

TRABAJO FIN DE GRADO SUPERIOR



CESUR SEVILLA CARTUJA

TORRENFLIX

**ADMINISTRACIÓN DE
SISTEMAS INFORMÁTICOS EN
RED**

PRESENTA

Fernando García Ayoso

TUTOR

Francisco Javier Fernández Ruiz

CURSO

2022/2023

Dedicatoria:

A mis padres por enseñarme los valores más importantes en la vida y por vuestra dedicación, amor y confianza.

A mis profesores, por mostrarme el camino hacia la excelencia académica y profesional. Vuestra experiencia, conocimiento y pasión por la informática han sido un verdadero ejemplo a seguir.

Y por último a Marta, mi fuente motivación y apoyo incondicional. Gracias por estar siempre. Eres lo mejor de mi vida.

Índice

1. Abstract	7
2. Justificación del proyecto	7
3. Introducción.....	8
4. Objetivos.....	9
5. Herramientas a utilizar	11
5.1 Herramientas Hardware	11
5.2 Herramientas Software	12
6. Desarrollo	19
6.1 - Instalar Raspberry Pi 4.....	19
6.2 - Configurar Raspberry Pi.....	26
6.3 - Instalar servidor de Torrent y configuración	35
6.4 - Controlar el usuario de Torrent	38
6.5 - Comprobar el funcionamiento del servicio	41
6.6 - Samba	43
6.7 - Visualizar en el resto de dispositivos.....	48
6.8 - Configurar red externa (DNS)	52
6.9 - Configurar Jellyfin.....	57
6.10 - Crear usuarios en Jellyfin	63
6.11 - Netdata	67
6.11.1 – Error Netdata.....	71
7. Conclusión	73
8. Bibliografía.....	74
9. Anexo	76

Índice de imágenes

Imagen 1: Herramientas hardware.....	11
Imagen 2; Raspberry Pi Installer	12
Imagen 3; Fing	13
Imagen 4; Docker	14
Imagen 5; Duck DNS.....	14
Imagen 6; Samba	15
Imagen 7; Torrent	16
Imagen 8; dlna	17
Imagen 9; Jellyfin	17
Imagen 10; Netdata	18
Imagen 11; micro-SD.....	19
Imagen 12; Raspberry Pi	19
Imagen 15; Sistema operativo	20
Imagen 16; Sistema operativo 32 bits.....	20
Imagen 13; Instalador Raspian	20
Imagen 14; Raspberry Pi Installer (2).....	20
Imagen 17; Elegir tarjeta microsd.....	21
Imagen 18; Ruta tarjeta	21
Imagen 19; Escribir instalador en la tarjeta.....	21
Imagen 20; Confirmar instalador en la tarjeta	22
Imagen 21; Proceso de instalación	22
Imagen 22; Proceso finalizado.....	22
Imagen 23; bootfs	23
Imagen 24; Archivo SSH	23
Imagen 25; Cable Ethernet.....	24
Imagen 26; Fing (2)	24
Imagen 27; IP Raspberry Pi.....	25
Imagen 28; Símbolo del sistema.....	25
Imagen 29; Conexión SSH exitosa.....	26
Imagen 30; Configuración SO Raspberry Pi.....	26
Imagen 31; Hostname	27
Imagen 32; Nombre hostname.....	27
Imagen 33; Configurar el idioma y la región	27
Imagen 34; Locale	27
Imagen 35; es_ES.UTF-8 UTF-8	28
Imagen 36; Timezone	28
Imagen 37; Timezone (2).....	28
Imagen 38; Wlan Country.....	28
Imagen 39; Country	29
Imagen 40; Finaliza la configuración del SO	29

Imagen 41; Cambio de contraseña	29
Imagen 42; Actualizar el sistema	30
Imagen 43; Reiniciar el sistema	30
Imagen 44; Reiniciar el sistema (2)	30
Imagen 45; Inicio del sistema	30
Imagen 46; Dispositivos conectados	31
Imagen 47; UUID ext4	31
Imagen 48; cd /mnt	32
Imagen 49; carpeta disco duro	32
Imagen 50; Archivo ftap	33
Imagen 51; Comprobación discos	33
Imagen 52; Reiniciamos el sistema	34
Imagen 53; Comprobación discos (2)	34
Imagen 54; Contenido en el interior del disco	34
Imagen 55; Carpeta Descargas	34
Imagen 56; Carpeta temp	35
Imagen 57; Instalar servidor torrent	35
Imagen 58; Detener el servicio	36
Imagen 60; Archivo de configuración	36
Imagen 59; Editar archivo de configuración	36
Imagen 61; Download-dir	37
Imagen 62; Incomplete-dir	37
Imagen 63; Incomplete-dir-enabled	37
Imagen 64; rcp-username	37
Imagen 65; rcp-password	37
Imagen 66; rcp-port	37
Imagen 67; rpc-whitelist-enabled	38
Imagen 68; Editar archivo transmission-daemon	38
Imagen 69; Archivo transmission-daemon	38
Imagen 70; transmission-daemon: user	39
Imagen 71; Editar archivo transmission-daemon.service	39
Imagen 72; Archivo transmission-daemon.service	39
Imagen 73; transmission-daemon.service: user	39
Imagen 74; Propietario transmission-daemon	40
Imagen 75; settings.json	40
Imagen 76; Propietario /home/pi/.config/transmission-daemon/	41
Imagen 77; IP:Puerto	41
Imagen 78; Torrent	42
Imagen 79; Elegir archivos	42
Imagen 80; Archivo subiéndose	42
Imagen 81; Archivo en /temp	43
Imagen 82; Instalar samba	43
Imagen 83; Detener configuración samba-ad-dc	43
Imagen 84; Archivo configuración samba-ad-dc	44
Imagen 85; Archivo samba-ad-dc	44

Imagen 86; Caracteres a añadir en samba-ad-dc	45
Imagen 87; Comando smbpasswd –a pi.....	45
Imagen 88; Activar el servicio.....	46
Imagen 90; \\IPraspberrypi\nombrecarpetacompartidasamba.....	46
Imagen 89; Conectar unidad de red	46
Imagen 91; Credenciales	47
Imagen 92; Carpeta.....	47
Imagen 93; Película.....	47
Imagen 94; Película (2).....	48
Imagen 95; Instalar minidlna	48
Imagen 96; Detener servicio minidlna	49
Imagen 97; Editar archivo configuración minidlna	49
Imagen 98; "user"	49
Imagen 99; "media_dir"	50
Imagen 100; "friendly_name"	50
Imagen 101; "inotify=yes"	50
Imagen 102; Iniciar servicio minidlna	50
Imagen 103; Router	52
Imagen 104; Puertos router	52
Imagen 105; Duckdns	53
Imagen 106; Instalar duckdns.....	53
Imagen 107; Instalar duckdns (2)	53
Imagen 108; Inicio de sesión	54
Imagen 109; Inicio de sesión correcto	54
Imagen 110; Conectar mediante SSH - duckdns.....	55
Imagen 111; Actualizar el sistema	57
Imagen 112; Instalar Docker.....	57
Imagen 113; Estado de Docker	58
Imagen 114; Crear archivo docker-compose.....	58
Imagen 115; Docker ps.....	60
Imagen 116; Inicio de sesión en Jellyfin	60
Imagen 117; Películas Jellyfin	61
Imagen 118; Peliculas Torrent	61
Imagen 119; Puerto 8096.....	62
Imagen 120; Comprobación película Jellyfin.....	62
Imagen 121; Descargar película.....	62
Imagen 122; Descargando película	62
Imagen 123; Película descargada	63
Imagen 124; Panel de control Jellyfin	63
Imagen 125; Usuarios Jellyfin	64
Imagen 126; Crear "usuario_1"	64
Imagen 127; NO permitir al usuario administrar el servidor.....	65
Imagen 128; Visualizar usuarios creados	65
Imagen 129; Opciones usuario_1	66
Imagen 130; Descarga usuario_1.....	66

Imagen 131; Visualizar película usuario_1	66
Imagen 132; Instalar Netdata	67
Imagen 133; Puertos Netdata	68
Imagen 134; Habilitar puerto Netdata	68
Imagen 135; Netdata interfaz.....	68
Imagen 136; Reproducción pelicula Netdata	69
Imagen 137; Comprobación subir película Netdata	70
Imagen 138; Comprobación subir película Netdata (2)	70
Imagen 139; Error Netdata.....	71

Índice de tablas

Tabla 1; Herramientas hardware	11
Tabla 2; SSH pi@IP	25
Tabla 3; Configurar Raspberry Pi	26
Tabla 4; Actualizar el sistema	30
Tabla 5; Dispositivos conectados	31
Tabla 6; formatear pendrive en ext4	31
Tabla 7; Editar archivo ftap	32
Tabla 8; Reinicio del sistema	33
Tabla 9; Instalar servidor torrent	35
Tabla 10; Detener el servicio	35
Tabla 11; Editar archivo de configuración	36
Tabla 12; Editar archivo transmission-daemon.service	39
Tabla 13; Actualizar daemon	40
Tabla 14; Propietario transmission-daemon	40
Tabla 15; Propietario /home/pi/.config/transmission-daemon/	40
Tabla 16; Instalar samba	43
Tabla 17; Detener configuración samba-ad-dc	43
Tabla 18; Archivo configuración samba-ad-dc	44
Tabla 19; Caracteres a añadir en samba-ad-dc	45
Tabla 20; Añadir el usuario pi a samba	45
Tabla 21; Activar el servicio	46
Tabla 22; Instalar minidlna	48
Tabla 23; Detener servicio minidlna	48
Tabla 24; Editar archivo configuración minidlna	49
Tabla 25; Iniciar servicio minidlna	50
Tabla 26; Reiniciar servicio minidlna	50
Tabla 27; Reiniciar servicio minidlna	50
Tabla 28; Instalar duckdns	54
Tabla 29; Instalar duckdns (3)	54
Tabla 30; Actualizar el sistema	57
Tabla 31; Insalar Docker	57
Tabla 32; Estado de Docker	57
Tabla 33; Crear archivo docker-compose	58
Tabla 34; Docker-compose	59
Tabla 35; Docker-compose up -d	60
Tabla 36; Netdata	67
Tabla 37; Puertos Netdata	67
Tabla 38; Comando error Netdata	71
Tabla 39; Comando error Netdata (2)	72

1. Abstract

Este proyecto se enfoca en la implementación de una Raspberry Pi como servidor NAS con torrent y Samba, y la configuración de un servidor Jellyfin con Docker. Además, se ha creado un DNS personalizado (fernandoayoso.duckdns.org) para acceder al contenido multimedia desde cualquier dispositivo conectado a internet.

La finalidad del proyecto es mejorar la experiencia del usuario en el acceso al contenido multimedia. Para ello, se han definido algunos objetivos específicos como la configuración del servidor NAS con Samba, la configuración del servidor torrent y Jellyfin con Docker, y la sincronización automática de las descargas de torrent con Jellyfin, además, se ha utilizado la herramienta Netdata, en la que se puede monitorear el rendimiento del sistema y, por tanto, se ha podido observar el estado de la Raspberry Pi en las subidas, descargas y cambios realizados.

La seguridad es otro aspecto fundamental en este proyecto, por lo que se han implementado medidas de seguridad para proteger la Raspberry Pi y el contenido multimedia almacenado en ella mediante un uso de credenciales.

Se ha proporcionado una solución tecnológica accesible e innovadora para tener acceso centralizado y eficiente al contenido multimedia desde cualquier dispositivo conectado a la red local, mejorando la experiencia del usuario y garantizando la seguridad del contenido almacenado en la Raspberry Pi.

2. Justificación del proyecto

La justificación principal para realizar este proyecto de configurar una Raspberry Pi como servidor NAS con torrent y samba, y también como servidor Jellyfin con Docker, es la necesidad de tener un acceso sencillo y centralizado a una gran cantidad de contenido multimedia en diferentes dispositivos.

Con la configuración de un servidor NAS, se puede almacenar todo el contenido multimedia, como películas y música, en un solo lugar accesible desde cualquier dispositivo conectado a la red local, lo que permite una mejor organización y administración de los archivos. Además, con la configuración del servidor Jellyfin, se puede compartir el contenido multimedia en diferentes dispositivos sin necesidad de copiar el contenido a cada dispositivo individualmente, ahorrando espacio en el dispositivo y mejorando la experiencia de usuario.

El uso de torrent como fuente de contenido multimedia también es beneficioso, ya que permite la descarga rápida de archivos grandes sin afectar demasiado el ancho de banda de la red. La capacidad de sincronizar automáticamente las descargas de torrent con Jellyfin

también es útil, ya que permite que todo el contenido esté disponible sin la necesidad de realizar transferencias manuales de archivos.

Además, la integración de Netdata en este proyecto también es importante, ya que proporciona una herramienta de monitoreo y análisis de rendimiento en tiempo real para la Raspberry Pi y los servicios que se ejecutan en ella. Esto permite conocer el uso de recursos y la salud de la Raspberry Pi, lo que puede ayudar en la identificación de problemas y en la optimización del rendimiento del servidor. La incorporación de Netdata en este proyecto también refuerza la seguridad, ya que puede detectar posibles amenazas o comportamientos anormales en el sistema.

En resumen, este servidor permite una mejor administración y acceso a una gran cantidad de contenido multimedia en diferentes dispositivos, lo que mejora la experiencia de usuario y facilita la vida cotidiana en el consumo de contenido. A su vez, resulta una ventaja competitiva ya que cualquier usuario autorizado que quiera acceder al contenido, podrá realizarlo y de manera gratuita.

3. Introducción

La tecnología ha avanzado de manera exponencial en los últimos años, lo que ha permitido el desarrollo de soluciones tecnológicas innovadoras y accesibles para cualquier persona interesada en aprovecharlas. Uno de estos avances es la Raspberry Pi, una pequeña placa que funciona como un ordenador completo, lo que la hace muy versátil y útil en diversos proyectos.

En este sentido, uno de los proyectos que se han vuelto cada vez más populares es la implementación de una Raspberry Pi como servidor NAS con torrent y samba, lo que permite tener un acceso centralizado y eficiente al contenido multimedia, incluyendo películas, series y música, desde cualquier dispositivo conectado a la red. Además, con la configuración de un servidor Jellyfin con Docker, se puede sincronizar automáticamente todas las descargas de torrent con Jellyfin, lo que permite tener todo el contenido multimedia disponible en un solo lugar, mejorando la experiencia del usuario. Todo esto se puede observar mediante el uso de Netdata, una herramienta que permite el monitoreo del rendimiento del sistema de la Raspberry Pi, incluyendo el uso de CPU, memoria y almacenamiento en tiempo real.

En este proyecto, se ha implementado una Raspberry Pi como servidor NAS con torrent y samba, y se ha configurado un servidor Jellyfin con Docker, lo que ha permitido tener acceso a todo el contenido multimedia de manera eficiente y centralizada. Además, se ha creado un DNS personalizado, fernandoayoso.duckdns.org, para tener un acceso más fácil y seguro al contenido multimedia desde cualquier dispositivo conectado a internet.

Este proyecto es una muestra del potencial de la tecnología y cómo se puede aprovechar para mejorar la experiencia del usuario en el acceso al contenido multimedia.

4. Objetivos

Objetivo general:

- Implementar una Raspberry Pi como servidor NAS con torrent y samba, y configurar un servidor Jellyfin con Docker y torrent, monitoreado con la herramienta Netdata, para tener un acceso centralizado y eficiente al contenido multimedia desde cualquier dispositivo conectado a la red.

Objetivos específicos:

- Asegurar la seguridad de la Raspberry Pi y del contenido multimedia almacenado en ella.
 - Cambiar la contraseña predeterminada de la Raspberry Pi.
 - Configurar un firewall para limitar el acceso a la Raspberry Pi desde la red.
 - Configurar el acceso SSH para que solo sea posible desde dispositivos autorizados.
 - Mantener la Raspberry Pi y las aplicaciones instaladas actualizadas para corregir posibles vulnerabilidades de seguridad.
- Configurar y mantener un servidor NAS con Samba para el acceso a archivos compartidos.
 - Instalar y configurar Samba en la Raspberry Pi.
 - Configurar las carpetas compartidas para el acceso desde cualquier dispositivo en la red.
 - Mantener el servidor NAS actualizado y funcionando de manera eficiente.
- Configurar un servidor torrent para descargar contenido multimedia.
 - Instalar y configurar un servidor torrent en la Raspberry Pi.
 - Configurar las descargas para que se realicen automáticamente en una carpeta compartida.
 - Configurar el servidor para limitar el uso del ancho de banda y optimizar la velocidad de descarga.
- Configurar un servidor Jellyfin con Docker para la gestión y sincronización del contenido multimedia.
 - Instalar y configurar Docker en la Raspberry Pi.
 - Descargar e instalar Jellyfin en Docker.
 - Configurar la sincronización automática de contenido multimedia desde la carpeta compartida de descargas torrent.

- Sincronizar automáticamente las descargas de torrent con Jellyfin.
 - Configurar el servidor Jellyfin para que detecte automáticamente los archivos multimedia descargados.
 - Configurar la sincronización automática de contenido multimedia desde la carpeta compartida de descargas torrent.
- Configurar un DNS personalizado para acceder al contenido multimedia desde cualquier dispositivo conectado a internet.
 - Configurar un servicio de DNS dinámico, como DuckDNS.
 - Configurar el router para redirigir el tráfico de internet al servidor Raspberry Pi.
 - Configurar el DNS personalizado para que apunte al servidor Raspberry Pi.
- Configurar una herramienta que permite el monitoreo del rendimiento del sistema
 - Configurar Netdata.
 - Observar cambios en la CPU de la Raspberry Pi a la hora de subir o reproducir películas.
 - Observar cambios en el almacenamiento de la Raspberry Pi a la hora de subir o reproducir películas.

5. Herramientas a utilizar

5.1 Herramientas Hardware

Dispositivo	Características	Precio aprox.	URL
Raspberry Pi 4	model B; 4 GB RAM	185,00 €	link
Cable Ethernet	cat 8; 0.5 M	4,00 €	link
Cable USB C	Cable USB C 3.1 A	4,00€	link
Tarjeta micro-SD 32 GB	Tarjeta micro-SD 32 GB	5,00€	link
Lector de tarjetas USB	Lector de tarjetas USB tarjeta SD/micro-SD	8,00€	link
Pendrive 512 GB	Memoria Flash USB 3.0 SanDisk Ultra de 512 GB	50,00€	link

Tabla 1; Herramientas hardware



Imagen 1: Herramientas hardware

5.2 Herramientas Software

- Raspberry Pi Installer

La herramienta [Raspberry Pi Installer](#) es una aplicación que permite instalar y configurar el sistema operativo Raspbian en una tarjeta de memoria SD para ser utilizada en una Raspberry Pi. La Raspberry Pi es un miniordenador de bajo costo y alto rendimiento que se utiliza para una variedad de proyectos, como la automatización del hogar, la robótica, el Internet de las cosas, la educación y el desarrollo de software.

La herramienta Raspberry Pi installer es fácil de usar y está disponible para sistemas operativos Windows, macOS y Linux. Permite descargar la imagen de Raspbian desde la página oficial de Raspberry Pi y escribir la imagen en la tarjeta de memoria SD. También permite configurar la red inalámbrica, definir el tamaño de la partición y configurar la Raspberry Pi para arrancar desde la tarjeta SD.



