

Seguridad y soberanía sobre las telecomunicaciones y datos



Ciclo Superior de Administración de Sistemas Informáticos en Redes

Jesús Rafael Romero Medina

IES Medina Ahazara

14/06/2023



Esta obra está sujeta a una licencia de Reconocimiento - No Comercial - Sin Obra Derivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL PROYECTO FINAL

| | |
|--|--|
| Título del trabajo: | Seguridad y soberanía sobre las telecomunicaciones y datos |
| Nombre del autor: | Jesús Rafael Romero Medina |
| Fecha de entrega (mm/aaaa): | 14/06/2023 |
| Área del Trabajo Final: | Seguridad, Redes, Administración de Sistemas |
| Ciclo Grado Superior: | Administración de Sistemas Informáticos en Red |
| Resumen del Trabajo (máximo 250 palabras): | |
| <p>Este proyecto busca proporcionar una solución efectiva para garantizar la seguridad y la autonomía en las telecomunicaciones y los datos. Al instalar estos servicios informáticos en la Raspberry Pi, se logra una seguridad, un mayor control sobre la privacidad y la protección de la información personal, brindando a las empresas una mayor tranquilidad y confianza en el entorno digital actual.</p> | |

ÍNDICE

| | |
|--|----|
| 1. Introducción..... | 5 |
| 1.1 Análisis de contexto..... | 6 |
| 1.2 Justificación..... | 7 |
| 1.3 Objetivos..... | 8 |
| 2. Propuesta de solución..... | 8 |
| 3. Conocimientos adquiridos en ASIR usados en el proyecto..... | 10 |
| 4. Análisis de Requisitos..... | 10 |
| 4.1 ¿Qué modelo de Raspberry Pi vamos a usar?..... | 11 |
| 5. Temporización..... | 12 |
| 5.1 Identificación de tareas..... | 13 |
| 5.2 Secuenciación de tareas..... | 14 |
| 6. Memoria técnica..... | 15 |
| 6.1 Instalación y configuraciones iniciales de Raspbian..... | 15 |
| - Instalación de Raspbian..... | 15 |
| - Configuración de SSH..... | 17 |
| - Expansión de tarjeta microSD..... | 18 |
| - Estableciendo IP estática en Raspberry Pi 3:..... | 20 |
| 6.2 Implantando múltiples servicios:..... | 23 |
| 6.2.1 Servicio de almacenamiento en la nube..... | 23 |
| - Introducción:..... | 23 |
| - Ventajas del almacenamiento en la nube:..... | 23 |
| - Nextcloud. ¿Qué es?..... | 24 |
| - Características de Nextcloud:..... | 24 |
| - Instalación y configuración de Nextcloud..... | 25 |
| - Instalación y configuración de Apache, PHP8.2 y sus módulos correspondientes:..... | 25 |
| - Descarga de NextCloud y configuración de Apache..... | 27 |
| - Creación de la base de datos:..... | 28 |
| - Configuración inicial de NextCloud:..... | 29 |
| - Configuraciones iniciales de Nextcloud:..... | 31 |
| - Configuraciones extras:..... | 35 |
| - Copias de seguridad Nextcloud:..... | 37 |
| 6.2.2 Sistema de videovigilancia..... | 40 |
| - Introducción:..... | 40 |
| - ¿Qué vamos a usar?..... | 40 |
| - Cámara de videovigilancia..... | 41 |
| - Software Motion..... | 41 |
| - Instalación y configuración de Software Motion..... | 42 |
| - Configuración aplicación Android:..... | 46 |
| 6.2.3 Servicio de monitorización de red..... | 48 |
| - Introducción:..... | 48 |
| - Nagios:..... | 48 |
| - ¿Qué características tiene Nagios?..... | 48 |
| - Preparando tu Raspberry Pi para Nagios..... | 49 |
| - Descargar y compilar Nagios..... | 50 |
| - Configuración de Nagios en la Raspberry Pi..... | 50 |
| - Inicio de Nagios en Raspberry Pi..... | 52 |

| | |
|--|----|
| - Instalación de los complementos de Nagios..... | 52 |
| - Conexión a la interfaz web de Nagios..... | 53 |
| - Agregar su host a Nagios..... | 54 |
| - Agregar un servicio a su host..... | 56 |
| 6.2.4 Servicio de detección de intrusos..... | 57 |
| - Introducción:..... | 57 |
| - Fail2ban:..... | 58 |
| - Servicios que soporta:..... | 58 |
| - Instalación y configuración de Fail2ban:..... | 58 |
| - Comprobación del funcionamiento:..... | 61 |
| 6.3 ¿Cómo instalar y usar No-IP en Raspberry Pi? (DNS Dinámico)..... | 62 |
| - ¿Qué es No-IP?..... | 62 |
| - ¿Cómo funciona?..... | 63 |
| - Creamos una cuenta..... | 63 |
| - Reenvío de puertos en Router:..... | 66 |
| - Instalación y configuración del cliente de actualización dinámica NO-IP..... | 66 |
| 6.4 Configuración SSL nuestro propio certificado autofirmado..... | 69 |
| 7. Estudio presupuestario..... | 71 |
| 8. Conclusiones finales..... | 72 |
| 8.1 Análisis DAFO..... | 72 |
| - Fortalezas:..... | 72 |
| - Debilidades:..... | 73 |
| - Oportunidades:..... | 73 |
| - Amenazas:..... | 74 |
| 8.2 Posibles mejoras..... | 74 |
| 9. Anexos..... | 74 |
| 10. Bibliografía y referencias..... | 75 |

1. Introducción

Para poder adaptarse a los nuevos modelos y formas de negocio del siglo XXI, no es suficiente que las empresas incorporen la tecnología, ya que ésta no produce por sí misma la digitalización. Es necesario un cambio generalizado en los procedimientos, rediseñar sus modelos subyacentes de negocio, así como sus métodos operativos.

La estrategia de la transformación digital permite mejorar la eficiencia en todos los procesos internos de la organización.

En las empresas existen tareas muy interesantes a llevar a cabo como ejemplo de transformación digital de procedimientos internos, si bien es cierto que el cliente no verá reflejado directamente en él mismo dichos cambios, se verá beneficiado de la mejora del rendimiento del negocio gracias a ellos, por lo que son imprescindibles.

En un mundo cada vez más digitalizado y conectado, la seguridad de las telecomunicaciones y los datos se ha convertido en una preocupación fundamental para las empresas. La creciente dependencia de terceros proveedores y el aumento de las amenazas cibernéticas han generado la necesidad de buscar soluciones que brinden mayor seguridad y autonomía en la gestión de la información empresarial.

La Raspberry Pi, una pequeña pero poderosa computadora de bajo costo, ofrece la flexibilidad y la capacidad necesaria para configurar y desplegar una amplia gama de servicios informáticos personalizados. Al aprovechar esta plataforma, se busca establecer una infraestructura tecnológica interna que permita a las empresas tener un mayor control sobre sus comunicaciones y datos, reducir la exposición a riesgos cibernéticos y proteger la información confidencial.

El resultado final será un ecosistema de servicios informáticos autosuficiente y seguro, que nos permitirá comunicarnos y almacenar datos de manera privada y confiable. Al eliminar la dependencia de terceros proveedores de servicios y tener el control total sobre nuestra infraestructura, podremos salvar nuestra privacidad y mantener la confidencialidad de nuestra información.

En resumen, este proyecto busca brindar una solución integral para garantizar la seguridad y autonomía en las telecomunicaciones y proteger nuestros datos. La instalación de estos servicios informáticos en la Raspberry Pi nos permitirá tener un control total sobre nuestras comunicaciones y datos, proporcionándonos una mayor tranquilidad en un mundo cada vez más interconectado.

1.1 Análisis de contexto

En la actualidad, las empresas enfrentan desafíos constantes relacionados con la seguridad de las telecomunicaciones y los datos. La creciente dependencia de las tecnologías de la información y la comunicación, así como el aumento de las amenazas cibernéticas, han generado una mayor preocupación por la protección de la información empresarial y la privacidad de los clientes. Para comprender el contexto en el que se desarrolla este proyecto, es importante analizar algunos factores relevantes:

- **Transformación digital:** En la actualidad, las empresas están inmersas en un proceso de transformación digital, adoptando tecnologías avanzadas como la inteligencia artificial, el big data, la computación en la nube etc. Esta transformación implica una mayor dependencia de las telecomunicaciones y los datos, lo que a su vez aumenta la necesidad de seguridad y soberanía sobre ellos.
- **Ciberataques y vulnerabilidades:** Las empresas se enfrentan a una amplia gama de ciberataques, como el malware, el ransomware y el phishing, que pueden comprometer la seguridad de sus sistemas y datos. Además, las vulnerabilidades en los dispositivos y las infraestructuras de red pueden ser explotadas por los ciberdelincuentes para acceder a información confidencial.
- **Soberanía digital y dependencia de proveedores:** Las empresas se enfrentan al desafío de garantizar su soberanía digital, especialmente en un contexto en el que pueden depender de proveedores extranjeros para servicios esenciales como infraestructura de telecomunicaciones, alojamiento de datos y software empresarial. Esto plantea preocupaciones sobre la seguridad, la confidencialidad y la dependencia tecnológica, lo que ha llevado a un mayor interés en desarrollar capacidades internas y soluciones de infraestructura propias.
- **Regulaciones de protección de datos:** Las empresas deben cumplir con regulaciones y leyes de protección de datos cada vez más estrictas, como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea y la Ley de Privacidad del Consumidor de California (CCPA). Esto implica implementar medidas de seguridad adecuadas para garantizar la privacidad y el manejo seguro de los datos de los clientes.
- **Eficiencia energética y sostenibilidad:** Cada vez más, las empresas están prestando atención a la eficiencia energética ya la reducción de su huella ambiental. La Raspberry Pi es conocida por su bajo consumo de energía, lo que la convierte en una opción favorable desde el punto de vista de la sostenibilidad y puede ayudar a las empresas a reducir su consumo de energía y costos operativos.

En este contexto, este proyecto se presenta como una solución viable para obtener seguridad y soberanía sobre las telecomunicaciones y los datos. Al aprovechar el potencial de esta plataforma, las empresas u organizaciones pueden establecer su propia infraestructura de comunicación y almacenamiento, reducir la dependencia de terceros proveedores, reducir también los costes económicos y mantener un mayor control sobre sus datos sensibles. Esto proporciona una capa adicional de seguridad y privacidad, mitigando los riesgos asociados con las amenazas cibernéticas y la dependencia de servicios externos.

1.2 Justificación

En la actualidad, la seguridad de las comunicaciones y la protección de los datos personales se han convertido en aspectos fundamentales en nuestra vida digital. El crecimiento exponencial de las amenazas cibernéticas y la vigilancia constante por parte de diferentes entidades han generado una creciente preocupación acerca de la privacidad y la autonomía en nuestras telecomunicaciones.

La dependencia de servicios externos y el almacenamiento en la nube plantean riesgos significativos, ya que nuestras comunicaciones y datos personales pueden estar expuestos a violaciones de privacidad, censura o incluso robo de información sensible. Además, los proveedores de servicios pueden tener acceso y control sobre nuestros datos, limitando nuestra autonomía y capacidad de gestionarlos de acuerdo con nuestras preferencias y necesidades.

En este contexto, surge la necesidad de buscar soluciones que permitan a las empresas mantener el control total sobre sus telecomunicaciones y datos, brindando mayor seguridad y autonomía. La Raspberry Pi, una plataforma de computación de bajo costo y altamente personalizable, se presenta como una opción ideal para lograr estos objetivos.

Este proyecto nace con la idea de implantar diversos procedimientos con el menor coste posible y el mayor número de ventajas.

La justificación principal del mismo es concienciar a las empresas de la posibilidad de realizar la transformación digital con las premisas del ahorro de costes en hardware, software y mantenimiento sin olvidar ni dejar de lado en ningún momento el disponer de una solución fiable, fácil de mantener y de gran calidad.

En resumen, este proyecto se justifica en la actualidad de las empresas debido a los problemas y las demandas en términos de seguridad, privacidad, autonomía y cumplimiento normativo. En este proyecto tenemos la solución para que las empresas pueden fortalecer su seguridad, proteger los datos, aumentar la confianza del cliente y lograr una mayor autonomía en la gestión de las telecomunicaciones y los datos, al tiempo que reduce los costos asociados.

1.3 Objetivos

El objetivo principal de este proyecto es implementar una solución tecnológica en la Raspberry Pi que mejore la eficiencia operativa, fortalezca la seguridad de la información y optimice las comunicaciones empresariales. Al combinar servicios como videovigilancia, nube privada, monitorización de red, sistema de llamadas y detección de intrusos, se busca optimizar los procesos internos, proteger los datos empresariales, garantizar una gestión eficiente de la red y mejorar las comunicaciones tanto internas como externas. Este objetivo central tiene como finalidad impulsar el rendimiento, la seguridad y la competitividad de la empresa en el entorno empresarial actual.

También implica fortalecer la protección de la información, reducir la exposición a riesgos cibernéticos, cumplir con las regulaciones de protección de datos y tener un mayor control sobre la infraestructura tecnológica.

2. Propuesta de solución

¿Qué vamos a realizar en nuestra Raspberry Pi?

- **Instalación de Sistema Operativo (Raspbian):** Raspbian es un sistema operativo basado en Linux diseñado específicamente para las Raspberry Pi. Necesitarás instalarlo en tu Raspberry Pi para comenzar a trabajar con ella.
- **Configuraciones iniciales de Sistema Operativo:** Después de instalar el sistema operativo, deberás realizar algunas configuraciones iniciales, como establecer una contraseña segura, configurar la conexión a Internet y actualizar el sistema.
- **Configuración de acceso remoto:** Configure el acceso remoto que le permitirá controlar su Raspberry Pi desde otro dispositivo, como una computadora portátil o un teléfono inteligente. Esto facilitará la administración de su Raspberry Pi sin tener que conectarla a un monitor y un teclado.

- **Servicio 1º: Sistema de videovigilancia (Motion):** Motion es una aplicación que convierte tu Raspberry Pi en un sistema de videovigilancia. Podrás conectar una cámara a la Raspberry Pi y utilizar Motion para grabar videos o detectar movimiento.
- **Servicio 2º: Crear nube privada (NextCloud):** NextCloud es una plataforma de almacenamiento en la nube que puedes instalar en tu Raspberry Pi. Te permitirá almacenar y sincronizar archivos, acceder a ellos desde cualquier lugar y compartirlos con otros de manera segura.
- **Servicio 3º: Sistema de monitorización de red (Nagios):** Nagios es una herramienta de monitorización de red que te permitirá supervisar el rendimiento y la disponibilidad de tus dispositivos y servicios de red. Podrás recibir alertas en caso de que algo no funcione correctamente.
- **Servicio 4º: Sistema de detección de intrusos (Fail2ban):** Fail2ban es una herramienta de seguridad que protege tu Raspberry Pi contra intentos de intrusión. Analiza los registros del sistema en busca de patrones sospechosos y bloquea automáticamente las direcciones IP de los atacantes.
- **Configuración DNS dinámica a través de NoIP:** NoIP es un servicio de DNS dinámico que te permite asignar un nombre de dominio a tu Raspberry Pi, incluso si tu dirección IP cambia. Podrá acceder a su Raspberry Pi utilizando un nombre de dominio en lugar de una dirección IP estática.

Todos los servicios anteriormente mencionados podrán ser accesibles desde la Red Local. Además realizaremos un “nateo” de los mismos en el Router para que puedan ser utilizados a través de Internet y, por ejemplo, poder ver desde nuestra casa, el servicio de videovigilancia implantado.

Nateo o traducción de direcciones de red o NAT: Es un mecanismo utilizado por Routers IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles. En nuestro caso realizaremos una redirección de puertos para poder acceder a los servicios desde fuera.

Esta técnica puede permitir que un usuario externo tenga acceso a un puerto en una dirección IP privada (dentro de una LAN) desde el exterior vía un Router con NAT activado.

La redirección de puertos permite que ordenadores remotos se conecten a un ordenador concreto dentro de una LAN privada.

3. Conocimientos adquiridos en ASIR usados en el proyecto

- Arquitectura de nuestro ordenador (Raspberry Pi)
- Instalación, gestión y administración de Software en Linux.
- Manejo y administración de repositorios.
- Gestión de ficheros.
- Manejo de consola de comandos.
- Gestión de usuarios y grupos.
- Administración de recursos.
- Principios de seguridad informática.
- Gestión de procesos e hilos de procesos.
- Administración de redes locales.
- Administración de redes nateadas.
- Administración remota Linux.
- Scripting en Linux.
- Gestión de copias de seguridad.
- Programación de tareas.

4. Análisis de Requisitos

Para realizar este proyecto con tu Raspberry Pi, necesitarás algunos requisitos básicos. Aquí tienes una lista de los elementos esenciales:

- **Raspberry Pi:** Necesitarás una Raspberry Pi. Puedes elegir entre diferentes modelos, como Raspberry Pi 4 o Raspberry Pi 3, dependiendo de tus necesidades y presupuesto.
- **Tarjeta microSD:** La Raspberry Pi utiliza una tarjeta microSD para almacenar el sistema operativo y los archivos. Asegúrate de obtener una tarjeta microSD de alta calidad con capacidad suficiente para tus necesidades (se recomienda al menos 16 GB).
- **Adaptador de corriente:** La Raspberry Pi necesita un adaptador de corriente para funcionar. Asegúrate de tener uno que proporcione la potencia adecuada para tu modelo de Raspberry Pi.

- **Cable HDMI:** Si planeas conectar la Raspberry Pi a un monitor, necesitarás un cable HDMI para establecer la conexión.
- **Teclado y ratón:** Si no vas a utilizar el acceso remoto de inmediato, necesitarás un teclado y un ratón USB para configurar la Raspberry Pi y realizar las primeras configuraciones.
- **Caja o carcasa:** Aunque no es estrictamente necesario, es recomendable utilizar una caja o carcasa para proteger tu Raspberry Pi de daños físicos y para mantener un aspecto ordenado.
- **Cámara:** Si planeas utilizar el servicio de videovigilancia con Motion, puedes adquirir una cámara compatible con la Raspberry Pi para capturar imágenes y videos.
- **Micrófono y altavoz (opcional):** Para el servicio de llamadas con Asterisk, puedes utilizar un micrófono y altavoz USB para realizar y recibir llamadas de voz.
- **Dispositivo de almacenamiento externo (opcional):** Si planeas configurar una nube privada con NextCloud y necesitas más capacidad de almacenamiento, puedes conectar un dispositivo de almacenamiento externo, como un disco duro USB, a tu Raspberry Pi.

Además de estos elementos físicos, necesitarás una conexión a Internet estable para acceder y administrar tu Raspberry Pi de forma remota, así como los cables y adaptadores necesarios para establecer las conexiones adecuadas.

4.1 ¿Qué modelo de Raspberry Pi vamos a usar?

La Raspberry Pi 3 Modelo B es una computadora de placa única desarrollada por la Fundación Raspberry Pi. Fue lanzado en febrero de 2016 y forma parte de la tercera generación de placas Raspberry Pi. Estas son algunas características y especificaciones clave de Raspberry Pi 3 Model B:

- **Procesador:** Está alimentado por un procesador ARM Cortex-A53 quad-core de 1,2 GHz y 64 bits.
- **Memoria:** Tiene 1 GB de RAM LPDDR2, que se comparte entre la CPU y la GPU.
- **Conectividad:** El Raspberry Pi 3 Modelo B incluye capacidades Wi-Fi (802.11n) y Bluetooth 4.2 integradas, lo que permite la comunicación inalámbrica.
- **Ethernet:** Dispone de un puerto Ethernet 10/100 para conectividad de red cableada.
- **Puertos USB:** Cuenta con cuatro puertos USB 2.0, lo que le permite conectar varios periféricos como teclados, ratones, almacenamiento externo, etc.

