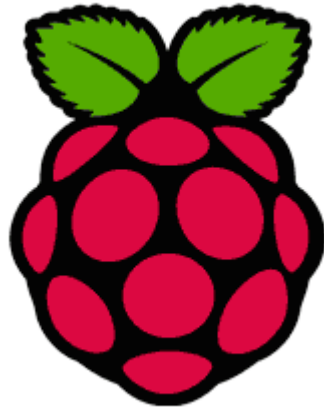




Institut Puig Castellar
Santa Coloma de Gramenet



Lanzador de emuladores retro en una consola Raspberry Pi

(Proyecto de desarrollo)

CFGM Sistemas Microinformáticos y Redes

Autores

Iván Fernández, Iker Aradillas, Eric Martin

Grupo

A

Curso académico

SMX2

Imagen de la portada

[Fuente](#)



Aquesta obra està subjecta a una llicència de
[Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de
Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

Resumen del proyecto

El proyecto a realizar consiste en la unión y configuración de diferentes componentes físicos y lógicos con el fin de obtener una consola de videojuegos, basada en Raspberry Pi, con la que el usuario pueda interactuar para la descarga de ROMs y el arranque en red de emuladores *retro*/antiguos. Aparte de lo anterior, mostraremos la configuración de un servidor *supermicro* con muchos otros servicios que den luz a nuestra página web dinámica. De cara a los objetivos para nuestro proyecto, destacamos el hecho de obtener una estructura del mismo que sea simple y con poco contenido, pero íntegra y totalmente funcional pese a la poca abundancia de funciones. Como objetivos a nivel de equipo, valoramos el hecho de obtener un trabajo durante el proyecto donde, aunque no sea perfectamente equitativo, cada miembro del equipo nunca está parado sin hacer nada. Por ende, para alcanzar el objetivo anterior, queremos aplicar una metodología estilo *waterfall* con una pizca de flexibilidad que esté complementada con diagramas de Gantt. Con estos diagramas, podemos visualizar claramente el flujo de trabajo de cada miembro y asegurarnos de que cada miembro tiene una carga de trabajo estable y continua.

Palabras clave

- Lanzador de emuladores
- Consola de videojuegos Raspberry Pi
- bash script
- Impresión 3D
- Servicios de red
- Servidores

Abstract

The project to be carried out consists of the union and configuration of different physical and logical components in order to obtain a video game console, based on Raspberry Pi, with which the user can interact for downloading ROMs and network boot of retro/old emulators. Apart from the above, we will show the configuration of a supermicro server with many other network services that give light to our dynamic web page. Regarding the objectives for our project, we emphasize the fact of obtaining a structure of the project that is simple and with little content, but complete and fully functional despite the low abundance of functions. As team level objectives, we value the fact of obtaining work during the project where, although not perfectly equitable, each team member is never standing around doing nothing. Therefore, to achieve the above objective, we want to apply a waterfall-style methodology with a pinch of flexibility that is complemented by gantt charts. With these diagrams, we can clearly visualize the workflow of each member and ensure that each member has a stable and continuous workload.

Keywords

- Emulator launcher
- Raspberry Pi video game console
- bash script
- 3D Print
- Network services
- Servers

Índice

1. Introducción al proyecto.....	10
1.1 Contexto.....	10
1.2 Justificación.....	11
1.3 Objetivos.....	11
1.31 Objetivo general.....	11
1.32 Objetivos específicos.....	11
1.4 Metodología de trabajo y planificación del proyecto.....	12
2. Descripción del proyecto.....	13
2.1 Análisis de requisitos.....	14
2.11 Requisitos funcionales.....	15
2.12 Requisitos no funcionales.....	16
2.2 Estructura del proyecto.....	16
2.21 Redes.....	16
2.22 Consola.....	17
2.23 Programas/servicios.....	18
2.3 Descripción de los componentes.....	19
2.31 Lógicos.....	19
2.31.1 Raspberry Pi OS Desktop 64 (Debian Bookworm).....	19
2.31.2 Input Remapper.....	21
2.31.3 Mesen2.....	23
2.31.4 Lanzador de emuladores (bash-scripting).....	32
2.31.5 MySQL.....	32
2.31.6 Apache2.....	33
2.31.7 Bind9.....	34
2.31.8 scp.....	34
2.31.9 Postfix.....	34
2.31.10 Dovecot.....	35
2.31.11 Kopia.....	36
2.32 Físicos.....	36
2.32.1 Raspberry Pi 5.....	36
2.32.2 Carcasa oficial para Raspberry Pi 5.....	38
2.32.3 Monitor BenQ.....	38
2.32.4 Cable HDMI estándar.....	39
2.32.5 Adaptador HDMI a microHDMI.....	39
2.32.6 Teclado y Ratón Logitech.....	39
2.32.7 Amplificador MAX98357A.....	40
2.32.8 Altavoz Weewooday 2W 8Ω.....	41
2.32.9 Cables dupont.....	42
2.32.10 Tarjeta microSDXC 64GB.....	42
2.32.11 Dualshock 4.....	42
2.32.12 Fuente de alimentación oficial de RPI (27W).....	43

2.32.13 Brazalet antiestática IFIXIT.....	44
2.32.14 Servidor Intel-Xeon (Supermicro).....	45
2.32.15 Cliente Desktop (Darwin).....	45
2.4 Definición de las funcionalidades.....	46
2.41 Descarga de ROMs a través del lanzador de emuladores.....	46
2.42 Lanzado de emuladores contenidos en la RPI 5.....	46
2.43 Acceso a una web de tienda.....	46
2.44 Envío de correos entre usuarios del dominio FIXMaster.....	46
2.5 Gastos de nuestro proyecto.....	47
3. Procesos de desarrollo.....	47
3.1 Aula Darwin (Servidor Xeon y Ordenador).....	47
3.11 Escoger servidor XEON.....	47
3.12 Discos.....	47
3.13 Instalación de Ubuntu Server 24.04.....	48
3.14 Configuración básica del servidor.....	49
3.15 Wireguard.....	49
3.16 Instalación IPMI.....	50
3.17 Instalación Kopia.....	52
3.17.1 Descargar.....	52
3.17.2 Crear el repositorio.....	53
3.17.3 Crear snapshot.....	53
3.17.4 Comprobación.....	54
3.17.5 Problemas encontrados.....	54
3.17.6 Solución.....	54
3.17.61 Eliminar repositorio antiguo.....	54
3.17.62 Crear repositorio por sftp.....	55
3.17.63 Crear snapshot.....	55
3.17.64 Comprobación.....	56
3.18 Apache2.....	56
3.18.1 Instalar Apache2.....	56
3.18.2 Configuración Apache2.....	56
3.18.21 Archivo 000-default-ssl.....	56
3.18.22 Archivo default-ssl.conf.....	57
3.18.23 Certificados.....	58
3.18.24 Activar Módulos.....	59
3.18.25 Comprobación.....	59
3.19 Bind9.....	60
3.19.1 Instalación Bind9.....	60
3.19.2 Configuración Bind9.....	60
3.19.21 Declaración de zona.....	60
3.19.22 Configuración archivo puig.es.hosts.....	60
3.19.3 Añadir Servidor de Correo.....	61
3.19.4 Reiniciar el servicio.....	61
3.19.5 Comprobación.....	61

3.19.6 Añadir DNS para VPN.....	62
3.19.62 DNS con vistas / Split Horizon.....	62
3.19.62.1 Archivo “/etc/bind/named.conf.local”.....	63
3.19.62.2 Archivo “guest/puig.es.hosts” (VPN).....	63
3.19.62.3 Archivo “trusted/puig.es.hosts” (Red LAN).....	63
3.110 Página web.....	64
3.110.1 Instalación Lamp Stack + Virtual Host.....	64
3.110.11 MySQL.....	64
3.110.12 PHP.....	65
3.110.13 Virtual Host + Permisos.....	65
3.110.4 Instalar Wordpress.....	65
3.110.5 Plugins Wordpress.....	68
3.110.51 MailPoet.....	70
3.110.52 WooCommerce.....	71
3.110.52.1 Categorías, Marcas y Productos.....	72
3.110.52.2 Páginas Tienda.....	76
3.110.52.3 Configuraciones.....	76
3.110.53 Polylang.....	80
3.110.54 WordFence.....	80
3.110.55 UnderConstruction.....	80
3.110.56 Cookie Notice & Compliance for GDPR / CCPA.....	81
3.110.57 JoinChat.....	81
3.110.57 WordFence Security.....	82
3.110.58 Yoast SEO.....	83
3.110.59 Contact Form 7.....	83
3.110.6 Apariencia / Temas y Menús.....	84
3.110.61 Apariencia / Temas.....	84
3.110.61.1 Cabecera / Pie de página.....	85
3.110.61.2 Diseño de páginas / entradas.....	86
3.110.61.3 Identidad de la web.....	86
3.110.61.4 WooCommerce.....	87
3.110.62 Menús.....	87
3.110.62.1 Personalización del tema Kadence.....	87
3.110.62.2 Ajustes de apariencia.....	88
3.110.62.3 Ubicaciones de los menús.....	88
3.110.63 Resultado Final Portada.....	89
3.111 Servidor de Correo.....	89
3.112 Prometheus, Grafana y Loki.....	91
3.112 Problemas obtenidos.....	96
3.112.1 Primer Problema.....	96
3.112.2 Segundo Problema.....	97
3.2 Consola de sobremesa.....	99
3.21 Conexiones eléctricas de la consola.....	99
3.22 Componentes y funcionalidades del lanzador de emuladores.....	101

3.22.1 Componentes.....	101
3.22.2 Funciones/casos de uso del script lanzador.....	102
3.23 Valoración de servicios para las funciones del script lanzador.....	103
3.23.1 Servicio para la descarga de ficheros vía red.....	103
3.23.2 Descripción técnica de la herramienta scp.....	104
3.23.3 Pruebas del servicio scp.....	104
3.24 Instalación y configuración de RPI OS.....	107
3.24.1 Instalación de RPI OS.....	107
3.24.2 Configuración de RPI OS.....	109
3.24.3 Autenticación por clave pública en las máquinas de darwin.....	110
3.24. bv4 Visor de escritorio remoto para la RPI 5: RPI-Connect.....	111
3.25 Instalación del emulador Mesen2 y pruebas de rendimiento.....	114
3.25.1 Instalación del emulador en la RPI 5.....	114
3.25.2 ROMs a usar para las pruebas.....	116
3.25.3 Creación de directorio temporal con las ROMs del supermicro en la RPI....	117
3.25.4 Ejecución de ROMs y apunte de problemas encontrados.....	117
3.26 Montaje del amplificador MAX98357A y configuración del software de Adafruit...	117
3.26.1 Preparación del hardware.....	117
3.26.2 Instalación y configuración del software.....	118
3.27 Instalación y pruebas del programa Input-Remapper.....	120
3.27.1 Instalación.....	120
3.27.2 Formas de uso.....	120
3.28 Desarrollo del lanzador de emuladores.....	122
3.28.1 Descarga de ROMs a utilizar.....	122
3.28.2 Programa de creación y ejecución de macros.....	123
3.28.21 Requisitos para el programa de macros a usar.....	123
3.28.22 Programas analizados.....	123
3.28.23 Programa escogido.....	123
3.28.3 Configuración de Input Remapper.....	124
3.28.31 Mapeos y macros teórico.....	124
3.28.32 Configuración de ambos presets.....	125
3.28.4 Programa de gestión de ventanas desde la CLI.....	127
3.28.41 Requisitos a analizar.....	127
3.28.42 Alternativa escogida.....	127
3.28.5 Funciones específicas de los scripts.....	128
3.28.51 Script lanzador.....	128
3.28.52 Script auxiliar 1: Control del proceso del emulador Mesen2.....	129
3.28.53 Script auxiliar 2: Arranque al iniciar la sesión en la terminal.....	130
3.28.6 Explicación de algunos bloques del código fuente final del lanzador.....	130
3.28.7 Backups del código fuente del lanzador.....	134
3.28.8 Automatización del arranque del lanzador.....	134
3.28.81 Ejecución en cada log in de la terminal.....	134