Imagen 2; Raspberry Pi Installer

- Fing

[Fing](#) es una aplicación de software para dispositivos móviles y de escritorio que permite escanear y analizar redes de computadoras. Fing puede utilizarse para descubrir y detectar dispositivos conectados a una red, así como para identificar posibles problemas de seguridad y rendimiento.

Con Fing, es posible realizar un escaneo completo de la red para identificar todos los dispositivos conectados, así como sus direcciones IP, nombres de host y otros detalles relevantes. También puede mostrar información detallada sobre cada dispositivo, como el tipo de dispositivo, la marca y modelo, la dirección MAC, el fabricante, el sistema operativo y el tiempo de actividad.

Fing también puede utilizarse para realizar pruebas de velocidad de la red y para identificar problemas de conectividad y rendimiento, como dispositivos que estén consumiendo demasiado ancho de banda o interferencias en la señal inalámbrica.



Imagen 3; Fing

- Docker

[Docker](#) es una herramienta de software que permite crear, implementar y ejecutar aplicaciones en contenedores virtuales. Los contenedores son entornos de ejecución aislados y autónomos que incluyen todo lo necesario para que una aplicación funcione, incluyendo el código, las bibliotecas y las dependencias.

La principal ventaja de Docker es que permite que las aplicaciones se ejecuten de manera consistente y portátil en diferentes entornos, como servidores locales, nubes públicas o privadas, y dispositivos IoT. Además, Docker simplifica el proceso de implementación y actualización de las aplicaciones, ya que los contenedores son autónomos y se pueden implementar y escalar de manera rápida y eficiente.

Docker utiliza una arquitectura cliente-servidor, donde el cliente es una interfaz de línea de comandos o una aplicación gráfica que se comunica con un daemon de Docker que se ejecuta en el host. El daemon se encarga de crear, gestionar y ejecutar los contenedores.

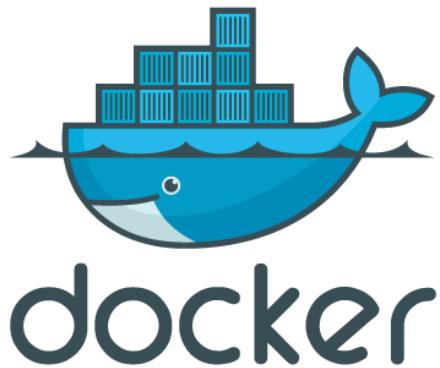


Imagen 4; Docker

- Duckdns

[DuckDNS](#) es una herramienta de servicio DNS dinámico gratuito que permite a los usuarios asignar un nombre de dominio personalizado a una dirección IP dinámica. Las direcciones IP dinámicas son asignadas por los proveedores de servicios de Internet (ISP) y cambian con cierta frecuencia, lo que dificulta su acceso remoto desde fuera de la red local.

Con DuckDNS, los usuarios pueden asignar un nombre de dominio personalizado a su dirección IP dinámica y acceder a su red desde cualquier lugar en Internet. DuckDNS actualiza automáticamente la dirección IP asignada por el ISP, lo que garantiza que el nombre de dominio personalizado esté siempre asociado con la dirección IP correcta.

Para utilizar DuckDNS, los usuarios deben registrarse en el sitio web de DuckDNS y crear un nombre de subdominio personalizado, como "nombre.duckdns.org". Luego, deben configurar un cliente DuckDNS en su red local para actualizar automáticamente la dirección IP asociada con ese subdominio cada vez que cambie.

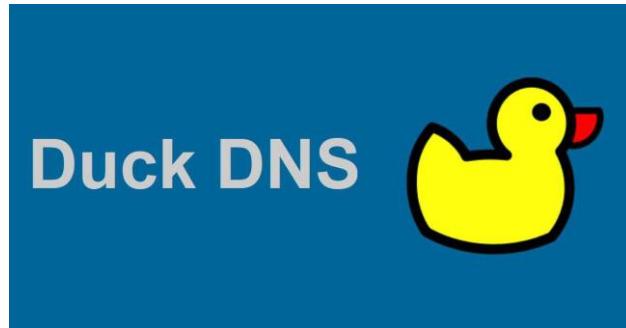


Imagen 5; Duck DNS

- Samba

[Samba](#) es una herramienta de software libre que proporciona la funcionalidad de compartir archivos entre sistemas operativos diferentes, incluyendo Linux, Windows y macOS. Samba permite que los sistemas operativos Linux y Unix actúen como servidores de archivos para sistemas operativos Windows y macOS, y viceversa.

Samba utiliza el protocolo SMB/CIFS (Server Message Block/Common Internet File System) para compartir archivos, impresoras y otros recursos entre los sistemas. SMB/CIFS es el protocolo utilizado por Microsoft Windows para compartir archivos en una red, lo que hace que Samba sea una herramienta muy útil para compartir archivos entre diferentes sistemas operativos.

Además de compartir archivos e impresoras, Samba también ofrece funciones avanzadas de autenticación y seguridad, como la integración con Active Directory de Microsoft, el control de acceso basado en usuarios y grupos, y la encriptación de datos durante la transferencia.

Samba también puede ser configurado para funcionar como un servidor de dominio, proporcionando servicios de autenticación y gestión de usuarios a través de un dominio de Windows.

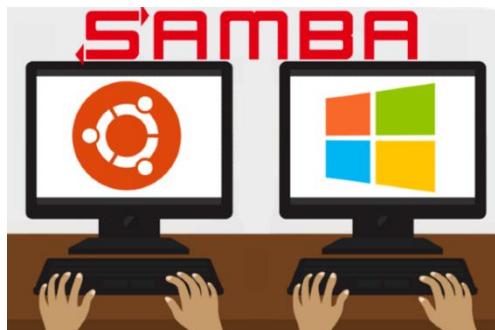


Imagen 6; Samba

- Torrent

[Torrent](#) es una herramienta de software que permite la descarga y compartición de archivos grandes a través de una red P2P (peer-to-peer). El protocolo torrent divide el archivo en pequeñas partes y las distribuye a través de múltiples usuarios que están descargando y compartiendo el archivo simultáneamente.

Cuando un usuario quiere descargar un archivo torrent, utiliza un cliente de torrent para conectarse a la red y descargar pequeñas partes del archivo de otros usuarios que ya lo están compartiendo. A medida que el usuario descarga más partes del archivo, también comparte las

partes que ya ha descargado, permitiendo que otros usuarios descarguen partes del archivo a través de su conexión a Internet.

La distribución de partes del archivo entre diferentes usuarios reduce la carga en un único servidor centralizado y también acelera la velocidad de descarga, ya que cada usuario comparte las partes del archivo que ya ha descargado con otros usuarios que están en el proceso de descargarlo.



Imagen 7; Torrent



Aunque la tecnología torrent es legal y puede utilizarse para compartir archivos legítimos, también se ha asociado con la descarga ilegal de material protegido por derechos de autor. Por lo tanto, es importante tener en cuenta que la descarga y el uso de archivos protegidos por derechos de autor sin autorización pueden ser ilegales y pueden tener consecuencias legales graves.

- Minidlna

[MiniDLNA](#) es una herramienta de software libre que permite la transmisión de archivos multimedia, como música, fotos y vídeos, desde un servidor DLNA (Digital Living Network Alliance) a un dispositivo de reproducción DLNA en una red local.

DLNA es un estándar de red que permite que los dispositivos electrónicos, como televisores inteligentes, consolas de videojuegos, reproductores de Blu-ray y teléfonos inteligentes, se comuniquen entre sí y comparten contenido multimedia a través de una red doméstica.

MiniDLNA es una herramienta muy útil para aquellos que desean transmitir contenido multimedia a través de una red local sin tener que copiar físicamente los archivos multimedia en cada dispositivo. En su lugar, MiniDLNA utiliza la biblioteca multimedia del servidor DLNA para compartir archivos multimedia en tiempo real a través de la red.

Además, MiniDLNA es fácil de configurar y utiliza muy poca memoria y CPU, lo que la hace una herramienta ideal para dispositivos con recursos limitados, como dispositivos NAS (Network Attached Storage), routers con capacidad de almacenamiento y dispositivos basados en Linux.



Imagen 8; dlna

- Jellyfin

[Jellyfin](#) es una herramienta de software libre y gratuita que permite la transmisión de archivos multimedia, como películas, programas de televisión, música y fotos, a través de una red local e Internet.

Jellyfin se basa en el código fuente del antiguo proyecto Emby, que solía ser una herramienta de software propietario. Sin embargo, Jellyfin es completamente de código abierto y gratuito, lo que significa que cualquier persona puede descargar, utilizar y modificar el código fuente de Jellyfin sin restricciones.

Jellyfin funciona como un servidor multimedia que aloja los archivos multimedia y los transmite a los dispositivos de reproducción compatibles, como televisores inteligentes, reproductores de Blu-ray, consolas de videojuegos, dispositivos móviles y computadoras. Jellyfin es compatible con una amplia gama de formatos de archivos multimedia y proporciona funciones avanzadas como la transcodificación, la sincronización de medios, la descarga de medios y la gestión de usuarios.

Además, Jellyfin cuenta con aplicaciones para múltiples plataformas, lo que permite a los usuarios acceder a su biblioteca multimedia desde cualquier lugar y en cualquier dispositivo.



Imagen 9; Jellyfin

- Netdata

Netdata es una herramienta de monitoreo de rendimiento en tiempo real que permite a los usuarios recopilar y visualizar datos de monitoreo del sistema en una interfaz web. Es intuitiva, lo que es genial para sistemas con recursos limitados.

Netdata puede monitorear una amplia variedad de métricas de sistema, incluyendo uso de CPU, RAM y disco, utilización de la red, actividad de servicios y aplicaciones, y mucho más. También puede ser utilizado para monitorear el estado de servidores web y bases de datos, proporcionando información valiosa sobre la salud y el rendimiento del sistema.

La herramienta es muy personalizable y se puede configurar para adaptarse a las necesidades específicas de cada usuario. Además, cuenta con una amplia gama de opciones de visualización de datos, lo que permite a los usuarios ver fácilmente las tendencias y el rendimiento en tiempo real.



Imagen 10; Netdata

6. Desarrollo

6.1 - Instalar Raspberry Pi 4

1. Insertamos la tarjeta micro-SD en el lector de tarjetas SD.



Imagen 11; micro-SD

2. Nos dirigimos hacia la página de [Raspberry Pi](#)

Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)



Imagen 12; Raspberry Pi

Fernando García Ayoso

Página 19

3. Descargamos el programa de instalación de Raspberry Pi para el sistema operativo que tengamos, en mi caso, Windows.



Imagen 14; Raspberry Pi Installer (2)

4. Como podemos observar en la imagen número cuatro, nos aparecen tres botones “Choose OS”, “Choose Storage” y “Write”. Clicamos en “Choose OS”, en el cual elegiremos el sistema operativo de nuestra Raspberry Pi.

Vamos a elegir el sistema operativo de 32-BIT sin escritorio

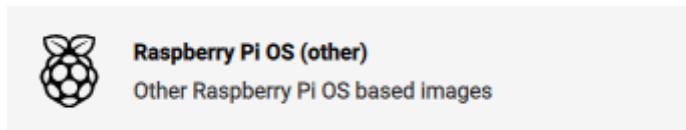


Imagen 15; Sistema operativo

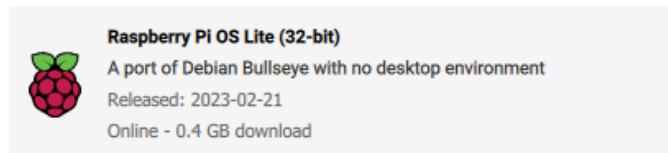


Imagen 16; Sistema operativo 32 bits

5. Cuando elegimos el sistema operativo, debemos elegir la tarjeta sd donde se va a escribir el sistema de la Raspberry Pi.



Imagen 17; Elegir tarjeta microsd



Imagen 18; Ruta tarjeta

6. Le damos a “Write” y confirmamos que queremos

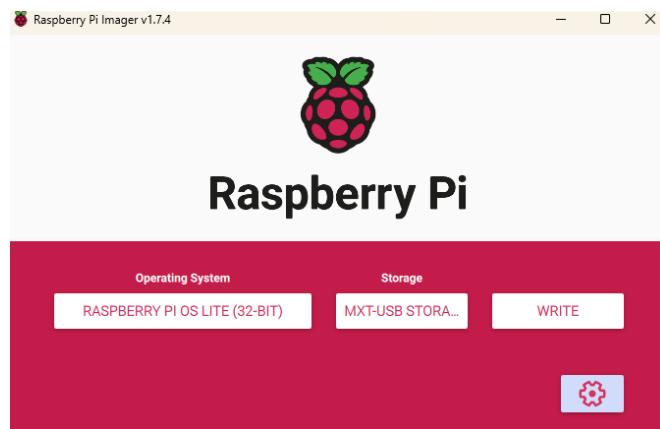


Imagen 19; Escribir instalador en la tarjeta

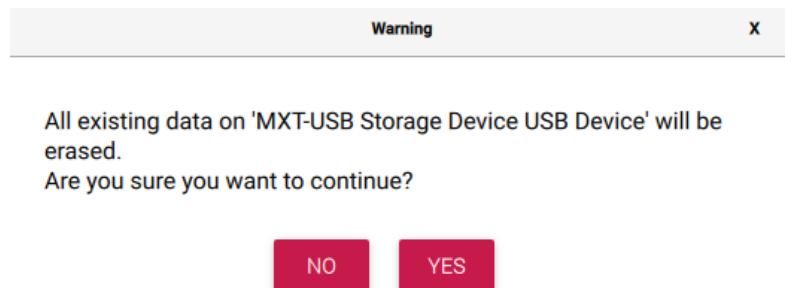


Imagen 20; Confirmar instalador en la tarjeta

7. A continuación se pondrá a escribir el sistema operativo de la Raspberry Pi en la tarjeta sd.

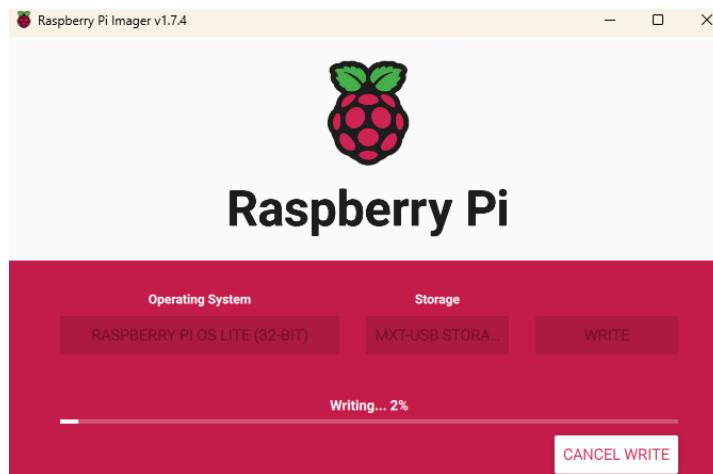


Imagen 21; Proceso de instalación

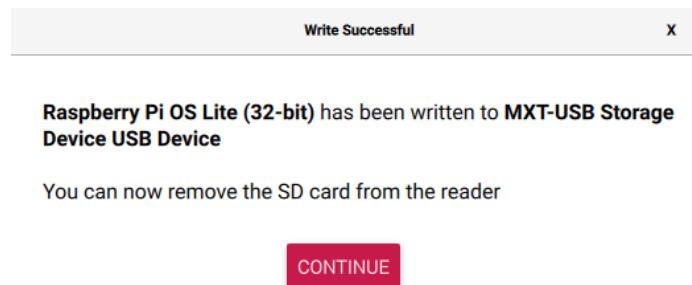


Imagen 22; Proceso finalizado

8. Una vez que haya terminado, preactivaremos el protocolo SSH para poder comunicarnos con la Raspberry Pi sin necesidad de conectarla a un monitor externo. Para ello, debemos dirigirnos al explorador de archivos->Este Equipo y nos debe aparecer una partición llamada “bootfs” que es con la que arranca el sistema operativo de la Raspberry Pi

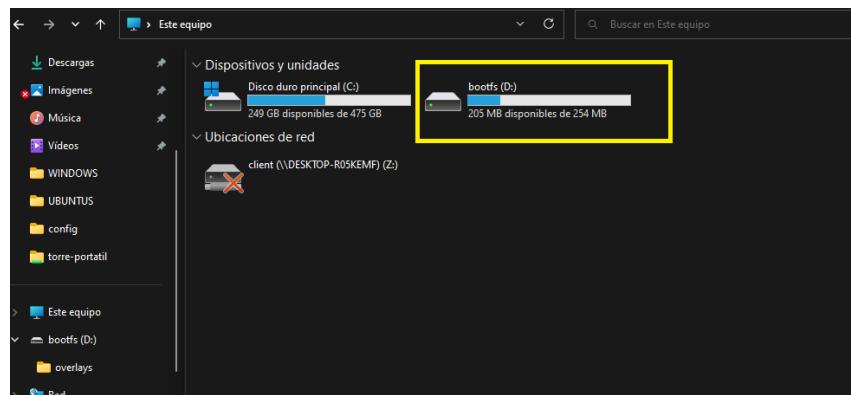


Imagen 23; bootfs

Si clicamos en ella, vamos a crear una archivo de texto llamado SSH, el cual debe estar SIN EXTENSIÓN de archivo.

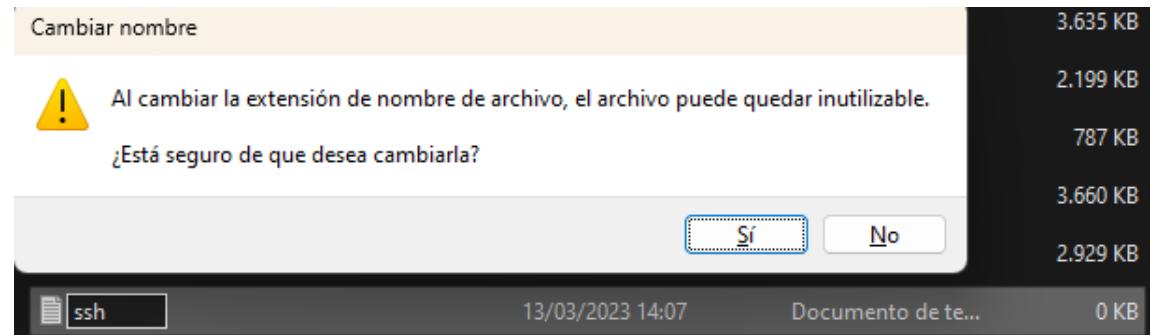


Imagen 24; Archivo SSH

9. Introducimos la tarjeta micro-sd a la Raspberry Pi con su fuente de alimentación y con un cable Ethernet para que nos dé IP.



Imagen 25; Cable Ethernet

10. Para comprobar la IP que tiene la Raspberry Pi, se utilizará la herramienta Fing, la cual, hemos explicado con anterioridad.

En esta herramienta, al conectar la Raspberry Pi, nos debe aparecer su información.