- **Video y audio:** la placa tiene un puerto HDMI de tamaño completo para conectarla a una pantalla, un conector de audio de 3,5 mm para salida de audio y admite salida de video compuesto y HDMI.
- **Almacenamiento:** Raspberry Pi 3 Model B se basa en tarjetas microSD para el almacenamiento principal. Tiene una ranura para tarjeta microSD donde puedes insertar la tarjeta que contiene el sistema operativo y tus archivos.

5. Temporización

A continuación, se presenta un posible cronograma de las actividades mencionadas para la Raspberry Pi:

Semana 1:

- Adquirir los elementos necesarios: Raspberry Pi, tarjeta microSD, adaptador de corriente, cables HDMI, teclado, ratón, caja o carcasa, y cualquier otro hardware adicional que requieras.
- Descargar la imagen de Raspbian y preparar la tarjeta microSD con el sistema operativo.
- Instalar Raspbian en tu Raspberry Pi y realizar las configuraciones iniciales del sistema operativo.

Semana 2:

- Configurar el acceso remoto a tu Raspberry Pi para poder administrarla de forma remota desde otro dispositivo.
- Instalar y configurar Motion para el sistema de videovigilancia. Conectar una cámara compatible y realizar las pruebas correspondientes.

Semana 3:

- Instalar y configurar NextCloud para crear tu nube privada en la Raspberry Pi. Configurar los usuarios, la sincronización de archivos y realizar pruebas de acceso remoto.

Semana 4:

- Instalar y configurar Nagios para el sistema de monitorización de red. Definir los dispositivos y servicios a monitorizar, establecer alertas y realizar pruebas de monitorización.

Semana 5:

- Instalar y configurar Fail2ban para el sistema de detección de intrusos. Configurar las reglas de bloqueo y realizar pruebas de detección de intrusos.

Semana 6:

- Configurar el DNS dinámico a través de NoIP. Crear una cuenta en NoIP, configurar la actualización dinámica de DNS en tu Raspberry Pi y verificar el funcionamiento.

Semana 7:

- Realizar pruebas finales de todos los servicios configurados y asegurarte de que funcionan correctamente.
- Realizar ajustes, optimizaciones y mejoras en la configuración según sea necesario.

5.1 Identificación de tareas

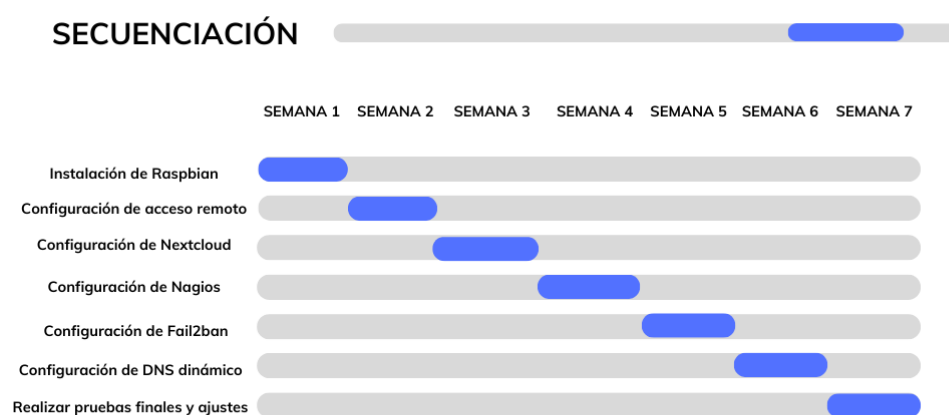
Estas serán nuestras tareas que realizaremos en el proyecto:

- Adquirir los elementos necesarios (Raspberry Pi, tarjeta microSD, adaptador de corriente, cables, teclado, ratón, etc.).
- Descargue la imagen de Raspbian y prepare la tarjeta microSD con el sistema operativo.
- Instala Raspbian en tu Raspberry Pi.
- Realizar las configuraciones iniciales del sistema operativo.
- Configure el acceso remoto a la Raspberry Pi.
- Instalar y configurar el software Motion para el sistema de videovigilancia.
- Conecte y configure una cámara compatible con Motion.
- Instale y configure NextCloud para crear su nube privada en la Raspberry Pi.
- Configurar usuarios y establecer la configuración de archivos en NextCloud.
- Instalar y configurar Nagios para el sistema de monitorización de red.
- Definir dispositivos y servicios a monitorizar en Nagios.
- Configurar alertas y notificaciones en Nagios.

- Instalar y configurar Fail2ban para el sistema de detección de intrusos.
- Configurar reglas de bloqueo en Fail2ban.
- Configurar el DNS dinámico a través de NoIP.
- Crear una cuenta en NoIP y configurar la actualización dinámica de DNS en la Raspberry Pi.
- Realizar pruebas finales de todos los servicios configurados.
- Realizar ajustes, optimizaciones y mejoras según sea necesario.

5.2 Secuenciación de tareas

Aquí tenemos una secuenciación en formato de diagrama de Gantt para el proyecto en tu Raspberry Pi. Ten en cuenta que el tiempo estimado para cada tarea es aproximado y puede variar según tus habilidades y disponibilidad de tiempo.



Planificación de mi proyecto en semanas

6. Memoria técnica

Vamos a proceder a la explicación de nuestro proyecto y de los pasos que hemos ido dando de una forma más técnica y detallando más las cosas. Empezemos:

6.1 Instalación y configuraciones iniciales de Raspbian

Es necesario dotar de un sistema operativo a nuestra Raspberry Pi, por lo que accederemos al apartado de descargas de su Web para descargar Raspbian, el sistema operativo oficial de Raspberry

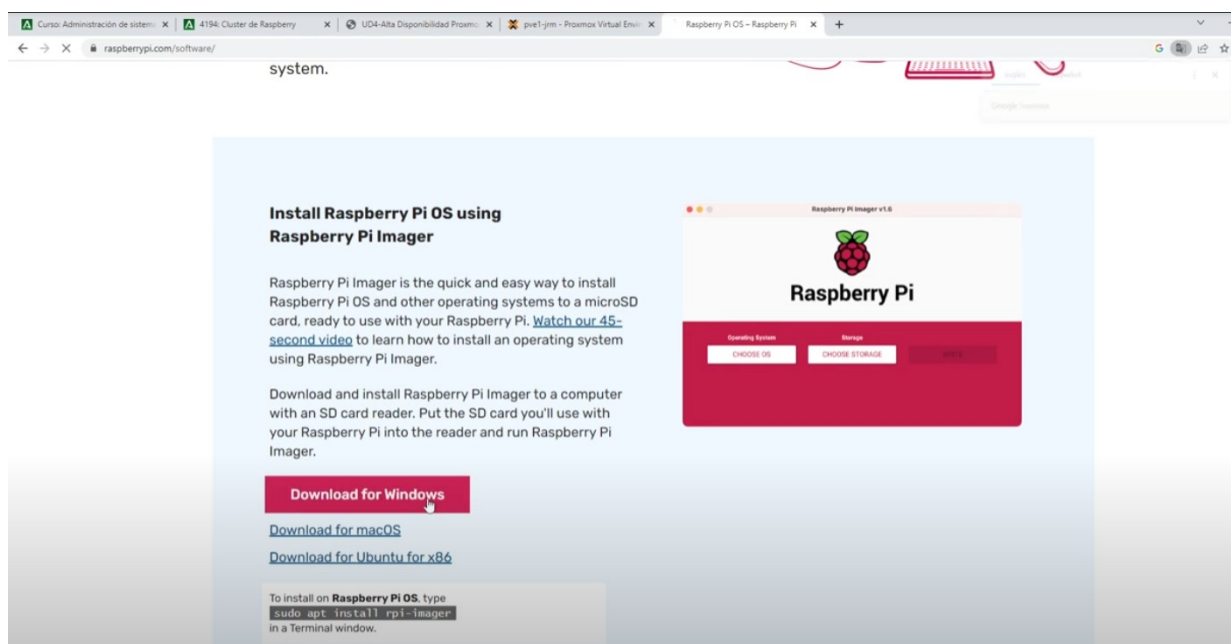
<https://www.raspberrypi.org/downloads/raspbian/>

En nuestro caso descargaremos la última versión de Raspbian.

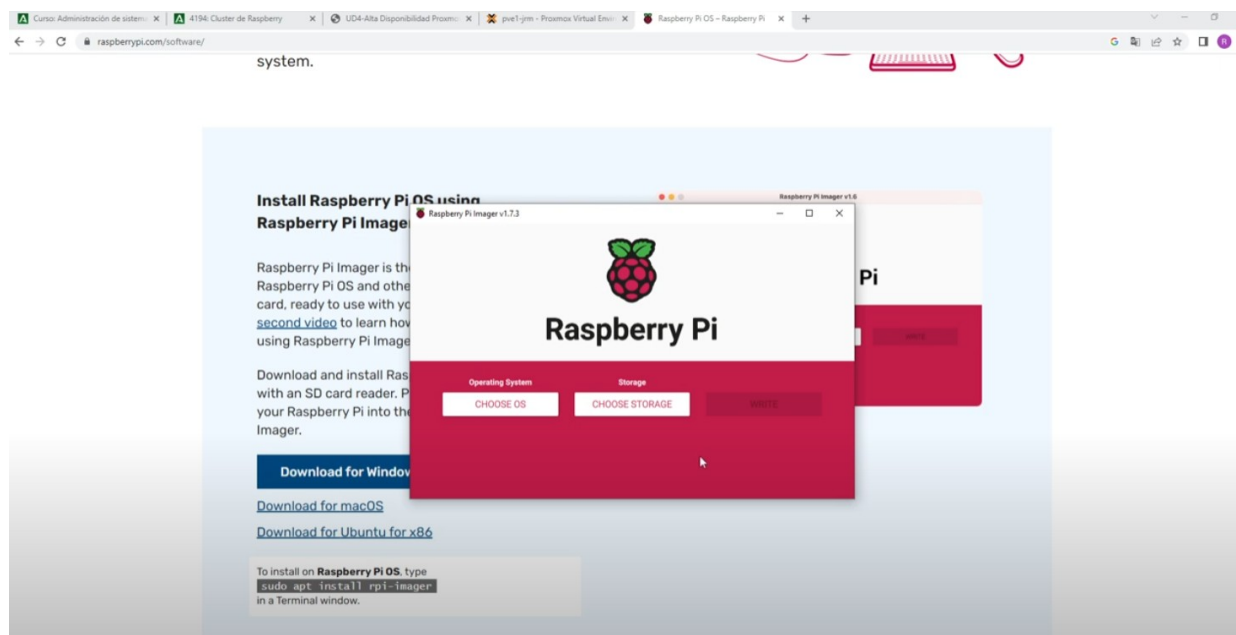
Es necesario disponer de una tarjeta microSD para instalar el S.O, que posteriormente introduciremos en la Raspberry Pi. En este proyecto usaré una tarjeta SAMSUNG de 128GB microSD HC I de clase 10.

- Instalación de Raspbian

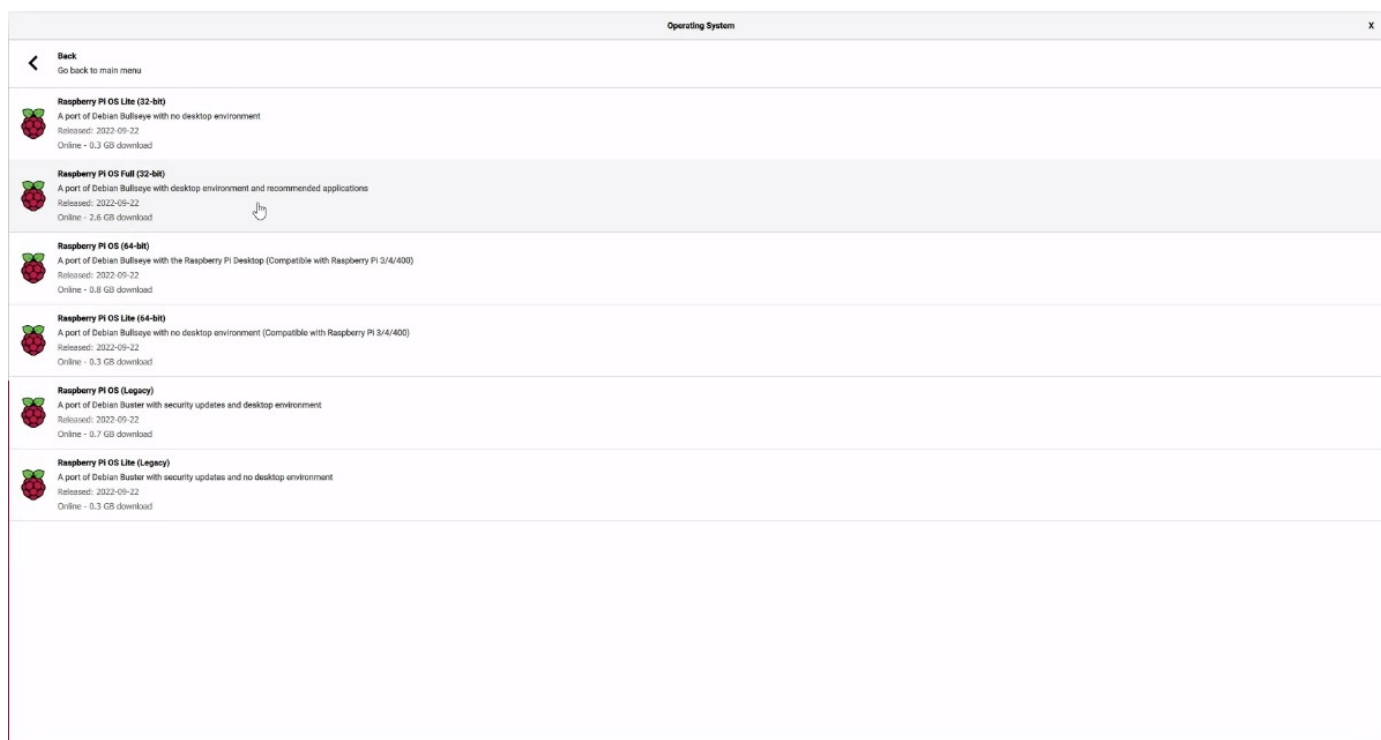
Descargamos e instalamos la versión última de Raspbian.



Abrimos el instalador, y procedemos con la instalación y comprobamos que haya funcionado correctamente y abrimos el software.



Elegimos que sistema operativo vamos a instalar. En este caso elegimos el sistema con escritorio, y de 64(bits).



- Configuración de SSH

A continuación, vamos a comenzar a realizar las configuraciones iniciales más recomendadas para nuestro pequeño servidor.

Habilitamos el SSH, le ponemos como nombre del host 'raspberrypi3' y le atribuimos un usuario y su contraseña.

Es recomendable cambiar la contraseña para intentar mejorar la seguridad de nuestro sistema, la contraseña por defecto es "raspberry", nosotros la cambiaremos por otra más segura.

The image shows a screenshot of the Raspberry Pi configuration tool (raspi-config) interface. The settings are as follows:

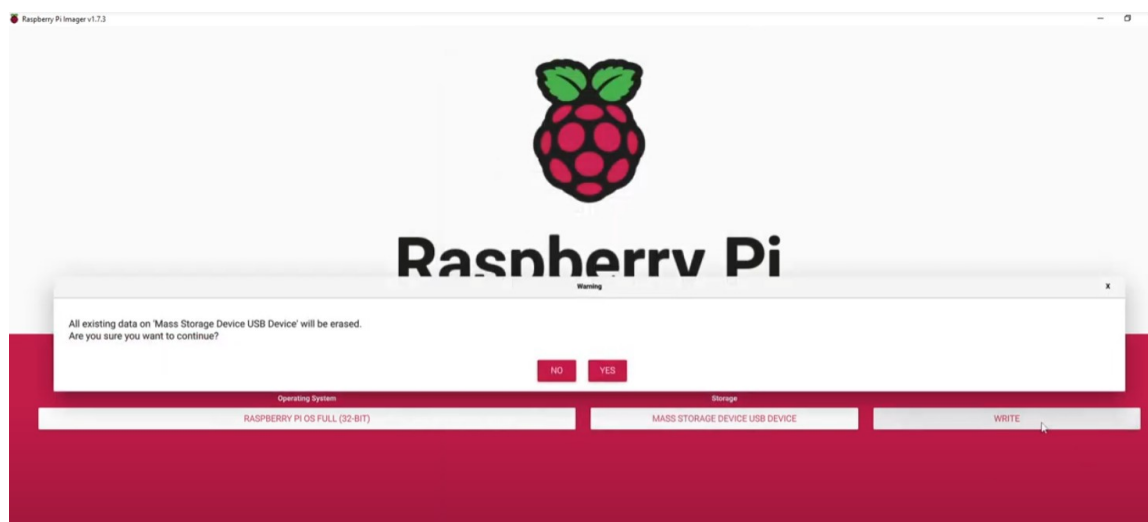
- ☒ Set hostname: `raspberrypi3`.local
- ☒ Enable SSH
 - ☒ Use password authentication
 - ☐ Allow public-key authentication only
 - Set authorized_keys for 'jrommed': _____
- ☒ Set username and password
 - Username: `jrommed`
 - Password: `••••••••`
- ☐ Configure wireless LAN
 - SSID: `sagemcom5B08-5G`
 - ☐ Hidden SSID
 - Password: `••••••••••`
 - ☐ Show password
 - Wireless LAN country: `GB`
- ☐ Set locale settings
 - Time zone: `Europe/Madrid`
 - Keyboard layout: `us`

Persistent settings

- ☐ Play sound when finished
- ☒ Eject media when finished
- ☒ Enable telemetry

At the bottom, there is a red button labeled "SAVE".

Pulsamos en write y confirmamos y empezara el proceso de escritura.



- Expansión de tarjeta microSD

A continuación, vamos a realizar una de las tareas más importantes a la hora de configurar nuestra Raspberry Pi, la expansión de la tarjeta microSD.

¿Qué significa expandir la tarjeta microSD? El sistema operativo instalado en nuestra tarjeta (después de flashear y arrancar la Raspberry Pi) solamente ocupa una pequeña parte del tamaño total de la tarjeta, ya que el S.O se instala en una partición con el espacio necesario (este espacio varía en función de la distribución flasheada).