3.28.82 Ejecución de la terminal en cada arranque.....	134
3.28.9 Pruebas del lanzador a nivel de usuario.....	137
3.28.91 Pruebas.....	137
3.28.92 Problemas encontrados durante las pruebas.....	138
3.3 Diseño 3D de la consola.....	139
3.31 Práctica Tinkercad.....	140
3.32 Componentes de la Consola en Tinkercad.....	140
3.33 Carcasa de la consola a base.....	142
3.34 Imprimir modelo 3D.....	143
4. Conclusión y cierre del proyecto.....	145
4.1 Conclusiones generales de proyecto.....	145
4.2 Consecución de los objetivos.....	145
4.3 Valoración de la metodología y planificación.....	145
4.4 Visión de futuro.....	145
5. Bibliografía.....	146
6. Anexo.....	146

1. Introducción al proyecto

1.1 Contexto

En un principio, nuestra idea de proyecto inicial era la de crear una consola portátil basada en el microcontrolador Raspberry Pi Pico. Este contendría un lanzador de emuladores programado mediante el lenguaje APL (vía Arduino IDE) con el que lanzaremos por red diferentes emuladores programados en C para este microcontrolador los cuales estarían almacenados en nuestro servidor Xeon de Darwin. Además de ello, dispondremos de una página web de tienda montada a partir de un gestor de contenido donde se podría surcar un catálogo de productos y realizar las correspondientes “ficticias” compras. Debido a lo anterior mencionado, nuestro servidor deberá ser configurado para el acceso por red y para que sea capaz de ofrecer los diferentes servicios montados. Para poder verificar los aspectos del servidor, haremos uso de un cliente en la misma red que el *supermicro* configurado para el uso de todas las funcionalidades del servidor. Por último, hacemos adjunto la idea de creación de cuentas de redes sociales en plataformas cómo *Mastodont*.

De todo lo presente aquí, la idea a descartar fue la arquitectura de nuestro proyecto basada en microcontrolador y, por ende, la programación del lanzador de emuladores con APL (Arduino Programming Language). La razón de ello es debido a que sería necesaria la adaptación de cada emulador a la arquitectura a usar con C. Para ello, necesitamos de conocimientos avanzados en el mismo y del propio hardware. Una justificación más extensa se puede encontrar en el anexo.

A raíz del descarte, decidimos implementar una arquitectura basada en Raspberry Pi de ordenador. Esto nos supone, además de una mayor facilidad de implementación de funciones al disponer de un SO, cambios en otros aspectos cómo la programación de nuestro lanzador, a realizar con *shell script*, y los emuladores a incorporar, principalmente para interfaces gráficas (en adelante se adjunta mayor detalle sobre la idea). Además de suponerse estas ventajas, también nos supuso el dilema sobre si hacer una consola de tipo portátil o sobremesa. Por diferentes y amplias razones decidimos hacer un modelo de sobremesa, aquel que presentamos en este documento. Adjuntamos también las razones de nuestra elección en el propio anexo. Por último, cabe resumir y aclarar que la idea presentada en este documento hace relación al cambio citado más los aspectos presentes en el primer párrafo de este apartado

Nota 1: La 1a versión de nuestro proyecto, 14/10/2024 - 7/1/2025, se encuentra adjunta y documentada en el anexo.

Nota 2: El proceso de valoración de los dos modelos de consola y la justificación de la elección del modelo de sobremesa se encuentra adjunto en el anexo.

1.2 Justificación

La principal razón para la realización de este proyecto es la puesta en práctica de los conocimientos adquiridos durante el ciclo de SMX y, también, los conocimientos propios derivados de actividades lúdicas realizadas en nuestro tiempo libre. Creemos que es un proyecto relevante para nosotros mismos ya que, además de ser objeto de motivación, incorpora una gran pizca de cada uno de los módulos tratados en el ciclo más otros aspectos que están estrechamente relacionados con el software estudiado en el mismo, cómo lo es el propio hardware y su montaje (Raspberry Pi y componentes). Además de utilizar herramientas vistas en el centro, incorporamos muchas otras herramientas libres seleccionadas, instaladas y configuradas a raíz de los conocimientos previos al proyecto, o bien adquiridos durante el proceso.

1.3 Objetivos

Nuestro objetivo será tener una consola de sobremesa a partir de Raspberry Pi dentro de una carcasa 3D diseñada por nosotros más un altavoz de salida incorporado en la misma. Al arrancar el SO, mostraría un menú con los diferentes emuladores lanzables y una lista de ROMs disponibles para la descarga. El usuario, en todo momento, solo interactúa con el mando de videoconsola conectado. Además de esto, otro de nuestros objetivos será la correcta configuración de nuestro servidor *supermicro*, y de todos los servicios adjuntos en el mismo (así como la tienda), y de nuestro cliente desktop del mismo.

1.31 Objetivo general

- Diseño y montaje de una estructura de Hardware con Raspberry Pi y otros componentes con un software lanzador de emuladores que permita el lanzamiento de emuladores libres.
- Configuración de un servidor intel xeon (supermicro) con un SO y múltiples servicios para linux.
- Configuración de un cliente desktop (linux) que nos permita verificar las funcionalidades de nuestro *supermicro*.
- Montaje de una página web de tienda mediante un gestor de contenidos.
- Creación de variedad de redes sociales.

1.32 Objetivos específicos

- Idear la estructura del hardware de nuestra consola y agrupar los diferentes componentes necesarios.
- Diseño e impresión 3D de una carcasa para nuestra consola

- Instalación de un SO libre en el servidor supermicro más los servicios requeridos por el lanzador de emuladores, la página web y otros de carácter general y básicos en el servidor (cómo sshd).
- Preparación de un cliente desktop en Darwin con el fin de poder verificar la funcionalidad de los montajes de servicios realizados en el servidor supermicro.
- Instalación de un gestor de contenido en el supermicro y su posterior configuración (catálogo de productos y falso proceso de facturación)
- Creación de cuentas en diferentes redes sociales, siendo una de ellas una libre y descentralizada.

1.4 Metodología de trabajo y planificación del proyecto

La metodología de trabajo a emplear en nuestro proyecto ha sido principalmente la llamada **waterfall** debido al carácter de las tareas planificadas, que bien pueden ser paralelas o secuenciales aunque siempre precedidas una detrás de la otra. Pese a esto, hemos decidido incorporar un poco de flexibilidad a esta metodología, permitiendo la reordenación, intercambio, o inserción de nuevas tareas a demanda de los miembros del equipo o de la situación actual o que nos envuelve. Para poder emplear esta metodología, hemos realizado...

- 1) **Hojas de ruta en google drive.** En este se ven presentes todas las tareas a realizar, quién las tiene que realizar y cuál es el orden de realización de las mismas. Además de esto, también nos permite realizar un seguimiento de las tareas que se van realizando marcando en verde las realizadas y en amarillo, o similar, aquellas en curso. Con ello podemos ver cómo progresa el proyecto.
- 2) **Diagrama de gantt con la herramienta *projectlibre*.** Con esta herramienta, hemos podido crear un diagrama de gantt que refleje solo las tareas generales de la hoja de ruta. Con ello podemos hacer un buen seguimiento de las etapas de nuestro proyecto y asegurarnos que nos mantenemos dentro de los plazos de entrega. Además de reflejar las tareas y el orden de ejecución, nos es posible asignar recursos (aquellos quienes realizan las tareas) y también calendarios de proyecto para cada miembro del equipo.

2. Descripción del proyecto

Consola con Raspberry Pi

Queremos realizar una consola mediante el uso de ordenadores Raspberry Pi. Estos son pequeños y compactos y son mucho más idóneos para la creación de una videoconsola que un ordenador de sobremesa (torre) o un portátil (teniendo en cuenta el aspecto). La Raspberry Pi estaría contenida dentro de su propia carcasa 3D diseñada por nosotros más una serie de módulos electrónicos incorporados dentro de la misma carcasa los cuales permiten salida de audio estéreo.

Nuestra Raspberry Pi tendría instalado un sistema operativo basado en Linux el cual ejecutará una serie de programas libres necesarios para la experiencia de juego. Entre estos tenemos

- Mapper de botones
- Lanzador de emuladores

Este último estaría programado por nosotros con *shell script* y dispondría de la función de descarga de juegos para los emuladores lanzables mediante un repositorio en nuestro servidor xeon. En relación al software de administración, también sería necesario la instalación de algunos servicios como **ssh** o algún tipo de **visor de escritorio remoto**.

Página web de tienda

Dispondremos de una página web de tienda configurada con un plugin de wordpress, cuyo gestor de contenido estará instalado en nuestro servidor supermicro. Dentro de la configuración de la tienda buscamos la creación de un catálogo de productos, un proceso ficticio de facturación y diversas secciones de descarga de material adicional respecto a productos de la tienda. Además, queremos implementar un plugin el cual incorpore un bot de tienda el cual envía correos a las direcciones registradas en los usuarios de la tienda.

Servidor supermicro

En un servidor xeon de la sala Darwin del IES Puig Castellar será utilizado para la instalación de los diferentes servicios necesarios para nuestro proyecto. Principalmente, nuestro servidor haría de servidor de,

- **Archivos.** Descarga de ROMs mediante el lanzador de emuladores en la consola.
- **Web.** Almacenaje del gestor de contenidos utilizado.
- **Correo.** Gestión de los correos electrónicos entre usuarios del mismo dominio. No incorporaremos ningún relay de internet y, por ende, la comunicación sólo podrá ser entre cuentas del mismo dominio de correo. Además, si queremos entablar estas comunicaciones entre estos usuarios en máquinas ajenas a la sala Darwin, será necesario configurar dichas máquinas como *peers* de nuestra VPN y asignarles nuestro DNS el cual incluirá resoluciones de nombre de los servicios smtp e imap a las IP correspondientes de la VPN (en este caso la IP de la VPN del servidor *supermicro*).
- **DNS.** Proporcionará resolución de nombres para nuestro MTA, MDA y el servidor apache2.

- **BBDD MySQL.** Almacenará la estructura de la base de datos (tablas...) diseñada para el gestor de contenidos a utilizar.

Estos servicios serán útiles para nuestro lanzador y para el gestor de contenidos a usar para nuestra página web de tienda. Otras características importantes del servidor será la instalación de servicios de administración remota como

- **openssh-server**
- **VPN con wireguard-tools.** Con ello podremos acceder a nuestro servidor incluso estando fuera de la red del Puig Castellar.
- **ipmi-tools.** Con esta herramienta seremos capaces de enviar señales de encendido/apagado/reinicio al ordenador BMC incorporado en el servidor xeon. Este pequeño ordenador en el supermicro dispone de una NIC dedicada y es a partir de esta NIC por donde se reciben los comandos de ipmi-tools que luego son procesados por el BMC. Según la señal, el BMC se encargará de encender o apagar el servidor correspondientemente.

Programas de seguridad también deberán ser instalados, como lo son,

- **Kopia.** Para la creación de instantáneas o snapshots de recuperación del estado de ficheros o directorios.
- **RAID 5.** Para evitar la pérdida de datos mediante la reconstrucción proporcionada por este nivel de raid.

Cliente desktop

Para asegurarnos del correcto funcionamiento de nuestro servidor xeon, tendremos que configurar un ordenador de sobremesa de los presentes en el taller del instituto (Darwin) el cual hará de cliente para cada uno de los servicios que hemos montado en el servidor. Con ello podremos verificar que funcionan bien. Además, debido a que el ordenador BMC del servidor Xeon no puede ser configurado con una VPN, será necesario este cliente para poder acceder al mismo y de ahí al propio BMC. Pues, estos se encuentran dentro de la misma red local.

Redes sociales

Otro punto a tener en cuenta sería la creación de redes sociales, poniendo como objetivo la creación de cuentas en herramientas libres y descentralizadas. Las cuentas estarían bajo el nombre de la empresa gestionada en dicha parte del crédito de síntesis (**FIXMaster**).