En la captura XX podemos observar que, efectivamente, hay un dispositivo en la red llamado “raspberrypi”

A screenshot of the Fing application interface. On the left is a sidebar with navigation links like 'Inicio', 'Reportes', 'Red', 'Seguridad', 'Internet', 'Gente', 'Herramientas', and 'Notificaciones'. Below that are buttons for 'PRODUCCIÓN DE BLOQUE...', 'Acceder a Premium...', 'Fing 3.2.8', and 'Fernando García Ayoso'. The main area is titled 'Dispositivos - MOVISTAR_ED19' and shows a table of network devices. The table has columns for 'TIPO', 'NOMBRE', 'DETALLES', 'CAMBIADO', and 'STATUS'. One row is highlighted for a 'raspberrypi' device, which is an 'Airs' model. Other devices listed include a Router (Movistar+ GPT-2542DNAC), a Smartphone (Oppo-A5 (A5)), a Generic device (Asus), a Computer (Apple iPhone 14 Pro Max), another Generic device (Asus Vivobook X712FB-X712FB), and several other common household items like TVs and tablets. The status column shows various connection types like 'Router', 'Mobile', 'Android', 'Windows 10 Pro', 'Windows 10', and 'Unknown'.

Imagen 26; Fing (2)

Si clicamos en él, observamos que la IP es 192.168.1.94

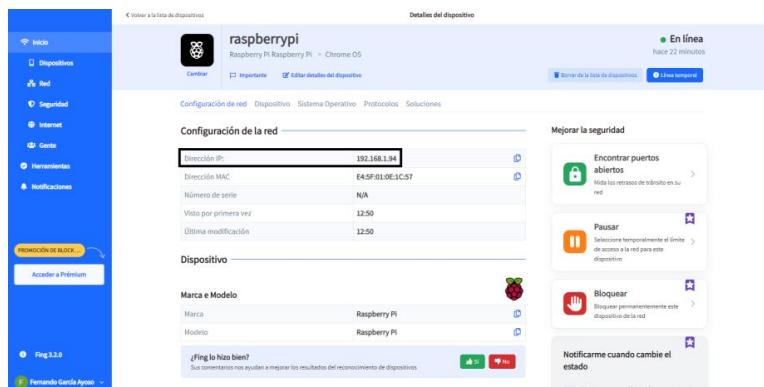


Imagen 27; IP Raspberry Pi

- Una vez hemos sabido que la IP de la Raspberry Pi es 192.168.1.92, nos conectamos por SSH. Para ello, abriremos el símbolo del sistema (cmd) como administrador:

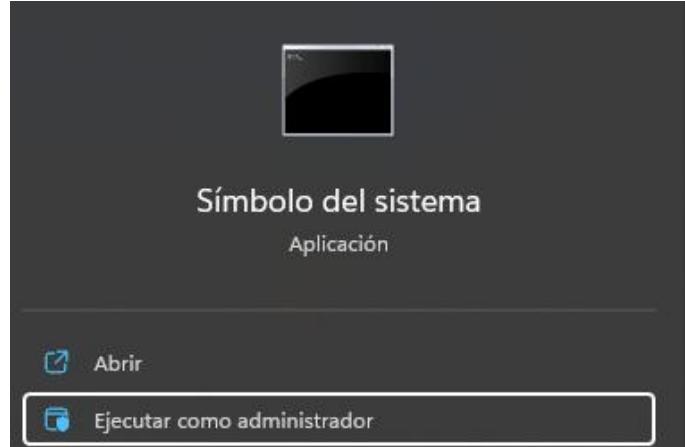


Imagen 28; Símbolo del sistema

- Una vez ejecutado el cmd, escribiremos:

```
SSH pi@192.168.1.94
```

Tabla 2; SSH pi@IP

Destacar que el usuario es pi y la contraseña es raspberry

```
C:\Users\ferna>ssh pi@192.168.1.94
The authenticity of host '192.168.1.94 (192.168.1.94)' can't be established.
ED25519 key fingerprint is SHA256:DFLvkrTYBxLKpQDFePvjZBZOsLGH//Wwt+1Zrp1iXgg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.94' (ED25519) to the list of known hosts.
pi@192.168.1.94's password:
Linux raspberrypi 5.15.84-v7l+ #1613 SMP Thu Jan 5 12:01:26 GMT 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~ $ |
```

Imagen 29; Conexión SSH exitosa

6.2 - Configurar Raspberry Pi

- Si escribimos:

```
|sudo raspi-config|
```

Tabla 3; Configurar Raspberry Pi

Aparece un menú en el que se encuentra la configuración del SO

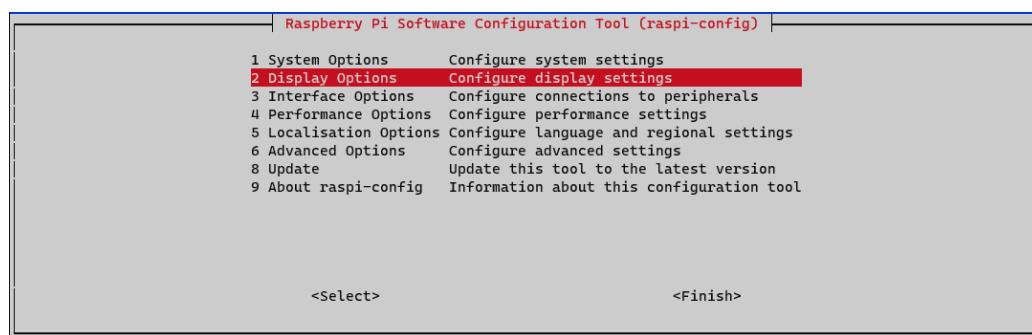


Imagen 30; Configuración SO Raspberry Pi

- Si nos dirigimos a “System options->Hostname”, cambiamos el nombre del servidor que queramos

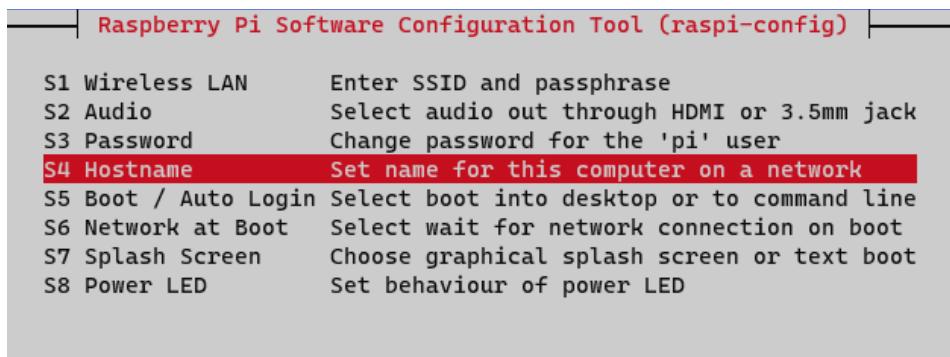


Imagen 31; Hostname

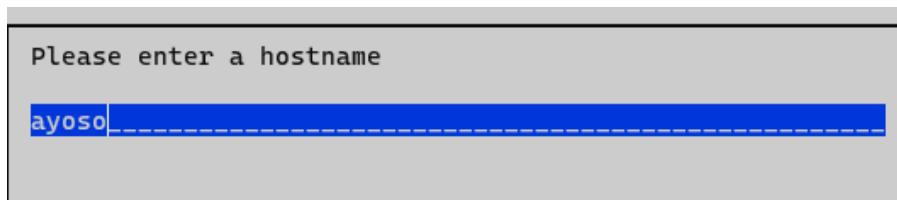


Imagen 32; Nombre hostname

- Nos dirigimos a “Localisation Options” y configuramos el idioma y la región.

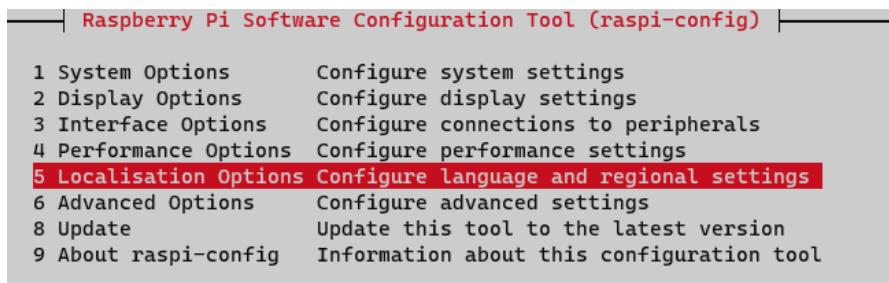


Imagen 33; Configurar el idioma y la región

“Locale”:

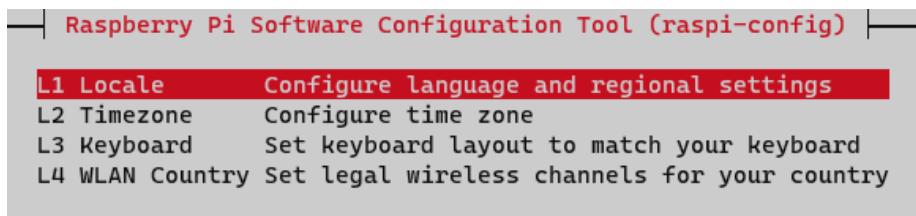


Imagen 34; Locale

Buscamos es_ES.UTF-8 UTF-8 y lo marcamos con el espacio:

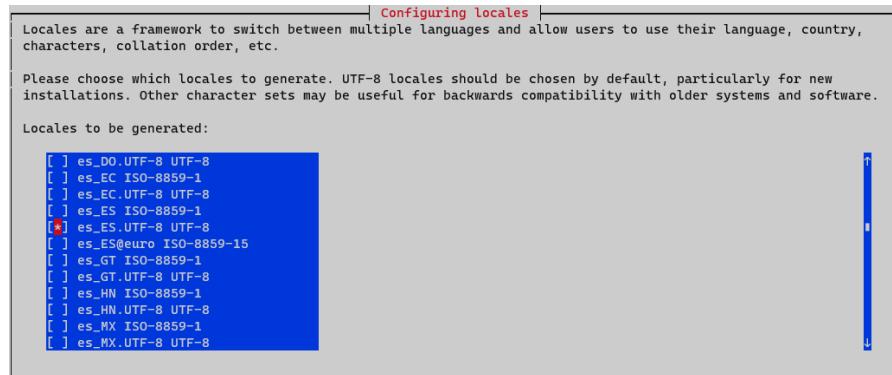


Imagen 35; es_ES.UTF-8 UTF-8

- A continuación, nos dirigimos a “Timezone” y configuramos la zona horaria:

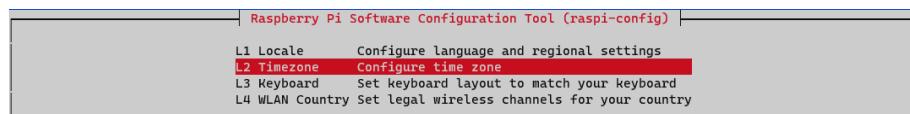


Imagen 36; Timezone

Seleccionamos “Madrid”



Imagen 37; Timezone (2)

- Luego le damos a “WLAN Country” en el que debemos elegir las frecuencias de la red wifi que se va a utilizar. En este caso elegimos “ES Spain”

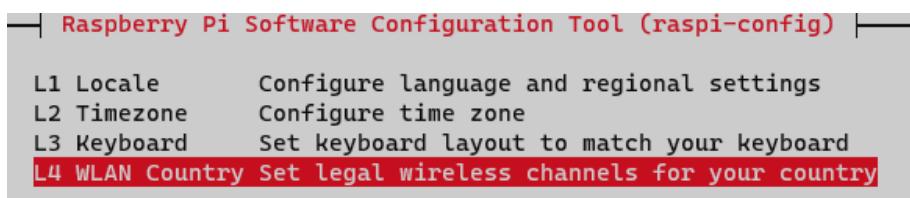


Imagen 38; Wlan Country

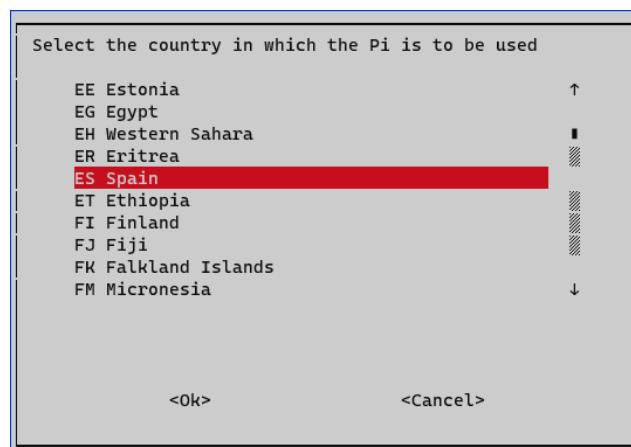


Imagen 39; Country

Una vez hayamos configurado lo anterior, seleccionamos “Finish”.

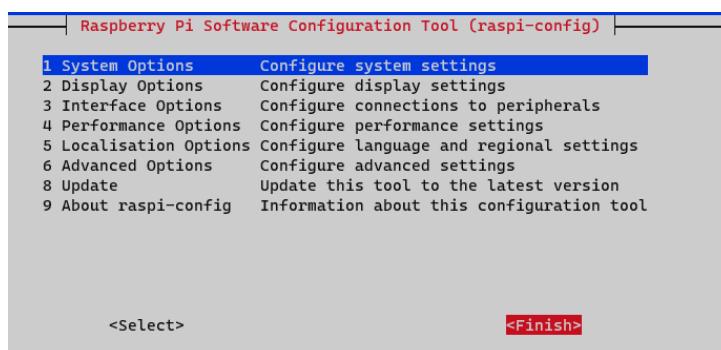


Imagen 40; Finaliza la configuración del SO

2. Podemos cambiar la contraseña con el comando `psswd`, en el que debemos escribir la contraseña antigua y posteriormente, la nueva que queramos. Destacar que el cambio de contraseña es necesario ya que la anterior contraseña era `raspberry` y es la utilizada comúnmente al iniciar una Raspberry Pi.

```
Changing password for pi.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

Imagen 41; Cambio de contraseña

3. Actualizamos el sistema con el comando:

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

Tabla 4; Actualizar el sistema

```
pi@ayoso:~ $ sudo apt-get update && sudo apt-get upgrade
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:2 http://archive.raspberrypi.org/debian bullseye InRelease [23.6 kB]
Get:3 http://archive.raspberrypi.org/raspbian bullseye/main armhf Packages [13.2 MB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [315 kB]
Fetched 13.6 MB in 14s (952 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Building state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  curl libcamera-apps-lite libcamera0 libcurl3-gnutls libcurl4 libtiff5 linux-libc-dev python3-libcamera
  python3-picamera2 python3-v4l2 raspberrypi-bootloader raspberrypi-kernel vcdbg
13 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 107 MB of archives.
After this operation, 35.8 kB of additional disk space will be used.
Do you want to continue? [y/n] y
Get:1 http://archive.raspberrypi.org/debian bullseye/main armhf raspberrypi-kernel armhf 1:1.20230306-1 [99.1 MB]
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf curl armhf 7.74.0-1.3+deb11u7 [258 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libcurl4 armhf 7.74.0-1.3+deb11u7 [310 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libtiff5 armhf 4.2.0-1+deb11u4 [270 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libcurl3-gnutls armhf 7.74.0-1.3+deb11u7 [306 kB]
18% [5 libcurl3-gnutls 8,192 B/306 kB 3%] [1 raspberrypi-kernel 17.5 MB/99.1 MB 18%]
```

Imagen 42; Actualizar el sistema

4. Reiniciamos el sistema con:

```
| sudo reboot |
```

Imagen 43; Reiniciar el sistema

```
pi@ayoso:~ $ sudo reboot
Connection to 192.168.1.94 closed by remote host.
Connection to 192.168.1.94 closed.
```

Imagen 44; Reiniciar el sistema (2)

```
C:\Users\ferna>ssh pi@192.168.1.94
pi@192.168.1.94's password:
Linux ayoso 5.15.84-v7l+ #1613 SMP Thu Jan 5 12:01:26 GMT 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 13 16:25:25 2023 from 192.168.1.90
pi@ayoso:~ $ |
```

Imagen 45; Inicio del sistema

5. Una vez hayamos realizado las actualizaciones, enchufamos el disco duro de 1 TB a la Raspberry Pi y mediante el comando que se muestra a continuación, observamos los dispositivos que se encuentran conectados:

Tipo de fichero

```
sudo blkid
```

Tabla 5; Dispositivos conectados

```
pi@ayoso:/ $ sudo blkid
/dev/mmcblk0p1: LABEL_FATBOOT="bootfs" LABEL="bootfs" UUID="CEE7-0DF0" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="48a9c3fc-01"
/dev/mmcblk0p2: LABEL="rootfs" UUID="ebadab55-bd70-46a9-8f18-995c136fb5b2" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="48a9c3fc-02"
/dev/sda1: LABEL="FE" BLOCK_SIZE="512" UUID="5C4C41F74C41CC88" TYPE="ntfs" PARTUUID="d637bf4e-01"
```

Imagen 46; Dispositivos conectados

Destacamos que el dispositivo con UUID="5C4C41F74C41CC88" TYPE="ntfs" que es el pendrive de 512 GB.

Lo más habitual es que si el pendrive se ha utilizado en Windows previamente, venga en formato "ntfs" que es el formato que le ha dado Windows a los dispositivos de almacenamiento externo por defecto. Podemos utilizarlo tal cual en Linux, el problema es que no nos da acceso a los permisos cuando lo montemos, entonces es un poco más difícil si vamos a acceder desde fuera.

Para formatear el pendrive en ext4 (lo cual es muy recomendable), debemos escribir los siguientes comandos:

```
sudo mkfs.ext4 /dev/sda1 -L FE
(siendo /dev/sda1 la interfaz donde está conectado)
```

Tabla 6; formatear pendrive en ext4

Si volvemos a realizar sudo blkid:

```
pi@ayoso:/ $ sudo blkid
/dev/mmcblk0p1: LABEL_FATBOOT="bootfs" LABEL="bootfs" UUID="CEE7-0DF0" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="48a9c3fc-01"
/dev/mmcblk0p2: LABEL="rootfs" UUID="ebadab55-bd70-46a9-8f18-995c136fb5b2" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="48a9c3fc-02"
/dev/sda1: LABEL="FE" UUID="3bac57c2-1fa4-495e-8b23-6f81cd42adab" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="d637bf4e-01"
```

Imagen 47; UUID ext4

Copiamos el UUID, que lo pegaremos en los próximos apartados

- Nos dirigimos a la ruta principal con cd / y realizamos un ls -l.

Observaremos una gran cantidad de directorios, en el que destaca el directorio mnt al cual nos dirigimos con cd /mnt