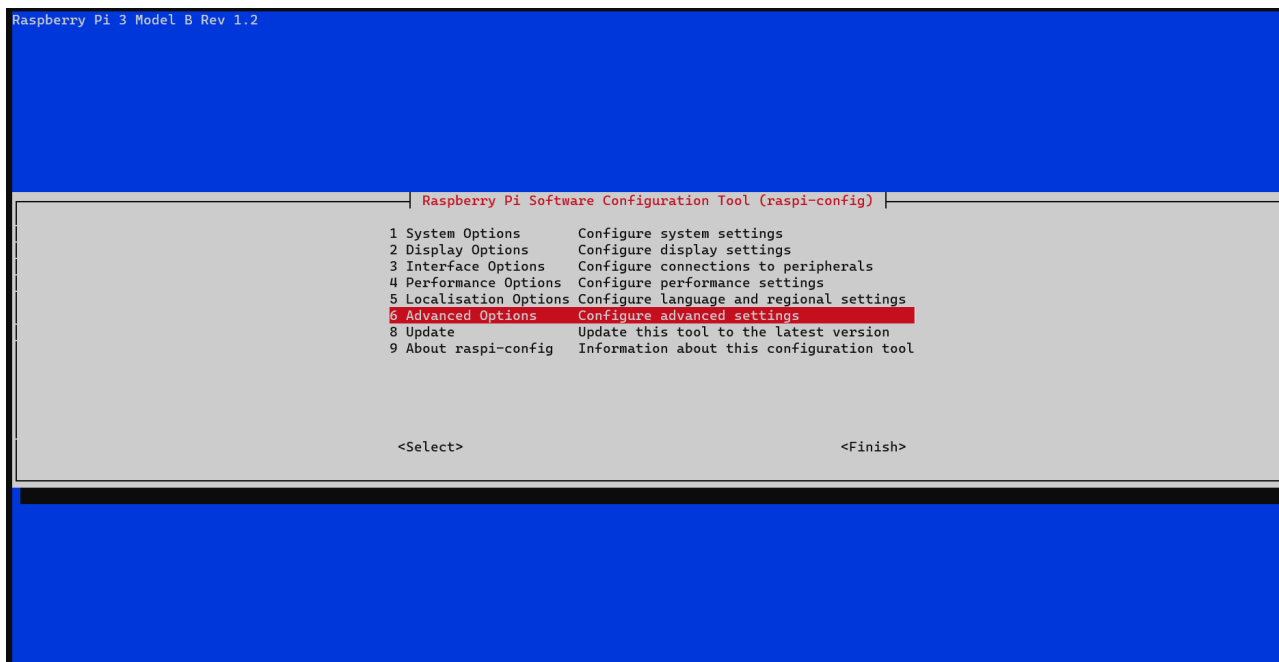
Esto deja la mayoría de espacio de la tarjeta sin utilizar. Para poder aprovecharlo, expandiremos la partición para ocupar toda la tarjeta, de esta forma dispondremos de más espacio para la realización completa del proyecto y servicios que lo componen.

Desde una terminal ejecutaremos el siguiente comando:

sudo raspi-config

Tras ejecutarlo, se nos abrirá la herramienta de configuración de Raspberry Pi, que, si bien es cierto que tiene muchas similitudes y semejanzas con el anterior menú gráfico, incluye ciertas opciones y características que el anterior carece de ellas, como la opción para expandir nuestra tarjeta.

- Seleccionaremos el menú de opciones avanzadas:



- Y elegiremos la opción "Expand Filesystem" para expandir nuestra tarjeta MicroSD.



Por último, el sistema nos pedirá un reinicio, tras ejecutarlo, podemos comprobar con el siguiente comando si la tarjeta MicroSD se ha expandido correctamente:

sudo free

```
jrommed@raspberrypi3:~ $ sudo free
              total        used        free      shared  buff/cache   available
Mem:          931432      248208      113160         3656       570064      614772
Swap:         102396         1280       101116
```

- Estableciendo IP estática en Raspberry Pi 3:

Algo muy importante es tener siempre localizada en nuestra red la Raspberry, además nos facilitará la tarea de implantar diversos servicios tener una IP estática en Raspberry Pi.

Como primer paso, debemos tener en cuenta el tipo de conexión a Internet con la que nuestra Raspberry se conecta a la red. Abriremos una consola de comandos en la cual ejecutaremos el comando ifconfig:

sudo ifconfig

```
jrommed@raspberrypi3:~ $ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.22 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a0d7:7f1d:5474:4d1e prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:4f:87:89 txqueuelen 1000 (Ethernet)
    RX packets 1560 bytes 294887 (287.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 305 bytes 51403 (50.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 20 bytes 2336 (2.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 2336 (2.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jrommed@raspberrypi3:~ $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether b8:27:eb:4f:87:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.22/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
        valid_lft 85819sec preferred_lft 75019sec
    inet6 fe80::a0d7:7f1d:5474:4d1e/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether b8:27:eb:1a:d2:dc brd ff:ff:ff:ff:ff:ff
jrommed@raspberrypi3:~ $
```

Dicho comando nos permite conocer información sobre la red, concretamente los adaptadores de red de los que disponemos y, si tenemos asignados o no dirección IP a alguno de los adaptadores. Deberemos conocer el adaptador que queremos establecer con IP estática, en mi caso será “”. Para establecer una IP estática a nuestra Raspberry Pi vamos a editar el archivo `/etc/dhcpd.conf`, para ello ejecutaremos el siguiente comando:

sudo nano /etc/dhcpd.conf

De tal manera que el archivo tendría que tener una estructura similar a la siguiente:

- **static ip_address**=(Aquí deberemos poner la dirección IP que queremos asignar).
- **static routers**=(Aquí deberemos poner la dirección IP de nuestro Router).
- **static domain_name_servers**=(Aquí deberemos poner las direcciones IP de los servidores DNS que deseamos usar, en este caso he puesto las dos de Google).

```
GNU nano 5.4 /etc/dhcpd.conf

# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# Most distributions have NTP support.
#option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
interface eth0
static ip_address=192.168.0.20/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=8.8.8.8 8.8.4.4

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0
```

A continuación, vamos a editar el archivo `/etc/wpa_supplicant/wpa_supplicant.conf`, para ello ejecutaremos el siguiente comando:

- **ssid**: deberemos añadir el nombre de nuestra red.
- **psk**: deberemos añadir la contraseña de nuestra red.
- **key_mgmt**: deberemos añadir el protocolo de seguridad que usará nuestra red.

```
GNU nano 5.4 /etc/wpa_supplicant/wpa_supplicant.conf

network={
    ssid="sagemcom5B08-5G"
    psk=
    key_mgmt=WPA-PSK/WPA2-PSK
}

country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
```

Con un poco de suerte y tras reiniciar nuestro sistema ya tendremos nuestra IP estática. Tan sólo bastará ejecutar el comando `ifconfig` para comprobar que nuestra IP asignada es la que anteriormente habíamos establecido en el archivo de configuración.

sudo ifconfig

```
jrommed@raspberrypi3:~ $ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.20 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a0d7:7f1d:5474:4d1e prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:4f:87:89 txqueuelen 1000 (Ethernet)
    RX packets 147 bytes 22409 (21.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 98 bytes 13206 (12.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 16 bytes 2044 (1.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 2044 (1.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jrommed@raspberrypi3:~ $ |
```

6.2 Implantando múltiples servicios:

A continuación, vamos a configurar y poner en servicio en la Raspberry Pi los siguientes servicios:

Servicio 1º: Sistema de videovigilancia (Motion).

Servicio 2º: Crear nube privada (NextCloud).

Servicio 3º: Sistema de monitorización de red (Nagios).

Servicio 4º: Sistema de detección de intrusos (Fail2ban).

6.2.1 Servicio de almacenamiento en la nube

- Introducción:

Proveniente del Inglés Cloud Storage, el almacenamiento en la nube es un modelo de almacenamiento de datos en el cual la información está alojada generalmente en múltiples servidores y en ocasiones en múltiples localizaciones.

Por lo general el entorno físico es administrado por una empresa de alojamiento. Los proveedores del almacenamiento en la nube son responsables de mantener los datos disponibles y accesibles y el entorno físico protegido.

- Ventajas del almacenamiento en la nube:

1. Acceder a tus documentos - Desde cualquier lugar.
2. Trabajar con otros - Desde cualquier lugar.
3. Copias de seguridad de los archivos.
4. Seguridad de los datos.
5. Ahorro de costes.

Acceso a los datos: La nube nos permite acceder a los datos desde cualquier lugar, generalmente accederemos a ellos mediante un software específico, una aplicación o una página Web.

Trabajar con otros: El almacenamiento en la nube nos permitirá trabajar con otras personas compartir archivos con grupos o usuarios concretos con los que deseamos trabajar conjuntamente.

Copias de seguridad de los datos: La mayoría de los proveedores de almacenamiento en la nube disponen de servicios de copias de seguridad para garantizar la disponibilidad de nuestra información.

Seguridad de los datos: Los proveedores de servicio garantizan la seguridad de nuestros datos y la disponibilidad de los mismos.

Ahorro de costes: Reducción de la inversión inicial, reducción de facturas de consumos energéticos, reducción de costes de mantenimientos y gestión, reducción de costes de hardware...

- Nextcloud. ¿Qué es?

Es un proyecto de software libre, creado inicialmente por el mismo creador de Owncloud, Frank Karlitschek, con el objetivo de que los usuarios recuperen el control sobre sus datos. La finalidad del producto es proporcionar a las organizaciones y a los particulares un control sobre su información y datos, facilitando la sincronización y el intercambio de ficheros entre dispositivos. Además, incorpora otras herramientas que permiten comunicarse por audio y vídeo vía WebRTC de manera segura.

- Características de Nextcloud:

- Software libre.
- La seguridad como prioridad.
- Gestionar el flujo de trabajo.
- Cliente para dispositivos móviles o de escritorio.
- Posibilidad de almacenamiento externo.
- Calendario y agenda de contactos.
- Llamadas de audio y vídeo seguras.
- Integración con Active Directory, LDAP, Kerberos...
- Contraseñas integradas.
- Cuotas de usuario.
- Monitorización de la actividad del servidor.
- Trackeo de los cambios en archivos.
- Visualización y edición de documentos con Collabora.

- Apps propias de NextCloud.
- Interfaz amigable.
- Previews de archivos.
- Fácil personalización y configuración.

- Instalación y configuración de Nextcloud.

Vamos a proceder a instalar Nextcloud en nuestra Raspberry Pi usando conjuntamente Apache2, PHP 8.2 y MariaDB para la base de datos.

Podemos configurar NextCloud con las siguientes bases de datos:

- SQLite.
- Mysql.
- MariaDB.
- PostgreSQL.

Usaremos una de las combinaciones más recomendadas: MariaDB + APACHE2 + PHP8.

- Instalación y configuración de Apache, PHP8.2 y sus módulos correspondientes:

Para instalar y configurar PHP y sus módulos en una Raspberry Pi 3 modelo B, puedes seguir los siguientes pasos:

Actualiza el sistema operativo de la Raspberry Pi utilizando el siguiente comando:

```
sudo apt-get update && sudo apt-get upgrade
```

Agrega el repositorio de PHP a la lista de fuentes de paquetes de apt usando el siguiente comando:

```
sudo apt-get install apt-transport-https lsb-release ca-certificates
```

```
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg  
https://packages.sury.org/php/apt.gpg
```

```
sudo sh -c 'echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main"  
> /etc/apt/sources.list.d/php.list'
```

El primer comando instala los paquetes ``apt-transport-https, lsb-release ca-certificates``, que son necesarios para descargar e instalar paquetes de forma segura desde repositorios HTTPS.

El segundo comando descarga la clave GPG para el repositorio de PHP y la guarda como archivo ``/etc/apt/trusted.gpg.d/php.gpg``. Esta clave se utiliza para verificar la integridad de los paquetes descargados del repositorio.

El tercer comando agrega un nuevo repositorio de software para PHP a la lista de fuentes del sistema. El comando ``$(lsb_release -sc)`` devuelve el nombre en clave de la versión del sistema, que se utiliza en la URL del repositorio. Luego, la URL del repositorio se agrega al archivo ``/etc/apt/sources.list.d/php.list``.

Después de ejecutar estos comandos, el sistema podrá descargar e instalar paquetes PHP desde el repositorio recién agregado.

Actualiza la lista de paquetes de apt usando el siguiente comando:

`sudo apt-get update`

A continuación, instalaremos apache2, mariadb-server y libapache2-mod-php

`sudo apt-get install apache2 mariadb-server libapache2-mod-php`

Después de la instalación de MariaDB server, el software pedirá la creación de una contraseña de root, en nuestro caso usaremos la contraseña 'root'. Es importante recordarla ya que necesitaremos conocerla durante la configuración de base de datos de Nextcloud. Ahora ejecutaremos el siguiente comando, que nos va a permitir instalar los siguientes paquetes:

`sudo apt install php-gd php-json php-mysql php-curl php-mbstring php-intl php-bcmath php-gmp php-imagick php-xml php-zip`

Estos paquetes son extensiones de PHP que agregan funcionalidades adicionales a tu instalación de PHP, lo que te permite trabajar con imágenes, bases de datos, solicitudes HTTP, internacionalización, manipulación de imágenes, XML, compresión y más.

- Descarga de NextCloud y configuración de Apache.

Nos dirigimos a la carpeta.

```
cd /var/www
```

Vamos a proceder a descargar e instalar la última versión de Nextcloud, por lo que la descargaremos de los repositorios oficiales:

```
sudo wget https://download.nextcloud.com/server/releases/latest.tar.bz2
```

Lo descomprimiremos con el siguiente comando:

```
sudo tar -xvf latest.tar.bz2
```

Ahora mismo deberíamos tener ubicada la carpeta de Nextcloud en /var/www/nextcloud Editaremos el siguiente archivo relativo a Apache, referenciando la ruta hacia la carpeta Nextcloud:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Editaremos el archivo añadiendo los siguientes parámetros justo debajo de

```
<VirtualHost*:80>
```

```
ServerName proyectoasirjrm.ddns.net
```

```
<Directory /var/www/nextcloud/>
```

```
Require all granted
```

```
AllowOverride All
```

```
Options FollowSymLinks MultiViews
```

```
<IfModule mod_dav.c>
```

```
Dav off
```

```
</IfModule>
```

```
</Directory>
```

```
</VirtualHost>
```

En general, esta configuración configura un host virtual para el dominio "proyectoasirjrm.ddns.net", sirviendo archivos desde el directorio "/var/www/nextcloud/" y otorgando acceso a todos los usuarios.

Guardamos los cambios y salimos de la edición del fichero. A continuación, estableceremos los permisos a Apache sobre la carpeta Nextcloud para permitir que el propietario tenga acceso completo, que el grupo tenga acceso de lectura y ejecución y que otros no tengan acceso al directorio. Con los siguientes comandos:

```
sudo chown www-data:www-data -R /var/www/nextcloud
```

```
sudo chmod 750 nextcloud -R
```

Vamos a activar los módulos necesarios para Apache, introduciendo uno por uno los siguientes comandos:

```
sudo a2enmod rewrite
```

```
sudo a2enmod headers
```

Para guardar y que se recojan correctamente los cambios reiniciaremos Apache:

```
sudo systemctl restart apache2
```

- Creación de la base de datos:

Para acceder a MariaDB usaremos el mismo comando que para iniciar MySQL:

```
sudo mysql -u root -p
```

Solicitará una contraseña, introduciremos la que pusimos en el proceso de instalación de MariaDB, en caso de no haberla establecido previamente, la contraseña por defecto es: "mariadb".

Una vez dentro del sistema ejecutaremos los siguientes comandos para crear la base de datos:

Creación de base de datos:

```
create Database nextcloud;
```

Creación de usuario:

```
create user jrommed@localhost identified by 'p@ssw0rd';
```

Concesión de privilegios al usuario para que pueda acceder a la base de datos:

```
grant all privileges on nextcloud.* to jrommed@localhost identified by 'p@ssw0rd';
```

Refrescamos los privilegios:

flush privileges;

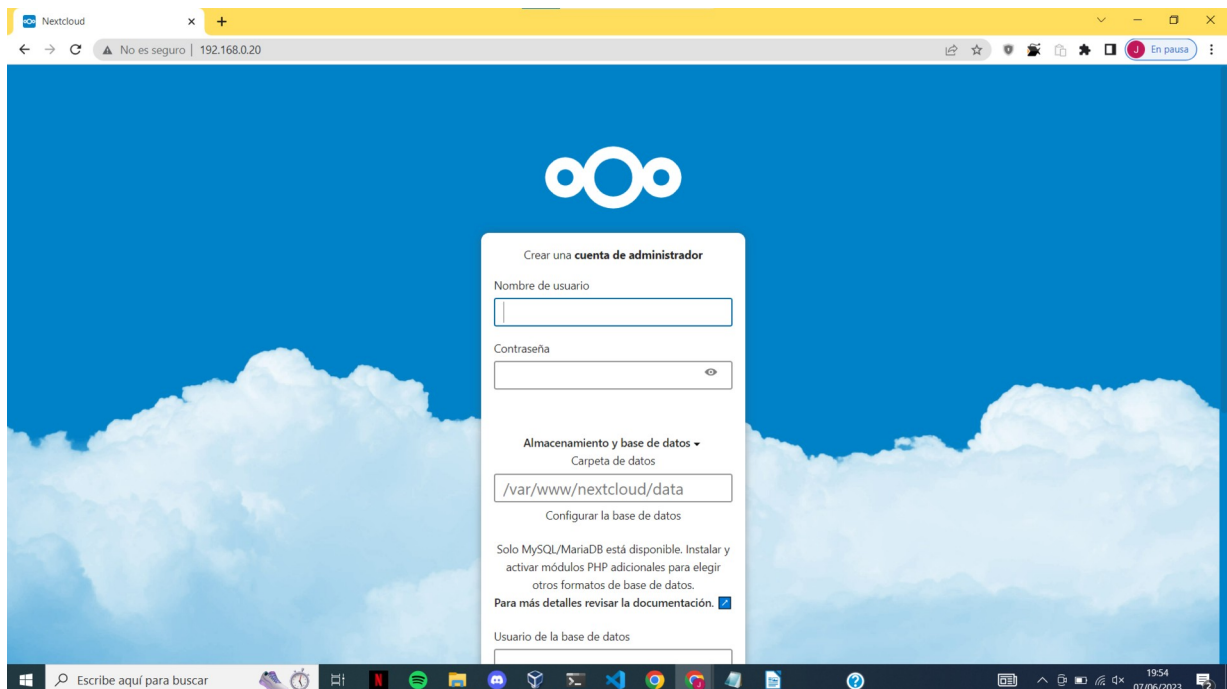
Saldremos del panel MariaDB ejecutando el siguiente comando:

\q

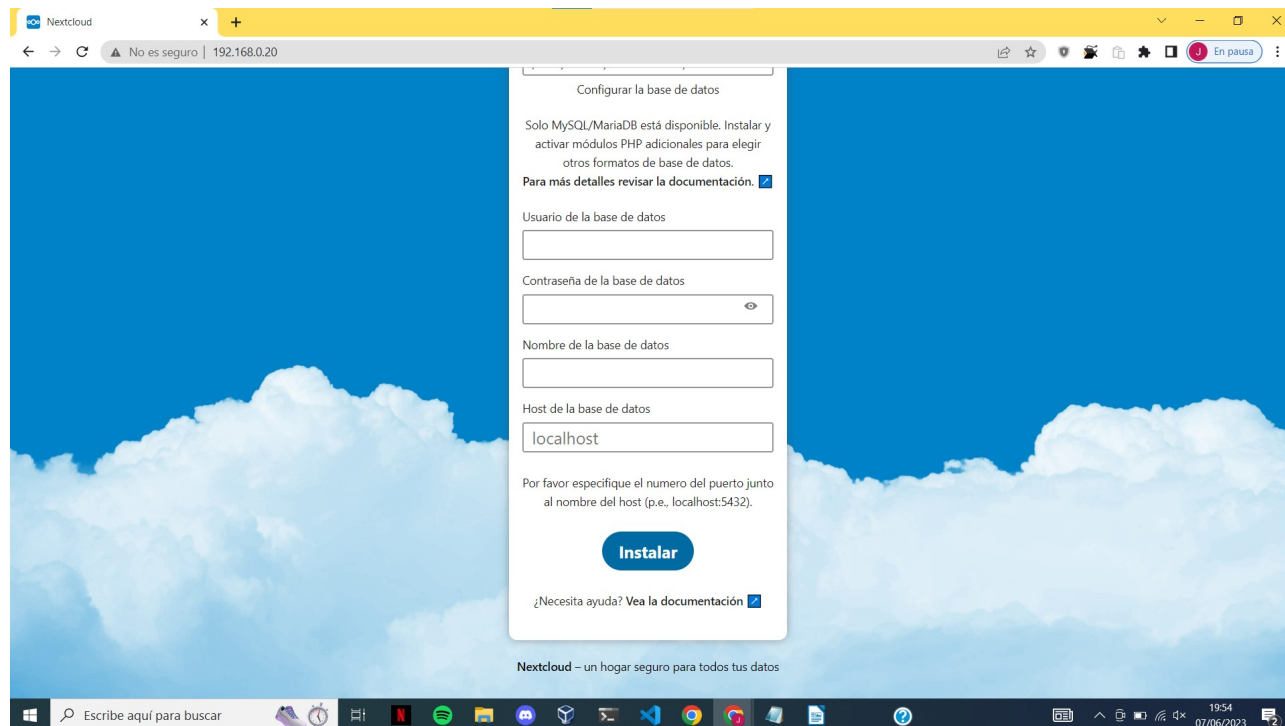
- Configuración inicial de NextCloud:

Una vez creada nuestra base de datos, entraremos a nuestro navegador Web poniendo la dirección IP de nuestra Raspberry PI:

<http://192.168.0.20/>



Una vez introducida la URL con la dirección IP de nuestra Raspberry, debemos crear una cuenta de administrador, como nombre de usuario estableceremos “administrador” y estableceremos una contraseña. Esta cuenta de usuario es la que utilizaremos para entrar en nuestro NextCloud.