2.1 Análisis de requisitos

Consola de sobremesa

- Interpretación de la documentación para el montaje e instalación del SO en una Raspberry Pi
- Añadido de periféricos externos de audio a la Raspberry Pi mediante los puertos GPIO
- Diseño e impresión de una carcasa 3D para el hardware ajustada a las medidas de los componentes físicos
- Configuración de un *mapper* de botones con el fin de poder usar un *gamepad* para la navegación del menú de nuestro lanzador de emuladores

- Desarrollo de un lanzador de emuladores con **shell-script** con un menú iterativo con dos opciones
 - Arranque de emuladores
 - Descarga de ROMs para los emuladores
- Instalación del software de administración remota necesario para el trabajo en casa u otras zonas alejadas del IES Puig Castellar
 - ssh
 - ipmi-tools
 - visor de escritorio remoto

Página web

- Creación de un catálogo de productos de informática
- Creación de una sección de descargas para cada producto
- Configuración de un método de pago alternativo
- Configuración de un plugin de envío automatizado de correos a usuarios registrados

Servidor Xeon

- Configuración de los siguientes servicios
 - VSFTP
 - apache2
 - DNS
 - SMTP e IMAP
 - MySQL
- Configuración de los siguientes programas para el acceso remoto
 - openssh-server
 - ipmi-tools
 - VPN con wireguard-tools u otras herramientas cómo **innernet-server**
- Configuración de los siguientes programas de seguridad **Kopia y RAID (5)**

Cliente Desktop (Darwin)

- Cliente de los servicios **VSFTP, apache2, DNS, SMTP e IMAP**.
- Acceso por **ssh** configurado.
- Que el cliente pueda acceder vía ssh al BMC del supermicro con el fin de poder enviarle comandos con **ipmi-tools** dentro de la LAN.

Redes sociales

- Creación de una cuenta de redes sociales bajo el nombre de nuestra empresa en 3 plataformas diferentes, siendo, al menos, una de ellas libre y descentralizada.
- Realización de un par o más de publicaciones en cada plataforma.

2.11 Requisitos funcionales

Los requisitos funcionales serían todos aquellos relacionados con,

- El montaje del hardware y el software implicado en nuestra consola
- El montaje y configuración de los servicios generales de nuestro *supermicro* así como los de administración y otras utilidades necesarias (seguridad).

- El montaje y configuración de nuestro cliente desktop en Darwin con el fin de verificar las funciones de nuestro *supermicro* (principal requisito funcional) y del encendido/apagado del mismo. Esto último es vital si queremos administrar eficientemente nuestro servidor desde cualquier lugar remotamente.

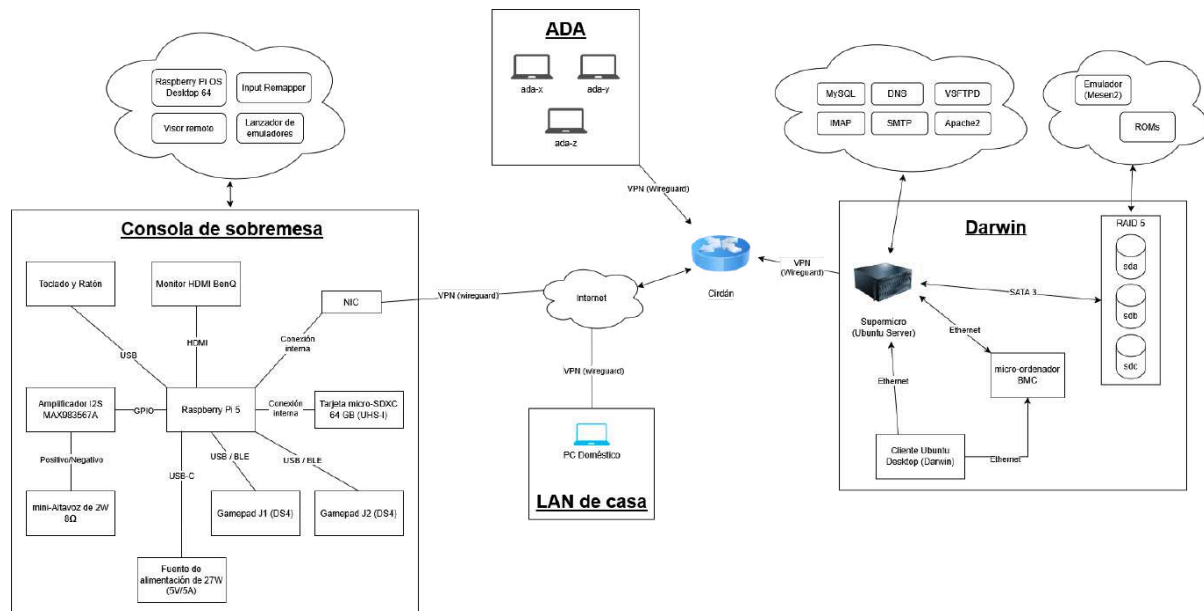
2.12 Requisitos no funcionales

Los requisitos no funcionales serían los siguientes,

- Configuración de la página web de tienda mediante uno o varios plugins de un gestor de contenido para *blogs* (wordpress).
- Creación de redes sociales y publicaciones varias.

2.2 Estructura del proyecto

2.21 Redes



En el anterior diagrama, podemos apreciar la presencia de todos los componentes físicos y lógicos, así como la interconexión de todos los componentes físicos del mismo. En nuestro proyecto, los diferentes componentes se verán repartidos en diferentes redes LAN. Las redes son representadas con un **cuadrado**, estas son:

- Doméstica (la de casa nuestra)
- Aula ada
- Taller o sala Darwin

Todas las redes locales se interconectan entre sí a través de internet. Para poder hacer esto, necesitamos de una red privada virtual configurada con wire guard-tools. En la VPN, encontramos como servidor a una máquina virtual del centro. Será este servidor el que actuará como relay entre los peers de la misma VPN. Por ende, para poder enlazar todas

nuestras máquinas, hemos tenido que configurarlas cómo peers de la VPN mediante una serie de claves privadas generadas por el instituto (proporcionadas por **Juan Morote**) más la clave pública del servidor. Mostramos los componentes de cada red a continuación y su forma de conectar con otros elementos.

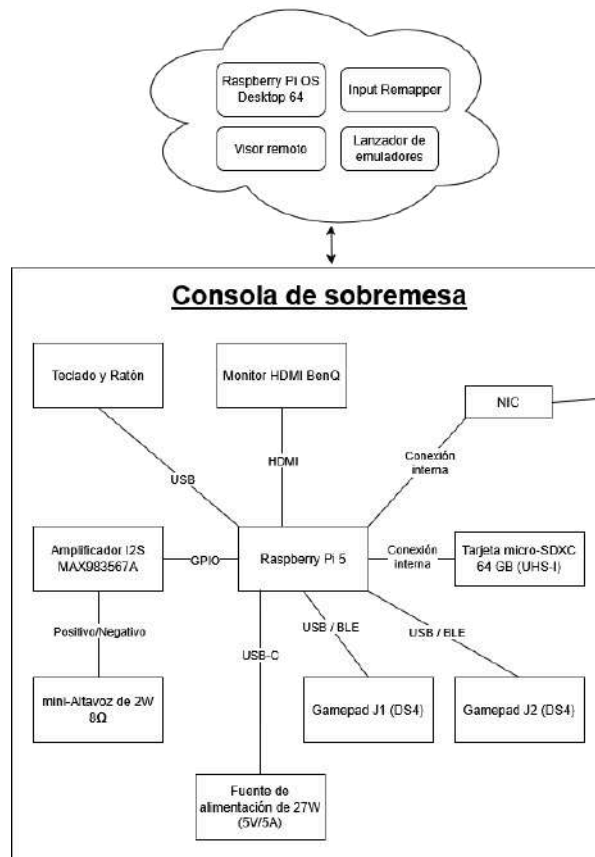
Doméstica (la de casa nuestra). Contiene los dispositivos que son utilizados para el desarrollo del proyecto fuera del entorno de trabajo habitual (Puig Castellar). El acceso a los hosts en el centro será mediante la VPN configurada.

Aula ada. Dentro del aula **ADA** del Puig Castellar, encontramos los ordenadores del centro con los que trabajamos durante las horas de síntesis. Estos, pese a estar en la red del Puig, deberán ser configurados con una VPN de manera nativa, o bien en una máquina virtual, si se quiere acceder a los hosts de otras aulas, cómo es el caso de las máquinas presentes en Darwin.

Taller o sala Darwin. Una de las áreas más importantes de nuestro proyecto, en ella se encuentra nuestro servidor *supermicro* más su BMC y el cliente desktop. Todos estos se conectan entre sí sin necesidad de una VPN gracias a que, además de estar en la misma LAN, en la red del puig si es posible hablar con los hosts de tu misma subred, pero no con hosts de otras subredes.

2.22 Consola

Otro aspecto clave a mencionar del diagrama es que la consola de sobremesa, esta no se encuentra incluida en ninguna de las redes presentes en el diagrama ya que no tendrá una red fija. Durante el transcurso del proyecto podrá ser desplazada a diferentes aulas del instituto o, incluso, a las LAN domésticas. A continuación, vamos a citar algunas de las conexiones de la consola.



Teclado y Ratón vía USB. Se hace uso de los puertos USB (Universal Serial Bus) incorporados en la propia placa.

Amplificador I2S vía GPIO. Las placas Raspberry Pi disponen de una serie de pines de entrada y salida para propósitos generales. Uno de ellos es la salida de audio a partir de un protocolo de transferencia de sonido digital llamado I2S. Por ello, para ahorrar espacio y a demanda de los módulos electrónicos a utilizar, haremos las conexiones mediante estos contactos eléctricos.

Altavoz vía Positivo/negativo. Hacemos referencia a las dos conexiones simples necesitadas por un altavoz a la salida analógica del amplificador MAX98357A.

Fuente de alimentación vía USB-C. La alimentación se realiza mediante un puerto USB de este tipo, única y exclusivamente de alimentación.

Monitor vía HDMI. Es la salida principal y más accesible de la gran mayoría de placas Raspberry Pi. La conexión mediante esta interfaz de vídeo nos garantiza una salida estándar y que no requiere de configuración.

NIC y slot para tarjetas micro-SD vía conexión interna. Con ello hacemos referencia a las pistas de la tarjeta electrónica principal que conectan estos puertos con la entrada y la salida de otros componentes internos.

2.23 Programas/servicios

Cómo se aprecia en los diagramas, todos los programas y servicios son comprendidos dentro de una nube, indicándonos que son componentes lógicos u abstractos y que, por lo tanto, no podemos tocar.

2.3 Descripción de los componentes

2.31 Lógicos

2.31.1 Raspberry Pi OS Desktop 64 (Debian Bookworm)

Descripción

El sistema operativo Raspberry Pi OS Desktop, anteriormente llamado Raspbian, es uno basado en Debian único para arquitecturas ARM de 64 o 32 bits (aunque dispone de una versión para x86). Este SO dispone de muchas de las características de Debian aunque con algunas modificaciones que exprimen los recursos del hardware con el fin de obtener una experiencia de uso mejorada. Este sistema operativo siempre dispondrá de dos líneas, una basada en la última *latest stable* de Debian (actualmente la 12), y otra línea con la versión *old stable* (actualmente la 11). A esta segunda “vieja” línea se le da el nombre de versión *Legacy* y estas se mantienen por la Fundación Raspberry Pi para aquellos usuarios con un software antiguo que podría dejar de funcionar en nuevas versiones. Dentro de estas dos líneas, encontramos diferentes alternativas

- **32 bits (ARM).** Compatible solo con 32 bits.
- **64 bits (ARM).** Compatible con arquitecturas de 64 y 32 bits.
- **Desktop Full.** Versión que incluye el entorno gráfico (PIXEL) más las aplicaciones recomendadas ya instaladas)
- **Desktop.** Incluye el entorno gráfico sin las apps. recomendadas.
- **Lite.** Solo incluye el SO, sin interfaz gráfica ni apps. recomendadas.