```
pi@ayoso:~ $ cd /
pi@ayoso:/ $ ls -l
total 60
lrwxrwxrwx  1 root root    7 feb 21 02:02 bin -> usr/bin
drwxr-xr-x  4 root root 4096 ene  1 1970 boot
drwxr-xr-x 16 root root 3800 mar 13 16:55 dev
drwxr-xr-x 82 root root 4096 mar 13 16:48 etc
drwxr-xr-x  3 root root 4096 feb 21 02:04 home
lrwxrwxrwx  1 root root    7 feb 21 02:02 lib -> usr/lib
drwx-----  2 root root 16384 feb 21 02:22 lost+found
drwxr-xr-x  2 root root 4096 feb 21 02:02 media
drwxr-xr-x  2 root root 4096 feb 21 02:02 mnt
drwxr-xr-x  2 root root 4096 feb 21 02:02 opt
dr-xr-xr-x 210 root root    0 ene  1 1970 proc
drwx-----  2 root root 4096 feb 21 02:02 root
drwxr-xr-x 24 root root 740 mar 13 17:21 run
lrwxrwxrwx  1 root root    8 feb 21 02:02 sbin -> usr/sbin
drwxr-xr-x  2 root root 4096 feb 21 02:02 srv
dr-xr-xr-x 12 root root    0 ene  1 1970 sys
drwxrwxrwt 10 root root 4096 mar 13 16:49 tmp
drwxr-xr-x 11 root root 4096 feb 21 02:02 usr
drwxr-xr-x 11 root root 4096 feb 21 02:23 var
```

Imagen 48; cd /mnt

- En el directorio /mnt crearemos la carpeta donde se “montará” el disco duro que tengamos, en este caso, el pendrive va a actuar de disco duro. Mount y mnt

```
pi@ayoso:/ $ cd mnt/
pi@ayoso:/mnt $ sudo mkdir disco
pi@ayoso:/mnt $ ls -l
total 4
drwxr-xr-x  2 root root 4096 mar 13 17:27 disco
pi@ayoso:/mnt $ |
```

Imagen 49; carpeta disco duro

- A continuación, editamos el archivo ftap que contiene y dicta los dispositivos que se van a montar en el arranque y se toman como nativos. Este archivo se edita con:

|sudo nano /etc/fstab|

Tabla 7; Editar archivo ftap

En este archivo pegaremos el UUID que habíamos copiado anteriormente. Destacar que pegaremos el UUID sin comillas.

Seguidamente, en la misma línea, escribimos donde queremos montar el disco, que en mi será en /mnt/disco con el formato ext4 defaults 0 0

Debe quedar así:

```
GNU nano 5.4                               /etc/fstab
proc      /proc          proc  defaults      0      0
PARTUUID=48a9c3fc-01  /boot        vfat  defaults      0      2
PARTUUID=48a9c3fc-02  /           ext4   defaults,noatime  0      1
# a swapfile is not a swap partition, no line here
# use dphys-swapfile swap[on|off] for that
UUID=3bac57c2-1fa4-495e-8b23-6f81cd42adab /mnt/disco ext4 defaults 0 0
```

Imagen 50; Archivo fstab

9. Reiniciamos el sistema con:

[sudo reboot]

Tabla 8; Reinicio del sistema

10. Si realizamos un df o df -h para ver los discos y el espacio libre, comprobaremos que NO se encuentra el disco que queremos

```
pi@ayoso:/mnt $ df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
/dev/root        29G    2,1G    26G    8% /
devtmpfs         1,8G      0    1,8G    0% /dev
tmpfs            1,9G      0    1,9G    0% /dev/shm
tmpfs            768M    1,1M    767M    1% /run
tmpfs            5,0M    4,0K    5,0M    1% /run/lock
/dev/mmcblk0p1    255M    50M    206M   20% /boot
tmpfs            384M      0    384M    0% /run/user/1000
```

Imagen 51; Comprobación discos

Por tanto, reiniciamos:

```
pi@ayoso:/mnt $ sudo reboot
Connection to 192.168.1.94 closed by remote host.
Connection to 192.168.1.94 closed.
```

Imagen 52; Reiniciamos el sistema

Si realizamos de nuevo un df -h, esta vez sí, nos aparece el pendrive:

S.Ficheros	Tamaño	Usados	Disp	Uso%	Montado en
/dev/root	29G	2,1G	26G	8%	/
devtmpfs	1,8G	0	1,8G	0%	/dev
tmpfs	1,9G	0	1,9G	0%	/dev/shm
tmpfs	768M	1,1M	767M	1%	/run
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
/dev/sda1	452G	28K	429G	1%	/mnt/disco
/dev/mmcblk0p1	255M	50M	206M	20%	/boot
tmpfs	384M	0	384M	0%	/run/user/1000

Imagen 53; Comprobación discos (2)

11. Si nos dirigimos hacia la ruta /mnt/disco y realizamos un ls observamos que sí hay contenido en su interior, por tanto, estaría correctamente montado:

```
pi@ayoso:~ $ cd /mnt/disco/
pi@ayoso:/mnt/disco $ ls
lost+found
pi@ayoso:/mnt/disco $ |
```

Imagen 54; Contenido en el interior del disco

12. Lo primero que vamos a hacer es crear una carpeta llamada descargas en el directorio /mnt/disco donde irán las descargas de los torrents que vamos a poner

```
pi@ayoso:/mnt/disco $ sudo mkdir Descargas
pi@ayoso:/mnt/disco $ ls -l
total 20
drwxr-xr-x 2 root root 4096 mar 13 21:34 Descargas
drwx----- 2 root root 16384 mar 13 21:14 lost+found
```

Imagen 55; Carpeta Descargas

Accedemos a la carpeta Descargas y aquí creamos una carpeta llamada temp que aquí va a ser donde se vayan almacenando los ficheros temporales que vaya descargando nuestro torrent

```
pi@ayoso:/mnt/disco $ cd Descargas/
pi@ayoso:/mnt/disco/Descargas $ sudo mkdir temp
pi@ayoso:/mnt/disco/Descargas $ ls -l
total 4
drwxr-xr-x 2 root root 4096 mar 13 21:36 temp
```

Imagen 56; Carpeta temp

6.3 - Instalar servidor de Torrent y configuración

1. Para instalar el servidor de torrent, escribiremos

```
sudo apt-get install transmission-daemon
```

Tabla 9; Instalar servidor torrent

```
pi@ayoso:/mnt/disco/Descargas $ sudo apt-get install transmission-daemon
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libminiupnpc17 libnatpmp1 transmission-cli transmission-common
Paquetes sugeridos:
 minissdpd natpmpc transmission-gtk
Se instalarán los siguientes paquetes NUEVOS:
 libminiupnpc17 libnatpmp1 transmission-cli transmission-common
 transmission-daemon
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 754 kB de archivos.
Se utilizarán 3.576 kB de espacio de disco adicional después de esta operación.
```

Imagen 57; Instalar servidor torrent

2. Una vez que se haya instalado, vamos a parar el servicio ya que vamos a editar los archivos de configuración.

Para detener el servicio, escribimos:

```
sudo service transmission-daemon stop
```

Tabla 10; Detener el servicio

```
pi@ayoso:/mnt/disco/Descargas $ sudo service transmission-daemon stop
```

Imagen 58; Detener el servicio

3. Editamos el archivo de configuración con:

```
|sudo nano /etc/transmission-daemon/settings.json|
```

Tabla 11; Editar archivo de configuración

```
pi@ayoso:/mnt/disco/Descargas $ sudo nano /etc/transmission-daemon/settings.json
```

Imagen 59; Editar archivo de configuración

Veremos un archivo tal que así:

```
GNU nano 5.4          /etc/transmission-daemon/settings.json
{
    "alt-speed-down": 50,
    "alt-speed-enabled": false,
    "alt-speed-time-begin": 540,
    "alt-speed-time-day": 127,
    "alt-speed-time-enabled": false,
    "alt-speed-time-end": 1020,
    "alt-speed-up": 50,
    "bind-address-ipv4": "0.0.0.0",
    "bind-address-ipv6": "::",
    "blocklist-enabled": false,
    "blocklist-url": "http://www.example.com/blocklist",
    "cache-size-mb": 4,
    "dht-enabled": true,
    "download-dir": "/var/lib/transmission-daemon/downloads",
    "download-limit": 100,
    "download-limit-enabled": 0,
    "download-queue-enabled": true,
    "download-queue-size": 5,
    "encryption": 1,
    "idle-seeding-limit": 30,
    "idle-seeding-limit-enabled": false,
    "incomplete-dir": "/var/lib/transmission-daemon/Downloads",
    "incomplete-dir-enabled": false,
    "lpd-enabled": false,
    "max-peers-global": 200,
    "message-level": 1,
    "peer-congestion-algorithm": "",
    "peer-id-ttl-hours": 6,
    "peer-limit-global": 200,
    "peer-limit-per-torrent": 50,
    "peer-port": 51413,
    "peer-port-random-high": 65535,
    "peer-port-random-low": 49152,
    "peer-port-random-on-start": false,
    "peer-socket-tos": "default",
    "pex-enabled": true,
```

Imagen 60; Archivo de configuración

4. En este archivo, vamos a dirigirnos al apartado donde nos dice: “download-dir:” y debemos escribir la ruta /mnt/disco/Descargas

```
"download-dir": "/mnt/disco/Descargas",
```

Imagen 61; Download-dir

5. Por otra parte, cambiamos también la ruta en el apartado de “incomplete-dir:”, pero esta vez, escribimos /mnt/disco/Descargas/tmp

```
"incomplete-dir": "/mnt/disco/Descargas/tmp",
```

Imagen 62; Incomplete-dir

6. Activamos que se use la carpeta temporal, cambiando “false” por “true” en la línea “incomplete-dir-enabled”

```
"incomplete-dir-enabled": true,
```

Imagen 63; Incomplete-dir-enabled

7. Más abajo, en la línea “rpc-username”, cambiamos el usuario a la aplicación.

```
"rpc-username": "fernandoayoso",
```

Imagen 64; rpc-username

8. Y obviamente, cambiamos su contraseña en la línea “rpc-password:”

```
"rpc-password": [REDACTED]
```

Imagen 65; rpc-password

9. Fijémonos bien en el puerto que viene, para acceder luego al programa. Se puede cambiar pero, en mi caso, voy a dejarlo tal y como viene

```
"rpc-port": 9091
```

Imagen 66; rpc-port

10. Y por último, en la línea "rpc-whitelist-enabled:", cambiamos "true" por "false" para que se pueda acceder desde cualquier dirección IP. Autenticar usuarios.

```
"rpc-whitelist-enabled": false,
```

Imagen 67; rpc-whitelist-enabled

6.4 - Controlar el usuario de Torrent

- Escribimos sudo nano /etc/init.d/transmission-daemon para controlar con qué usuario se lanza el daemon de este programa:

```
pi@ayoso:/mnt/disco/Descargas $ sudo nano /etc/init.d/transmission-daemon
```

Imagen 68; Editar archivo transmission-daemon

Nos aparecerá un fichero tal que así:

```
GNU nano 5.4          /etc/init.d/transmission-daemon
#!/bin/sh -e
### BEGIN INIT INFO
# Provides:          transmission-daemon
# Required-Start:    $local_fs $remote_fs $network
# Required-Stop:     $local_fs $remote_fs $network
# Default-Start:    2 3 4 5
# Default-Stop:     0 1 6
# Short-Description: Start or stop the transmission-daemon.
# Description:       Enable service provided by transmission-daemon.
### END INIT INFO

NAME=transmission-daemon
DAEMON=/usr/bin/$NAME
USER=debian-transmission
STOP_TIMEOUT=30

export PATH="${PATH:+$PATH:}/sbin"

[ -x $DAEMON ] || exit 0

[ -e /etc/default/$NAME ] && . /etc/default/$NAME

. /lib/lsb/init-functions

start_daemon () {
    if [ ${ENABLE_DAEMON} != 1 ]; then
        log_progress_msg "(disabled)"
        log_end_msg 255 || true
    else
        start-stop-daemon --start \
            --chuid $USER \
            $START_STOP_OPTIONS \
            --exec $DAEMON -- $OPTIONS || log_end_msg $?
        log_end_msg 0
    fi
}
```

Imagen 69; Archivo transmission-daemon

2. Como podemos observar, hay un apartado “user”. Debemos cambiarlo por el usuario de la Raspberry Pi, que en este caso, es pi

```
NAME=transmission-daemon
DAEMON=/usr/bin/$NAME
USER=pi
STOP_TIMEOUT=30
```

Imagen 70; transmission-daemon: user

3. Guardamos y a continuación, realizamos otro sudo nano pero a este directorio:

```
|sudo nano /etc/systemd/system/multi-user.target.wants/transmission-daemon.service|
```

Tabla 12; Editar archivo transmission-daemon.service

```
pi@ayoso:/mnt/disco/Descargas $ sudo nano /etc/systemd/system/multi-user.target.wants/transmission-daemon.service
```

Imagen 71; Editar archivo transmission-daemon.service

Nos aparecerá un fichero tal que así:

```
/etc/systemd/system/multi-user.target.wants/transmission-daemon.service
[Unit]
Description=Transmission BitTorrent Daemon
After=network.target

[Service]
User=debian-transmission
Type=notify
ExecStart=/usr/bin/transmission-daemon -f --log-error
ExecStop=/bin/kill -s STOP $MAINPID
ExecReload=/bin/kill -s HUP $MAINPID
NoNewPrivileges=true

[Install]
WantedBy=multi-user.target
```

Imagen 72; Archivo transmission-daemon.service

4. Como podemos observar, hay un apartado “user”. Debemos cambiarlo por el usuario de la Raspberry Pi, que en este caso, es pi

```
[Service]
User=pi
```

Imagen 73; transmission-daemon.service: user

5. Guardamos y regarcamos el daemon con el comando:

```
sudo service transmission-daemon start
```

Tabla 13; Actualizar daemon

6. Cambiamos el propietario y el grupo a toda la carpeta de /transmission-daemon con el comando:

```
sudo chown -R pi:pi /etc/transmission-daemon/
```

Tabla 14; Propietario transmission-daemon

```
pi@ayoso:/mnt/disco/Descargas $ sudo chown -R pi:pi /etc/transmission-daemon/
```

Imagen 74; Propietario transmission-daemon

7. Creamos un enlace simbólico desde el archivo settings.json en el directorio /etc/transmission-daemon/ hasta el directorio /home/pi/.config/transmission-daemon/.

```
pi@ayoso:~/.config $ sudo ln -s /etc/transmission-daemon/settings.json /home/pi/.config/transmission-daemon/
```

Imagen 75; settings.json

El propósito de esto es permitir que la aplicación de servidor de descarga de torrents Transmission Daemon (transmission-daemon) pueda acceder al archivo settings.json desde su nuevo lugar en /home/pi/.config/transmission-daemon/ en lugar de su ubicación original en /etc/transmission-daemon/.

8. Cambiamos el propietario y el grupo del directorio /home/pi/.config/transmission-daemon/

```
sudo chown -R pi:pi /home/pi/.config/transmission-daemon/
```

Tabla 15; Propietario /home/pi/.config/transmission-daemon/

```
pi@ayoso:~/config $ sudo chown -R pi:pi /home/pi/.config/transmission-daemon
```

Imagen 76; Propietario /home/pi/.config/transmission-daemon/

Este comando cambia el propietario y el grupo del directorio /home/pi/.config/transmission-daemon/ y todos sus archivos y subdirectorios a pi:pi, es decir, el usuario y el grupo de usuario llamados "pi". El modificador -R significa que se aplica el cambio de permisos de manera recursiva a todos los archivos y subdirectorios dentro del directorio especificado.

El propósito de esto es asegurarse de que el usuario "pi" tenga los permisos adecuados para acceder y modificar los archivos en el directorio /home/pi/.config/transmission-daemon/, que se necesita para que la aplicación Transmission Daemon funcione correctamente.

6.5 - Comprobar el funcionamiento del servicio

1. Abrimos el navegador y escribimos la IP de la Raspberry Pi junto con su puerto, en mi caso, 192.168.1.94:9091

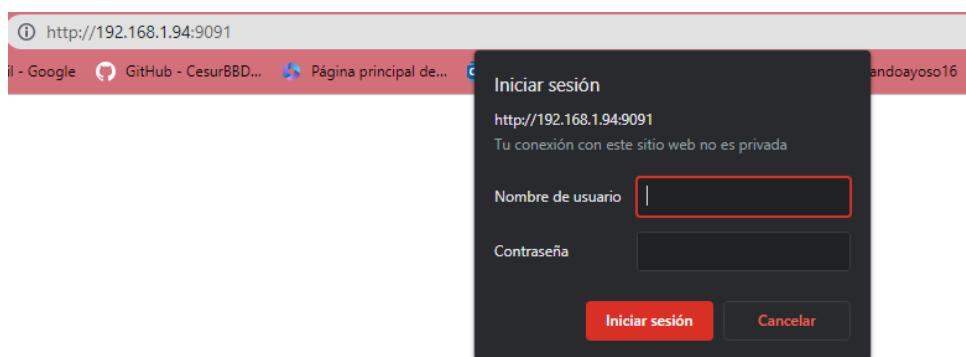


Imagen 77; IP:Puerto

Como es lógico, nos pedirá nuestro usuario y contraseña que anteriormente hemos configurado.

2. Una vez hayamos introducido nuestras credenciales, nos saldrá una ventana como esta:



Imagen 78; Torrent

3. Si le damos al botón superior izquierdo, nos saldrá un menú en el que debemos clicar sobre el botón de subir archivos y elegir el archivo que queramos subir.

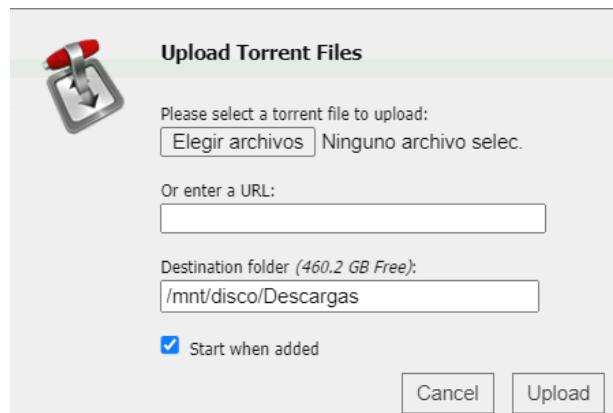


Imagen 79; Elegir archivos



Imagen 80; Archivo subiéndose

Como podemos observar, la película Big Buck Bunny se encuentra en el directorio Descargas.