En directorio de datos, seleccionaremos la carpeta donde queremos guardar todos nuestros archivos, por defecto será:

`/var/www/nextcloud/data`

Si queremos guardar los datos en un disco duro, dirigiremos la ruta hacia la carpeta automontada del disco, ejemplo `/media/disco1`.

Ahora añadiremos los datos con los que anteriormente creamos la base de datos:

USUARIO BASE DE DATOS:

`jrommed`

CONTRASEÑA DE LA BASE DE DATOS:

`p@ssw0rd`

NOMBRE DE LA BASE DE DATOS:

`nextcloud`

HOST DE LA BASE DE DATOS:

Localhost

Pulsaremos sobre “Completar la instalación”, esperaremos unos minutos mientras se crean las tablas necesarias para crear el sistema y accederemos automáticamente a nuestro NextCloud.

Aumentaremos el límite de memoria de PHP, por un posible aviso que no salga en nuestro NextCloud. Iremos a al siguiente archivo:

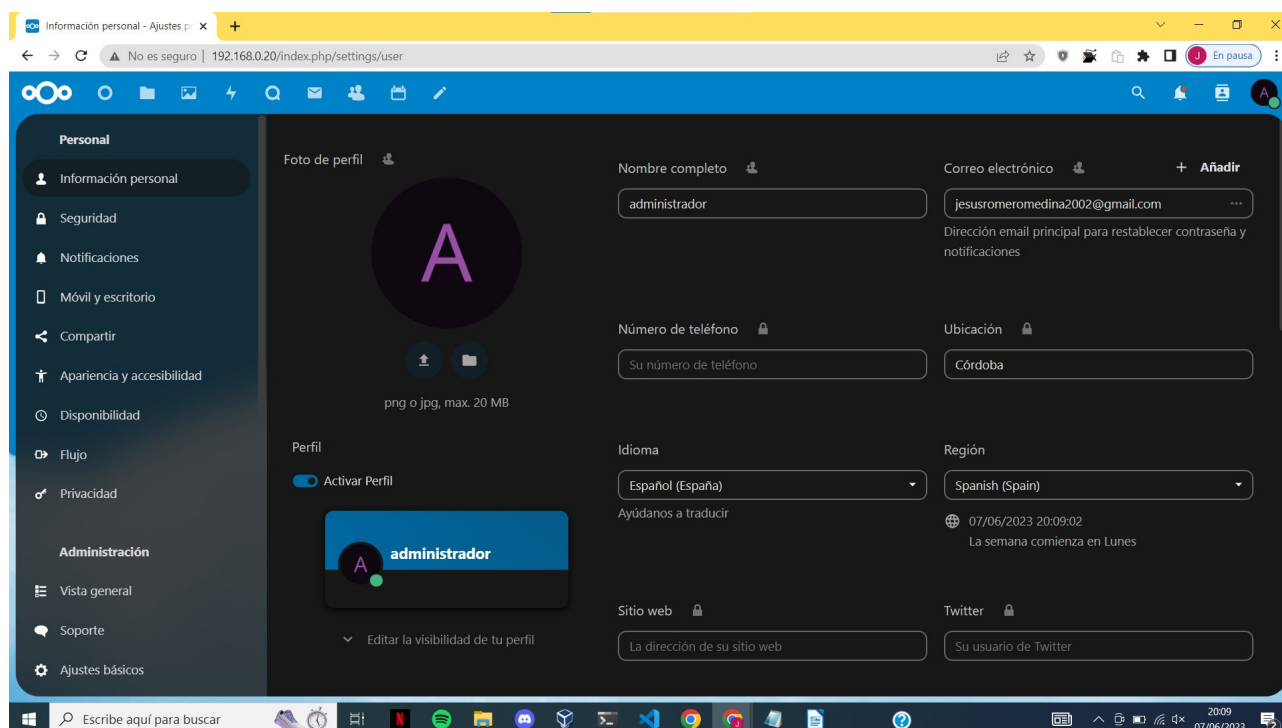
```
sudo nano /var/www/nextcloud/.htaccess
```

Y añadimos la siguiente linea debajo de PHP 8:

```
php_value memory_limit 512M
```

- Configuraciones iniciales de Nextcloud:

En la esquina superior derecha haremos clic en “administrador” y a continuación en “personal”. En el usuario administrador podemos establecer un nombre completo y un correo electrónico para la recuperación de contraseña y notificaciones. También podremos restablecer nuestra contraseña.

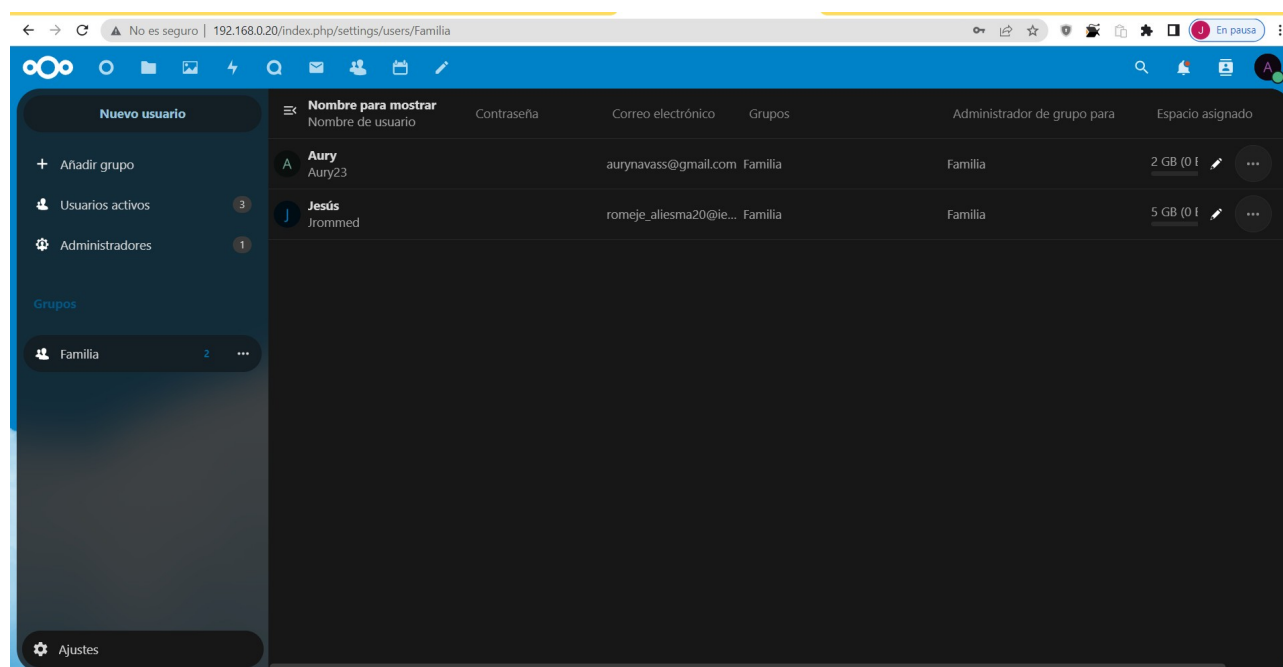


También existe un apartado muy interesante denominado “Actividad” en el cual podemos establecer cuándo y cómo se nos notifican los cambios y actividades referentes a los archivos. Se podrá configurar la frecuencia con la que se envían los correos electrónicos.

Al final de esta pantalla podremos ver nuestra ID de Nube Federada, en nuestro caso administrador@192.168.0.20

Si volvemos al menú, podemos observar el apartado “Usuarios”, en el cual trataremos todo lo referente a la gestión de usuarios y grupos de Nextcloud. Vamos a crear el grupo “Familia”, en el que como usuarios de ejemplo añadiremos los siguientes: “Aurea María” y “Jesús”.

Una vez creados los usuarios y asignarlos al grupo configuraremos una cuota de uso máximo de disco por usuario, estableciendo así 2 GB a “Aurea María”, y 5 GB a “Jesús”.



En el menú también podemos acceder a la sección “Ayuda”, en la cual tenemos a nuestra disposición toda la documentación oficial referente a Nextcloud.

Por último, accederemos posiblemente al menú más importante, el de administración. Una vez dentro recorreremos las pestañas que lo componen. En la pestaña “Ajustos básicos” se muestra Cron, el programador de tareas, permite ejecutar tareas a través de AJAX, Webcron y Cron.

Ajax es la opción predeterminada, desafortunadamente es el sistema menos confiable, cada vez que un usuario visita la página Nextcloud, se ejecuta un solo trabajo en segundo plano. La ventaja de este mecanismo es que no requiere acceso al sistema ni registro con un servicio de terceros. La desventaja reside en la comparación con el servicio Webcron, que requiere visitas regulares a la página para que se active.

Webcron apunta a un servicio externo Webcron ,como por ejemplo;

<https://www.easycron.com/> en el que asegurarás que los trabajos en segundo plano se ejecutarán regularmente. Para usar este tipo de servicio con el servidor, debemos ser capaz de acceder al servidor usando internet. Por ejemplo:

URL a introducir: `http[s]://<dominio-del-servidor>/nextcloud/cron.php`

Usar la característica **Cron** del sistema operativo es el método preferido para ejecutar tareas regulares. Este método habilita la ejecución programada de tareas sin las inherentes limitaciones que el servidor Web pudiera tener.

En la pestaña “**Vista general**” podemos también ver la versión que disponemos de Nextcloud y el canal de actualización pudiendo elegir entre las opciones stable, daily, beta y production.

La pestaña “**Sistema**” permite ver la carga de la CPU de nuestro servidor, así como el uso de memoria. También podemos ver datos como los usuarios activos, el almacenamiento, la versión de PHP y de la base de datos. Podemos monitorizar dicha información desde una herramienta externa usando la siguiente url:

<http://192.168.0.20/ocs/v2.php/apps/serverinfo/api/v1/info>

Llegamos a una de las pestañas más interesantes del menú de administración, la pestaña “**Seguridad**” que nos permitirá cifrar los datos de nuestra nube privada.

Pulsaremos sobre “Habilitar cifrado en el servidor”. El sistema nos advierte que la encriptación por sí sola no garantiza la seguridad del sistema, que aumenta el tamaño de los archivos y que siempre debemos tener una copia de nuestros datos. Habiéndolas leído pulsaremos sobre el botón “Habilitar cifrado”.

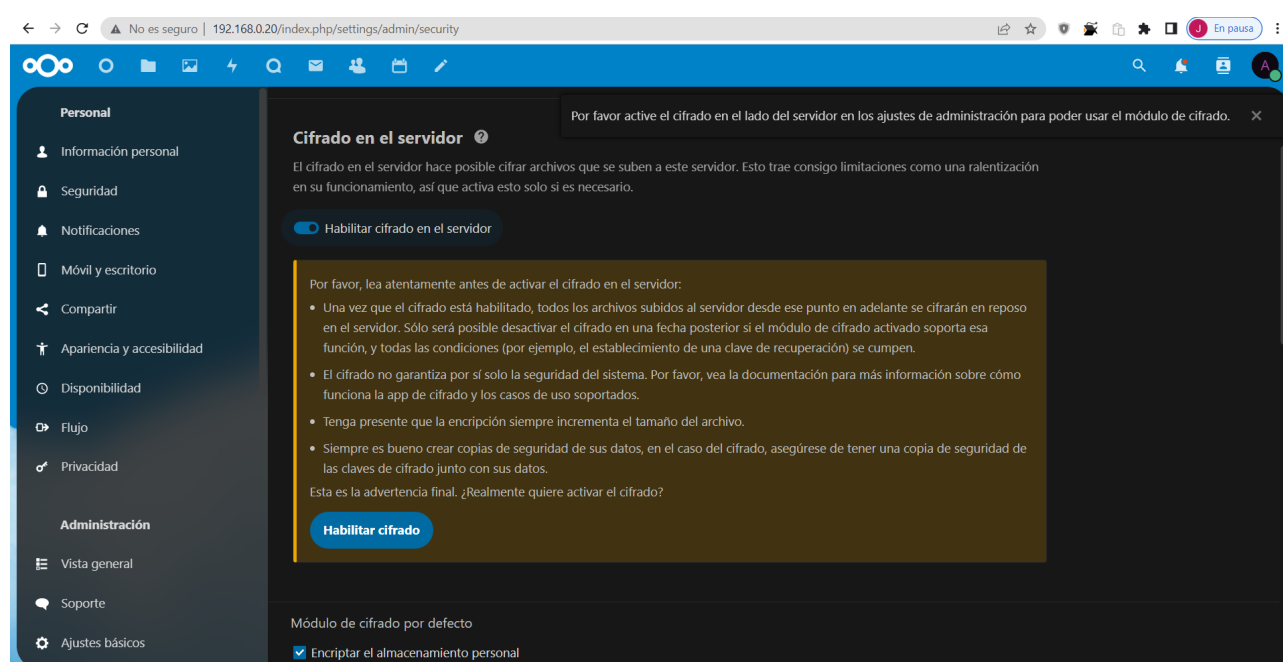
A continuación, vamos a habilitar un módulo de cifrado en el menú de aplicaciones. Lo activaremos para continuar el proceso.

Volveremos al menú administración, a la pestaña “**Seguridad**”, en ella veremos que ahora aparece el módulo que acabamos de activar en nuestro servidor. El sistema nos advierte que la aplicación de encriptación está habilitada pero las claves no están inicializadas, para inicializarlas tan sólo debemos de salir del sistema y volver a iniciar sesión.

Una vez hecha dicha acción, y habiendo vuelto al menú Administración>Seguridad, comprobaremos que sale una casilla con un tic de verificación que establece que se encriptará el almacenamiento personal. Al activar esa opción se encriptarán todos los archivos almacenados en la memoria principal.

Por último, vamos a habilitar las claves de recuperación de ficheros. Si uno de los usuarios pierde su contraseña de acceso a Nextcloud, sus archivos serían irre recuperables. Para evitar esto se creará una clave de recuperación. Si accedemos a la sección Seguridad de la página de administración podremos establecer dicha clave.

Tras todos los cambios realizados en el apartado Seguridad, tendremos nuestra nube encriptada, y la configuración debería ser similar a la siguiente:



La pestaña “**Registros**” permite llevar un registro de los errores que suceden en el sistema.

Desde la pestaña “**Ajustes básicos**” podremos configurar acciones tan importantes como el envío de correos electrónicos para las notificaciones. Será necesario establecer ciertos parámetros tales como la autenticación, servidor SMTP, datos de autenticación...

Tras haber finalizado todas las anteriores opciones habremos terminado de configurar Nextcloud.

- Configuraciones extras:

Vamos a realizar unas configuraciones extras para nuestro NextCloud como las siguientes:

- Aumento del tamaño máximo de carga de Nextcloud:

De forma predeterminada, PHP tiene un límite de carga muy bajo, tan bajo que solo es de 2 MB. Para cambiar esto, necesitamos modificar el `php.ini` archivo y aumentar el límite. Un sistema de almacenamiento en la nube no sería muy útil si solo pudiera cargar archivos de 2MB.

Para comenzar, debemos comenzar a editar el archivo de configuración con el siguiente comando:

```
sudo nano /etc/php/8.2/apache2/php.ini
```

Ahora necesitamos encontrar y reemplazar las siguientes dos líneas.

```
post_max_size = 8M
```

```
upload_max_filesize = 2M
```

Por estas siguientes líneas:

```
post_max_size = 1024M
```

```
upload_max_filesize = 1024M
```

Ahora podemos guardar y salir del archivo.

Ahora necesitamos reiniciar Apache2 para obligarlo a leer el archivo de configuración actualizado. Podemos hacerlo fácilmente con el siguiente comando:

```
sudo service apache2 restart
```

Ahora debería poder reiniciar su navegador web y comenzar una nueva carga para ver que el tamaño máximo de carga se ha incrementado con éxito.

- Mover la carpeta de datos de Nextcloud:

Con Nextcloud ahora instalado de forma segura, ahora podemos modificar la configuración para que sea más segura y un poco más útil. Una de las primeras cosas que debemos hacer es mover el directorio de datos para que no quede en nuestro directorio accesible desde la web.

Para comenzar, hagamos nuestro nuevo directorio donde almacenaremos nuestros archivos de datos. Para hacerlo más fácil, crearemos una nueva carpeta `/var/nextcloudy` moveremos nuestra carpeta de datos allí. Cree la carpeta ejecutando el siguiente comando:

```
sudo mkdir -p /var/nextcloud
```

Con nuestra nueva carpeta que creamos, ahora moveremos nuestro directorio de datos a ella, esto es fácil de hacer gracias al comando `mv`.

Tenga en cuenta que su sistema Nextcloud estará fuera de servicio mientras movemos el archivo y luego ajustamos el archivo de configuración.

Para comenzar el movimiento, escriba el siguiente comando:

```
sudo mv -v /var/www/nextcloud/data /var/nextcloud/data
```

Ahora, con los archivos movidos, ahora podemos modificar la `datadirectory` configuración para que apunte a nuestro nuevo directorio. Primero, cambiemos al directorio de configuración de Nextcloud con el siguiente comando:

```
cd /var/www/nextcloud/config
```

Ahora podemos copiar el archivo de configuración para hacer una copia de seguridad del archivo, podemos hacer esto con el siguiente comando:

```
sudo cp -p config.php config.php.bk
```

Finalmente, abramos el `config.php` archivo para editarlo usando `nano`.

```
sudo nano config.php
```

Dentro de este archivo necesitamos cambiar la siguiente línea:

```
'datadirectory' => '/var/www/nextcloud/data',
```

Por esta:

```
'datadirectory' => '/var/nextcloud/data',
```

Ahora podemos guardar y salir del archivo. Como última precaución, debemos asegurarnos de que el `www-data` usuario aún tenga propiedad sobre nuestra nueva carpeta.

```
sudo chown -R www-data:www-data /var/nextcloud/data
```

Ahora debería poder actualizar su navegador web y todos sus archivos deberían mostrarse exactamente como estaban anteriormente.

- Copias de seguridad Nextcloud:

A continuación, vamos a implementar unos scripts que automáticamente gestionarán copias de seguridad.