Toma de recursos de hardware necesarios

Para garantizar un buen funcionamiento de este sistema operativo en la placa Raspberry Pi, vamos a declarar los aspectos de **CPU, RAM y espacio en disco duro** necesarios.

CPU. Según los documentos oficiales, cómo la página web de descarga de las diferentes versiones de Raspberry Pi OS, vemos una serie de [placas compatibles](#) en relación a la versión **NO** legacy de Raspberry Pi OS Desktop de 64 bits. A partir de esto, deducimos que cualquiera de las placas presentes en esta lista serían idóneas para el funcionamiento de este sistema operativo. Si estas se muestran en la lista, es debido a que sus CPUs soldadas en placa son totalmente compatibles. No será necesario preocuparse por los tiempos de CPU consumidos por el SO ya que estos no suelen superar el 2-3% en procesadores modernos.

RAM. Al no mostrarse la RAM recomendada en los documentos oficiales de Raspberry Pi, hemos decidido basarnos en los valores proporcionados para la distribución en la que deriva RPi OS. Es decir, **Debian 12**. Si revisamos la [documentación oficial](#) de este SO para arm64, veremos que la RAM recomendada es de **2GB**.

Espacio en disco duro. [La documentación oficial de Raspberry Pi](#) nos indica que lo recomendado sería una unidad de almacenamiento no volátil (sólo en el caso de microSD) de entre **32GB hasta máximo 2TB (debido a limitaciones con la tabla de particiones MBR)**.

Cometido

Con este SO podremos operar correctamente los recursos del hardware y, además, obtener un mejor rendimiento debido al *boost* del hardware que obtenemos gracias a la programación del SO. Además, la versión Desktop nos garantiza una instalación limpia desde un inicio (no incluye aplicaciones/servicios recomendados), dándonos la opción de instalar solo lo que queramos.

Porqué de su elección

Para el descarte entre los diferentes SSOO Linux, nos basamos en los siguientes requisitos:

Sea una versión de escritorio o *desktop*. Esto es necesario para que los emuladores de consolas puedan ser instalados y ejecutados, pues estos requieren de una interfaz gráfica.

Incorpore el núcleo Linux y esté basado en Debian. Los sistemas Linux se caracterizan por disponer de la seguridad de la buena programación del Kernel Linux. Además, ya disponemos de experiencia previa en sistemas basados en Debian y es algo que nos facilita el trabajo.

Compatibilidad con el sistema de paquetes *flatpak*. Muchos programas libres, así como emuladores están disponibles con este gestor de paquetes. Además de ello, flatpak siempre es una muy buena alternativa a APT en caso de querer instalar versiones más nuevas e incluso sin permisos de root.

Sin mucho software adicional instalado por defecto. Queremos una distribución con el software con el fin de comenzar con un espacio libre en el disco. Con ello garantizamos un mejor rendimiento de la tarjeta-SD que, además, suelen estar más limitadas en términos de velocidad de lectura y escritura.

No requiera de muchos recursos del ordenador (tiempo de CPU, memoria RAM...). Con ello mejoramos el rendimiento de los programas que ejecutamos en la RPi cómo lo serán los emuladores.

Que aproveche el hardware de la RPi. Algunos sistemas están optimizados para uno o varios hardwares específicos, factor que aumenta el rendimiento.

Para dispositivos ARM. La gama de Pi está basada en la arquitectura de CPU ARM, siendo esto un requisito fundamental.

Existencia de una versión LTS activa y mantenida por una comunidad/organización.

Con esto nos aseguramos...

- De disponer de una documentación amplia y actualizada.
- De evitar problemas de ciberseguridad.
- De reducir el riesgo de aparición de *bugs* o fallos.
- De evitar problemas de compatibilidad con paquetes esenciales para el funcionamiento de otros programas.

Los anteriores requisitos fueron introducidos en la [siguiente tabla](#) con el fin de hacer un análisis más rápido y eficiente. A partir de esto, llegamos a un empate entre **Ubuntu Desktop 24.04 (ARM)** y **Raspberry Pi OS Desktop (ARM64)**. Ambas son alternativas muy fiables y reconocidas y que permiten aumentar el rendimiento del hardware solo por la forma en la se estructura su código fuente.

Ahora bien, sabemos que la comunidad y la empresa (Canonical) detrás del SO Ubuntu se ve más involucrada en el desarrollo de la versión para ordenadores **x86**, y menos en su versión para **ARM**.

En cambio, sabemos que la Raspberry Pi Foundation se centra más en su gama de ordenadores **ARM**. Por ello, pese a que los dos SSOO cumplen con todos los requisitos, creemos que la mejor opción para una Raspberry Pi será **Raspberry Pi OS Desktop 64 (Debian Bookworm)**. Este se encuentra ligeramente por delante a lo que nos podría ofrecer Ubuntu Desktop y es este ligero aumento el que puede suponer un mejor funcionamiento de la consola, sobre todo teniendo en cuenta la limitación de recursos de este tipo de hardware comparado con un ordenador convencional de sobremesa.

Origen

Será necesario obtenerlo mediante la descarga en la página [web oficial de Raspberry Pi](#).

2.31.2 Input Remapper

Descripción

Input Remapper es una herramienta libre para sistemas Linux que permite traducir la entrada de alguno de los periféricos detectados por el sistema operativo a cualquier otro tipo de salida. Dispone de un modo gráfico y otro en modo texto para aquellos sistemas sin GUI. Esta última característica puede ser muy útil si no se pretende que el usuario haga cambios en la configuración de los dispositivos de E/S.

Toma de recursos de hardware necesarios

Para ello, primero vamos a instalar Input Remapper en nuestra máquina virtual Ubuntu Desktop 24.04. Según la documentación oficial, existe un paquete *debian* que poder instalar en Ubuntu (el cual tenemos que descargar manualmente de la última **release** en GitHub con wget). Para instalar el paquete .deb, **sudo apt install -f /ruta/a/paquete.deb**

```
usuario@lana:~$  
usuario@lana:~$ sudo apt install -f ./input-remapper-2.1.1.deb
```

A la hora de pasar a la práctica y, cómo era de esperar, Input Remapper es un programa que no requiere muchos recursos. Podemos ver esto claramente con la herramienta **htop** en la máquina virtual

Main		I/O											
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command		
42407	root	20	0	28344	7168	6016	S	0.0	0.2	0:00.01	sudo input-remapper-gtk		
42415	root	20	0	28344	2676	1408	S	0.0	0.1	0:00.00	sudo input-remapper-gtk		

Debido a esto, los únicos requisitos mínimos que podríamos establecer serían aquellos necesarios para la instalación de este programa en Raspberry Pi OS Desktop.

Cometido

A nivel de usuario, nuestra consola solo podría ser operada a través de un *gamepad*. Por ello será necesario instalar un *mapper* de gamepads que, además de detectar nuestro mando, sea capaz de interpretar qué pulsaciones de botones se corresponden con qué pulsaciones en el teclado. En un principio, esto solo será necesario para nuestro lanzador con el fin de poder navegar por el mismo antes de lanzar algún emulador por red, ya que, en el momento que uno de estos se viese lanzado, el script terminaría la ejecución del mapeador base para evitar problemas con el mapeador nativo del emulador.

La razón por la que queremos que sea posible almacenar diferentes configuraciones en ficheros es debido a que, seguramente, algunos emuladores no dispongan de un mapeador nativo. En este caso, tendríamos que añadir una funcionalidad al script la cual se encargue de intercambiar la configuración de botones del mapeador nativo según el emulador que se arranque. Si es uno con soporte nativo, se terminaría la ejecución. Si no, se mantendría el proceso vivo y se intercambiaría la configuración del mismo.

Porqué de su elección

Los requisitos que hemos escogido para la elección de este programa son los siguientes,

1. Libre y de código abierto.
2. Actualizado recientemente.
3. Con buena documentación.
4. Compilado por el desarrollador para sistemas Linux (Kernel 6.6) ARM64.
5. Que pueda almacenar varias configuraciones en varios archivos.

Mostramos el análisis a continuación,

1. Libre y de código abierto. Bajo la licencia GPL v3.0 de la FSF.

2. Actualizado recientemente. Existen cambios recientes en el repositorio de GitHub. Además, la última release fue liberada en febrero de 2024.

3. Con buena documentación. El repositorio de GitHub incluye una sección de uso con la documentación necesaria para el uso de la interfaz gráfica del programa o bien en modo texto.

4. Compilado por el desarrollador para sistemas Linux (Kernel 6.6) ARM64. En la documentación oficial se muestra el proceso para varias distribuciones entre las cuales NO está Raspberry Pi OS. Pese a esto, existe un método de instalación para cualquier otra distribución haciendo uso de un script de Python (el cual requiere de Python3 instalado y algunas dependencias). Para ello, se necesita de los siguientes programas. Veremos para cada uno de ellos si,

a) Se encuentran disponibles en los repositorios de Debian. La última versión de [Raspberry Pi OS Desktop 64 tiene configurados los repositorios de la versión de Debian](#) de la cual deriva. En este caso, la versión 12 (Bookworm). Por ello tendremos que buscar si el paquete se incluye ya compilado en los repositorios de la versión 12. Actualmente, la

versión 12 de Debian es la versión **stable** y, por lo tanto, será necesario buscar estos paquetes dentro de los repositorios **stable**.

b) Sino, revisar en las webs oficiales si ya se encuentra un binario compilado para el sistema

c) En caso contrario, revisamos si el código fuente está disponible y que herramientas son necesarias para compilar

Dependencias

- **python3**. Se encuentra disponible la versión [3.11 para arm64](#) en los repositorios debian-bookworm
- **python3-evdev**. Se encuentra disponible la versión [1.6.1 para arm64](#) en los repositorios debian-bookworm. La versión mínima requerida por **InputRemapper** es la 1.3.0, por ello no hay problema.
- **python3-dev**. Se encuentra disponible la [versión 3.11 para arm64](#) en los repositorios debian-bookworm.
- **python3-pydantic**. Se encuentra disponible sin versionado [para todas las arquitecturas](#) en los repositorios debian-bookworm.
- **python3-pydbus**. Se encuentra disponible sin versionado [para todas las arquitecturas](#) en los repositorios debian-bookworm.
- **python3-psutil**. Se encuentra disponible la versión [5.9.4 para arm64](#) en los repositorios debian-bookworm.
- **gtksourceview4**. Se encuentra disponible sin versionado [para todas las arquitecturas](#) en los repositorios debian-bookworm.
- **git**. Se encuentra disponible sin versionado [para todas las arquitecturas](#) en los repositorios debian-bookworm.

Por lo tanto, deducimos que si es posible instalar InputRemapper en RPi OS Desktop 64

5. Que pueda almacenar varias configuraciones en varios archivos. Si revisamos la documentación oficial, vemos que la configuración de los gamepads se almacenan en archivos **.json** dentro de su propio directorio. A raíz de esto, podríamos hacer la configuración para cada caso necesario y manipular la ubicación de los mismos a demanda gracias a nuestro script.

Origen

Todas las partes necesarias para el funcionamiento del programa pueden ser encontradas en los repositorios de Debian 12 (Bookworm), o bien en el [repositorio oficial del creador](#).

2.31.3 Mesen2

Descripción

Mesen2 es un emulador de NES libre y de código abierto multiplataforma. Este emulador, además de ser capaz de emular la consola NES, también es capaz de emular SNES, Game Boy, Game Boy Advance, PC Engine y SMS/Game Gear. Nosotros solo nos centraremos en la emulación de NES.

En un principio, el desarrollador se encargó de mantener dos versiones

- **Mesen: Emulador de NES**
- **Mesen-S: Emulador de SNES**

Con el tiempo se detuvo el desarrollo y, más adelante, el autor decidió retomarlo y programar una nueva versión que, además de añadir algunas características que mejoran el rendimiento y la precisión en la emulación, incluye los sistemas emulables entre sus dos antiguos emuladores más la inclusión de algunos sistemas nuevos como los antes mencionados.