```
pi@ayoso:~ $ cd /mnt/disco/Descargas/
pi@ayoso:/mnt/disco/Descargas $ ls
'Big Buck Bunny'  temp
```

Imagen 81; Archivo en /temp

6.6 - Samba

- Al no poder visualizar un video con la terminal, procederemos a instalar el programa Samba que lo que va a hacer es poder compartir carpetas de forma fácil y sencilla para dispositivos Windows y Mac

Para ello, ejecutamos el siguiente comando:

```
sudo apt-get install samba samba-common-bin
```

Tabla 16; Instalar samba

```
pi@ayoso:/mnt/disco/Descargas $ sudo apt-get install samba samba-common-bin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  attr ibverbs-providers libavahi-client3 libboost-iostreams1.74.0 libboost-thread1.74.0 libcephfs2 libcurl2 libgfan0
  libgfprpc0 libgfrpc0 libglusterfs0 libibverbs1 libldb2 librados2 librdmacm1 libtdb1 liburing1 python3-cffi-backend
  python3-cryptography python3-dnspython python3-gpg python3-ldb python3-markdown python3-pygments
  python3-requests-toolbelt python3-samba python3-talloc python3-tdb python3-yaml samba-common samba-dsdb-modules
  samba-libs samba-vfs-modules tdb-tools
```

Imagen 82; Instalar samba

- Tal y como hemos hecho antes, detenemos el servicio para editar el archivo de configuración que se encuentra en /etc/samba/smb.conf. Para detener el servicio samba, lo realizaremos con el comando

```
sudo service samba-ad-dc stop
```

Tabla 17; Detener configuración samba-ad-dc

```
pi@ayoso:/mnt/disco/Descargas $ sudo service samba-ad-dc stop
```

Imagen 83; Detener configuración samba-ad-dc

3. Como hemos comentado en el paso anterior, hay que editar el archivo de configuración. Para ello, escribimos:

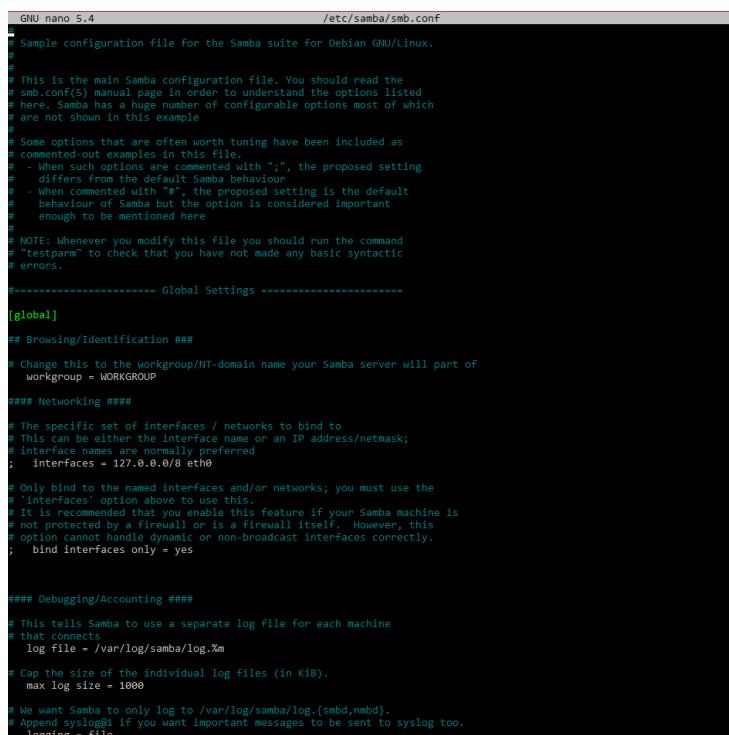
```
sudo nano /etc/samba/smb.conf
```

Tabla 18; Archivo configuración samba-ad-dc

```
pi@ayoso:/mnt/disco/Descargas $ sudo nano /etc/samba/smb.conf
```

Imagen 84: Archivo configuración samba-ad-dc

Nos debe aparecer un archivo como este:



```
GNU nano 5.4          /etc/samba/smb.conf
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
# - When such options are commented with ";", the proposed setting
#   differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the default
#   behaviour of Samba but the option is considered important
# enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
#----- Global Settings -----
[global]
## Browsing/Identification ##
# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = WORKGROUP
### Networking ####
# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
;   interfaces = 127.0.0.0/8 eth0

# Only bind to the named interfaces and/or networks; you must use the
# "interfaces" option above to use this
# It is recommended that you disable this feature if your Samba machine is
# not protected by a firewall or is a firewall itself. However, this
# option cannot handle dynamic or non-broadcast interfaces correctly.
;   bind interfaces only = yes

### Debugging/Accounting ####
# This tells Samba to use a separate log file for each machine
# that connects
log file = /var/log/samba/log.%m
# Cap the size of the individual log files (in Kib).
max log size = 1000
# We want Samba to only log to /var/log/samba/log.{smbd,nmbd}.
# Append syslog@1 if you want important messages to be sent to syslog too.
#logon drive = f:
```

Imagen 85; Archivo samba-ad-dc

4. Nos dirigimos hasta el final del documento en el que vamos a añadir una serie de caracteres:

```
[nombrecarpetacompartidasamba]
comment = descripción
Directory path = /mnt/disco/Descargas
browseable = Yes
writeable = Yes
only guest = no
create mask = 0644
directory mask = 0755
public = no
```

Tabla 19; Caracteres a añadir en samba-ad-dc

En mi caso, va a quedar tal que así:

```
[filefernandoayoso]
comment = carpetacompartida
path = /mnt/disco/Descargas_
browseable = Yes
writeable = Yes
only guest = no
create mask = 0644
directory mask = 0755
public = no
```

Imagen 86; Caracteres a añadir en samba-ad-dc

5. A continuación, añadimos el usuario pi a samba. Para ello:

```
sudo smbpasswd -a pi
```

Tabla 20; Añadir el usuario pi a samba

```
pi@ayoso:/mnt/disco/Descargas $ sudo smbpasswd -a pi
New SMB password:
Retype new SMB password:
Added user pi.
pi@ayoso:/mnt/disco/Descargas $ _
```

Imagen 87; Comando smbpasswd -a pi

Como podemos observar, nos pide una contraseña. No es obligatorio poner la misma que la del usuario pero es recomendable, así no nos tenemos que aprender dos contraseñas diferentes.

- Activamos el servicio:

```
sudo systemctl restart smbd
```

Tabla 21; Activar el servicio

```
pi@ayoso:/mnt/disco/Descargas $ sudo systemctl restart smbd
```

Imagen 88; Activar el servicio

- Añadimos Samba a nuestro Windows. Para ello, abrimos el explorador de archivos y buscamos “Conectar unidad de red”

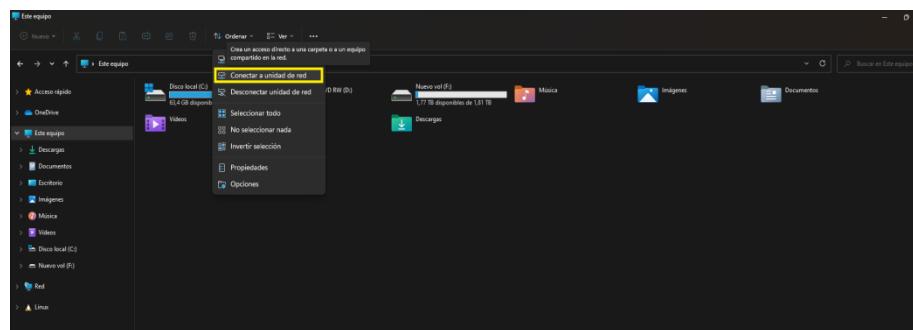


Imagen 89; Conectar unidad de red

Aparecerá una ventana en la que tendremos que elegir la letra que quedamos en el apartado de “unidad” y en el de “carpeta” escribiremos:
<\\IPraspberrypi\\nombrecarpetacompartidasamba>

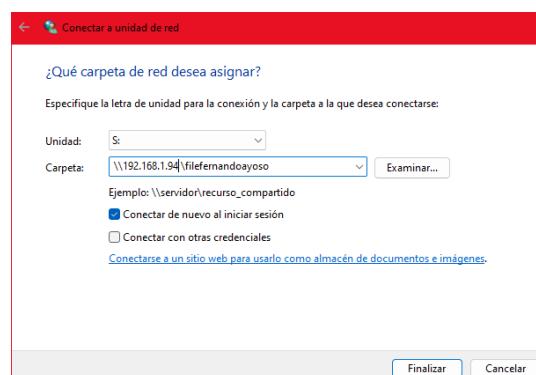


Imagen 90; <\\IPraspberrypi\\nombrecarpetacompartidasamba>

Nos pedirá que introduzcamos la contraseña:

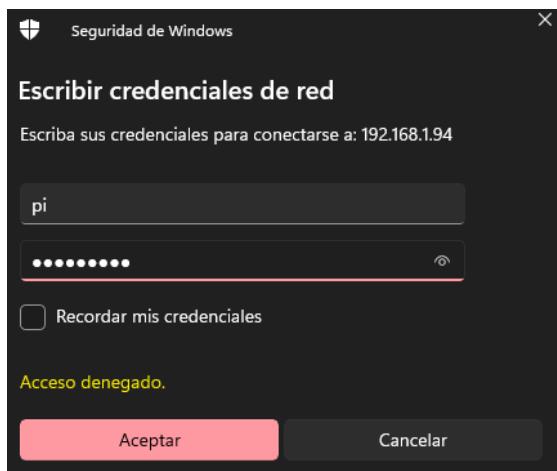


Imagen 91; Credenciales

8. Como podemos observar, tenemos la carpeta “Big Buck Bunny” y “temp”.

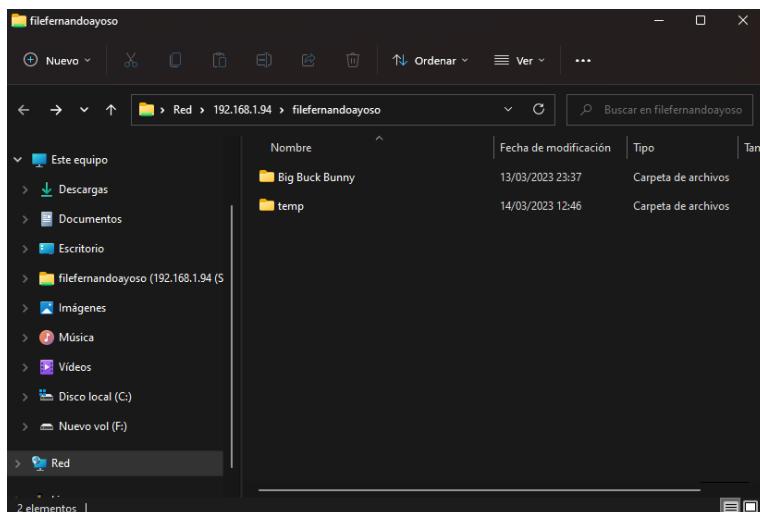


Imagen 92; Carpeta

Si entramos en la carpeta “Big Buck Bunny”, podremos reproducir la película

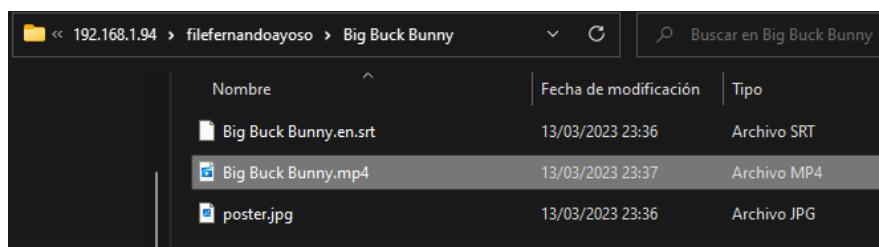


Imagen 93; Película



Imagen 94; Película (2)

6.7 - Visualizar en el resto de dispositivos

- Una vez que podemos visualizar videos y carpetas en el ordenador, vamos a instalar minidlna para poder visualizarlos en el resto de dispositivos. Así pues, instalamos:

sudo apt-get install minidlna

Tabla 22; Instalar minidlna

```
D:\ayoso\entorno\descargas$ sudo apt-get install minidlna
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ASe instalarán los siguientes paquetes adicionales:
  fontconfig libaacs0 libao0m libavcodec58 libavutil56 libbdplus0 libbluray2 libcairo-gobject2 libchromaprint1 libcodec2-0.9 libdatrie1
  libdav1d4 libdrm-amdgpu libdrm-nouveau2 libdrm-radeon1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgl1 libgl1-mesa-dri libglapi-mesa
  libglvnd0 libglx-mesa0 libglx0 libgme0 libgraphite2-3 libgsml libharfbuzz0b libid3tag0 libl1vmm1 libmp3lame0 libmpg123-0 libnorm1 libopenjp2-7 libopenmpt0
  libopus0 libpango-1.0-0 libpangocairo-1.0-0 libpangofc2-1.0-0 libpgeom-5.3-0 libpixman-1-0 librabitmq4 librsvg2-2 librsvg2-common libsensors-config libsensors5
  libshine3 libsnappy1v5 libodium23 libsoxr0 libspeex1 libsr1.4-gnutls libssh-gcrypt-4 libswresample3 libthai-data libtheora0 libtwolame0 libubrread0
  libva-drm2 libva-x11-2 libvba2 libvpdau-va-g11 libvpdau libvorbisfile2 libvpv6 libvulkani libvavpack1 libwayland-client0 libxi1-xcb1 libx264-160
  libx265-192 libxcb-dr12-0 libxcb-dr13-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxdamage1 libxfixed3
  libxrender1 libxshmfence1 libxvidcore libxf86vm1 libz3-4 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers mesa-vdpau-drivers mesa-vulkan-drivers
  ocl-icd-libopencl1 shared-mime-info va-driver-all vdpau-driver-all
Paquetes sugeridos:
  libbluray-bdj opus-tools librsvg2-bin lm-sensors speex opencl-icd nvidia-legacy-390xx-vdpau-driver nvidia-legacy-340xx-vdpau-driver
Se instalarán los siguientes paquetes NUEVOS:
  fontconfig libaacs0 libao0m libavcodec58 libavutil56 libbdplus0 libbluray2 libcairo-gobject2 libchromaprint1 libcodec2-0.9 libdatrie1
  libdav1d4 libdrm-amdgpu libdrm-nouveau2 libdrm-radeon1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgl1 libgl1-mesa-dri libglapi-mesa
  libglvnd0 libglx-mesa0 libglx0 libgme0 libgraphite2-3 libgsml libharfbuzz0b libid3tag0 libl1vmm1 libmp3lame0 libmpg123-0 libnorm1 libopenjp2-7 libopenmpt0
  libopus0 libpango-1.0-0 libpangocairo-1.0-0 libpangofc2-1.0-0 libpgeom-5.3-0 libpixman-1-0 librabitmq4 librsvg2-2 librsvg2-common libsensors-config libsensors5
  libshine3 libsnappy1v5 libodium23 libsoxr0 libspeex1 libsr1.4-gnutls libssh-gcrypt-4 libswresample3 libthai-data libtheora0 libtwolame0 libubrread0
  libva-drm2 libva-x11-2 libvba2 libvpdau-va-g11 libvpdau libvorbisfile2 libvpv6 libvulkani libvavpack1 libwayland-client0 libxi1-xcb1 libx264-160
  libx265-192 libxcb-dr12-0 libxcb-dr13-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxdamage1 libxfixed3
  libxrender1 libxshmfence1 libxvidcore libxf86vm1 libz3-4 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers mesa-vdpau-drivers mesa-vulkan-drivers minidlna
0 actualizados, 102 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 69,7 MB de archivos.
Se utilizarán 243 MB de espacio de disco adicional después de esta operación.
Desea continuar? [S/n]
```

Imagen 95; Instalar minidlna

- Como hemos visto hasta ahora, detenemos el servicio para editar sus archivos de configuración.

sudo service minidlna stop

Tabla 23; Detener servicio minidlna

```
pi@ayoso:/mnt/disco/Descargas $ sudo service minidlna stop
```

Imagen 96; Detener servicio minidlna

3. Editamos el archivo de configuración con el comando:

```
sudo nano /etc/minidlna.conf
```

Tabla 24; Editar archivo configuración minidlna

El archivo es como este:

```
GNU nano 5.4                               /etc/minidlna.conf
# This is the configuration file for the MiniDLNA daemon, a DLNA/UPnP-AV media
# server.
#
# Unless otherwise noted, the commented out options show their default value.
#
# On Debian, you can also refer to the minidlna.conf(5) man page for
# documentation about this file.
#
# Specify the user name or uid to run as (root by default).
# On Debian system command line option (from /etc/default/minidlna) overrides this.
#user=minidlna

# Path to the directory you want scanned for media files.
#
# This option can be specified more than once if you want multiple directories
# scanned.
#
# If you want to restrict a media_dir to a specific content type, you can
# prepend the directory name with a letter representing the type (A, P or V),
# followed by a comma, as so:
# * "A" for audio (eg. media_dir=A,/var/lib/minidlna/music)
# * "P" for pictures (eg. media_dir=P,/var/lib/minidlna/pictures)
# * "V" for video (eg. media_dir=V,/var/lib/minidlna/videos)
# * "PV" for pictures and video (eg. media_dir=PV,/var/lib/minidlna/digital_camera)
media_dir=/var/lib/minidlna

# Set this to merge all media_dir base contents into the root container
# (The default is no.)
#merge_media_dirs=no

# Path to the directory that should hold the database and album art cache.
#db_dir=/var/cache/minidlna

# Path to the directory that should hold the log file.
#log_dir=/var/log/minidlna

# Type and minimum level of importance of messages to be logged.
#
# The types are "artwork", "database", "general", "http", "inotify",
# "metadata", "scanner", "ssdp" and "tivo".
#
# The levels are "off", "fatal", "error", "warn", "info" or "debug".
#
# "off" turns off logging entirely, "fatal" is the highest level of importance
# and "debug" the lowest.
```

Imagen 97; Editar archivo configuración minidlna

4. Debemos desmarcar la línea donde se visualiza “user” y cambiar el usuario que viene por defecto por nuestro usuario, es decir, por pi:

```
user=pi
```

Imagen 98; “user”

5. Debemos cambiar la línea donde se visualiza “media_dir” y escribir la ruta donde se encuentre nuestra carpeta con vídeos, que en mi caso es /mnt/disco/Descargas

```
media_dir=/mnt/disco/Descargas
```

Imagen 99; “media_dir”

6. Desmarcamos la línea “friendly_name” y escribir un nombre que será visible para todos los dispositivos para reproducir contenido

```
# Defaults to "hostname.us"  
friendly_name=serverayoso
```

Imagen 100; “friendly_name”

7. Por último, desmarcamos la línea “inotify=yes”

```
# Automatic di  
inotify=yes
```

Imagen 101; “inotify=yes”

8. Guardamos e iniciamos el servicio:

```
sudo service minidlna start
```

Tabla 25; Iniciar servicio minidlna

```
pi@ayoso:/mnt/disco/Descargas $ sudo service minidlna start
```

Imagen 102; Iniciar servicio minidlna

9. Escribimos este comando el cual es necesario tras reiniciar el servidor DLNA para forzar el primer escaneo de la carpeta que hayamos puesto como librería