Siguiendo las premisas de ahorro de costes, usaremos un disco duro por conexión USB como dispositivo hardware donde se almacenarán las copias.

Lo primero que debemos de hacer es formatear nuestro disco duro. Ejecutaremos el siguiente comando para conocer los dispositivos de almacenamiento de los que dispone el sistema.

```
sudo fdisk -l | grep /dev/
```

Se detectan los discos `"/dev/mmcbkl0"` (la tarjeta microSD de la Raspberry Pi) y `"/dev/sda"` (el disco duro que he conectado mediante USB en el que realizaremos las copias).

A continuación, ejecutaremos el siguiente comando para seleccionar el disco `/dev/sda`:

```
sudo fdisk /dev/sda
```

Pulsaremos la tecla `"n"` para crear una nueva partición y elegiremos la opción `"p"` (primaria):

Escribiremos `"1"` ya que es la primera partición que creamos en el disco duro.

A continuación, estableceremos los valores por defecto en lo relativo a los sectores del disco.

Pulsaremos la tecla `"t"` para establecer el tipo de partición. Puesto que la vamos a formatear en el formato de archivos `"Ext.4"` estableceremos `"Linux"` como tipo de partición. Para conocer el listado de tipos de particiones y sus códigos hexadecimales asociados tan sólo bastará con ejecutar la tecla `"L"`.

En nuestro caso estableceremos como código el `"83"`, el código Hexadecimal de `"Ext4"`.

Para finalizar con la herramienta de particionado `"fdisk"`, teclearemos `"w"` para escribir los cambios en la tabla de particionado.

Si volvemos a ejecutar el siguiente comando:

```
sudo fdisk -l | grep /dev/
```

Comprobaremos que se ha creado correctamente la partición en el disco `"/dev/sda"`, con la nomenclatura `"/dev/sda1"`.

Por último, necesitamos darle formato a dicha partición por lo que ejecutaremos el siguiente comando:

```
sudo mkfs.ext4 /dev/sda1
```

A continuación, debemos montar permanentemente la partición en una carpeta. Crearemos la carpeta “hddnextcloud” en /mnt/ con el siguiente comando:

```
sudo mkdir /mnt/hddnextcloud
```

Una vez creada la carpeta vamos a montarla permanentemente, para ello editaremos el fichero /etc/fstab con el siguiente comando:

```
sudo nano /etc/fstab
```

Añadiendo la siguiente línea:

```
/dev/sda1 /mnt/hddnextcloud ext4 defaults 0 0
```

Es necesario reiniciar el sistema para que recoja los cambios y el disco duro quede montado permanentemente en la carpeta /mnt/hddnextcloud

```
sudo shutdown -r now
```

Tras reiniciar el sistema ejecutaremos el siguiente comando para comprobar que la carpeta se ha montado automáticamente:

```
sudo mount | grep /sda
```

Crearemos varios scripts: backupnextcloud.sh y borradobackupnextcloud.sh

Script backupnextcloud.sh

El script backupnextcloud.sh tiene tres funciones principales:

- Crear la carpeta correspondiente a la copia de seguridad en /mnt/hddnextcloud con el nombre “copianextcloud” y la marca de la fecha actual.
- Crear copia de seguridad de la carpeta /var/www/nextcloud en /mnt/hddnextcloud/copianextcloud(fechaactual) con el nombre “nextcloud-dirbcp” y la marca de la fecha actual.
- Crear copia de seguridad de la base de datos MariaDB en /mnt/hddnextcloud/copianextcloud(fechaactual) con el nombre nextcloud-sqlbcp y la marca de la fecha actual.

```
#!/bin/bash
```

```
#Creamos la carpeta
```

```
mkdir /mnt/hddnextcloud/copianextcloud `date +"%Y%m%d"` 2>/dev/null
```

```
#Copia de la carpeta de Nextcloud
```

```
rsync -Aax /var/www/nextcloud/ /mnt/hddnextcloud/copianextcloud `date
```

```
+"%Y%m%d"`/nextcloud-dirbkp `date +"%Y%m%d"` / 2>/dev/null
```

```
#Copia de la base de datos
```

```
mysqldump --single-transaction -u root -pp@ssw0rd nextcloud >
```

```
/mnt/hddnextcloud/copianextcloud `date +"%Y%m%d"`/nextcloud-sqlbkp `date
```

```
+"%Y%m%d"`.bak 2>/dev/null
```

Ubicaremos el script en la carpeta /bin y asignaremos los permisos 755

7 = Lectura+Escritura+Ejecución

5 = Lectura+Ejecución

Para ello ejecutaremos el siguiente comando:

```
sudo chmod 755 backupnextcloud.sh
```

Podemos comprobar el correcto funcionamiento del script, ejecutándolo independientemente de dónde se esté situado en la consola, es decir no es necesario estar ubicado en la carpeta donde se encuentra el script. Esto sucede al almacenar un script en /bin/

Script borrado backupnextcloud.sh

El script borrado backupnextcloud.sh tiene una función principal:

- Busca en el directorio /mnt/hddnextcloud la existencia de alguna carpeta creada hace más de 7 días. Las carpetas que encuentre serán borradas. Es decir, las copias de seguridad nunca sobrepasarán los 7 días de duración en el sistema.

```
sudo find /mnt/hddnextcloud/ -type d -name 'copia*' -mtime +7 -exec rm -rf {} \; 2>/dev/null
```

Ubicaremos el script en la carpeta /bin y asignaremos los permisos 755

7 = Lectura+Escritura+Ejecución

5 = Lectura+Ejecución

Para ello ejecutaremos el siguiente comando:

```
sudo chmod 755 backupnextcloud.sh
```

Al igual que el anterior script, al haberlo ubicado en la carpeta /bin/ se podrá ejecutar desde cualquier lugar de la consola de comandos.

Programación automática ejecución Scripts:

A continuación, vamos a programar la ejecución automática de los scripts anteriormente creados, de tal manera que diariamente se ejecutará el script de borrado de las copias que tienen más de 7 días a las 21:00 pm y diariamente también una copia de seguridad de todo el sistema Nextcloud a las 21:05 pm.

Para programar la ejecución de los scripts ejecutaremos el siguiente comando. Y añadiremos las siguientes líneas:

sudo crontab -e

```
# m h dom mon dow   command
0 21 * * * /bin/borradobackupnextcloud.sh
5 21 * * * /bin/backupnextcloud.sh
```

6.2.2 Sistema de videovigilancia

- Introducción:

A continuación, vamos a implantar un sistema de videovigilancia a través de nuestro pequeño servidor, algo esencial hoy en día en la mayoría de los negocios, debido a la necesidad de proteger físicamente nuestro entorno.

Si bien es cierto que cada día la seguridad lógica cobra más y más sentido en nuestra vida, no podemos dejar de lado la seguridad física, por lo que, creo que no se puede hablar de seguridad física y no hablar de un sistema de videovigilancia. En este caso constará de una cámara de seguridad configurada para emitir a través de internet y poder ser visualizada desde cualquier lugar.

- ¿Qué vamos a usar?

Para la implantación de este sistema usaremos nuestro mini servidor Raspberry Pi, una cámara y el software “motion”.

Puesto que ya hemos dedicado mucho tiempo a hablar de la Raspberry no indagaremos más en ella, pero sí dedicaremos un par de párrafos a la cámara y el software elegido para la implantación de dicho sistema.

- Cámara de videovigilancia

La cámara elegida para la implantación de un sistema como éste podría ser cualquiera que dispusiese de una conexión por USB, pero puesto que una de las principales premisas del proyecto es el ahorro de costes, vamos a usar una Webcam de segunda mano es decir, que este ya usada pero funciona correctamente, más tarde lo veremos.

En este caso vamos a usar la Cámara SunplusIT Inc USB 2.0 es un dispositivo plug-and-play que se puede conectar fácilmente a una computadora a través de un puerto USB 2.0. Admite una resolución de hasta 640 x 480 píxeles y una velocidad de fotogramas de hasta 30 fotogramas por segundo. La cámara web también cuenta con un micrófono incorporado para la entrada de audio.

El Cámara SunplusIT Inc USB 2.0 es compatible con varios sistemas operativos, incluidos Windows, Mac OS y Linux. Se puede usar para videoconferencias, reuniones en línea, grabación de video y transmisión en vivo.

En general, la Cámara SunplusIT Inc USB 2.0 es una cámara web básica adecuada para el uso diario que ofrece una calidad de imagen y sonido decente a un precio asequible.

- Software Motion

Motion es un programa que monitoriza la señal de vídeo de una o más cámaras y es capaz de detectar si una parte significativa de una imagen ha cambiado. En otras palabras, es capaz de detectar movimiento.

El programa está escrito en C y hecho para el sistema operativo Linux.

Motion está basado en la línea de comandos. No tiene ninguna interfaz gráfica de usuario. Todo el setup se basa en archivos de configuración (archivos simples de texto que pueden ser editados con cualquier editor de texto plano).

Principales características de Motion:

- Opción de importar vídeo desde múltiples cámaras.
- Guardar imágenes cuando la señal de vídeo de la cámara detecte movimiento.
- Crear archivos de vídeo que contengan el evento cuando la cámara detecta movimiento.
- Ejecutar un programa externo cuando el movimiento es detectado.

- Ejecutar un programa externo al comienzo de un evento de varios movimientos detectados.
- Ejecutar un programa externo al final de un evento de varios movimientos detectados. Ejecutar un programa externo cuando una imagen es guardada.
- Streaming de vídeo en directo.
- Hacer fotografías de manera automatizadas o regulares en un intervalo de tiempo.
- Hacer fotografías de manera automatizadas o regulares en un intervalo de tiempo usando cron.
- Alimentar eventos a una base de datos MySQL, PostgreSQL o SQLite3.
- Configurable por el usuario y definido por el usuario en la pantalla.
- Visualización mediante una simple interfaz Web.
- Control automático de reducción de ruido en imagen.
- Altamente configurable la introducción de texto sobre las imágenes.
- Altamente configurable la definición de los nombres de archivos de las imágenes y archivos de vídeos guardados.

- Hardware soportado por Motion:

Motion soporta la entrada de vídeo a través de dos tipos de recursos: Los dispositivos standard video4linux (/dev/video0) y las cámaras de red. Motion no dispone de drivers para cámaras. Si el dispositivo funciona correctamente con otro software común de vídeo, funcionará con Motion y viceversa. En ocasiones es conveniente primero hacer funcionar la cámara con otro software y después utilizar esas opciones de conexión con Motion.

- Instalación y configuración de Software Motion

Para comenzar vamos a realizar una serie de comprobaciones básicas:

Comprobar que el sistema está actualizado:

sudo apt-get update

sudo apt-get upgrade

Comprobar que el sistema detecta la cámara:

sudo lsusb

```
jrommed@raspberrypi3:~$ sudo lsusb
Bus 001 Device 004: ID 0806:0806 SunplusIT Inc USB 2.0 Camera
Bus 001 Device 003: ID 0424:ec00 Microchip Technology, Inc. (formerly SMSC) SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Microchip Technology, Inc. (formerly SMSC) SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
jrommed@raspberrypi3:~$ |
```

(Como vemos en la imagen, el sistema ha detectado en el Bus 001 con identificador 0806:0806 nuestra cámara con el nombre Cámara SunplusIT Inc USB 2.0).

Si nuestra Raspberry Pi ha detectado correctamente nuestra cámara podemos continuar con el siguiente paso, la instalación del Software Motion, para ello ejecutaremos el siguiente comando:

sudo apt-get install motion

Una vez instalado motion vamos a proceder a editar su respectivo archivo de configuración: **sudo nano /etc/motion/motion.conf**. Editaremos el archivo configurándolo tal y cómo se muestra a continuación.

Daemon on

stream localhost off

stream_maxrate 100

Width 640

Height 480

Framerate 100

picture_output off

Esto es lo que significa cada una de estas configuraciones:

Daemon on: Esto activa el modo daemon para el servidor o el software. Un daemon es un proceso en segundo plano que se ejecuta continuamente y maneja las solicitudes de otros programas o clientes.

stream_localhost off: Esto desactiva la opción de restringir la transmisión solo al host local. Si está activado, significaría que solo se puede acceder a la transmisión desde la misma máquina en la que se ejecuta el servidor o el software.

stream_maxrate 100: Esto establece la tasa de bits máxima para la transmisión en 100 kbps (kilobits por segundo). La tasa de bits determina la cantidad de datos que se transmiten por segundo durante la transmisión.

Width 640 y Height 480: estas configuraciones especifican las dimensiones del cuadro de video en píxeles. En este caso, el marco tiene 640 píxeles de ancho y 480 píxeles de alto.

Framerate 100: Esto establece la velocidad de fotogramas del vídeo en 100 fotogramas por segundo. Una velocidad de cuadro más alta significa que el video parece más fluido, pero también requiere más ancho de banda y potencia de procesamiento.

Output_pictures off: Esto desactiva la opción de guardar cuadros individuales del video como imágenes. Si está activado, significaría que el video se guarda como una serie de imágenes en lugar de una transmisión de video continua.

Seguidamente, vamos a configurar la autenticación para aportarle un extra de seguridad a nuestro sistema de videovigilancia. Para ello, y trabajando sobre el mismo archivo editaremos las siguientes líneas estableciéndolas de la siguiente manera.

```
# Set the authentication method (default: 0)
```

```
# 0 = disabled
```

```
# 1 = Basic authentication
```

```
# 2 = MD5 digest (the safer authentication)
```

```
stream_auth_method 2
```

```
# Authentication for the stream. Syntax username: jrommed
```

```
# Default: not defined (Disabled)
```

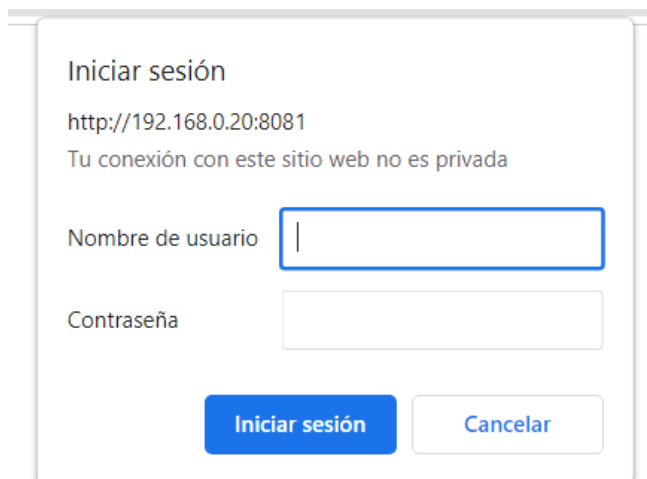
```
stream_authentication administrador:p@ssw0rd2023
```

Por último, iniciaremos el servicio motion con el siguiente comando:

sudo motion -n

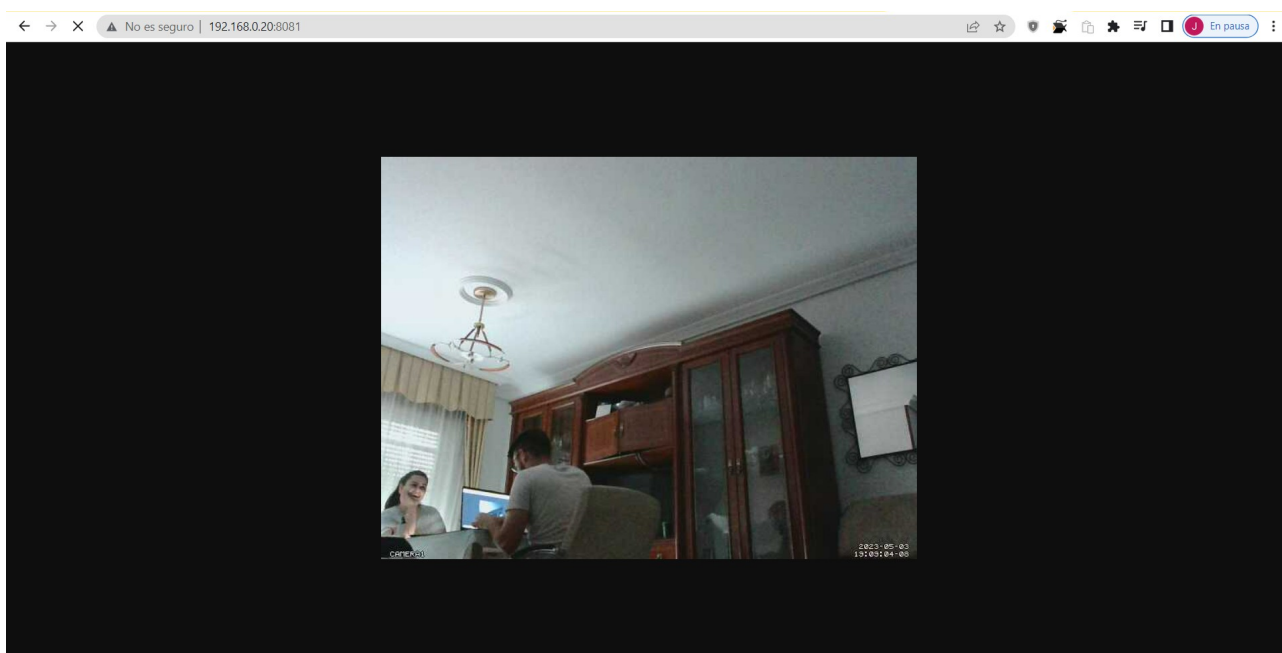
Para comprobar el correcto funcionamiento de nuestra cámara de seguridad nos bastará con acceder a la dirección <http://192.168.0.20:8081/> (Es decir la dirección IP de la Raspberry y el puerto que usa el Software Motion, en nuestro caso el puerto 8081. Dicho puerto puede configurarse a través del archivo de configuración /etc/motion/motion.conf.

Al acceder a la página Web se nos solicitarán nuestras credenciales de acceso al sistema:



A login dialog box titled "Iniciar sesión". It displays the URL "http://192.168.0.20:8081" and a warning "Tu conexión con este sitio web no es privada". Below this are two input fields: "Nombre de usuario" and "Contraseña". At the bottom are two buttons: "Iniciar sesión" (blue) and "Cancelar" (white with blue border).

Una vez introducidas nuestras credenciales de acceso podremos ver el Streaming de nuestro sistema de videovigilancia.



Debido a un problema arbitrario que hace que el servicio motion se congele puntualmente crearemos un script llamado **reiniciarmotion.sh** para usarlo en caso de que lo detectemos. El script contiene las siguientes líneas:

```
proceso=$(ps -aux | grep motion | grep -v grep | head -n 1 | tr -s " " ":" | cut -d ":" -f2)
```

```
if [ ! -z "$proceso" ]; then
```

```
    kill -9 $proceso
```

```
    sleep 10
```

```
fi
```

```
motion -n
```

Este script busca y termina cualquier instancia en ejecución del programa "motion" y luego vuelve a ejecutarlo en primer plano.

Asegúrese de que el archivo tenga permisos de ejecución. Puedes otorgar los permisos obtenidos el siguiente comando:

```
sudo chmod +x reiniciarmotion.sh
```

Ejecuta el script con el siguiente comando:

```
sudo ./reiniciarmotion.sh
```

- Configuración aplicación Android:

Como tarea complementaria vamos a instalar en nuestro dispositivo Android una aplicación que nos permita visualizar la cámara de seguridad sin necesidad de acceder a la misma a través del navegador Web.

Desde el mercado de apps de Android (Play Store) buscaremos la aplicación **"IP Cam Viewer Basic"**.