Cabe resaltar que este emulador no tiene soporte, al menos según la documentación existente de sus dos anteriores versiones, para gamepads bluetooth. Por lo tanto, esto se volverá un requisito opcional tanto para la Raspberry Pi como para los *gamepads*.

Toma de recursos de hardware necesarios

Debido a la falta de información por parte del desarrollador, creemos que lo mejoría sería revisar estos datos nosotros mismos. Para ello podríamos hacerlo directamente en una MV con Raspberry Pi OS Desktop, aunque, de igual manera, no podremos hacer esto ya que este SO es para sistemas ARM y el host utilizado para realizar el proyecto está montado bajo **AMD64**. VirtualBox solo se encarga de virtualizar recursos y no permite la virtualización de sistemas operativos para una arquitectura diferente a la del host que mantiene las MMVV. Para ello sería necesario emular la arquitectura con algún programa como **QEMU** o similares.

Debido a esto, tenemos dos opciones

- 1) Realizar la prueba en una Ubuntu Desktop 24.04 LTS
- 2) Realizar la prueba en la versión de Raspberry Pi OS para **x86**.

La segunda opción nos permitiría hacer una mejor aproximación a lo que sería el producto final. De igual manera, esta versión para **x86** se encuentra muy desactualizada ya que,

- **Incorpora el kernel linux 5.10**
- **Para sistemas de 32 bits**
- **Se lanzó en 2022.**

Por esto, creemos que la mejor opción será realizar las pruebas en una **MV Ubuntu Desktop 24.04 LTS ya que,**

- Tiene el mismo juego de instrucciones que el SO que instalaremos finalmente (64 bits)
- Tiene una versión del kernel similar a la de Raspberry Pi OS Desktop 64 (basado en Debian12). Ubuntu dispone de la **6.8** mientras que la última versión de RPi OS dispone de la **6.6**.

Instalación de Mesen2

La build utilizada en la consola será la build basada en **NET8**. Por ello, vamos a instalar la misma en nuestra MV Ubuntu Desktop 24.04 LTS. La build de NET8 disponible para

sistemas Linux x86 solo se encuentra disponible en formato **Appimage**, lo cual nos permitirá instalar el emulador rápidamente. Pese a esto, necesitamos de estas dos librerías,

- **NET8**
- **SDL2**

Instaladas las dos librerías, solo nos quedaría instalar el paquete en formato Appimage desde el repositorio, darle permisos de ejecución y lanzarlo. [Información sobre cómo y que es Appimage](#).

NET8

En la documentación oficial vemos que ubuntu 24.04 LTS es una de las distribuciones aceptadas y que la versión 8 de NET se encuentra en los repositorios. Al instalarlo mediante apt, se instalará la librería y sus dependencias automáticamente.

Instalaremos el NET 8 SDK ya que este incluye, además de las herramientas de desarrollador, las herramientas necesarias para ejecutar aplicaciones hechas con esta librería.

```
apt install dotnet-sdk-8.0
```

```
usuario@lana:~$ apt search dotnet-sdk-8.0
Ordenando... Hecho
Buscar en todo el texto... Hecho
dotnet-sdk-8.0/noble-updates,noble-security,now 8.0.115-0ubuntu1~24.04.1 amd64 [instalado]
.NET 8.0 Software Development Kit
```

SDL2

En la documentación oficial vemos que todos los sistemas unix-like con un kernel Linux 2.6+ son compatibles. Podemos instalar SDL2 en sistemas basados en debian (cómo Ubuntu 24.04) con el comando

```
apt install libsdl2-2.0-0
```

```
usuario@lana:~$ apt search libsdl2-2.0-0
Ordenando... Hecho
Buscar en todo el texto... Hecho
libsdl2-2.0-0/noble,now 2.30.0+dfsg-1build3 amd64 [instalado]
Simple DirectMedia Layer
```

Instalación del paquete Appimage

1. Descargamos el paquete, del repositorio oficial, llamado “Linux x64 - Appimage” de la sección “.NET builds” en la MV con wget.

2. Descomprimimos.

```
usuario@lana:~/mesen2$ ls -lh
total 130M
-rw-rw-r-- 1 usuario usuario 65M abr 24 14:41 'Mesen (Linux x64 - AppImage).zip'
-rwxr-xr-x 1 usuario usuario 65M abr 24 12:41 Mesen.AppImage
usuario@lana:~/mesen2$
```

3. Damos permisos de ejecución

```
-rwxr-xr-x 1 usuario usuario 65M abr 24 12:41 Mesen.AppImage
```

4. Lanzamos

```
usuario@lana:~/mesen2$ ./Mesen.AppImage
dlopen(): error loading libfuse.so.2

AppImages require FUSE to run.
You might still be able to extract the contents of this AppImage
if you run it with the --appimage-extract option.
See https://github.com/AppImage/AppImageKit/wiki/FUSE
for more information
```

A la hora de hacerlo, vemos que **FUSE** es requerido para lanzar aplicaciones **Appimage**. En la documentación oficial de Appimage, nos dice que las distribuciones de Linux suelen venir con la herramienta FUSE por defecto y que, en caso de no funcionar, posiblemente tengamos que instalar FUSE2 (cómo se muestra en la salida de error de la imagen anterior).

El primer paso en la documentación hace referencia a sistemas operativos previos a la Ubuntu 22.04. Por lo tanto, saltamos al siguiente paso donde lo primero que tenemos que hacer es revisar si tenemos instalada la versión **FUSE 3**. En caso de ser así, deberemos instalar la versión 2 de FUSE.

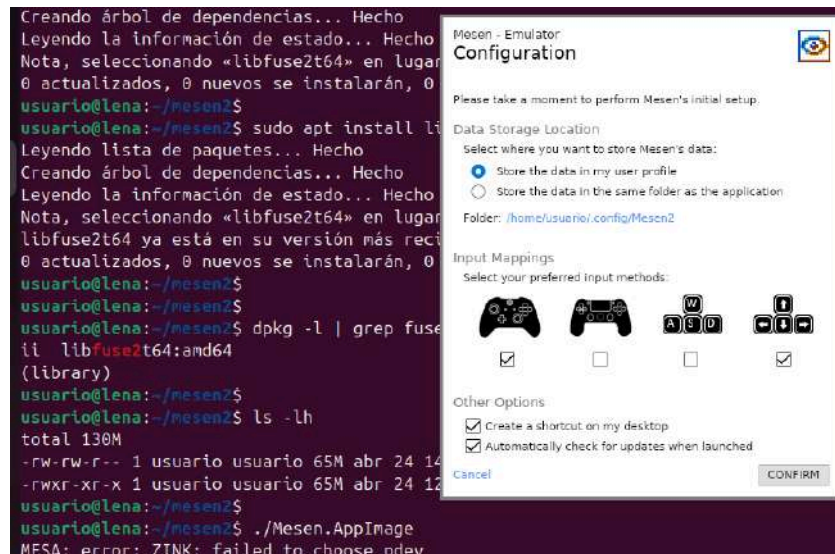
```
usuario@lana:~/mesen2$ dpkg -l | grep fuse3
ii  fuse3                      3.14.0-5build1          amd64
(3.x version)
ii  libfuse3-3:amd64          3.14.0-5build1          amd64
(library) (3.x version)
usuario@lana:~/mesen2$
```

Vemos que si está instalado, por lo tanto, descargamos **FUSE2** con **sudo apt-get install libfuse2**

```
usuario@lana:~/mesen2$ dpkg -l | grep fuse2
ii  libfuse2t64:amd64         2.9.9-8.1build1         amd64
(library)
usuario@lana:~/mesen2$
```

5. Volvemos a lanzar

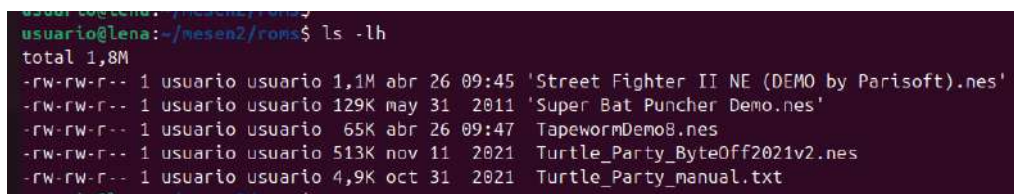
Veremos cómo ahora ya no hay ningún error.



Descarga de ROMs para NES

Realizaremos la prueba con los siguientes demos de juegos gratuitos de NES hechos por la comunidad ([fuente](#))

- [Street Fighter II NE](#)
- [Turtle Party](#)
- [Tapeworm Disco Puzzle](#)
- [Super Bat Puncher](#)



Ejecución de los juegos y toma de recursos utilizados

Para cada juego, tomaremos nota de la CPU y RAM utilizada por el emulador ejecutando la ROM tanto en la MV con “**monitor del sistema**” cómo en el host con el “**administrador de tareas**”. Para ello, vamos a tomar nota de la velocidad de CPU y tamaño de RAM del host y, acto seguido, los recursos del host que dedicamos a la MV. Esto nos será útil para poder deducir el tiempo de CPU utilizado por la MV.

Nuestra máquina host Windows 10 dispone de,

- **CPU** = 4 núcleos 4 hilos a 3.60GHz
- **RAM** = 13,9GB / 12,9 GiB

Los recursos del host que estamos virtualizando para la MV son,

- **CPU** = 2 núcleos 2 hilos a 3.60GHz
- **RAM** = 4096MB / 3906MiB

Street Fighter II NE

Recursos consumidos en la MV por el proceso padre

- CPU 2 cores, 2 threads: 90% → **3.24GHz**
- RAM (máx 4096MB): **155 MB**

Recursos consumidos por la MV en el host

- CPU 4 cores, 4 threads: 54% → **1.94GHz**

El valor que nosotros observamos en el host acerca del consumo de CPU estaría compuesto, además de por el propio emulador más la ROM en ejecución, de otros programas abiertos, en la MV, en el momento de la captura de estos valores como **el monitor del sistema, la terminal, el entorno gráfico y el sistema operativo**. Para extraer el tiempo que consume solo el emulador de ese 54% de los recursos del host que consume la MV, hacemos la siguiente operación

$90 * 1.94 / 100$. Que nos da como resultado **1.74GHz**.

Si queremos saber el % de ese 54% que solo se corresponde con el emulador, podemos hacer otra regla de 3

54% → 1.94GHz

?% → 1.74GHz

$1.74 * 54 / 1.94$. Nos da como resultado **48%**

Por lo tanto, el tiempo de CPU consumido en el host (W10) por solo el emulador y la ROM es de **1.74Ghz (48%)**.

Turtle Party

Recursos consumidos en la MV por el proceso padre

- CPU 2 cores, 2 threads: 85% → **3.06GHz**
- RAM (máx 4096MB): **172 MB**

Recursos consumidos por la MV en el host

- CPU 4 cores, 4 threads: 59% → **2.12GHz**

El consumo de solo el emulador más la ROM en el host sería de → **1.80GHz (50%)**

Tapeworm Disco Puzzle

Recursos consumidos en la MV por el proceso padre

- CPU 2 cores, 2 threads: 86% → **3.09GHz**
- RAM (máx 4096MB): **246 MB**

Recursos consumidos por la MV en el host

- CPU 4 cores, 4 threads: 58% → **2.08GHz**

El consumo de solo el emulador más la ROM en el host sería de → **1.78GHz (49%)**

Super Bat Puncher

Recursos consumidos en la MV por el proceso padre

- CPU 2 cores, 2 threads: 85% → **3.06GHz**
- RAM (máx 4096MB): **273 MB**

Recursos consumidos por la MV en el host

- CPU 4 cores, 4 threads: 61% → **2.19GHz**

El consumo de solo el emulador más la ROM en el host sería de → **1.86GHz (51%)**

Decisión de los requisitos

Analizados los resultados, creemos que los requisitos recomendados para este emulador son,

CPU (4 núcleos, 4 hilos a 3.60GHz) → 1.86GHz (o lo que es lo mismo, un 51% de los recursos de CPU totales de la máquina).