```
sudo minidlna -R
```

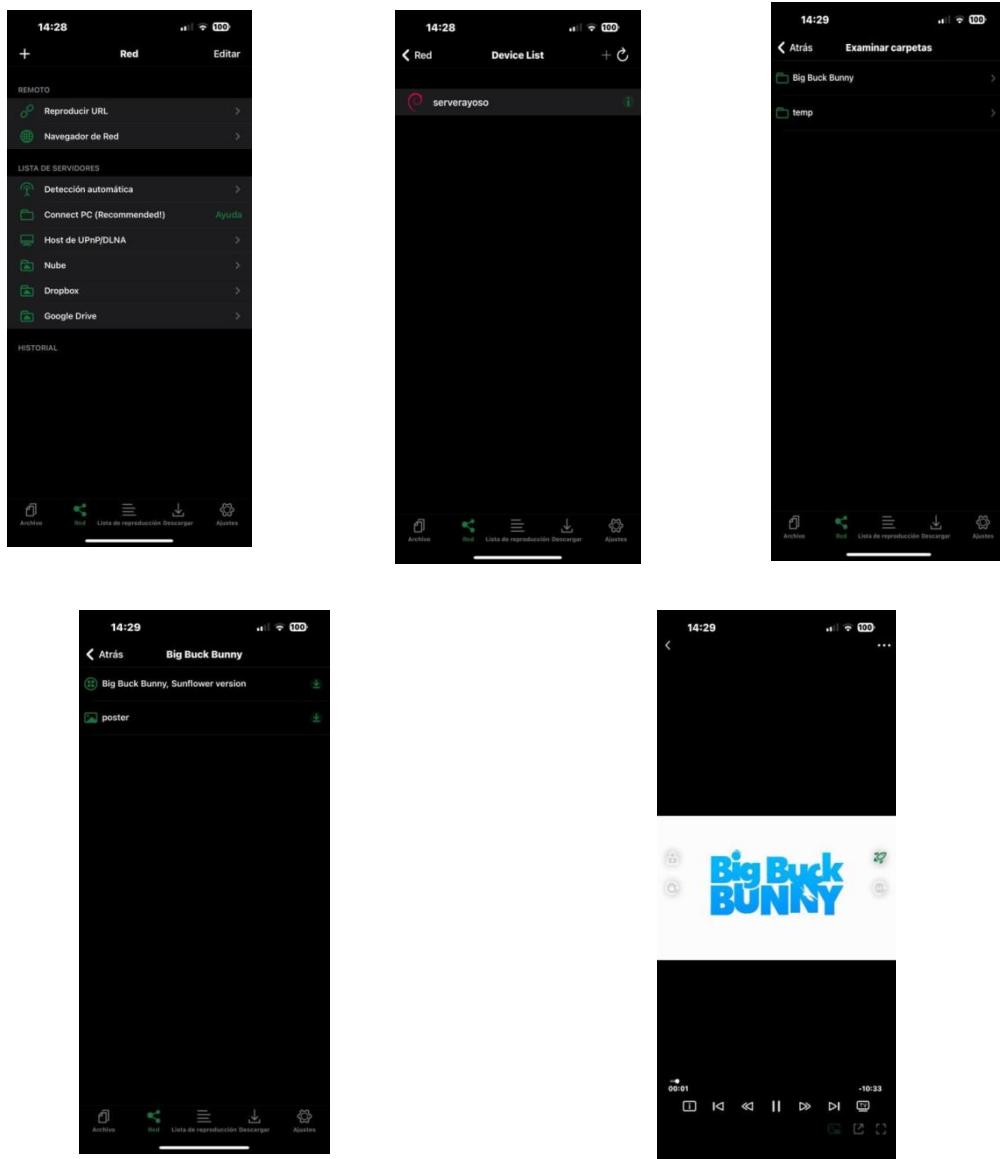
Tabla 26; Reiniciar servicio minidlna

```
pi@ayoso:/mnt/disco/Descargas $ sudo minidlnad -R
```

Tabla 27; Reiniciar servicio minidlna

10. Para su comprobación, se ha utilizado un dispositivo IOS mediante la aplicación de OPlayer¹.

Se ha procedido a escanear la red local, nos aparecerá el servidor, y las carpetas con la película.



¹ Oplayer es una aplicación móvil que permite a los usuarios reproducir videos en diferentes formatos en sus dispositivos iOS y Android

6.8 - Configurar red externa (DNS)

Todo lo realizado anteriormente se ha configurado a nivel de red local dominada por un router. Para acceder a nivel externo del router, hay que abrir unos puertos en él.

Existe una forma sencilla pero menos segura (utilizando una DMZ) y una forma más segura pero menos sencilla (redireccionando puertos)

La DMZ a groso modo significa que todo lo que venga desde fuera se va a redirigir a una IP interna de nuestra red local. Esto es inseguro ya que estaremos redirigiendo todos los puertos y todas las configuraciones, programas y demás, a internet de nuestra Raspberry Pi. Esto ocasiona que todas las contraseñas y programas no sean seguras.

Para la redirección de puertos, redirige un puerto del exterior a un puerto del interior. La transmisión de archivos (ftp) que utiliza el puerto 22, redirigimos el puerto 22 de nuestro router al puerto 22 de la Raspberry Pi, desde fuera, atacando al puerto 22, estaríamos atacando a la Raspberry Pi. Podemos cambiar el puerto exterior, por ejemplo, al 9000, y cuando ataquemos al puerto 9000 exterior, estaríamos atacando al puerto 22 de la Raspberry Pi.

Vamos a realizarlo, como es lógico, por la redirección de puertos. Para ello, escribimos en el navegador 192.168.1.1 y entramos en la configuración del router:

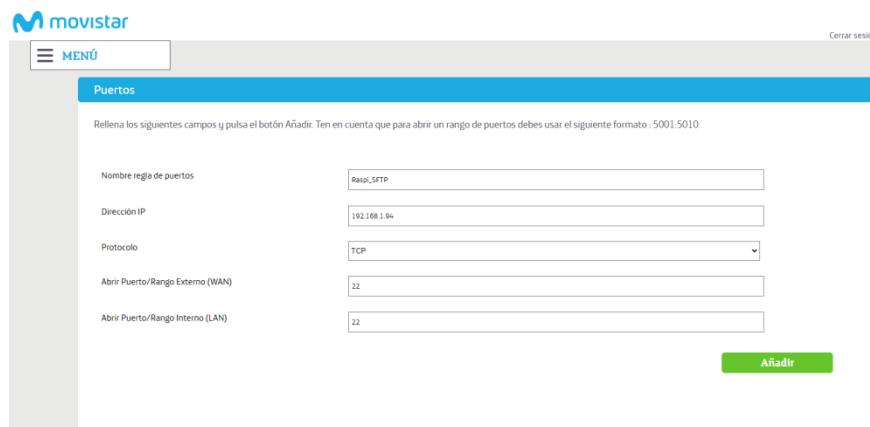


Imagen 103; Router

Nombre raspi_SFTP	Protocolo TCP	Puerto/Rango Externo 22:22	Puerto/Rango Interno 22:22	Dirección IP 192.168.1.94	Activar <input checked="" type="checkbox"/>
transmision	TCP	9091:9091	9091:9091	192.168.1.94	<input checked="" type="checkbox"/>

Imagen 104; Puertos router

La dirección IP externa irá cambiando con el tiempo, porque una dirección IP pública vale dinero, así que utilizaremos la web <https://www.duckdns.org/> que nos va a facilitar un dominio con la actualización automática de la IP.

Así que creamos el dominio que queramos, en mi caso será: <http://fernandoayoso.duckdns.org> y ya estaría registrado

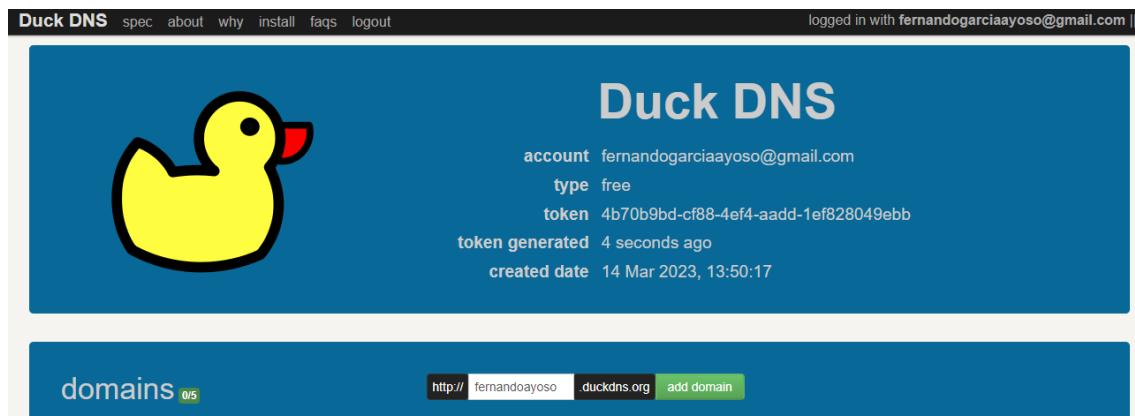


Imagen 105; Duckdns

A continuación, en la parte superior, clicamos en “install”:

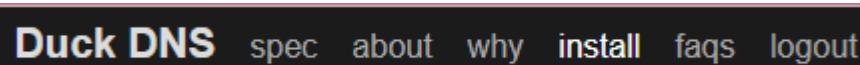


Imagen 106; Instalar duckdns

Marcamos la opción de “Linux cron” y abajo seleccionamos el dominio que queremos instalar, que en mi caso sería fernandoayoso

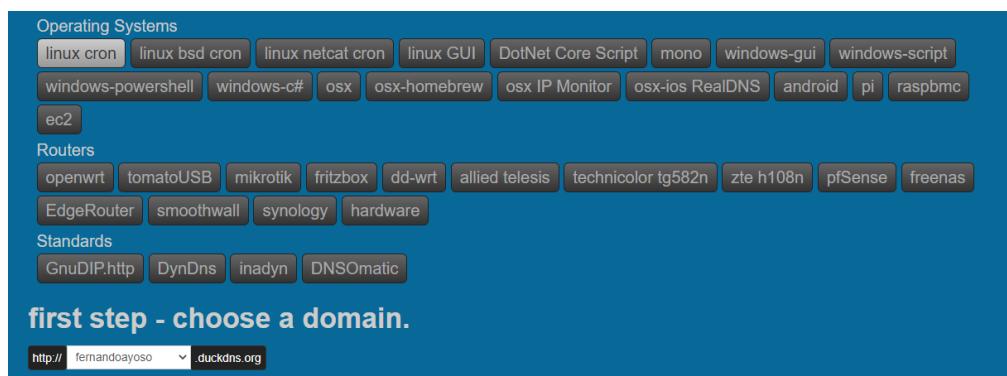


Imagen 107; Instalar duckdns (2)

Una vez seleccionado el dominio, se nos generarán una serie de instrucciones que debemos seguir:

```
mkdir duckdns
cd duckdns
sudo nano duck.sh
echo url="https://www.duckdns.org/update?domains=fernandoayoso&token=4b70b9bd-
cf88-4ef4-aadd-1ef828049ebb&ip=" | curl -k -o /duckdns/duck.log -K -
sudo chmod 700 duck.sh
crontab -e
*/5 * * * * ~/duckdns/duck.sh >/dev/null 2>&1
./duck.sh
cat duck.log
```

Tabla 28; Instalar duckdns

Al realizar “cat duck.log” nos debe aparecer un mensaje de OK

```
pi@ayoso:/duckdns $ cat duck.log
OKpi@ayoso:/duckdns $ exit
```

Tabla 29; Instalar duckdns (3)

En teoría, si abrimos el navegador y escribimos <http://fernandoayoso.duckdns.org:9091/> nos debe salir un cuadro para que pongamos las credenciales

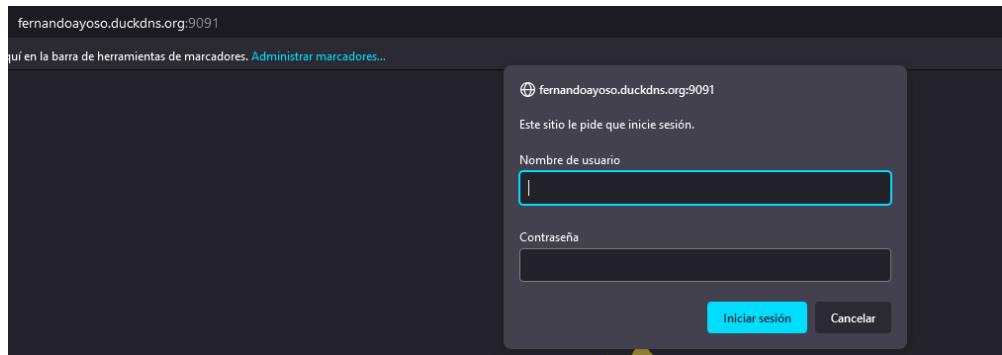


Imagen 108; Inicio de sesión

Si las credenciales son correctas, debe aparecer tal que así:

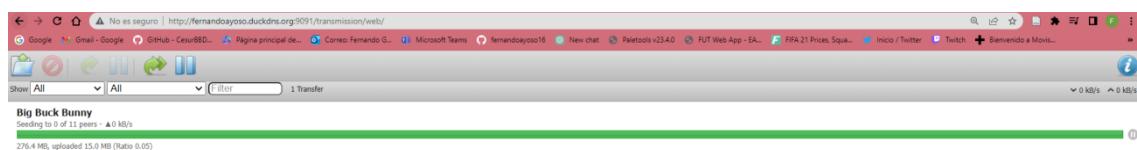


Imagen 109; Inicio de sesión correcto

También podemos conectarnos mediante SSH en la terminal:

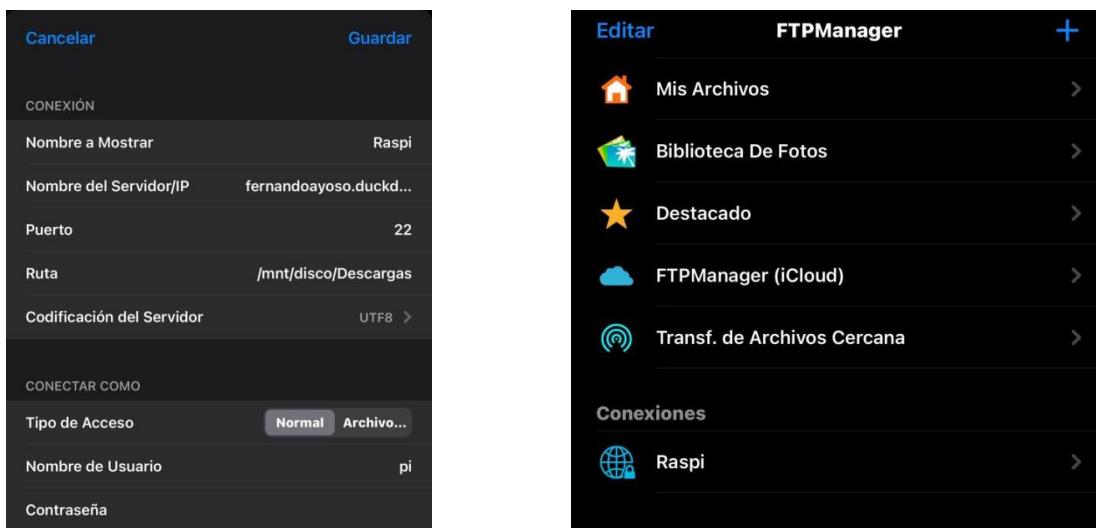
```
C:\Users\Equipo>ssh pi@fernandoayoso.duckdns.org
pi@fernandoayoso.duckdns.org's password:
Linux ayoso 5.15.84-v7l+ #1613 SMP Thu Jan 5 12:01:26 GMT 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

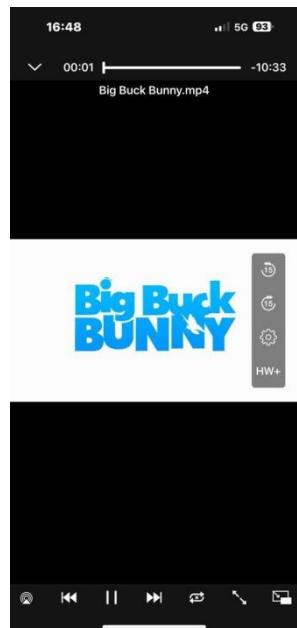
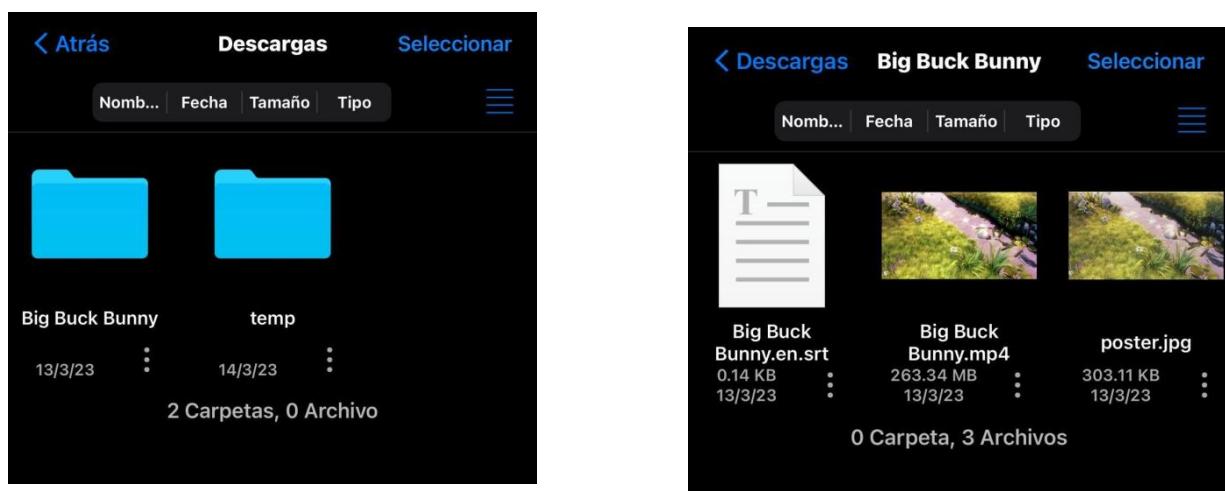
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar 14 16:07:08 2023 from 2.140.222.228
pi@ayoso:~ $ -
```

Imagen 110; Conectar mediante SSH - duckdns

Y, por supuesto, podemos utilizar nuestro dispositivo móvil. En mi caso, al tener un Iphone, he descargado la aplicación FTPManager² en el cual hemos configurado el puerto, el DNS y las credenciales. Como podemos observar, nos salen las carpetas Big Buck Bunny y temp. En la carpeta Big Buck Bunny podemos reproducir la película.



² FTP Manager es un software de gestión de archivos que permite a los usuarios transferir archivos entre un ordenador y un servidor FTP (File Transfer Protocol) de manera rápida y fácil.



6.9 - Configurar Jellyfin

1. Actualizamos el sistema:

```
[sudo apt-get update]
```

Tabla 30; Actualizar el sistema

```
pi@ayoso:~ $ sudo apt-get update
Obj:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Obj:2 http://archive.raspberrypi.org/debian bullseye InRelease
Leyendo lista de paquetes... Hecho
pi@ayoso:~ $ |
```

Imagen 111; Actualizar el sistema

2. Instalamos Docker:

```
[sudo apt-get install docker.io]
```

Tabla 31; Insalar Docker

```
pi@ayoso:~ $ sudo apt-get install docker.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apparmor cgroupsfs-mount containerd git git-man liberror-perl libintl-perl libintl-xs-perl libmodule-find-perl
  libmodule-scandeps-perl libproc-processtable-perl libsort-naturally-perl libterm-readkey-perl needrestart runc tini
Paquetes sugeridos:
  apparmor-profiles-extra apparmor-utils containernetworking-plugins docker-doc aufs-tools btrfs-progs debootstrap
  rinse rootlesskit xfsprogs zfs-fuse | zfsutils-linux git-daemon-run | git-daemon-sysvinit git-doc git-el git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn needrestart-session | libnotify-bin iucode-tool
Paquetes recomendados:
  criu
Se instalarán los siguientes paquetes NUEVOS:
  apparmor cgroupsfs-mount containerd docker.io git git-man liberror-perl libintl-perl libintl-xs-perl
  libmodule-find-perl libmodule-scandeps-perl libproc-processtable-perl libsort-naturally-perl libterm-readkey-perl
  needrestart runc tini
0 actualizados, 17 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 51,2 MB de archivos.
Se utilizarán 230 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] |
```

Imagen 112; Instalar Docker

Y observamos su estado mediante el comando:

```
[sudo systemctl status docker]
```

Tabla 32; Estado de Docker

```
pi@ayoso:/ $ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-03-15 21:56:32 CET; 7min ago
     TriggeredBy: ● docker.socket
      Docs: https://docs.docker.com
 Main PID: 2652 (dockerd)
    Tasks: 10
      CPU: 3.207s
     CGroup: /system.slice/docker.service
             └─2652 /usr/sbin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

mar 15 21:56:29 ayoso dockerd[2652]: time="2023-03-15T21:56:29.080956251+01:00" level=info msg="cResolverWrapper: send"
mar 15 21:56:29 ayoso dockerd[2652]: time="2023-03-15T21:56:29.081928770+01:00" level=info msg="ClientConn switching ba"
mar 15 21:56:29 ayoso dockerd[2652]: time="2023-03-15T21:56:29.453308695+01:00" level=warning msg="Unable to find memor"
mar 15 21:56:29 ayoso dockerd[2652]: time="2023-03-15T21:56:29.454613825+01:00" level=info msg="Loading containers: sta"
mar 15 21:56:30 ayoso dockerd[2652]: time="2023-03-15T21:56:30.812165102+01:00" level=info msg="Default bridge (dockero"
mar 15 21:56:31 ayoso dockerd[2652]: time="2023-03-15T21:56:31.545862324+01:00" level=info msg="Loading containers: don"
mar 15 21:56:31 ayoso dockerd[2652]: time="2023-03-15T21:56:31.738849657+01:00" level=info msg="Docker daemon" commit=>
mar 15 21:56:31 ayoso dockerd[2652]: time="2023-03-15T21:56:31.741129657+01:00" level=info msg="Daemon has completed in"
mar 15 21:56:32 ayoso systemd[1]: Started Docker Application Container Engine.
mar 15 21:56:32 ayoso dockerd[2652]: time="2023-03-15T21:56:32.115457083+01:00" level=info msg="API listen on /run/doc<br>Lines 1-21 (END)
```

Imagen 113; Estado de Docker

3. Una vez que hayamos instalado Docker, crearemos un archivo .yml llamado docker-compose mediante el comando:

```
| sudo nano docker-compose.yml
```

Tabla 33; Crear archivo docker-compose

```
pi@ayoso:/ $ cd mnt/  
pi@ayoso:/mnt $ ls  
disco docker-compose.yml local storage
```

Imagen 114; Crear archivo docker-compose

4. El docker-compose que se va a proceder a ejecutar tendrá esta estructura:

```
version: '3'

services:
  transmission:
    container_name: transmission
    image: linuxserver/transmission
    restart: always
    networks:
      - jellyfin
    volumes:
      - /mnt/disco/torrents:/downloads
      - /mnt/disco/transmission/config:/config
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Madrid
```

```

ports:
- 9092:9091
- 51414:51413
- 51414:51413/udp

jellyfin:
  container_name: jellyfin
  image: jellyfin/jellyfin
  restart: always
  networks:
    - jellyfin
  volumes:
    - /mnt/disco/Descargas:/config
    - /mnt/disco/Descargas:/transcode
    - /mnt/disco/Descargas:/data
    - /mnt/disco/Descargas:/media/Movies
  environment:
    - TZ=Europe/Madrid

ports:
- 8096:8096
- 8920:8920
networks:
  jellyfin:
    driver: bridge

```

Tabla 34; Docker-compose

Este archivo docker-compose contiene la configuración para levantar dos contenedores: uno para Transmission (un cliente BitTorrent) y otro para Jellyfin (un servidor de medios).

En cuanto a la configuración de Transmission, se especifica que el contenedor se llamará "transmission", se usará la imagen "linuxserver/transmission" y siempre se reiniciará en caso de fallo.

También se especifica que el contenedor estará en la red "jellyfin" y se montarán dos volúmenes en el host: uno para los archivos descargados y otro para la configuración de Transmission. Además, se establecen las variables de entorno PUID, PGID y TZ para el usuario y grupo del contenedor y la zona horaria.

En cuanto a la configuración de Jellyfin, se especifica que el contenedor se llamará "jellyfin", se usará la imagen "jellyfin/jellyfin" y siempre se reiniciará en caso de fallo. También se especifica que el contenedor estará en la red "jellyfin" y se montarán varios volúmenes en el host para la configuración de Jellyfin, transcodificación de archivos, datos y películas. Además, se establece la variable de entorno TZ para la zona horaria.

Finalmente, se define una red llamada "jellyfin" que será utilizada por ambos contenedores y se especifica que su driver será "bridge".

Además, se exponen los puertos 9091, 51413 (TCP y UDP) para Transmission y los puertos 8096 y 8920 para Jellyfin en el host, lo que permitirá el acceso a las interfaces web de ambos servicios desde fuera del contenedor.

5. Ejecutamos el archivo mediante el comando:

```
[ docker-compose up -d ]
```

Tabla 35; Docker-compose up -d

6. Si realizamos un docker ps, se observarán los contenedores funcionando:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
c250a2fe800	linuxserver/transmission	"/init"	40 minutes ago	Up 40 minutes	0.0.0.0:9092->9091/tcp, :::9092->9091/tcp, 0.0.0.0:51414->51413/tcp, 0.0.0.0:51414->51413/udp, :::51414->51413/tcp, :::51414->51413/tcp
13/udp	transmission	"jellyfin/jellyfin"	40 minutes ago	Up 40 minutes (healthy)	0.0.0.0:8096->8096/tcp, :::8096->8096/tcp, 0.0.0.0:8920->8920/tcp, :::8920->8920/tcp

Imagen 115; Docker ps

7. Iniciamos el navegador y escribimos la IP: 192.168.1.94:8096 o bien, mediante <http://fernandoayoso.duckdns.org:8096/> En ella, encontraremos una web en la que tendremos que configurar la ruta donde se encontrará la carpeta, el idioma, etc. Una vez que se realicen las configuraciones previas, nos pedirá un inicio de sesión.

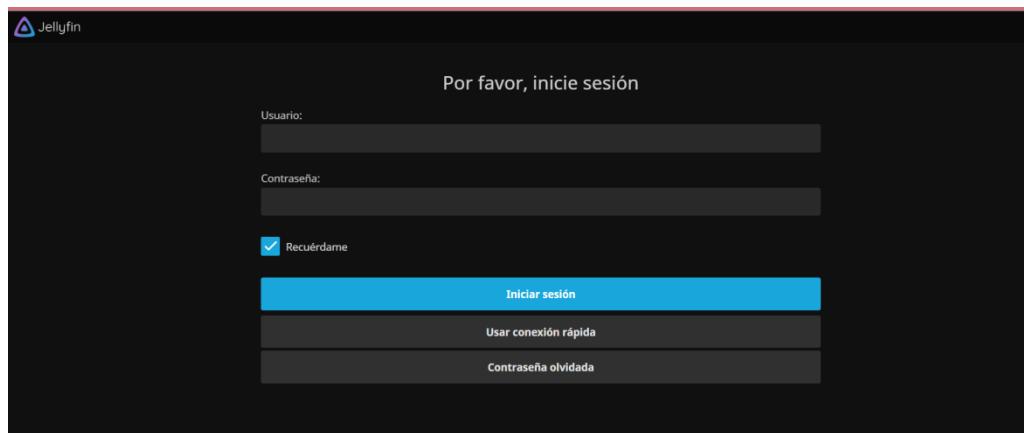


Imagen 116; Inicio de sesión en Jellyfin

8. Al iniciar sesión, podremos encontrar las películas que se encuentran en el servidor de transmission:

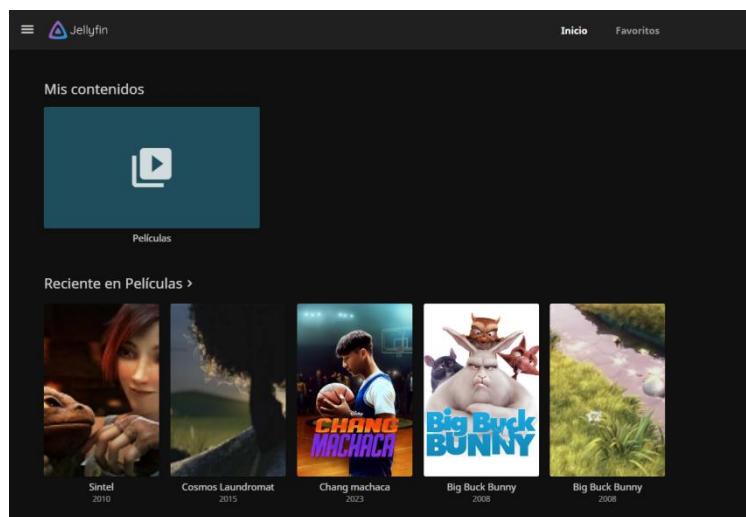


Imagen 117; Películas Jellyfin

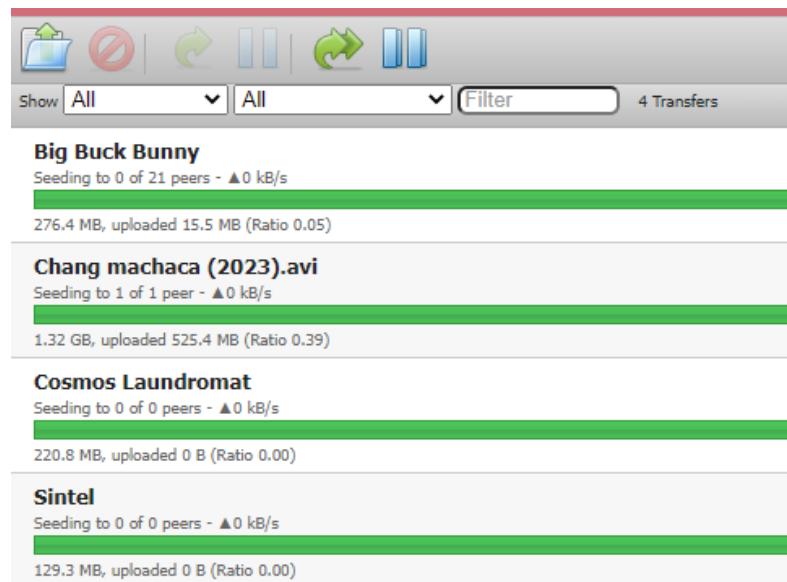


Imagen 118; Películas Torrent

9. En el router (192.168.1.1) añadiremos una regla de puerto (8096) para permitir que Jellyfin se comunique con Internet a través de este puerto específico. Esto permitirá que el tráfico de red llegue a la aplicación a través del puerto 8096 y permitirá que los dispositivos externos accedan a la biblioteca multimedia alojada en el servidor.



Imagen 119; Puerto 8096

10. Si iniciamos una película, se puede comprobar que esta inicia de forma correcta:



Imagen 120; Comprobación película Jellyfin

11. También, se puede descargar la película que queramos. Para ello, clic derecho sobre la película que se quiera descargar

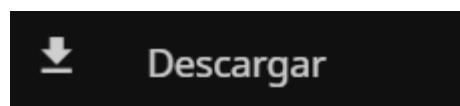


Imagen 121; Descargar película



Imagen 122; Descargando película

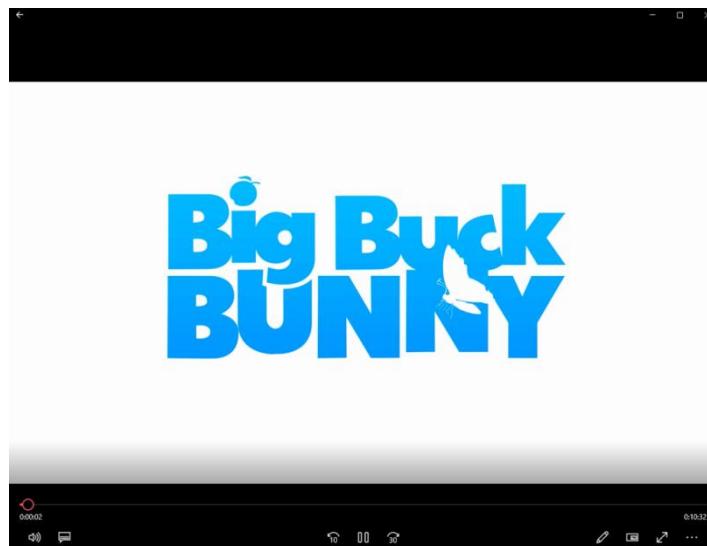


Imagen 123; Película descargada

6.10 - Crear usuarios en Jellyfin

1. Nos dirigimos a panel de control

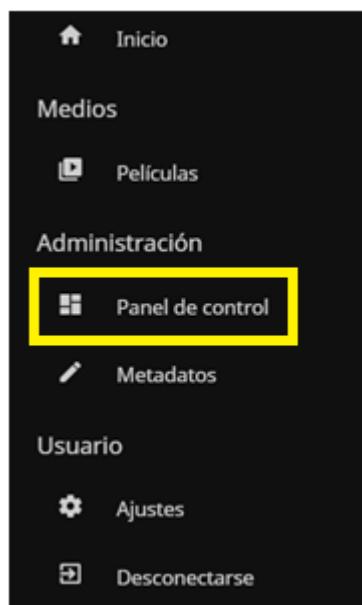


Imagen 124; Panel de control Jellyfin

2. Apartado de usuarios

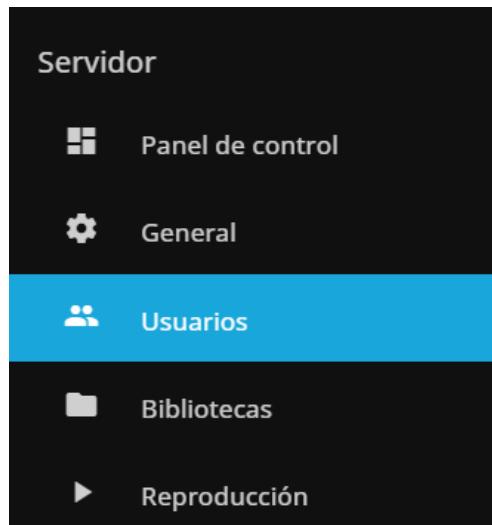


Imagen 125; Usuarios Jellyfin

3. Creamos el usuario: usuario_1 y guardamos.

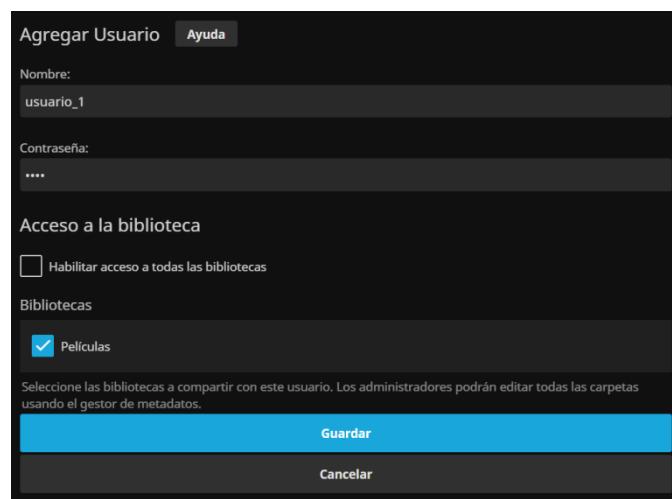


Imagen 126; Crear "usuario_1"

- Una vez creado, nos aparecerán una multitud de opciones, en el que se debe asegurar que NO esté marcada: "Permitir a este usuario administrar el servidor"

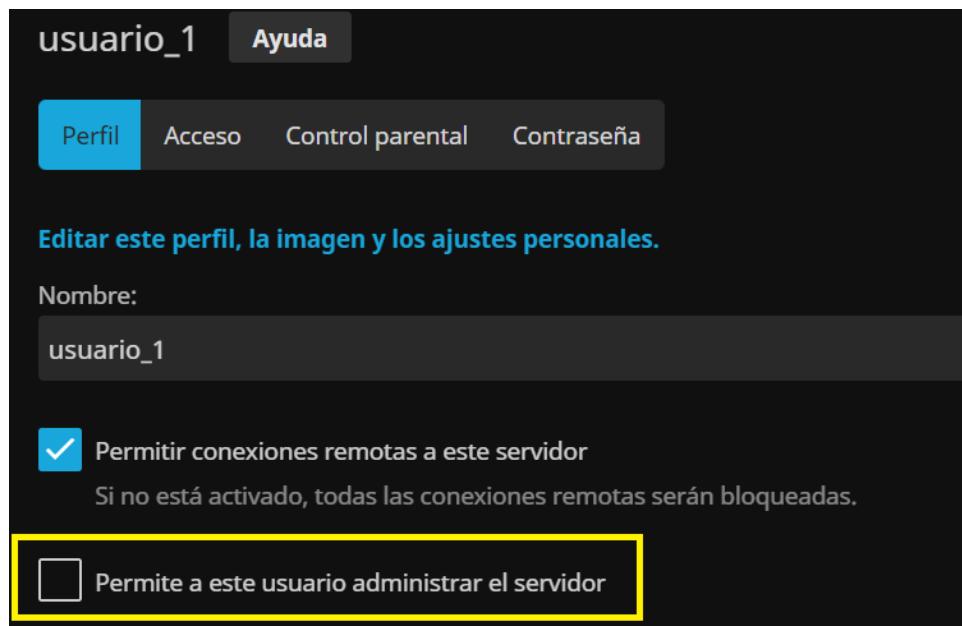


Imagen 127; NO permitir al usuario administrar el servidor

- Como podemos observar, podemos visualizar los usuarios que hemos creado:

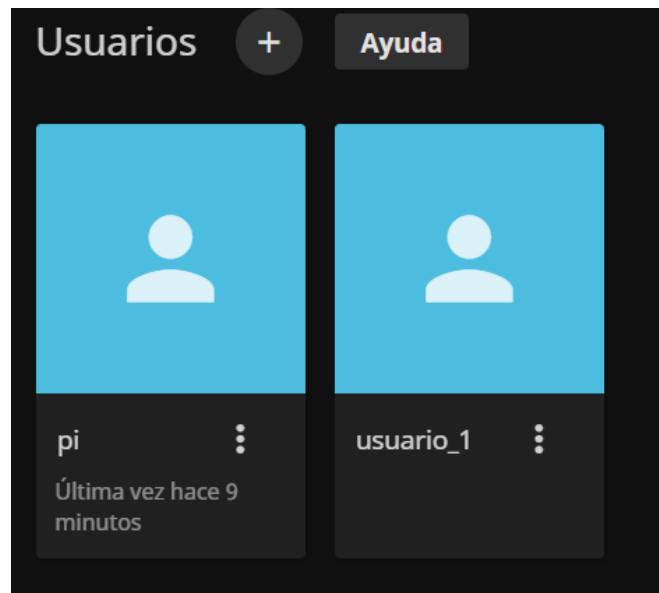


Imagen 128; Visualizar usuarios creados

6. Si iniciamos sesión con el usuario_1, podemos observar de que puede observar y descargarse películas, pero nunca borrarlas.

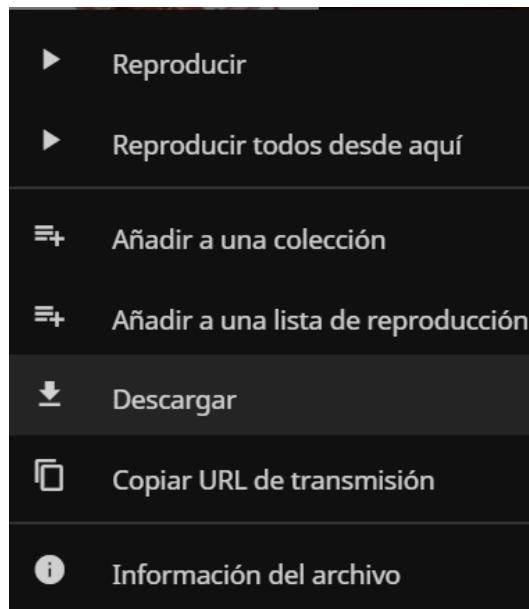


Imagen 129; Opciones usuario_1



Imagen 130; Descarga usuario_1



Imagen 131; Visualizar película usuario_1

6.11 - Netdata

Netdata es una herramienta de monitoreo de rendimiento en tiempo real que permite a los usuarios recopilar y visualizar datos de monitoreo del sistema en una interfaz web. Es intuitiva, lo que es genial para sistemas con recursos limitados.

Netdata puede monitorear una amplia variedad de métricas de sistema, incluyendo uso de CPU, RAM y disco, utilización de la red, actividad de servicios y aplicaciones, y mucho más. También puede ser utilizado para monitorear el estado de servidores web y bases de datos, proporcionando información valiosa sobre la salud y el rendimiento del sistema.

La herramienta es muy personalizable y se puede configurar para adaptarse a las necesidades específicas de cada usuario. Además, cuenta con una amplia gama de opciones de visualización de datos, lo que permite a los usuarios ver fácilmente las tendencias y el rendimiento en tiempo real.

Para instalar Netdata, se utilizará el siguiente comando en la terminal:

```
bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

Tabla 36; Netdata

Imagen 132; Instalar Netdata

Netdata, por defecto, asigna el puerto 19999, el cual se puede comprobar con este comando:

```
sudo netstat -tulnp | grep 19999
```

Tabla 37; Puertos Netdata

pi@ayoso:~ \$ sudo netstat -tulnp grep 1999	tcp 0 0 0.0.0.0:1999 0.0.0.0:*	LISTEN 7868/netdata
	tcp6 0 0 :::1999 :::*	LISTEN 7868/netdata

Imagen 133; Puertos Netdata

Por tanto, en el router, debemos habilitar el puerto 1999.



