junio 2023].
- [32] EcuRed, «OpenSSL,» [En línea]. Available: <https://www.ecured.cu/OpenSSL>. [Último acceso: 11 Junio 2023].
- [33] Universidad Veracruzana, «Nuevo parche de seguridad OpenSSL,» [En línea]. Available: https://www.uv.mx/infosegura/general/noti_openssl-2/. [Último acceso: 11 Junio 2023].
- [34] OnwCloud, «Deployment Recommendations,» [En línea]. Available: https://doc.owncloud.com/server/next/admin_manual/installation/deployment_recommendations.html. [Último acceso: 12 Junio 2023].
- [35] OwnCloud, «System Requirements,» [En línea]. Available: https://doc.owncloud.com/server/next/admin_manual/installation/system_requirements.html. [Último acceso: 12 Junio 2023].
- [36] AskUbuntu, «php-curl not installed in php 7.4,» [En línea]. Available: <https://askubuntu.com/questions/1282005/php-curl-not-installed-in-php-7-4>. [Último acceso: 13 Junio 2023].
- [37] DigitalOcean, «How To Install PHP 7.4,» 31 Marzo 2021. [En línea]. Available: <https://www.digitalocean.com/community/tutorials/how-to-install-php-7-4-and-set-up-a-local-development-environment-on-ubuntu-20-04>. [Último acceso: 13 Junio 2023].
- [38] J. Ibañes, «Montaje de una nube segura para uso de una Intranet OTAN y Sistemas de Mando y Control,» Unir, 2017. [En línea]. Available: <https://reunir.unir.net/handle/123456789/6595>. [Último acceso: 15 Junio 2023].

7 ANEXOS

Las listas de los Anexos se muestran a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 28 de agosto de 2023

De mi consideración:

Yo, ANDRÉS FERNANDO REYES CASTRO, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN SERVIDOR DE ALMACENAMIENTO Y COMPARTICIÓN DE ARCHIVOS EN LA NUBE DEL TIPO OPEN SOURCE elaborado por el estudiante KEVIN JOEL CABRERA QUINCHE de la carrera en TECNOLOGÍA 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 12%.

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.

LINK: [Reporte Similitud TIC Cabrera Quinche Kevin Joel.pdf](#)

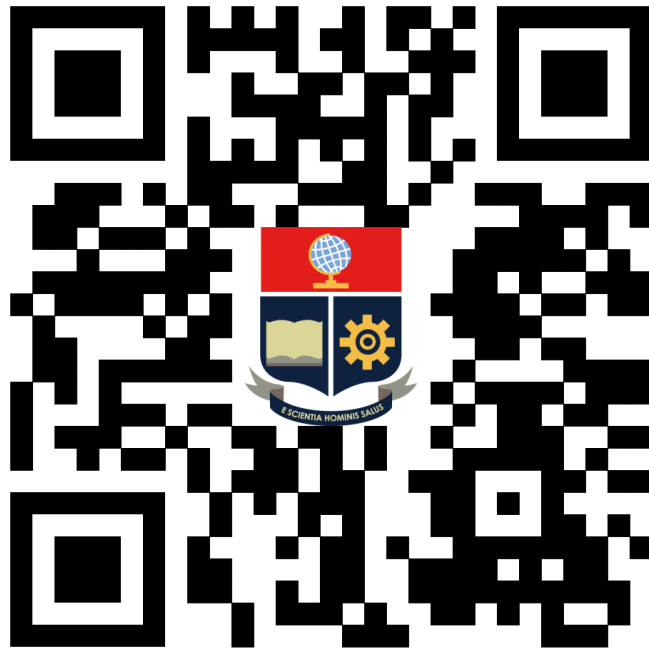
Atentamente,

ANDRÉS FERNANDO REYES CASTRO

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces



Anexo II.I Código QR del video de implementación y pruebas de funcionamiento.

Anexo II.II Enlace del video para descargar.

[TIC_Video Explicativo_Cabrera Quinche Kevin Joel.mp4](#)