RAM → 273MB

A la hora de comparar este valor con otras CPUs, hay que tener en cuenta que el consumo de CPU total del programa, según la referencia anterior, puede variar según la cantidad de núcleos de la CPU y la velocidad de reloj de los mismos. Para poder calcular cuál sería el consumo de tiempo en otras CPUs a partir de este valor podemos hacer una regla de tres.

$$4 \text{ (3.60GHz)} \rightarrow 51\%$$

$$x \text{ (yGHz)} \rightarrow ?\%$$

$$\left(\frac{\text{núcleos}_{\text{dst}}}{\text{frecuencia}_{\text{dst}}} \right) * \text{uso de CPU total (\%)} \text{ en el valor de referencia} / \left(\frac{\text{núcleos}_{\text{ref}}}{\text{frecuencia}_{\text{ref}}} \right) = \text{uso de CPU total en la CPU}_{\text{dst}} (\%)$$

En nuestro caso... (según nuestra referencia).

- **uso de CPU total en el valor de referencia** = 51
- **núcleos_{ref}** = 4
- **frecuencia_{ref}** = 3.60

Cometido

El propósito de este emulador será cumplimentar la funcionalidad de arranque por red de nuestro lanzador de emuladores. Además de esto, formará parte de uno de los principales casos de uso para nuestro proyecto, que es el *gameplay*.

Porqué de su elección

Antes de la definición de los requisitos, mostramos una serie de aspectos a tener en cuenta. En un principio, nuestra intención es que los ejecutables binarios de los emuladores sean lanzados por red a través de nuestro script. Estos binarios se almacenarán en el servidor, y serán solicitados por el cliente a demanda a través de nuestro lanzador de emuladores hecho con **shell-scripting**. En el caso de que alguno de los emuladores necesitará de alguna librería para su ejecución, deberán de instalarse en la Raspberry Pi y permanecer en la misma. Siendo esto un requisito más para el usuario final si este quiere hacer uso de nuestro script. Concluido esto, mostramos ahora los requisitos en los que nos basamos y el posterior análisis para este emulador.

Libre, gratuito y de código abierto. Debido a la naturaleza del proyecto, es importante que estos se encuentren bajo una licencia que los haga libres y totalmente gratuitos.

Actualizado recientemente. Evitaremos problemas de compatibilidad con el software moderno y actualizado de nuestro SO.

Documentación para la instalación/compilado. Creemos que es el aspecto más importante en relación a la documentación necesaria. En caso de no haber documentación de otros aspectos, no lo creemos un gran problema, ya que estos programas disponen de una interfaz gráfica y, por ende, son intuitivas.

Ejecutable y librerías compatibles con sistemas Linux 6.6 ARM64. Habría que revisar que tanto el emulador como las librerías necesarias han sido desarrolladas pensando en dar soporte, entre muchos otros, al sistema que nosotros necesitamos que sea compatible.

1. Ejecutable compilado para Linux 6.6 ARM64. **Sino...**
2. Código fuente disponible para compilar
3. Existencia de un compilador capaz de compilar ese código fuente para Linux 6.6 ARM64
4. Librerías disponibles en los repositorios de Debian 12 (Bookworm stable) **Sino...**
5. Librerías compiladas para Linux 6.6 ARM64. **Sinó...**
6. Código fuente para compilar
7. Existencia de un compilador capaz de compilar código fuente para Linux 6.6 ARM64.

[No esencial] Permite la conexión de mandos por USB y/excepto Bluetooth y su mapeado de botones. El soporte nativo para mandos de videoconsola nos facilitará la implementación de los gamepads físicos en los mismos. En caso de haber soporte nativo, solo tendríamos que terminar la ejecución del interpretador base del SO a la hora de lanzar el emulador por red para que no haya problemas por el hecho de tener dos programas similares funcionando sobre el mismo periférico.

Libre, gratuito y de código abierto. Licenciado bajo la GPL 3.0. Su código fuente, escrito en C++ y C#, está disponible en su repositorio.

Actualizado recientemente. Se realizaron cambios recientes en el repositorio de GitHub, hace 4 días a fecha de **21/04/2025**. Por lo tanto, deducimos que se sigue manteniendo.

Documentación para la instalación/compilado. En el repositorio de GitHub hay presentes algunos pasos que ilustran el proceso de compilado.

Ejecutable y librerías compatibles con sistemas Linux 6.6 ARM64. Vemos que el código fuente de emulador Mesen2 está hecho en conjunto con la librería SDL2 y NET8, ambas son librerías que se utilizan para el desarrollo de código multiplataforma. Tendremos que verificar que hay soporte para Linux 6.6 arm64 por parte de estas dos librerías.

SDL2. Disponible para las distribuciones de Linux que incorporan el kernel Linux 2.6 o superior. [Fuente](#)

NET8. Existen varias versiones para esta librería, la librería SDK para desarrollo y la .NET para tiempos de ejecución. Según la [documentación oficial](#), vemos que si está disponible NET8 para Linux, aunque serán necesarias las siguientes dependencias para los sistemas basados en Debian (cómo es el caso de Raspberry Pi OS).

- **libc6:** Disponible para linux arm64. Se encuentra en los repositorios debian 12. [Fuente](#)
- **libgcc1:** Esta librería solo se encuentra en los repositorios de Debian 10 (buster). Pese a esto, encontramos una versión actualizada llamada **libgcc-s1** disponible en los repositorios de **Debian 12**. En caso de que esta versión no diese efecto, se puede probar a añadir el repositorio de **buster** en **bookworm** e instalar el paquete **libgcc1** en la versión ([guía orientativa](#)).
- **libgssapi-krb5-2:** Disponible en los repositorios de Debian 12 para arm64. [Fuente](#)
- **libc70.** En los repositorios de Debian no encontramos esta misma versión, pero si la librería **libc72** para los repositorios Debian 12. Pese a no ser la misma versión, creemos que vale la pena utilizarla por lo siguiente
 - Versión superior que seguramente incluya mejoras respecto a la anterior sin eliminar rasgos importantes de la versión 70. Pues, no hay mucha distancia entre versiones
 - La versión pura de Debian 12 si necesita de **libc72** en vez de **libc70**. Al estar nuestro SO basado en la misma versión de Debian, creemos que esto es un punto a favor.
- **libstdc++6:** Disponible en los repositorios de Debian 12 para arm64. [Fuente](#)
- **zlib1g:** Disponible en los repositorios de Debian 12 para arm64. [Fuente](#)

Ejecutable compilado para Linux 6.6 ARM64. En el [repositorio de github](#) oficial disponemos de un binario ya compilado para linux arm64.

Librerías disponibles en los repositorios de Debian 12 (Bookworm stable). Demostrado anteriormente

[No esencial] Permite la conexión de mandos por USB y/excepto Bluetooth y su mapeado de botones. Este emulador es una fusión de los dos anteriores emuladores **Mesen y Mesen-S**. Si revisamos la documentación de estos dos emuladores por separado, esto dado a que Mesen2 no dispone de una documentación, veremos que sí había soporte para gamepads. Por lo tanto deducimos que si existe soporte nativo en el mismo.

Origen

Los componentes necesarios serán descargados a través de los repositorios oficiales de Debian 12 y/o a raíz de los enlaces adjuntos en el [repositorio de github](#) para este emulador.

2.31.4 Lanzador de emuladores (bash-scripting)

Descripción

bash es la *shell* o intérprete de comandos de Linux. Con los comandos de esta shell se pueden automatizar múltiples acciones y procesos mediante la creación de scripts secuenciales, iterativos e incluso inteligentes mediante el uso de condicionales.

Cometido

La función de **bash** es nuestro proyecto será ser el lenguaje de alto nivel utilizado para la programación de nuestro lanzador. Lanzador el cual, cómo ya se mencionó anteriormente, se encargará de proporcionar al usuario un sistema de menú-submenú iterativo que dará la opción al usuario de **descargar ROMs almacenadas en nuestro supermicro** y de **arrancar por red nuestro emulador Mesen2**. Lo que nosotros llamamos *arranque por red* hace referencia la descarga y posterior inserción del binario principal del mismo directamente en RAM. Con ello conseguimos que el programa se ejecute sin tener que estar físicamente en almacenamiento no volátil del dispositivo.

Porqué de su elección

La razón para el uso de **shell-script** es debido a la previa experiencia en la creación de programas con este tipo de lenguaje. Además, con ello podremos interconectar las funcionalidades de nuestros servicios con nuestro lanzador de manera fácil y similar a cómo se haría en un caso real con el uso de APIs. Sabemos que en ciertos casos siempre será mejor el uso de otros lenguajes de alto nivel cómo C o similares, pero no hemos querido invertir en ello debido a la falta de tiempo y la poca o casi nula experiencia en lenguajes de alto nivel para propósito general (cómo C o Python).

Origen

Bash es una de las herramientas imprescindibles de cualquier sistema operativo Linux. Disponemos de él gracias a la instalación de un SO basado en Debian.

2.31.5 MySQL

Descripción

MySQL es un gestor de bases de datos ampliamente utilizado para aplicaciones/servicios/gestores de contenidos, los cuales permanecen en un sistema linux. Este gestor es libre, gratuito, reconocido, moderno, versátil y dispone de una utilidad en la

línea de comandos para el uso avanzado. De igual manera, existen herramientas basadas en la web, cómo **phpMyAdmin**, que permiten administrar gráficamente esta herramienta.

Cometido

El propósito de esta utilidad será el de ser la base de datos del gestor de contenido que vamos a utilizar para nuestra tienda. En este caso, un plugin de **woocommerce** que funciona con **wordpress** mediante el uso de algunas APIs.

Porqué de su elección

Lo hemos escogido, pese a haber otros gestores de BBDD compatibles con wordpress, debido a la compatibilidad con nuestro servidor supermicro y, también, la experiencia previa sobre la creación de BBDD, usuarios y gestión de privilegios obtenida en **Aplicaciones Web** (M08).

Origen

Los repositorios de Ubuntu server son la alternativa más viable para la instalación del binario principal y dependencias en el orden correspondiente.

2.31.6 Apache2

Descripción

Apache2 es un servidor web libre y gratuito ampliamente utilizado en empresas u otras instituciones. Este dispone de la capacidad crear diferentes hosts virtuales, los cuales actúan cómo pequeños servidores web dentro del propio apache. Apache2, en la actualidad, soporta los protocolos HTTP/1.1 y HTTP/2 (con una posible ampliación a HTTP/3 y QUIC cuando estos protocolos se encuentren en una versión estable). Además, también se disponen de diferentes módulos que se activan de manera global para todos los virtual hosts, pudiendo llamar a las directivas de estos módulos en los virtual hosts que las requieran. Por último, cabe mencionar que apache2 soporta la creación de virtual hosts basados en IP y en nombre (gracias a un servidor DNS).

Cometido

Este, simplemente, almacenará todo el código en **php** del gestor de contenidos utilizado y, a su vez, se encargará de enviar el HTML generado dinámicamente al navegador web del cliente que lo solicita.

Porqué de su elección

Similar a lo ocurrido con MySQL, se requiere de un servidor web si queremos poder ofrecer nuestra web dinámica con wordpress. La razón de haber escogido apache2 se debe al amplio manejo obtenido del mismo durante el módulo de **Serveis de Xarxa**. Además, apache2 es un servicio moderno que se centra sobre todo en ser escalable y tener una amplia disponibilidad.

Origen

Podemos instalarlo a través de internet mediante el uso de los repositorios oficiales de **ubuntu**.

2.31.7 Bind9

Descripción

Bind9 es el servicio DNS utilizado para la resolución de nombres del dominio de nuestro proyecto (**FIXMaster**). Este servicio es gratuito y de código abierto y es uno de los más utilizados en todo el mundo dentro de sistemas Linux. Con este servicio, es posible definir zonas directas o inversas que mantienen un dominio y las correspondientes resoluciones para ese dominio. Además, se ofrece la posibilidad de configurar que varios DNS se encargan de una misma zona con el fin de reducir el consumo de recursos sobre una sola máquina, esto es lo que se conoce como **master-slave**. En este modelo, el maestro define la autoridad y varios esclavos y, además, los esclavos deben declarar en local quién es su maestro (aquel del cual reciben la configuración de la zona).