Imagen 134; Habilitar puerto Netdata

Por tanto, en mi caso, debemos escribir en el navegador: <http://fernandoayoso.duckdns.org:1999/>

Al acceder a la dirección IP y puerto especificados, o mediante el DNS pre-configurado, se visualizará una interfaz de usuario visual en el navegador que permitirá interactuar con Netdata.

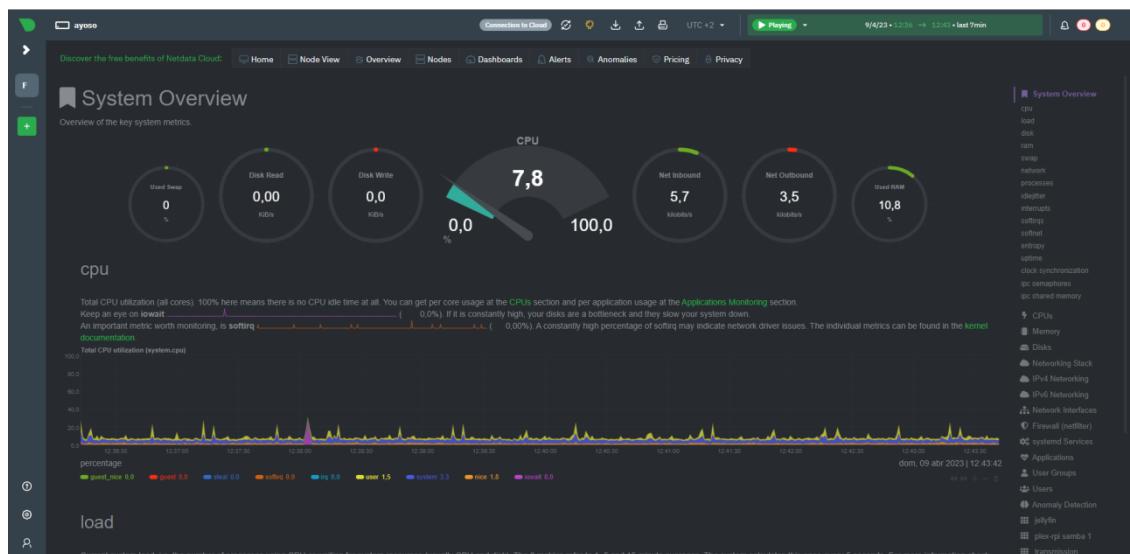


Imagen 135; Netdata interfaz

Como se puede observar en la interfaz gráfica de Netdata, existen diferentes secciones que permiten visualizar distintos aspectos del rendimiento del sistema. A continuación, se describen algunas de las secciones más comunes:

- Home: es la página de inicio donde se pueden encontrar estadísticas generales del sistema y un resumen de los diferentes módulos de Netdata instalados.
- Node View: proporciona una vista detallada de las métricas de rendimiento de un solo nodo. Aquí se pueden ver estadísticas de CPU, memoria, discos, red, entre otros.
- Overview: es una vista general de todos los nodos conectados, lo que permite monitorear múltiples sistemas a la vez.
- Nodes: esta opción proporciona información detallada sobre todos los nodos conectados, como la salud del sistema, las alertas, el uso de la red y la actividad de E/S.

- Dashboards: aquí se pueden encontrar paneles predefinidos que permiten ver la información de monitoreo de diferentes sistemas, como servidores web, bases de datos, etc.
- Alerts: en esta sección se pueden configurar las alertas para notificar al usuario cuando se alcanzan ciertos umbrales de rendimiento.
- Anomalies: aquí se pueden detectar anomalías y tendencias inusuales en el rendimiento del sistema.
- Pricing: esta opción muestra los planes de precios y las características para aquellos usuarios que deseen utilizar Netdata en un entorno empresarial.
- Privacy: proporciona información sobre la política de privacidad y seguridad de Netdata.

Estas son solo algunas de las secciones disponibles en Netdata, pero hay muchas más que permiten visualizar otros aspectos del rendimiento del sistema.



Imagen 136; Reproducción película Netdata

Cuando se reproduce una película en Jellyfin, es común observar un aumento en el uso de la CPU de la Raspberry Pi. Esto se debe a que la Raspberry Pi está descodificando el contenido de video y luego lo está enviando a la pantalla. El proceso de decodificación requiere una gran cantidad de procesamiento y, por lo tanto, aumenta la carga en la CPU de la Raspberry Pi. Es importante tener en cuenta que el grado de aumento de la carga en la CPU depende de la resolución y la calidad del video que se está reproduciendo. Por lo tanto, es posible que se observe una carga de CPU más alta al reproducir videos de alta definición o de alta calidad en Jellyfin.



Imagen 137; Comprobación subir película Netdata

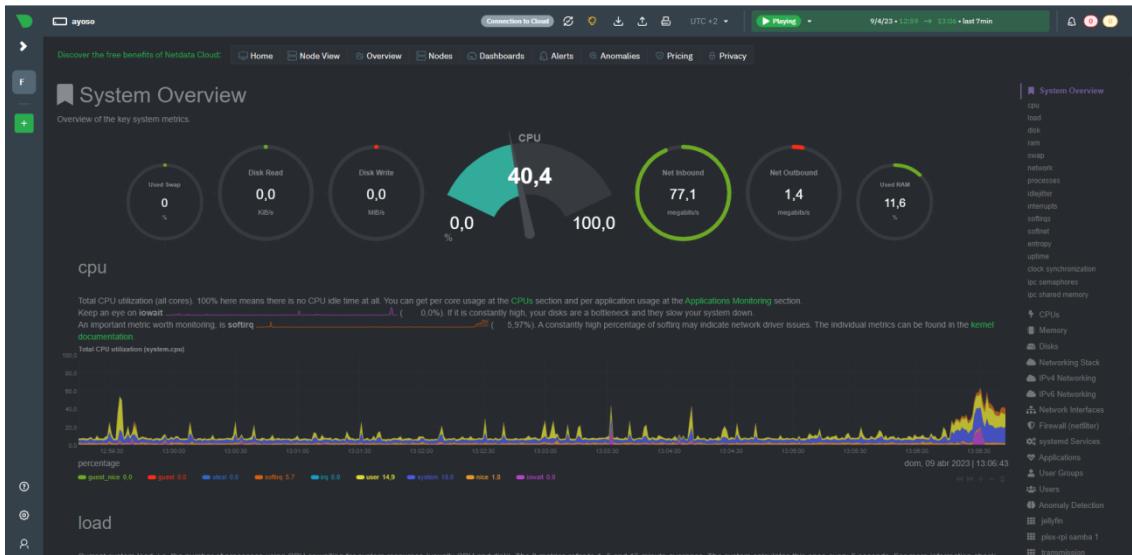


Imagen 138; Comprobación subir película Netdata (2)

Al subir una película a Torrent, la Raspberry Pi puede aumentar su uso de CPU. Esto se debe a que la aplicación necesita el archivo de video para que sea compatible con diferentes dispositivos y resoluciones. La transcodificación (conversión de un archivo multimedia digital, como un archivo de vídeo) puede ser un proceso intensivo en recursos, especialmente si la película está en una resolución alta o tiene una tasa de bits alta. Es importante tener en cuenta que cuanto más potente sea la Raspberry Pi, más rápido será el proceso de transcodificación y menos afectará el rendimiento general del dispositivo. Además, es recomendable elegir un formato de archivo que no requiera transcodificación para evitar problemas de rendimiento.

6.11.1 – Error Netdata

Cabe la posibilidad que si no iniciamos el servidor durante varios días, Netdata deje de funcionar o muestre un mensaje como este:

```
● netdata.service - Real time performance monitoring
   Loaded: loaded (/lib/systemd/system/netdata.service; enabled; vendor preset: enabled)
   Active: activating (auto-restart) (Result: exit-code) since Mon 2023-04-17 21:48:50 CEST; 13s ago
     Process: 4284 ExecStartPre=/bin/mkdir -p /opt/netdata/var/cache/netdata (code=exited, status=0/SUCCESS)
     Process: 4285 ExecStartPre=/bin/chown -R netdata /opt/netdata/var/cache/netdata (code=exited, status=0/SUCCESS)
     Process: 4286 ExecStartPre=/bin/mkdir -p /run/netdata (code=exited, status=0/SUCCESS)
     Process: 4287 ExecStartPre=/bin/chown -R netdata /run/netdata (code=exited, status=0/SUCCESS)
     Process: 4288 ExecStart=/opt/netdata/usr/sbin/netdata -P /run/netdata/netdata.pid -D (code=exited, status=1/FAILURE)
      Main PID: 4288 (code=exited, status=1/FAILURE)
        CPU: 446ms

abr 17 21:48:50 ayoso [4288]: NETDATA SHUTDOWN: in      0 ms, remove pid file - next: free openssl structures
abr 17 21:48:50 ayoso [4288]: NETDATA SHUTDOWN: in      0 ms, free openssl structures - next: remove incomplete shutdown file - next: exit
abr 17 21:48:50 ayoso [4288]: NETDATA SHUTDOWN: in      0 ms, remove incomplete shutdown file - next: exit
abr 17 21:48:50 ayoso [4288]: NETDATA SHUTDOWN: completed in 106 ms - netdata is now exiting - bye bye...
```

Imagen 139; Error Netdata

Este error muestra que el servicio de Netdata está activando y reiniciando automáticamente, pero no se está iniciando correctamente y se está deteniendo. El proceso principal de Netdata ha salido con un estado de fallo, lo que indica que se ha producido un problema en la ejecución.

Una posible causa de este problema podría ser un conflicto con otro servicio que ya esté utilizando el puerto 19999, que es el puerto por defecto utilizado por Netdata. También podría haber un problema con la instalación o la configuración de Netdata.

Para resolverlo, se puede observar los registros de Netdata en el archivo */var/log/netdata/error.log* para obtener más información sobre el problema o simplemente se debería escribir este comando para ver si se produce algún error o mensaje de advertencia adicional:

```
wget -O /tmp/netdata-kickstart.sh https://my-netdata.io/kickstart.sh && sh /tmp/netdata-kickstart.sh --stable-channel --claim-token M5mvIUI9DAoBM42m6McWn8KB7Px17QMcaWoeEs9tuvKOusbCejKAHRKIfhulArMaatcgHtdJ4iZgEpoxqOs_bfhWOoMEpgAHP7MUq2bE0ZkFYpF-RsPfGOPil6aUtMhx5SvPF6I --claim-url https://app.netdata.cloud
```

Tabla 38; Comando error Netdata

Este comando descarga el archivo kickstart.sh desde <https://my-netdata.io/kickstart.sh> y lo guarda en el directorio temporal /tmp/netdata-kickstart.sh. A continuación, ejecuta el archivo kickstart.sh utilizando el shell de Unix, pasando algunos argumentos a la línea de comando.

Los argumentos que se pasan a kickstart.sh configuran Netdata para usar el canal estable (--stable-channel) y proporcionan un token de reclamo (--claim-token) y una URL de reclamo (--claim-url). Estos detalles son necesarios para que Netdata se registre y se pueda acceder a través de la nube de Netdata.

Posteriormente, escribiremos este otro:

```
sudo /opt/netdata/usr/sbin/netdata -D
```

Tabla 39; Comando error Netdata (2)

Este, inicia la ejecución del servicio de Netdata en modo daemon (en segundo plano) con el argumento "-D", lo que significa que se ejecutará como un proceso en segundo plano.

Una vez que se ejecutan los dos comandos, Netdata se instalará y configurará automáticamente en el servidor, y estará disponible en <http://fernandoayoso.duckdns.org:19999> (en mi caso) para monitorear el rendimiento del sistema. También se registrará en la nube de Netdata, lo que permitirá acceder a los datos de rendimiento en línea desde cualquier lugar.

7. Conclusión

El proyecto realizado ha sido un éxito en la implementación de una solución tecnológica eficiente y segura para el acceso centralizado al contenido multimedia. La Raspberry Pi ha sido utilizada de manera óptima, aprovechando su capacidad como servidor NAS con Samba, servidor torrent y servidor Jellyfin con Docker. Además, se ha implementado un DNS personalizado para mejorar la accesibilidad desde cualquier dispositivo conectado a internet.

La sincronización automática de las descargas de torrent con Jellyfin ha mejorado significativamente la experiencia del usuario al eliminar la necesidad de transferir manualmente los archivos descargados a la biblioteca multimedia. La implementación de medidas de seguridad ha sido un aspecto fundamental en el proyecto para garantizar la protección del contenido multimedia almacenado en la Raspberry Pi.

Además, la integración de Netdata en este proyecto también ha sido importante, ya que ha proporcionado un manejo del monitoreo y análisis de rendimiento en tiempo real para la Raspberry Pi y los servicios que se ejecutan en ella.

Esto ha permitido conocer el uso de recursos y la salud de la Raspberry Pi, lo que ha ayudado en la identificación de problemas y en la optimización del rendimiento del servidor.

Se ha logrado proporcionar una solución tecnológica innovadora y accesible para el acceso centralizado al contenido multimedia, mejorando significativamente la experiencia del usuario y garantizando la seguridad del contenido almacenado.

Más allá de los objetivos específicos, este proyecto también ha fomentado el aprendizaje y la aplicación de habilidades técnicas importantes como la configuración de un servidor NAS, la implementación de medidas de seguridad, el uso de Docker para la implementación de servicios, y la sincronización automática de descargas de torrent con un servidor multimedia.

Además, la implementación de un DNS personalizado ha permitido mejorar la accesibilidad al contenido multimedia, lo que puede tener un impacto positivo en el uso y disfrute del mismo por parte del usuario.

En conclusión, con la realización de este proyecto se ha pretendido demostrar el potencial de la Raspberry Pi como un dispositivo versátil y de medio-bajo coste para la implementación de soluciones tecnológicas innovadoras y eficientes en diferentes ámbitos, incluyendo el almacenamiento y acceso a contenido multimedia.

8. Bibliografía

Bricolaje. (s. f.). *Cómo conectar un disco duro a la Raspberry Pi (y por qué debería hacerlo)*.

<https://es.ephesossoftware.com/articles/diy/how-to-connect-a-hard-drive-to-raspberry-pi-and-why-you-should.html>. Artículo

Daniel González Gutiérrez. (2021, 25 febrero). *NAS Casero con Raspberry Pi (desde 0) 2021*

<https://www.youtube.com/watch?v=5nIDqWHNkWc>. VideoYouTube

Duck DNS. (s. f.). <https://www.duckdns.org/>

Fernández, Y. (2021, 19 enero). *Cómo crear tu propio Netflix con un servidor DLNA, Google Drive o Plex*. Xataka. <https://www.xataka.com/basics/como-crear-tu-propio-netflix-servidor-dlna-google-drive-plex>. Artículo

Ltd, R. P. (s. f.). OS –. Raspberry Pi. <https://www.raspberrypi.com/software/>

Netdata: Monitoring and troubleshooting transformed. (s. f.). <https://www.netdata.cloud/>

PabloKBS.(2023, 6 Febrero). *GitHub - pablokbs/plex-rpi*. <https://github.com/pablokbs/plex-rpi>. GitHub

Pelado Nerd. (2018, 20 diciembre). *DESCARGÁ TUS SERIES AUTOMÁTICAMENTE con RTORRENT y DOCKER*. https://www.youtube.com/watch?v=pZArt1yEw_o. VideoYouTube

Pelado Nerd. (2023, 7 febrero). *Mi server PLEX en DOCKER COMPLETO*.
<https://www.youtube.com/watch?v=7LiHtL-veCc>. VideoYouTube

Pelado Nerd. (2023b, febrero 7). *Mi server PLEX en DOCKER COMPLETO*.

<https://www.youtube.com/watch?v=7LiHtL-veCc>. VideoYoutube

revhardware.com. (2022, 19 diciembre). *Jellyfin y Raspberry PI: un sistema gratuito para administrar sus medios*. Hardware Revolucionario.

<https://revhardware.com/general/jellyfin-y-raspberry-pi-un-sistema-gratuito-para-administrar-sus-medios/>. Artículo

Sanchez, I. D. (2023, 10 marzo). *12 mejores webs de torrents en 2023 | Seguras y funcionan.*

vpnMentor. <https://es.vpnmentor.com/blog/los-10-mejores-sitios-web-de-torrents-de/>.Artículo>

W. (2022, 3 noviembre). Post author: WunderTech. <https://www.wundertech.net/setup-wireguard-on-a-raspberry-pi-vpn-setup-tutorial/>.Artículo>

WebTorrent. (s. f.).*Free Torrents - Public Domain, Creative*

Commons.<https://webtorrent.io/free-torrents.Artículo>

9. Anexo

Se informa que este proyecto ha sido subido al repositorio GitHub y se encuentra disponible en el siguiente enlace: https://github.com/fernandoayoso/TFG_FernandoGarciaAyoso.git

Además, se ha creado un video explicativo del proyecto que ha sido subido a YouTube y se encuentra disponible en el siguiente enlace: <https://youtu.be/ZJiUetTJVxw>

Se invita a todos los interesados a visitar los enlaces proporcionados para conocer más detalles sobre el proyecto y su implementación.