Añadiremos el tipo de cámara, en nuestro caso seleccionaremos “URL jpeg/mjpeg genérica” y configuraremos los datos referentes estableciéndoles tal y cómo se muestran en la siguiente imagen:

Añadir/Editar cámara IP

Nombre: Camera Raspberry

Crear: Generic URL Find

Modelo: Generic URL (e.g. http://x.com/image.j..)

URL: http://192.168.0.20:8081

Enter the full URL to an image or mjpeg stream (eg. <http://x.com:9000/image.jpg>). Add 'refreshrate' parameter to slow update (eg. <http://x.com/image.jpg?refreshrate=2> for 2 second updates).

User: administrador Pswd:

Mas opciones Cancel Test Guardar

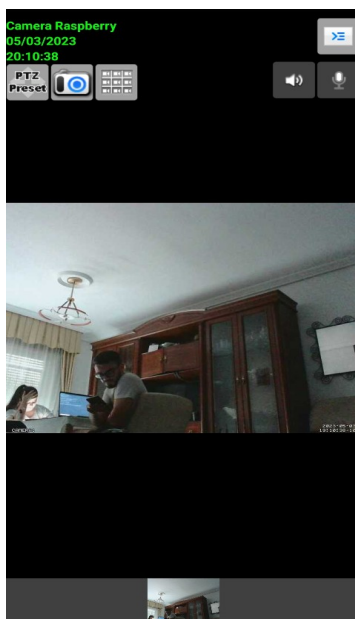
Uso:

Si desea que su cámara/dvr sea visible tanto desde el interior como desde el exterior de su red local, lea el siguiente post:
<http://hit-mob.com/forums/viewtopic.php?f=8&t=47>

Para los DVRs añada una cámara por cada canal. Ponga el número del canal en el campo "Ch.#".

Una vez que todas sus cámaras hayan sido configuradas, es recomendable utilizar la opción "Exportar" para guardar la configuración a la tarjeta SD.

Pulsaremos sobre el botón test para comprobar la conexión con nuestra cámara de videovigilancia, debiendo obtener un mensaje de confirmación de la conexión. Para finalizar si volvemos a la página principal de la aplicación comprobaremos como obtenemos a la perfección la imagen de la cámara de seguridad:



6.2.3 Servicio de monitorización de red

- **Introducción:**

La monitorización de redes consiste en el uso de un sistema que constantemente monitoriza una red de ordenadores buscando componentes lentos o fallidos, notificando al administrador de la red (vía email, teléfono, u otras alarmas) en caso de cortes.

Normalmente las únicas métricas de medición son tiempo de respuesta, disponibilidad y tiempo de funcionamiento, aunque las métricas de consistencia y fiabilidad están empezando a ganar popularidad.

- **Nagios:**

Nagios es una aplicación de software de monitoreo de sistemas informáticos, monitoreo de redes y monitoreo de infraestructuras de código abierto. Está diseñado para monitorear sistemas, redes y servicios y proporcionar alertas en caso de problemas o fallas.

Nagios puede monitorear una amplia gama de dispositivos y servicios, incluidos servidores, conmutadores, enrutadores, aplicaciones y más. Puede comprobar la disponibilidad y la capacidad de respuesta de los servicios de red, supervisar las métricas del sistema, como el uso de la CPU, el espacio en disco y la utilización de la memoria, y realizar comprobaciones personalizadas en función de requisitos específicos.

- **¿Qué características tiene Nagios?**

- **Capacidades de monitoreo:** Nagios le permite monitorear la salud y el rendimiento de varios dispositivos, aplicaciones y servicios. Admite métodos de monitoreo tanto activos como pasivos.
- **Alertas y notificaciones:** cuando Nagios detecta un problema o una interrupción del servicio, puede enviar notificaciones por correo electrónico, SMS u otros métodos a los administradores del sistema u otros contactos designados. Esto permite una respuesta oportuna a los problemas y minimiza el tiempo de inactividad.
- **Interfaz web:** Nagios proporciona una interfaz basada en web llamada Nagios Core, que permite a los administradores configurar y administrar el sistema de monitoreo, ver información de estado, reconocer alertas y realizar otras tareas administrativas.

- **Extensibilidad:** Nagios es altamente extensible y personalizable. Admite el uso de complementos, que son pequeños programas que realizan comprobaciones o acciones específicas. Existe una amplia gama de complementos desarrollados por la comunidad disponibles para monitorear diferentes tipos de dispositivos y servicios.
- **Informes y visualización:** Nagios ofrece capacidades de informes para generar informes históricos y análisis de tendencias. También se puede integrar con herramientas gráficas como Graphite o Grafana para crear representaciones visuales de datos de rendimiento.
- **Monitoreo distribuido:** Nagios admite una arquitectura de monitoreo distribuido, lo que le permite monitorear múltiples sistemas y redes desde una ubicación centralizada. Esto es particularmente útil para implementaciones a gran escala y entornos distribuidos.

En general, Nagios es una poderosa solución de monitoreo que ayuda a los administradores de sistemas y equipos de TI a realizar un seguimiento del estado y el rendimiento de su infraestructura y responder rápidamente a cualquier problema o incidente. Se usa ampliamente en organizaciones pequeñas y grandes para garantizar la disponibilidad y confiabilidad de los sistemas y servicios críticos.

- Preparando tu Raspberry Pi para Nagios

Antes de comenzar, asegúrenos de que nuestro sistema operativo esté completamente actualizado. Para actualizar todo, debemos escribir los siguientes dos comandos en la terminal.

```
sudo apt update
```

```
sudo apt full-upgrade
```

Una vez que su Raspberry Pi haya terminado de actualizarse, ahora podemos instalar los paquetes que usaremos para ejecutar Nagios.

Ejecute el siguiente comando para instalar todos los paquetes que necesitamos.

```
sudo apt install -y autoconf build-essential wget unzip apache2 apache2-utils  
libapache2-mod-php php libgd-dev snmp libnet-snmp-perl gettext libssl-dev  
wget bc gawk dc libmcrypt-dev
```

Este comando instala varios paquetes que necesitamos. Estos paquetes incluyen el compilador que necesitamos para compilar el software de Nagios.

- Descargar y compilar Nagios

Para comenzar, primero vamos a cambiar al directorio /tmp. Este directorio es donde descargaremos, extraeremos y compilaremos el código fuente de Nagios.

```
cd /tmp
```

Ahora podemos descargar el código fuente de Nagios a nuestra Raspberry Pi ejecutando el siguiente comando.

```
sudo wget -O nagios.tar.gz
```

```
https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
```

Este comando se usará para descargar el código fuente de Nagios a nuestro directorio /tmp.

Una vez que el archivo ha terminado de descargarse, podemos extraerlo ejecutando el siguiente comando.

```
tar xzf nagios.tar.gz
```

Ahora cambie al directorio de Nagios y configure el software para la compilación.

```
cd /tmp/nagioscore-nagios-4.4.6/
```

```
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

Pasamos en la ruta a donde queremos que se almacene nuestra configuración de apache2.

Ahora compilemos Nagios ejecutando el siguiente comando.

```
sudo make all
```

Este proceso puede llevar algún tiempo ya que necesita compilar todo el código de Nagios.

- Configuración de Nagios en la Raspberry Pi

Hagamos uso del comando make para crear el usuario y el grupo que Nagios necesita para ejecutarse. También agregaremos el usuario www-data al grupo nagios creado por nuestro comando make.

```
sudo make install-groups-users
```

```
sudo usermod -a -G nagios www-data
```

A continuación, instale los archivos binarios compilados en nuestro sistema operativo utilizando el siguiente comando.

```
sudo make install
```

También podemos usar el comando `make` para instalar el servicio de Nagios y configurarlo para que se inicie en el arranque. Ejecute el siguiente comando para instalar el demonio central de Nagios.

`sudo make install-daemoninit`

Ahora podemos ejecutar el siguiente comando para configurar el directorio de comandos externo.

`sudo make install-commandmode`

Nuestro próximo paso es copiar el archivo de configuración de muestra nuevamente usando el comando `make`.

`sudo make install-config`

Estos archivos de configuración son necesarios para que funcione Nagios. Sin los archivos de configuración, el software no se cargará.

Nuestro penúltimo paso es instalar los archivos de configuración de Apache.

Este comando instalará los archivos de configuración necesarios en el directorio que especificamos cuando configuramos el archivo `MAKE`.

También usaremos dos comandos `a2enmod` para asegurarnos de que los módulos de Apache requeridos estén habilitados.

`sudo make install-webconf`

`sudo a2enmod rewrite`

`sudo a2enmod cgi`

En nuestro paso final, crearemos un usuario de Apache que utilizará para acceder a la interfaz de Nagios en su Raspberry Pi.

El siguiente comando creará un usuario llamado `nagiosadmin`. Se le pedirá que especifique una contraseña para este usuario.

`sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin`

Se debe llamar al usuario `nagiosadmin` para satisfacer la configuración predeterminada de Nagios.

- Inicio de Nagios en Raspberry Pi

Nuestro primer paso es reiniciar el servidor web Apache ejecutando el siguiente comando.

sudo systemctl restart apache2

Reiniciar Apache permitirá cargar nuestros nuevos archivos de configuración.

A continuación, habilite el servicio de Nagios e inícielo ejecutando los siguientes dos comandos.

sudo systemctl enable nagios

sudo systemctl start nagios

Al habilitar el servicio, permitiremos que Nagios se inicie en el arranque de su Raspberry Pi.

Puede verificar que Nagios se haya iniciado en su Raspberry Pi ejecutando el siguiente comando.

sudo systemctl status nagios

Si todo funciona según lo previsto, debería ver el siguiente texto en la terminal.

Active: active (running)

Este texto nos dice que el servicio está activo y se está ejecutando actualmente.

- Instalación de los complementos de Nagios

Para que Nagios funcione correctamente, necesitamos instalar sus complementos. Sin él, el software de Nagios tendrá muy poca funcionalidad.

Primero cambie a nuestro directorio /tmp.

cd /tmp

Ahora que estamos en el directorio /tmp, podemos descargar los complementos de Nagios ejecutando el siguiente comando.

sudo wget -O nagios-plugins.tar.gz <https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.3.3/nagios-plugins-2.3.3.tar.gz>

Ahora extraiga el código fuente del complemento a nuestro directorio actual usando el siguiente comando.

tar xzf nagios-plugins.tar.gz

Nuestro siguiente paso es cambiar a nuestro directorio recién creado y configurar los complementos para la compilación.

```
cd /tmp/nagios-plugins-2.3.3
```

```
sudo ./configure
```

Una vez que se haya completado el proceso de configuración, podemos compilar los complementos de Nagios ejecutando el siguiente comando.

```
sudo make
```

Dependiendo de su Raspberry Pi, este proceso puede llevar algún tiempo. Sin embargo, debería ser mucho más rápido que compilar el código base de Nagios.

Termine este proceso instalando los complementos de Nagios ejecutando el siguiente comando.

```
sudo make install
```

Para asegurarse de que Nagios cargue los nuevos complementos, reinicie el software ejecutando el siguiente comando.

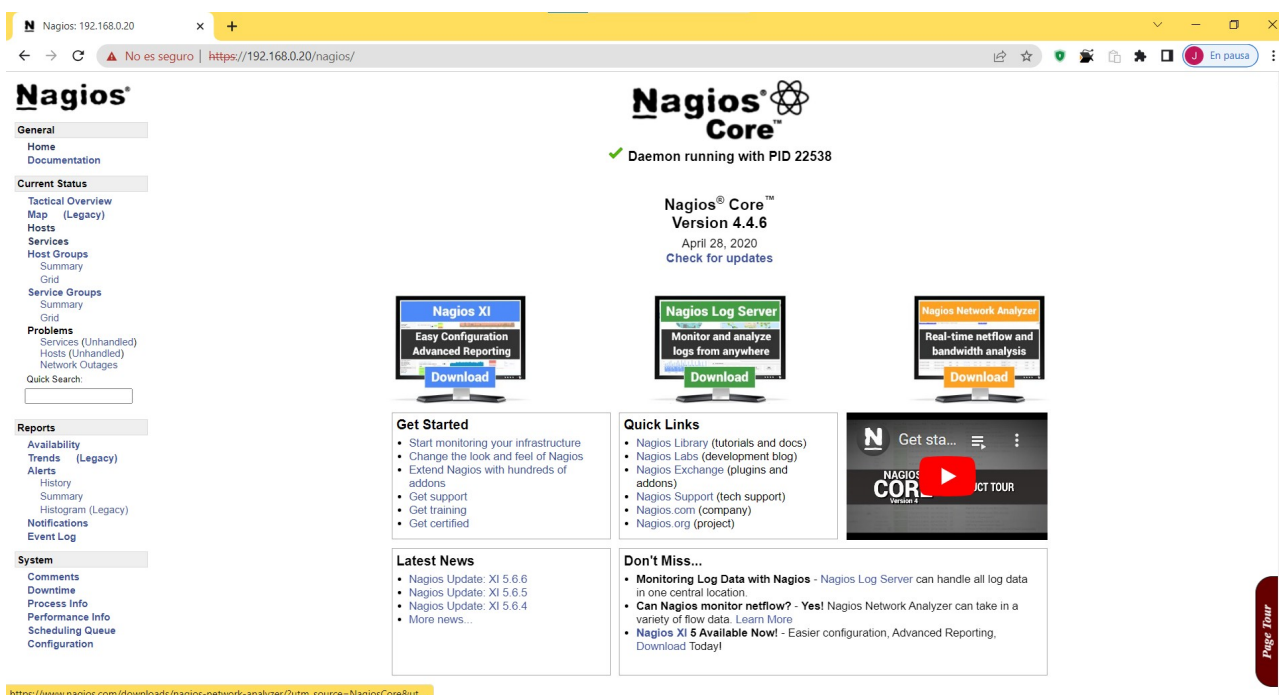
```
sudo systemctl restart nagios
```

- Conexión a la interfaz web de Nagios

Ahora que hemos instalado todo, ahora podemos probar que la interfaz web de Nagios está en línea y funcionando.

Para conectarse a la interfaz web de Nagios, deberá ir a la dirección IP de su Pi seguida de /nagios. Sería así: <http://192.168.0.20/nagios>

Tras una conexión exitosa, debería ser recibido por la página de inicio central de Nagios, todo ejecutándose desde su Raspberry Pi.



- Agregar su host a Nagios

Para ello, necesitaremos crear un archivo de configuración donde definiremos los distintos detalles del dispositivo que queremos monitorizar.

Comencemos cambiando a la carpeta de configuración de "objetos" para Nagios.

Esta carpeta, de forma predeterminada, contiene algunos archivos de configuración de muestra. Estos archivos pueden ser útiles para ver cómo configurar Nagios.

```
cd /usr/local/nagios/etc/objects
```

Ahora vamos a crear un archivo de configuración donde especificaremos la configuración de nuestro host. Para este ejemplo, llamaremos a este archivo pimylifeuphost.cfg.

```
sudo nano host.cfg
```

Dentro de este archivo, necesitamos ingresar las siguientes líneas. Deberá asegurarse de especificar la dirección IP del dispositivo en el que desea realizar las comprobaciones.

```
define host {
    use                linux-server          ; Host group to use
    host_name          usuario                ; Name of this host
    alias              us                     ; Alias
    address            192.168.0.25          ; IP Address
}
```

Esta configuración básica nos permitirá al menos monitorear si el dispositivo está en línea enviándole una solicitud de ping.

A continuación, debemos modificar nuestra configuración de Nagios para que sepa leer nuestro nuevo archivo de configuración. Comience a modificar el archivo de configuración ejecutando el siguiente comando.

sudo nano /usr/local/nagios/etc/nagios.cfg

Dentro de este archivo, busque la siguiente línea y agregue nuestra nueva línea de configuración debajo.

Encontrar

cfg_file=/usr/local/nagios/etc/objects/templates.cfg

Agrega abajo

cfg_file=/usr/local/nagios/etc/objects/host.cfg

Ahora podemos reiniciar el servicio Nagios en nuestra Raspberry Pi usando el siguiente comando.

sudo systemctl restart nagios

Ahora debería poder ver su nuevo host yendo a la página " Hosts " en el panel de control de Nagios.

The screenshot shows the Nagios web interface at the URL <https://192.168.0.20/nagios/>. The interface includes a navigation menu on the left with options like General, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, and Reports. The main content area displays the 'Host Status Totals' and 'Service Status Totals' summaries. The 'Host Status Totals' summary shows 2 Up, 0 Down, 0 Unreachable, and 0 Pending hosts. The 'Service Status Totals' summary shows 8 OK, 0 Warning, 0 Unknown, 0 Critical, and 0 Pending services. Below these summaries, the 'Host Status Details For All Host Groups' table is displayed, showing details for two hosts: 'localhost' and 'usuario'. Both hosts are in the 'UP' status, with the last check performed on 06-11-2023 at 01:18:28. The status information for both hosts indicates 'PING OK - Packet loss = 0%, RTA = 0.24 ms'.

| Host | Status | Last Check | Duration | Status Information |
|-----------|--------|---------------------|---------------|--|
| localhost | UP | 06-11-2023 01:18:28 | 0d 0h 25m 42s | PING OK - Packet loss = 0%, RTA = 0.24 ms |
| usuario | UP | 06-11-2023 01:18:52 | 0d 0h 0m 18s+ | PING OK - Packet loss = 0%, RTA = 56.05 ms |

Actualmente, Nagios solo realizará una simple verificación de ping para ver si los hosts están en línea.

- Agregar un servicio a su host

Para esta guía, configuraremos dos servicios que usarán los comandos `check_http` y `check_ssh` en nuestro host. Ambos comandos se pueden usar para hacer bastante, pero usaremos sus usos más básicos.

Para poder agregar estos nuevos comandos, debemos ir y modificar el archivo de host que creamos anteriormente. Podemos comenzar a modificar este archivo ejecutando el siguiente comando en nuestra Raspberry Pi.

`sudo nano /usr/local/nagios/etc/objects/host.cfg`

Dentro de este archivo, necesitaremos agregar una nueva sección al final que defina nuestros servicios. Cada servicio debe definirse en un bloque separado, por lo que para este tutorial crearemos dos de estos bloques.