Cometido

Cómo se puede deducir, bind9 será necesario para la resolución de nombres necesaria en otros servicios como lo sería nuestro apache2 y los servicios asociados al correo electrónico como postfix y dovecot.

Porqué de su elección

Razón similar a casos anteriores, se debe a que es una herramienta libre, fácil de usar, compatible con Linux, moderna y muy eficiente a la hora de cumplir con su trabajo. Se actualiza constantemente y eso la hace libre de errores que puedan dispersar nuestro flujo de trabajo habitual.

Origen

La vía más fácil sería mediante la instalación con paquetes debian y el gestor aptitude (apt). Pese a esto, existe la posibilidad de instalar el paquete debian manualmente o, incluso, de compilar el código fuente para Linux mediante el uso de **gnu-compiler-collections**.

2.31.8 scp

Descripción

Esta herramienta es una utilidad del demonio **openssh-server** el cual permite copiar archivos de la máquina local a una remota o entre dos máquinas remotas desde la máquina local.

Cometido

La función de este protocolo será transferir las ROMs de su repositorio en el servidor a nuestra RPI 5.

Porqué de su elección

Lo hemos escogido, ya que es un protocolo con un uso sencillo y que puede ser implementado fácilmente con scripts de *bash*.

Origen

Esta herramienta forma parte del servicio **openssh-server**.

2.31.9 Postfix

Descripción

Postfix es un servidor de correo electrónico o *MTA* libre y de código abierto que sirve para la transferencia de correo hacia otros servidores SMTP o bien IMAP/POP3. Este servicio en concreto hará uso de la dirección del destinatario con el fin de saber si este debe ser enviado al MTA del dominio de ese correo, o bien aceptarlo (dominio destino igual al dominio controlado) y dejarlo al correspondiente MDA (IMAP/POP3). Con postfix, por defecto, solo se aceptan los correos que vengan de usuarios del mismo dominio que controla.

Cometido

Nuestro objetivo principal para nuestro postfix será la transferencia de correo al MDA de nuestra empresa. Este será su único cometido, ya que todas las comunicaciones serán entre usuarios del mismo dominio y, por lo tanto, no será necesario transferir el correo a otros MTAs externos. Esto último podría ser realizado en el envío configurando nuestro DNS para resolver dominios como **gmail.com**, pero bien los correos hacia nuestra MTA no funcionarán ya que no tenemos control de las resoluciones de los DNS que los MTAs de estos dominios utilizan.

Otro rol fundamental será la de poner de alta un usuario de tipo **noreply** que será asignado a un bot de nuestra tienda virtual que se encargue de enviar correos automáticos a los usuarios registrados en la tienda. Obviamente, según lo citado anteriormente, esto sólo funcionará si el correo electrónico de ese usuario es del mismo dominio que la cuenta usada para el bot (en este caso, sería el dominio de nuestra empresa **FIXMaster**).

Porqué de su elección

Cómo muchos de los servicios anteriores, además de ser la herramienta libre y gratuita aprendida durante el curso, creemos que es una buena opción debido a que es un servicio que puede ser instalado y configurado rápidamente gracias a las preguntas realizadas durante la instalación.

Origen

Su vía más accesible será mediante los repositorios configurados en Ubuntu Server 24.04.1 LTS.

2.31.10 Dovecot

Descripción

Dovecot es el servicio complementario de SMTP que se encarga de almacenar el correo para un cierto usuario del dominio que controla. Serán luego estos usuarios los que se conecten a dovecot, en este caso con el protocolo IMAP, y se descarguen el correo entrante en su bandeja de entrada. Dovecot permite el cambio de formato de los archivos que contienen los correos a uno u otro según el tipo de bandeja configurada en el MUA (solo en el caso de mailutils). Los correos para el usuario con **maildir** serán repartidos, en dovecot, en diferentes archivos en vez de contenerlos todos en uno solo (cómo ocurre en el caso de mailbox con mailutils).

Cometido

A diferencia de postfix, dovecot se encargará de almacenar el correo para los usuarios de nuestro dominio. El correo almacenado sólo será de usuarios del dominio para otros

usuarios del mismo dominio ya que **no** tenemos en mente el establecimiento de un relay para el envío y posterior recepción de correo de otros MTA de internet como **gmail**.

Porqué de su elección

Dovecot, además de ser libre y ser la herramienta por excelencia enseñada en el centro, es uno de los servicios más fáciles de poner en marcha. Pues, lo único que se debe hacer es instalarlo y realizar cambios, opcional, en la configuración respecto al tipo de bandeja de entrada utilizado por los MUA mailutils.

Origen

Descarga accesible mediante los repositorios de ubuntu server con **aptitude**.

2.31.11 Kopia

Descripción

Kopia es una herramienta de backup que permite la creación de instantáneas o *snapshots*. Estas *fotos* son tomadas solo de la configuración de un cierto fichero o directorio y, por ende, veremos que estos *backups* se completan mucho más rápidamente. Estas snapshots són después almacenadas en un repositorio, el cual puede ser de diferentes tipos. De ahí, si queremos restaurar algún dato, tendremos que montar estas *snapshots* cómo si de una partición se tratase con el fin de poder acceder a su contenido.

Cometido

Kopia será la herramienta de backup que usaremos en nuestro servidor *supermicro* con el fin de evitar la pérdida de datos frente a todo tipo de problemas. Estará configurado para que realice *snapshots* de los directorios más importantes cada un tiempo programado y los almacene en un repositorio de tipo SFTP ubicado en nuestro cliente en Darwin. También haremos un uso de esta herramienta en casos puntuales antes de realizar cambios en el servidor que puedan romperlo.

Porqué de su elección

Kopia es la herramienta más rápida y eficaz, además de ser libre, para la creación de backups pequeños. Este tipo de backups se vuelven mucho más accesibles que cualquier tipo de backup diferencial o incremental el cual requeriría de las copias anteriores.

Origen

En este caso, su instalación se debe de hacer manualmente mediante un fichero comprimido, el cual contiene el ejecutable binario de este programa. Será necesario revisar que la versión instalada es compatible con nuestro kernel y arquitectura (linux amd64).

2.32 Físicos

Consola de sobremesa

2.32.1 Raspberry Pi 5

Descripción

La Raspberry Pi es un ordenador de tipo SBC (Single Board Computer) de bajo consumo que ofrece diversas características, todas en una misma placa. Esta serie de placas son famosas por poder adaptarse a múltiples propósitos, ya sea de desarrollo o bien para simple uso personal. Para el ámbito de desarrollo, se nos proporcionan una serie de pines de propósito general llamados GPIO que son muy útiles para proyectos de electrónica y, además, pueden ser controlados mediante programas escritos con diversos lenguajes de programación. El modelo concreto de Raspberry Pi 5 nos ofrece una potencia de cálculo suficiente para los emuladores y programas que introduciremos. A continuación citamos las diferentes características técnicas.

Raspberry Pi 5		
Spec	Contenido	Cantidad
CPU Clock	2.4GHz	
Arquitectura de CPU	Quad-core ARMv8 Cortex-A76 de 64 bits (SoC)	x4
GPU Clock	800MHz	
RAM	SDRAM LPDDR4-3200 de 2/4/8/16 GB	x1
NIC	WLAN Wi-Fi 5 (2.4 y 5 GHz)	x1
	Gigabit Ethernet PoE (requiere de un hat por separado)	x1
Bluetooth	5.0 BLE	x1
Salida de vídeo	micro-HDMI (4k)	x2
Puertos	USB 3.0	x2
	USB 2.0	x2
	PCIe 2.0	x1
	Ranura micro-SD (SDR104)	
	GPIO	x40
Otros	Pulsador de goma de encendido/apagado	x1
	Conector ventilador PWM	x1
Alimentación	5V/5A (27W)	

Cometido

El propósito de nuestra Raspberry Pi es que sea el núcleo de nuestra consola de sobremesa. Contendrá todo el software, desarrollado por nosotros y por otros

programadores, necesario para la experiencia del usuario

Porqué de su elección

Los requisitos establecidos para la elección del modelo són los siguientes,

- x1 Puerto HDMI
- x1 slot para micro-SD
- x2 Puerto USB 3.0
- x2 Puerto USB 2.0/3.0
- x1 Puerto RJ-45 Gigabit ethernet
- x1 Chip funciones inalámbricas
 - Wi-Fi
 - Bluetooth (opcional)
- Que sea un modelo presente en la lista de aquellos compatibles con el sistema operativo Raspberry Pi OS with Desktop (64)
- **Mínimo 4GB de RAM.** Con ello garantizamos un buen funcionamiento del sistema incluso con el SO y Mesen2 en ejecución.
- Que la CPU incorporada en ese modelo no comporte un consumo por parte del emulador mayor o igual al **80%**. (para saber este dato hace falta utilizar la fórmula explicada en la toma de recursos de **Mesen2**)

A raíz del análisis en la [siguiente hoja de cálculo](#), descartamos varios modelos, quedándonos al final con el modelo 5 de RPi.

Origen

Este componente fue adquirido por el centro (IES Puig Castellar) con el fin de poder proveernos este material a nuestro grupo y futuros grupos de proyecto en el instituto.

2.32.2 Carcasa oficial para Raspberry Pi 5

Descripción

La carcasa oficial de Raspberry Pi 5 está diseñada para minimizar el daño por golpes y, entre otras cosas, descargas electrostáticas. Esta incorpora un ventilador, el cual puede ser conectado al puerto PWM de la Raspberry Pi 5, cuya velocidad se controla según la temperatura de la CPU.



Cometido

Esta carcasa la usaremos durante el periodo de tiempo que no esté disponible nuestro propio diseño 3D con el fin de garantizar nuestra seguridad y la de la RPi.

Porqué de su elección

En un principio no teníamos en mente la presencia del mismo en el proyecto, ya que este fue adquirido por el centro en conjunto con la RPi 5. De todas formas, nos será útil para su manipulación durante el proceso de diseño e impresión de nuestra propia carcasa 3D.

Origen

Cómo se ha mencionado, este componente fue adquirido por el centro en conjunto con la RPi 5.

2.32.3 Monitor BenQ

Descripción



Monitor LED el cual dispone de una interfaz HDMI y VGA. Permite el ajuste de la fuente de video, cambio de brillo, entre otros aspectos...

Cometido

Su cometido en el proyecto será la visualización de los gráficos renderizados por la GPU de la Raspberry Pi.

Porqué de su elección

Es ideal para nuestro proyecto debido a sus dimensiones y a la existencia de una fuente de alimentación ya incorporada en el mismo. Además, dispone de una entrada de vídeo apta con las salidas presentes en nuestra RPi.

Origen

Material del centro, más específicamente, de la sala Darwin.

2.32.4 Cable HDMI estándar

Descripción. Para la comunicación del puerto de vídeo de la RPI con el monitor haremos uso de un cable HDMI estándar.

Cometido. Enlace con el que transmitir el video entre la RPi y el monitor.



Porqué de su elección. Dispone de las clavijas y la interfaz correspondiente con las del monitor y la RPi.

Origen. Material de un miembro del equipo (Iván).

2.32.5 Adaptador HDMI a microHDMI

Cometido. Permitir la salida de vídeo de los puertos microHDMI de la RPi.

Porqué de su elección. Este componente será necesario ya que la Raspberry Pi ofrece unos puertos de tipo micro HDMI y nuestro cable solo tiene un conector HDMI estándar en ambos extremos. Con un adaptador de este tipo en alguno de los extremos ya nos será posible enviar vídeo por el mismo.

Origen. Material comprado a través de internet por los miembros del equipo.

2.32.6 Teclado y Ratón Logitech

Descripción técnica. Teclado QWERTY Español alámbrico USB tipo A y ratón alámbrico USB de tipo A.

Cometido. Con ellos podremos operar nuestra RPI a nivel técnico y de desarrollo.