```
define service {
    use                local-service
    host_name          usuario
    service_description SSH
    check_command       check_ssh
}

define service {
    use                local-service
    host_name          usuario
    service_description HTTP
    check_command       check_http
}
```

Con estos dos bloques, Nagios verificará automáticamente el estado de los puertos SSH y HTTP que se ejecutan en nuestro host especificado.

Ahora deberíamos verificar que nuestros cambios de configuración sean válidos. Podemos hacerlo ejecutando el siguiente comando. Este comando le indicará al software Nagios que verifique el contenido de los archivos de configuración.

`sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`

Si todo se ha ingresado correctamente, verá "Things look okay" aparecer el texto en la línea de comando.

Ahora podemos reiniciar Nagios para que nuestros cambios se carguen en nuestra Raspberry Pi.

`sudo systemctl restart nagios`

Ahora, cuando abra la interfaz web de Nagios, vaya al menú " Servicios ". Dentro de esta página, podrá ver el estado de sus servicios SSH y HTTP de su nuevo host.

The screenshot shows the Nagios web interface at the URL <https://192.168.0.20/nagios/>. The interface includes a sidebar with navigation links such as General, Current Status, Tactical Overview, Hosts, Services, Host Groups, Service Groups, Problems, Quick Search, Reports, and System. The main content area displays the 'Service Status Details For All Hosts' table, which lists the status of various services for the host 'localhost'.

| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|-----------|-----------------|--------|---------------------|---------------|---------|---|
| localhost | Current Load | OK | 06-11-2023 01:25:58 | 0d 0h 34m 7s | 1/4 | OK - load average: 0.04, 0.02, 0.07 |
| | Current Users | OK | 06-11-2023 01:26:36 | 0d 0h 33m 29s | 1/4 | USERS OK - 3 users currently logged in |
| | HTTP | OK | 06-11-2023 01:27:13 | 0d 0h 32m 52s | 1/4 | HTTP OK: HTTP/1.1 301 Moved Permanently - 516 bytes in 0.002 second response time |
| | PING | OK | 06-11-2023 01:27:55 | 0d 0h 32m 14s | 1/4 | PING OK - Packet loss = 0%, RTT = 0.23 ms |
| | Root Partition | OK | 06-11-2023 01:28:28 | 0d 0h 36m 37s | 1/4 | DISK OK - free space: / 109671 MB (95.31% inode=98%) |
| | SSH | OK | 06-11-2023 01:29:06 | 0d 0h 35m 59s | 1/4 | SSH OK - OpenSSH_8.4p1 Debian-5+deb11u1 (protocol 2.0) |
| | Swap Usage | OK | 06-11-2023 01:29:43 | 0d 0h 35m 22s | 1/4 | SWAP OK - 100% free (99 MB out of 99 MB) |
| | Total Processes | OK | 06-11-2023 01:25:21 | 0d 0h 34m 44s | 1/4 | PROCS OK: 56 processes with STATE = RSZDT |
| | HTTP | OK | 06-11-2023 01:26:18 | 0d 0h 1m 29s+ | 1/4 | HTTP OK: HTTP/1.1 200 OK - 10945 bytes in 0.098 second response time |
| | SSH | OK | 06-11-2023 01:27:18 | 0d 0h 1m 28s+ | 1/4 | SSH OK - OpenSSH_8.9p1 Ubuntu-3ubuntu0.1 (protocol 2.0) |

Con suerte, en esta etapa, ahora tendrá el software Nagios funcionando en su Raspberry Pi.

6.2.4 Servicio de detección de intrusos

- Introducción:

Un servicio de detección de intrusos (IDS, por sus siglas en inglés Intrusion Detection System) es un componente crucial en la seguridad de una red o sistema informático. Su objetivo principal es identificar y responder a actividades sospechosas o maliciosas que podrían dañar la integridad, la confidencialidad o la disponibilidad de los datos y recursos de una organización.

- Fail2ban:

Fail2ban es una aplicación escrita en Python usada para la prevención de intrusos en un sistema, que actúa penalizando o bloqueando las conexiones remotas que intentan accesos por fuerza bruta.

Fail2ban busca en los registros (logs) de los programas que se especifiquen las reglas que el usuario decida para poder aplicar una penalización. La penalización puede ser bloquear la aplicación que ha fallado en un determinado puerto, bloquearla para todos los puertos, etc. Las penalizaciones, así como las reglas, son definidas por el usuario.

Habitualmente, si las IP de ataque se prohíben por un lapso prudencial de tiempo, la sobrecarga de red provocada por los ataques baja y, también se reduce la probabilidad de que un ataque de fuerza bruta basada en diccionarios tenga éxito.

Después de una sucesión de intentos fallidos, Fail2ban (en función de la configuración determinada por el usuario) decidirá la acción a realizar sobre la IP que originó el problema. Puede simplemente notificar por e-mail del suceso, denegar el acceso a la IP atacante, bloquearla en determinados puertos y habilitarla en otros, etc

- Servicios que soporta:

Actualmente Fail2ban establece filtros para Apache, sshd, qmail, vsftpd, lighttpd, Postfix y Courier Mail Server.

- Instalación y configuración de Fail2ban:

Para instalar Fail2ban tan sólo es necesario la ejecución del siguiente comando:

```
sudo apt-get install fail2ban
```

Tras instalarlo vamos a proceder a editar su archivo de configuración para establecer ciertos parámetros.

```
sudo nano /etc/fail2ban/jail.conf
```

Comprobaremos que el valor ignoreip esté establecido tal y cómo se muestra a continuación, para que el tráfico proveniente de la red local no lo banee.

```
ignoreip = 192.168.1.0/24
```

Tal y como se ha mencionado anteriormente, Fail2ban permite monitorizar múltiples servicios, en nuestro caso, y para testear su correcto funcionamiento, configuraremos los valores referentes a ssh estableciéndolos tal y como se muestran a continuación:

```
[ssh]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
```

A continuación, instalaremos los paquetes “ssmtp” y “mailutils”, el software necesario para permitirnos enviar correos electrónicos desde nuestro servidor:

sudo apt-get install ssmtp mailutils

Vamos a editar el archivo /etc/ssmtp/ssmtp.conf , en él estableceremos la configuración de nuestro proveedor de correo electrónico, en mi caso Gmail:

sudo nano /etc/ssmtp/ssmtp.conf

Editaremos la línea “mailhub” estableciéndola tal y cómo se muestra a continuación:

mailhub=smtp.gmail.com:587

Y añadiremos el siguiente código al final del archivo:

Usuario y contraseña de la cuenta Gmail. Usuario con el prefijo "@".

Ej. usuario@ejemplo.com

AuthUser=mensajesraspberry@gmail.com

AuthPass=p@ssw0rd2023

Esto es para usar conexiones STARTTLS

UseSTARTTLS=YES

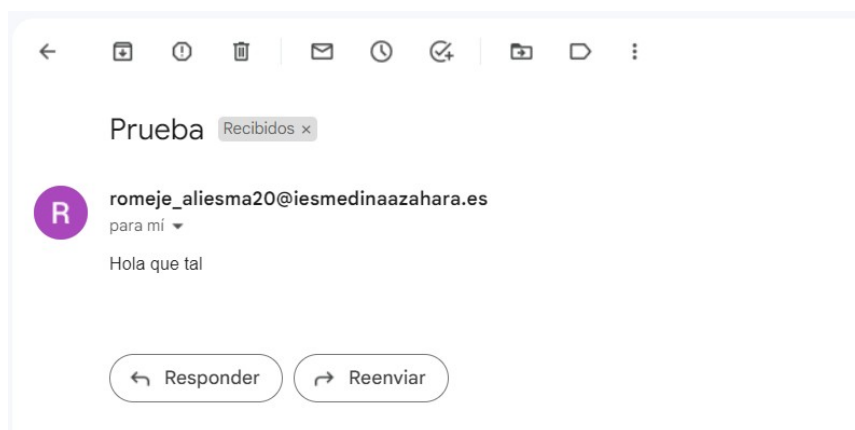
Para que funcione correctamente el envío de mensajes a través de un servidor Gmail, necesitamos habilitar la opción de Google “Permitir que aplicaciones menos seguras accedan a las cuentas”. Para ello accederemos al siguiente link:

<https://myaccount.google.com/lesssecureapps?pli=1>

Y estableceremos en “Sí” la opción “Permitir el acceso de aplicaciones menos seguras”.

A continuación, ejecutaremos el siguiente comando para comprobar si funciona correctamente el envío de un correo electrónico desde nuestro sistema.

```
sudo echo "Hola que tal" | mail -s "Prueba" romeje_aliesma20@iesmedinaazahara.es
```



En el archivo `/etc/fail2ban/jail.conf` modificaremos la siguiente línea por nuestro correo electrónico:

```
destemail = romeje_aliesma20@iesmedinaazahara.es
```

A continuación, buscaremos la siguiente línea:

```
action = %(action_)s
```

Y la sustituiremos por la siguiente:

```
action = %(action_mw)s
```

Para finalizar reiniciaremos el servicio Fail2ban.

```
sudo /etc/init.d/fail2ban restart
```

La aplicación ya estaría configurada para banear a los usuarios que mediante el protocolo SSH realizasen 5 intentos fallidos de conexión.

- Comprobación del funcionamiento:

Vamos a comprobar el correcto funcionamiento de Fail2ban, haremos la comprobación desde la aplicación Android “JuiceSSH”, un poderoso cliente SSH para Android. Configuraremos la nueva conexión con la Raspberry Pi:

← Nueva Conexión ✓

AJUSTES BÁSICOS

Alias: Raspberry Pi

Tipo: SSH

Dirección: 192.168.0.20

Identidad: pi

AJUSTES AVANZADOS

Puerto: 22

Conectar Vía: (Opcional)

Ejecutar Snippet: (Opcional)

Retroceso: Por defecto (envía DEL)

GRUPOS

AÑADIR A GRUPO

Una vez hayamos añadido la nueva conexión procederemos a usarla. La App nos solicitará una contraseña, nosotros la inventaremos para fallar en el proceso de autenticación hasta 5 veces (el parámetro establecido para banear una IP).

Fallo de Autenticación

Por favor introduce la contraseña para pi:

passinventado

☒ Mostrar Contraseña ☐ Recordar Contraseña

CANCELAR ACEPTAR

Tras cinco intentos fallidos de conexión usando la contraseña inventada, el sistema denegará el intento de conexión.

Conexión Fallida

Connection refused

¿Deseas reintentarlo?

NO SÍ

Además, inmediatamente recibiremos un correo electrónico notificando sobre el baneo de la IP implicada. En este caso la IP 192.168.0.X (la de mi dispositivo Android).

La IP ha sido baneada durante el tiempo establecido en el parámetro bantime del archivo /etc/fail2ban/jail.conf.

Comprobar baneo IP:

Si ejecutamos el siguiente comando con la IP de nuestro sistema y devuelve algún valor, nuestra IP habrá sido baneada:

```
sudo iptables -L -n | grep '192.168.0.X'
```

Quitar baneo a una IP específica:

Para eliminar el baneo a una IP específica debemos ejecutar el siguiente comando:

```
sudo fail2ban-client set ssh unbanip 192.168.0.X
```

(Donde la IP 192.168.0.X sería la IP del terminal en el que hemos sido baneados). Por último, reiniciaremos el servicio Fail2ban para que el sistema recoja los cambios realizados:

```
sudo /etc/init.d/fail2ban restart
```

6.3 ¿Cómo instalar y usar No-IP en Raspberry Pi? (DNS Dinámico)

Para finalizar el proyecto, vamos a realizar la externalización de los servicios que componen la infraestructura.

Utilizaremos el servicio <https://www.noip.com/>. No-IP es un proveedor de DNS dinámico que dispone de versión gratuita durante 30 días. Es un sistema que permite redireccionar automáticamente el dominio que hayamos configurado con nuestra IP externa real. Activaremos también el servicio cliente de actualización dinámica, que permitirá a NO-IP redireccionar nuestra IP y modificar el registro DNS en caso de que se produzca algún cambio en la IP externa.

- ¿Qué es No-IP?

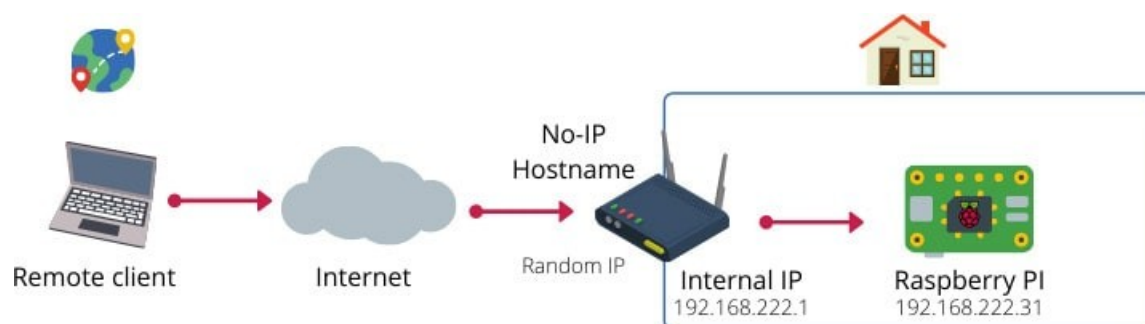
No-IP es un proveedor de servicios de DNS dinámico. La idea es vincular un nombre de dominio (o un subdominio en el plan gratuito) a su dirección IP.

Es particularmente útil si su dirección IP pública cambia regularmente. Le brinda una forma de acceder a su servidor doméstico desde cualquier lugar, incluso si no tiene una dirección IP estática.

- ¿Cómo funciona?

La idea en ese tipo de arquitectura es usar el servicio No-IP para obtener un nombre de host, que siempre redirigirá a su dirección IP pública.

Gracias a este proveedor, puedes alojar cualquier servicio que quieras en casa (en tu Raspberry Pi o no). Si necesitas una imagen para comprender mejor, así es como funciona:



El cliente remoto puede estar en cualquier parte del mundo y acceder al nombre de host creado en No-IP (veremos cómo crear uno justo después de eso).

Su enrutador en casa redirigirá el tráfico a Raspberry Pi o a otro host en la red, según lo que desee hacer.

Para configurar todo, hay 3 pasos principales, que explicaré en este tutorial:

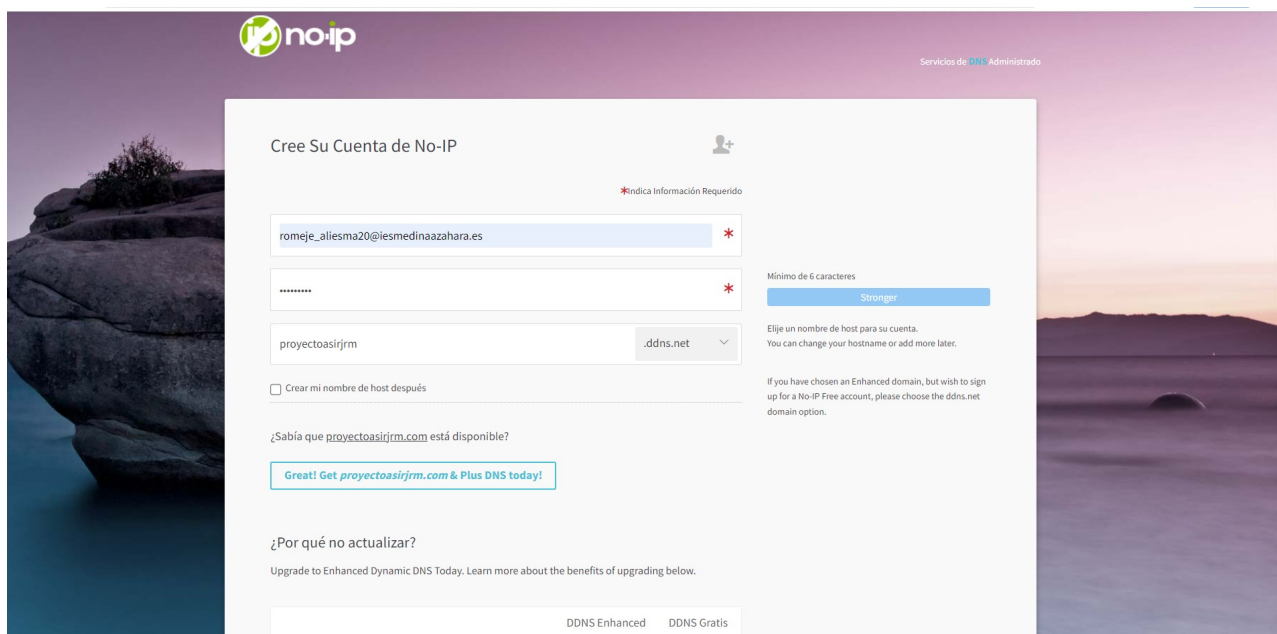
- Crea tu cuenta y elige un nombre de host
- Descargue e instale el cliente No-IP en su Raspberry Pi
- Configure el reenvío de puertos en su enrutador , para redirigir cualquier acceso a su host.

- Creamos una cuenta

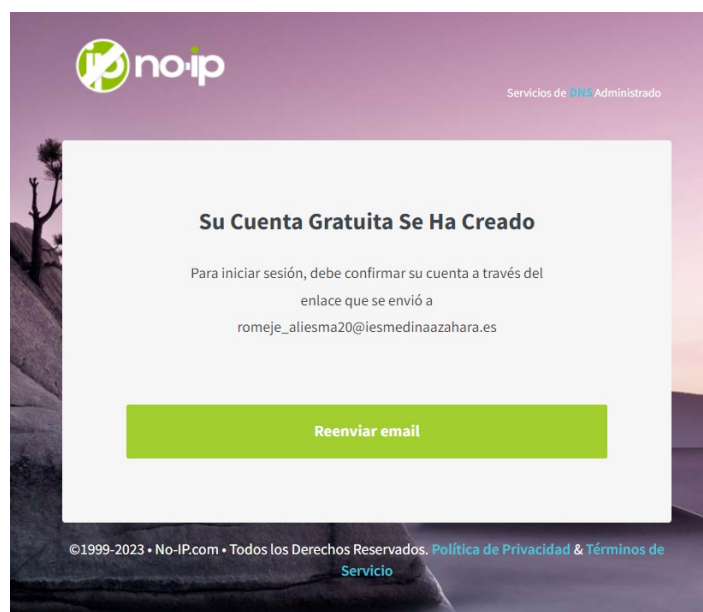
Entonces, sea cual sea su objetivo, el primer paso es crear una cuenta y elegir un nombre de host en NoIP.com:

- Ir al sitio web de No-IP
- Haga clic en "Registrarse" en el menú superior

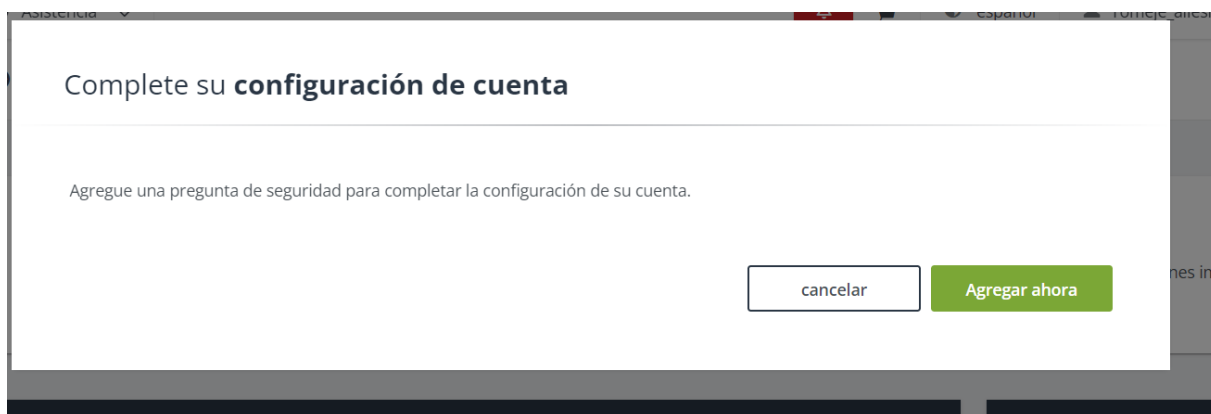
- Complete el formulario con su dirección de correo electrónico, contraseña y elija un nombre de host:



- Es posible que reciba una advertencia si está utilizando caracteres especiales en su contraseña.
- Haga clic en "Registro gratuito" para crear su cuenta.
- Luego, deberá confirmar su dirección de correo electrónico, haciendo clic en el correo electrónico:

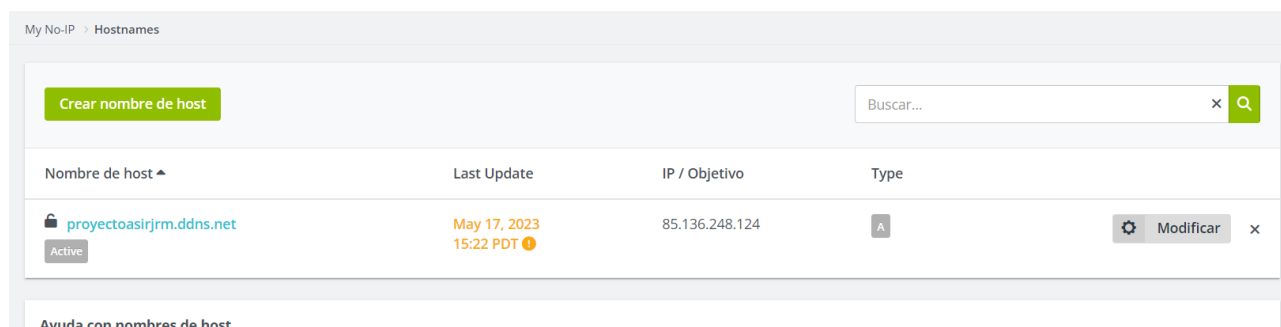


- Haga clic en "Confirmar cuenta".
- Aparece el sitio web, pidiéndole que descargue el cliente. Puede omitir este paso por ahora, ya que probablemente no sea el que necesita para Raspberry Pi. Eso lo veremos más tarde.
- Entonces entra en tu cuenta. Donde, debe establecer un nombre de usuario y elegir una pregunta de seguridad , o no funcionará:



Haga clic en "Agregar ahora" y haga esto de inmediato.

En su cuenta, también puede ver que su nombre de host aún no está actualizado. Sin embargo, puede funcionar, ya que, de forma predeterminada, la dirección IP se establece en la que utilizó para crear su cuenta:



Necesitamos instalar el cliente para sincronizar su dirección IP y eliminar esta advertencia. Hay varias opciones disponibles. Nosotros vamos a instalarlo en una Raspberry Pi.

- Reenvío de puertos en Router:

Para acceder a los servicios desde el exterior debemos configurar nuestro Router, para permitir el acceso a dispositivos de nuestra red local.

Accederemos a nuestro Router a través de la interfaz Web, para ello introduciremos la dirección del mismo en la barra de direcciones del navegador.

Seguidamente, iniciaremos sesión en el sistema mediante las credenciales de acceso.








En la siguiente imagen se muestra el resumen final de la configuración del reenvío de puertos realizado en nuestro Router.

Los puertos que han sido habilitados son los siguientes:

Puerto 8081: Cámara de videovigilancia.

Puerto 80: Nextcloud e Nagios.

Redirección de puertos

| Nombre del servicio | Dirección IP | Protocolo | Puerto LAN | Puerto público | | | |
|---|--------------|-----------|------------|----------------|---|---|---|
| Videovigilancia | 192.168.0.20 | TCP/UDP | 8081 | 8081 |  |  |  |
| NextC/Icing2 | 192.168.0.20 | TCP/UDP | 80 | 80 |  |  |  |
|  | | | | | | | |

- Instalación y configuración del cliente de actualización dinámica NO-IP

A continuación, vamos a instalar y configurar el cliente de actualización dinámica NO-IP, esta acción permitirá poder actualizar nuestra cuenta (y por consiguiente el dominio asociado) con la IP externa de nuestra organización, en caso de que haya cambiado.

Debe elegir una carpeta para descargar y compilar No-IP.

Comenzaremos situándonos en la carpeta “/home/” en la que descargaremos dicho software:

```
cd /home/jrommed
```

Crearemos la siguiente carpeta y nos iremos hacia ella.

```
sudo mkdir noip
```

```
cd noip
```

Descargaremos el cliente de actualización dinámica NO-IP:

```
sudo wget https://www.noip.com/client/linux/noip-duc-linux.tar.gz
```

Descomprimiremos y extraeremos dicho software:

```
sudo tar -zxvf noip-duc-linux.tar.gz
```

Vaya a la nueva carpeta creada: el número de versión en el nombre de la carpeta puede cambiar, así que asegúrese de usar el correspondiente a sus archivos extraídos.

```
cd noip-2.1.9-1
```

Una vez en esta carpeta, use los siguientes comandos para compilar e instalar No-IP:

```
sudo make
```

```
sudo make install
```

Obviamente, debe ingresar su dirección de correo electrónico y contraseña para vincular su dispositivo con su cuenta , pero también la frecuencia de actualización (con qué frecuencia el cliente actualizará su dirección IP en No-IP).

Se parece a esto:

```
jrommed@raspberrypi3:/home/noip/noip-2.1.9-1 $ sudo make install
if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi
if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi
cp noip2 /usr/local/bin/noip2
/usr/local/bin/noip2 -C -c /tmp/no-ip2.conf

Auto configuration for Linux client of no-ip.com.

Please enter the login/email string for no-ip.com romeje_aliesma20@iesmedinaazahara.es
Please enter the password for user 'romeje_aliesma20@iesmedinaazahara.es' *****

Only one host [proyectoasirjrm.ddns.net] is registered to this account.
It will be used.
Please enter an update interval:[30] 30
Do you wish to run something at successful update?[N] (y/N) n

New configuration file '/tmp/no-ip2.conf' created.

mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf
jrommed@raspberrypi3:/home/noip/noip-2.1.9-1 $ |
```

Por cierto, el intervalo de actualización se establece en minutos .Una vez completado, su dirección IP debe actualizarse en el sitio web.

Después de usarlo durante unas horas, noté varios problemas con el cliente No-IP en Raspberry Pi.

Cuando instala No-IP, puede elegir algunas cosas en el asistente, pero si algo no funciona o desea cambiar algo, no puede ejecutar "make install" nuevamente.

Entonces, si necesita actualizar algo (su contraseña o el intervalo de actualización, por ejemplo), aquí hay algunos comandos que puede usar.

El comando principal es:

sudo noip2 <option>

Entonces, las siguientes opciones están disponibles:

- -C: crea una nueva configuración (debes detener No-IP antes de hacer esto)
- -c <ruta>: usa otro archivo de configuración
- -S: muestra la configuración actual
- -U <minutos>: cambiar el intervalo de actualización
- -u <nombre de usuario> -p <contraseña>: actualice las credenciales de su cuenta
- -i <IP>: fuerza una dirección IP específica
- -l <interfaz>: si tiene Wi-Fi y Ethernet habilitados, por ejemplo, puede especificar qué interfaz usar
- -h: muestra todas las opciones (ayuda)

Por ejemplo, quería cambiar el intervalo de actualización de mis pruebas, para ver si funciona correctamente cuando cambio de una conexión a otra.

Entonces, he hecho esto:

sudo noip2 -U 15

Cinco es el mínimo que puede elegir, el valor predeterminado es 30 minutos.

Después de reiniciar, noté que la dirección IP ya no se actualiza. El cliente No-IP no se inicia automáticamente al arrancar. Hay muchas maneras de solucionar esto.

Pero elegí la más fácil:

Abra su configuración crontab:

```
sudo crontab -e
```

Pegue esta línea en él:

```
@reboot /usr/local/bin/noip2
```

Eso es todo, No-IP ahora se iniciará automáticamente en el arranque.

6.4 Configuración SSL nuestro propio certificado autofirmado

Ahora realmente deberíamos trabajar en la configuración de su servidor Raspberry Pi para que se ejecute a través de HTTPS y no de HTTP simple.

Antes de generar el certificado, primero hagamos un directorio para almacenarlo.

```
sudo mkdir /etc/apache2/ssl
```

Después crearemos el certificado, válido para 3 años, con el siguiente comando:

```
sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -out  
/etc/apache2/ssl/server.crt -keyout /etc/apache2/ssl/server.key
```

Crearé la clave del certificado y además nos realizará varias preguntas. Responderemos a ellas con datos ficticios, menos en la parte en la que nos pide el «Common Name» donde tendremos que meter el dominio que tengamos (si tenemos uno) o en mi caso el dominio que me ofrece mi servicio de DNS dinámico. ¡No sirven direcciones IP!

Todavía no reinicializaremos Apache. Ahora deberemos instalar el servicio SSL para Apache:

```
sudo a2enmod ssl
```

Ahora hacemos accesible el sitio ssl creando un enlace simbólico a la carpeta correspondiente:

```
sudo ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-  
enabled/000-default-ssl.conf
```

Y editamos este archivo de configuración ya sea con el editor nano o el que prefieras:

```
sudo nano /etc/apache2/sites-enabled/000-default-ssl.conf
```

Cambiaremos estas dos líneas por encima de la línea `</VirtualHost>`

```
SSLCertificateFile /etc/apache2/ssl/server.crt
```

```
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

Ahora ya si queremos, reiniciamos el servicio de Apache para que tome en cuenta estas modificaciones:

```
sudo service apache2 restart
```

Con eso ya podríamos acceder mediante https a la dirección de nuestro dominio. Al acceder el navegador nos dará una advertencia, ya que el certificado no está firmado por una agencia certificadora (CA), ya que nos lo hemos «autofirmado».

Un paso adicional opcional para garantizar que tenga la mejor seguridad para su configuración de Nextcloud es hacer cumplir SSL para que no se pueda realizar ninguna conexión a través de HTTP, si se realiza una conexión, lo redirigirá a HTTPS.

Podemos hacer esto haciendo algunos cambios en nuestra configuración de apache, para comenzar editamos el archivo predeterminado con el siguiente comando:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Reemplace todo el texto de este archivo con el siguiente código. Básicamente, esto redirigirá todo el tráfico HTTP a su equivalente HTTPS.

```
<VirtualHost *:80>
```

```
    RewriteEngine On
```

```
    RewriteCond %{HTTPS} off
```

```
    RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
```

```
</VirtualHost>
```

Ahora, antes de que esto funcione, debemos habilitar el módulo de redirección y reiniciar apache. Podemos lograr esto fácilmente ejecutando los siguientes dos comandos:

```
sudo a2enmod rewrite
```

```
sudo service apache2 restart
```

Ahora, ir a su Raspberry Pi en HTTP debería redirigir automáticamente a la versión HTTPS.

Para agregar su dominio/IP necesitamos modificar el archivo de configuración de NextCloud, podemos hacerlo ejecutando el siguiente comando:

```
sudo nano /var/www/nextcloud/config/config.php
```

Dentro de este archivo, verá un bloque de texto como el siguiente. Esta es una matriz de todos los dominios confiables a través de los cuales permite que Nextcloud opere.

Por ahora, solo debe incluir la dirección IP local de su Raspberry Pi. Agregaremos nuestro nuevo dominio/IP al final de esta matriz.

Para nuestro ejemplo, agregaremos proyectoasirjrm.ddns.net a la matriz. Esto significa que necesitamos incrementar la ID de la matriz y agregar el nombre de dominio. Una vez que haya agregado uno nuevo, debería verse como a continuación. Repita este procedimiento para cualquier nueva IP o dominio con el que desee que Nextcloud pueda operar.

```
'trusted_domains' =>  
array (  
    0 => '192.168.1.105',  
    1 => 'proyectoasirjrm.ddns.net',  
),
```

7. Estudio presupuestario

A continuación, se detallan los costes de realización del proyecto, no se han incluido los costes asociados al personal informático que desempeña estas tareas.

| Descripción | Precio Unitario | Importe Total |
|---------------------------------|-----------------|-----------------|
| Raspberry Pi 3 model B | 60,00 € | 60,00 € |
| Aukru Caja + Disipador de Claor | 7,50 € | 7,50 € |
| MicroSD 128GB Samsung | 12,00 € | 12,00 € |
| Cámara SunplusIT Inc USB 2.0 | 2,00 € | 2,00 € |
| Mano de obra | 72,00 €/jornada | 72,00 € |
| Total | | 153,50 € |

En este caso el coste de mano de obra conllevaría a 72€/jornada ya que hemos elegido contratar a un personal con un sueldo base de 1600€ al mes y en la zona geográfica de Córdona, Andalucía.

Además de los costes iniciales, también es importante considerar los costes en términos de consumo de energía eléctrica, posibles actualizaciones o mejoras futuras, y cualquier gasto de mantenimiento o sustitución de componentes a lo largo del tiempo.

Recuerda que estos costes son solo estimaciones y pueden variar según tu ubicación geográfica, los proveedores y las opciones que elijas.

8. Conclusiones finales

Es requisito indispensable para la finalización de este curso, la realización de este proyecto. No dudé ni un segundo en usar como elemento principal la Raspberry Pi.

Si bien es cierto que, al comenzar, no conocía en profundidad las grandes posibilidades que ofrece este dispositivo, mis ganas, curiosidad e ilusión me impulsaron a implicarme al 100% en este trabajo.

Siempre surgen dudas... Muy probablemente una de las principales, y que me ha acompañado durante todo este tiempo, ha sido la siguiente: "¿Será suficientemente potente este hardware para soportar lo que quiero implantar?".

A medida que iba haciendo el proyecto, implementando diversos servicios, configurando y programando diversas tareas, me iba dando cuenta de la realidad. La Raspberry Pi es un hardware con un sistema operativo extremadamente optimizado, que gestiona eficientemente sus recursos y que iba a permitir realizar a la perfección todas las ideas que deseaba implementar.

En general, este proyecto ha logrado satisfacer los objetivos propuestos al brindar a los usuarios la seguridad y la autonomía deseadas en sus telecomunicaciones y datos. Al tener el control total sobre la infraestructura y los servicios implementados en la Raspberry Pi, los usuarios pueden proteger su privacidad, reducir la exposición a amenazas cibernéticas y mantener un mayor control sobre su información personal.

8.1 Análisis DAFO

Este es el análisis DAFO que vamos hemos realizado para nuestro proyecto:

- **Fortalezas:**

- **Versatilidad de la Raspberry Pi:** La Raspberry Pi es un dispositivo altamente versátil que permite realizar una amplia gama de proyectos y configuraciones.

- **Comunidad y soporte:** Existe una gran comunidad de usuarios de Raspberry Pi y abundante documentación en línea, lo que facilita la resolución de problemas y la obtención de ayuda.
- **Configuración personalizada:** Puedes adaptar la configuración del sistema y los servicios según tus necesidades y preferencias.
- **Control sobre las comunicaciones y los datos:** La instalación de estos servicios informáticos en la Raspberry Pi proporciona a los usuarios un mayor control y autonomía sobre sus comunicaciones y datos.

- Debilidades:

- **Limitaciones de hardware:** La Raspberry Pi tiene recursos limitados en comparación con computadoras de escritorio o servidores más potentes, lo que puede afectar el rendimiento en proyectos más exigentes.
- **Conexión a Internet requerida:** Muchos de los servicios propuestos en el proyecto requieren una conexión a Internet estable para funcionar correctamente.
- **Curva de aprendizaje:** Si eres nuevo en el uso de Raspberry Pi y la configuración de servicios, es posible que enfrentes una curva de aprendizaje inicial para familiarizarte con los conceptos y configuraciones necesarios.

- Oportunidades:

- **Aprendizaje y experiencia:** Este proyecto te brinda la oportunidad de aprender sobre la configuración y administración de servicios en una Raspberry Pi, lo que te ayudará a desarrollar habilidades técnicas y experiencia en proyectos de IoT y servidores caseros.
- **Personalización:** Puedes personalizar cada servicio según tus necesidades específicas, permitiéndote adaptarlos a tus requerimientos particulares.
- **Potencial de expansión:** A medida que adquieras experiencia y conocimientos, puedes expandir este proyecto y explorar otros servicios y configuraciones en tu Raspberry Pi.
- **Reducción de costos:** La Raspberry Pi es una solución de hardware económica y de bajo consumo de energía, lo que puede ayudar a reducir los costos asociados con la infraestructura tecnológica.

- Amenazas:

- **Incompatibilidad de hardware o software:** Al trabajar con una amplia gama de servicios y software, puede haber casos de incompatibilidad entre diferentes componentes o versiones, lo que podría generar desafíos adicionales.
- **Estabilidad del sistema:** Dependiendo de la configuración y los servicios utilizados, podría haber desafíos en cuanto a la estabilidad y el rendimiento del sistema, especialmente al ejecutar múltiples servicios simultáneamente.
- **Amenazas cibernéticas en constante evolución:** Las amenazas cibernéticas están en constante evolución y se vuelven cada vez más sofisticadas. Existe la posibilidad de que los servicios instalados puedan ser vulnerables a nuevos ataques y técnicas de intrusión, lo que podría comprometer la seguridad y la protección de los datos.

8.2 Posibles mejoras

Existen varias mejoras posibles que podríamos considerar para ampliar y mejorar nuestro proyecto:

- **Integrar servicios adicionales:** Explora otros servicios y aplicaciones que se pueden ejecutar en la Raspberry Pi, como un servidor VPN para tener ese control en la red.
- **Seguridad reforzada:** Implementa medidas adicionales de seguridad para proteger tu Raspberry Pi y los servicios configurados. Esto puede incluir la configuración de cortafuegos, la implementación de autenticación de dos factores, el uso de certificados SSL para conexiones seguras, entre otras medidas.
- **Automatización y control:** Utiliza la Raspberry Pi para automatizar tareas y controlar dispositivos.
- **Integración de un servidor de correo electrónico** con el servicio de almacenamiento en la nube (NextCloud).
- **Subir un repositorio a GitHub** para poder compartir todo este proyecto con las empresas que estén interesadas en él.

9. Anexos

[- Anexo sobre la Raspberry Pi y los diferentes modelos que hay.](#)

10. Bibliografía y referencias

Sitio web: Raspberry Pi Forums URL: <https://www.raspberrypi.org/forums/>

Subreddit: Raspberry Pi URL: https://www.reddit.com/r/raspberry_pi/

Artículo: "Cómo instalar No-IP en Raspberry Pi" Sitio web: Pi My Life Up URL: <https://raspberrytips.com/install-no-ip-raspberry-pi/>

Artículo: "Raspberry Pi Nextcloud Server: tu propio almacenamiento en la nube" Sitio web: Pi My Life Up URL: <https://pimylifeup.com/raspberry-pi-nextcloud-server/>

Artículo: "17 Consejos de seguridad para la Raspberry Pi" Sitio web: Raspberry Tips URL: <https://raspberrytips.es/17-consejos-seguridad-para-la-raspberry-pi/>

Artículo: "Cómo instalar Nextcloud en Raspberry Pi" Sitio web: Raspberry Tips URL: <https://raspberrytips.es/instalar-nextcloud-raspberry-pi/>

Artículo: "Cómo instalar el último PHP en Raspberry Pi" Sitio web: Pi My Life Up URL: <https://pimylifeup.com/raspberry-pi-latest-php/>

Artículo: "Nagios en Raspberry Pi: una guía completa" Sitio web: Raspberry Tips URL: <https://raspberrytips.com/nagios-raspberry-pi/>

Artículo: "Cómo instalar Nagios en Raspberry Pi" Sitio web: Pi My Life Up URL: <https://pimylifeup.com/raspberry-pi-nagios/>

Artículo: "Sistema de videovigilancia con Motion en Raspberry Pi" Sitio web: Geekland URL: <https://geekland.eu/sistema-de-videovigilancia-motion/>

Artículo: "Configurar un certificado SSL autofirmado para mi Raspberry Pi con Raspbian" Sitio web: Victorhck in the Free World URL: [https://victorhckinthefreeworld.com/2018/07/25/configurar-un-certificado-ssl-autofirmado-para -mi-raspberry-pi-con-raspbian/](https://victorhckinthefreeworld.com/2018/07/25/configurar-un-certificado-ssl-autofirmado-para-mi-raspberry-pi-con-raspbian/)

Artículo: "Cómo configurar Let's Encrypt SSL en Raspberry Pi" Sitio web: Pi My Life Up URL: <https://pimylifeup.com/raspberry-pi-ssl-lets-encrypt/>

Artículo: "Cómo configurar Fail2Ban en Raspberry Pi" Sitio web: Pi My Life Up URL: <https://pimylifeup.com/raspberry-pi-fail2ban/>