Porqué de su elección. La marca correspondiente proporciona material cómodo y duradero, aspectos útiles para el trabajo no remoto en la RPi.

Origen. Material del aula Darwin.



2.32.7 Amplificador MAX98357A

Descripción técnica

El MAX98357A es un módulo que permite la amplificación de audio digital I2S. Este amplificador admite como entrada la salida de audio digital I2S de nuestro dispositivo, la cual amplifica (con una ganancia máxima de 15 db) y envía a través del protocolo I2S al convertor DAC incorporado en la misma placa. El trabajo de este chip será transformar el audio digital I2S a analógico, el cual poder inyectar directamente a nuestro altavoz. Este dispone de los siguientes pines de 2,54mm

- **LRC.** Por este se indica si el audio de entrada es para el canal izquierdo o el derecho.
- **BCLK.** Por este se reciben las señales que sincronizan el envío de los datos.
- **DIN.** Por este pin se envían los datos tanto de entrada como de salida.
- **SD.** Con este pin, podremos apagar según la señal enviada.
- **GAIN.** Con este pin, podremos ajustar la ganancia según la resistencia a la que expongamos al mismo.
- **V_{in}.** Entrada de alimentación de 3.3 a 5.5V
- **GND.** Toma de tierra o polo negativo.

Además, podemos apreciar otros dos pines más en la parte superior que son para la salida del altavoz. Las diferentes salidas de audio pueden ser **stereo, mono left, mono right**. Cabe recalcar que según el voltaje de alimentación y la resistencia del altavoz conectado, obtendremos una potencia u otra de nuestro amplificador. Estos valores son,

- 5V into 4Ω - 3W max
- 3.3V into 4Ω - 1.3W max
- 5V into 8Ω - 1.8W max
- 3.3V into 8Ω - 0.8W max

Cometido. El propósito de este amplificador será recibir la salida de audio de la Raspberry Pi a través de los pines GPIO correspondientes, amplificarla y enviarla al altavoz analógico de bobina móvil. Resumidamente, será el audio incorporado de nuestra consola.

Porqué de su elección.

Para escoger un módulo concreto, verificamos los siguientes requisitos para cada posible modelo.

- Módulo en placa PCB
- Hileras de pines de 2,54 (aptas para cables dupont de 2,54).
- Salida (positivo y negativo) para, mínimo, un altavoz
- Tipo I2S
- Conversor DAC incorporado en placa
- Debe existir algún controlador del DAC para el SO a instalar (RPi OS Desktop 64, Debian 12).
- Entrada de voltaje apta en relación a la salida de voltaje de los pines GPIO de la RPi seleccionada
- Ganancia de unos 15 dB

Dicho análisis se encuentra en la [siguiente tabla](#).

Origen.

Material de un miembro del equipo (Iván).

2.32.8 Altavoz Weewooday 2W 8Ω

Descripción técnica. Como su nombre dicta, este altavoz de bobina móvil es apto para una salida de 2W y tiene una resistencia de 8 ohms. Dispone de un diámetro muy reducido, de unas 1,1 pulgadas.

Cometido. Producirá el sonido amplificado del MAX98357 a través de las señales analógicas enviadas al mismo.

Porqué de su elección. La elección de este modelo fue dada a partir del cumplimiento de los siguientes requisitos,

- **De bobina móvil**
- **8 ohmios a 2 o 3 W (vatios).** Las razones son las siguientes,
 - Primera: Si alimentamos el MAX98357 con 3.3V, podremos tener una salida de máxima 0.8W si el altavoz es de 8 ohmios, lo cual nos proporcionará suficiente volumen a cómo mucho 1 metro de distancia.
 - Segunda: Conectar un altavoz de 8 ohms al amplificador alimentado con 3.3V supondría un consumo de corriente de 240mAh por parte del amplificador más el consumo de corriente del amplificador solo por el hecho de estar conectado a una fuente de energía. Ahora bien, este es un consumo de corriente que consideramos seguro teniendo en cuenta que los pines de 3.3V no dan más de **500mAh**. Si el altavoz comprado fuese de 4 ohmios, entonces la potencia de salida máxima sería de unos **1.3W**, lo cual implica un consumo de **400mAh más el del propio amplificador**. Este valor estaría muy cerca del límite del pin de 3.3V y podría suponer un riesgo para el ordenador. Por ello creemos que lo anterior contado es la mejor opción.

- **Con cables de positivo y negativo ya soldados**

Dicho análisis se encuentra en la [tabla siguiente](#)

Origen. Material de un miembro del equipo (Iván).

2.32.9 Cables dupont

Descripción técnica. Este tipo de cables permiten el establecimiento de conexiones eléctricas a través de pines de 2,54mm.

Cometido. La función de estos componentes será la de establecer las conexiones eléctricas entre los pines del módulo amplificador MAX98357A y los pines GPIO correspondientes de la RPi 5.

Porqué de su elección. Creemos que es la mejor forma de establecer conexión entre estos dos componentes (el amplificador y la RPi) por el hecho de que no es necesario ningún tipo de soldadura. Obviamente, debido a lo anterior, las conexiones son menos íntegras pero para una máquina inamovible es más que suficiente.

Origen. Aportación por un miembro del equipo (Iván).

2.32.10 Tarjeta microSDXC 64GB

Descripción técnica. La siguiente tarjeta microSD tiene un espacio de almacenamiento de 64GB. Debido a las marcas en el frontal de la microSD, podemos saber que la velocidad de escritura y lectura puede llegar hasta un máximo de 104MB/s (respectivo de la clase UHS-I). Además, sabemos que la velocidad mínima de escritura será de unos 30MB/s gracias a la marca U3.

Cometido. Será el almacenamiento no volátil principal de nuestra RPi. Pues, estos SBC no disponen de un disco duro integrado.

Porqué de su elección. Para la elección analizamos los siguientes requisitos. Dicho análisis de los mismos [a continuación](#).

- Mínimo 32GB a máximo 2TB
- **Interfaz de bus DDR50 o SDR104.** Nuestro modelo de RPi solo soporta este tipo de interfaces. Preferiblemente, es mejor la interfaz SDR104, cómo se dijo anteriormente, ya que ofrece un bus de E/L más rápido.

Origen. Aportado por un miembro del equipo (Iván).

2.32.11 Dualshock 4

Descripción técnica. Este *gamepad* de la consola PlayStation 4 ofrece una cruceta, 4 botones laterales, dos *joysticks*, 4 gatillos y 2 botones centrales. Dispone de un puerto microUSB que puede ser utilizado para la conexión del mando y la carga de la batería de 800mAh a 5V.

Cometido. Con ello daremos al usuario una forma de interactuar con la consola. Será necesario solo la cruceta, botones centrales y dos de los botones laterales con el fin de poder *simular* el *gamepad* de la NES.

Porqué de su elección. Para su elección analizamos los siguientes requisitos,

- Interfaz USB con conector correspondiente a los puertos USB de la RPi
- Interfaz Bluetooth (opcional).
- Que disponga de cruceta situada arriba a la izquierda
- Que disponga de dos botones centrales.
- Que disponga de 4 botones situados a la derecha.

Dicho análisis se encuentra en la siguiente tabla.

Origen. Aportados por dos miembros del equipo (Iván e Iker)

2.32.12 Fuente de alimentación oficial de RPI (27W)

Descripción técnica. La fuente de alimentación a utilizar es la oficial para este tipo de ordenadores, más específicamente, para Raspberry Pi 5. Dispone de una potencia total de 27W.

Cometido. Con ello podremos alimentar el núcleo de nuestra consola y todos sus periféricos.

Porqué de su elección. Antes de poder establecer los requisitos, tuvimos que definir el consumo de corriente de cada componente.

	Cantidad	Componente	Corriente	Voltaje
Raspberry Pi 5	x1	Placa sin interfaces	800mAh	5V
	x1	HDMI	50mAh	5V
	x1	microSDXC	20 - 100mAh	3.3V
	x1	Amplificador	242,2mAh	3.3V
	x1	Teclado y ratón	100mAh	5V
	x2	Dualshock 4 (USB 3.0)	1600mAh (800mAh cada uno)	5V
TOTAL			2.892,2mAh	

Amplificador: El consumo que tendrá alimentado con 3.3V y conectado a un altavoz de 9 ohms será de **240mAh**. Lo sabemos gracias a que en la [documentación oficial de Adafruit](#) se muestra que la potencia de salida es de **0.8W**, lo único a hacer fue el hecho de dividir **5 / 0.8**, que nos da **0.24 A (240mAh)**.

Ahora bien, el consumo de reposo del amplificador sería de unos **2.4mAh**, lo cual conviene sumar al valor anterior. Por lo tanto, el valor total de consumo será de **242,4 mAh**.

Dualshock 4: El consumo de corriente cuando estamos cargando un DS4 es de máximo unos 800mAh. Dependiendo del tipo de USB, el amperaje variará.

- **USB 2.0: Máximo unos 500mAh.** Esto es debido a que la corriente máxima que puede proporcionar este estándar son solo **500mAh**.
- **USB 3.0: Máximo unos 800mAh.** Con USB 3.0, la corriente máxima que se puede proporcionar es de 900mAh. De todas formas, el consumo solo será de 800mAh, ya que es lo máximo que puede exigir este gamepad al cargarse.

Hay que tener en cuenta que los valores máximos para cada estándar también dependen del hardware que los mantiene. Por último, creemos que es mejor tener en cuenta el valor de carga del DS4 con USB 3.0, ya que nunca sabremos cuando el usuario va un puerto USB u otro (2.0 o 3.0) de la Raspberry Pi 5 (ya que dispone de dos puertos por cada tipo). [Más información.](#)

Hecho lo anterior, pasamos a la declaración de los requisitos y su posterior análisis en la [siguiente tabla.](#)

- Conector de carga para la Raspberry Pi correspondiente con el conector de carga en la misma
- Valores de corriente y voltaje correspondientes con el modelo escogido según la [tabla siguiente.](#)
- Que el consumo de cada periférico se encuentre por debajo del límite permitido por el hardware con la fuente de alimentación seleccionada

Origen. Componente adquirido por el centro en conjunto con la RPi 5 y la carcasa oficial.

2.32.13 Brazaletes antiestática IFIXIT

Descripción técnica. El brazaletes dispone de una pieza metálica que hace contacto con la piel y, a su vez, hace contacto con un cable de cobre adjunto a unas pinzas que están adheridas al componente a tratar.

Cometido. Evitar el riesgo de daño por descarga electrostática a la hora de la manipulación de nuestra RPi 5.

Porqué de su elección. Económico y de una marca reconocida.

Origen. Adquirido por los miembros del equipo (online).

Darwin

2.32.14 Servidor Intel-Xeon (Supermicro)

Descripción técnica. Este servidor de 4U, nosotros usamos sólo una unidad, es un servidor retirado de la UPC, pero que todavía sigue en funcionamiento. Este dispone,

- De 3 bahías para discos duros por unidad
- 2 puertos LAN por unidad. Uno de ellos pertenece a dicha unidad, mientras que la segunda interfaz es perteneciente al microordenador BMC.

Cometido. Nuestro servidor supermicro tendrá numerosos cometidos en el proyecto. Principalmente...

- Almacenará el catálogo de ROMs de nuestro lanzador de emuladores.
- Hará de servidor DNS y resolverá los nombres de nuestro dominio.
- Hará de servidor apache2 y almacenará nuestra web de tienda realizada con **wordpress**.
- Hará de servidor de **BBDD mySQL** y almacenará la base de datos de nuestro gestor de contenidos.
- Hará de **MTA** y **MDA** (postfix y dovecot) para nuestro dominio de empresa. Con ellos los usuarios dados de alta podrán enviarse correos entre e-mails del mismo sitio.
- Realizará un resguardo (*snapshots*) de la información relevante de nuestro servidor con Kopia.
- Recopilará **Logs** y **métricas** del Cliente Desktop y del servidor y los graficará mediante **grafana**, usando Loki y Prometheus.
- Constará de **wireguard** para que la consola se descargue las ROMs mediante la red con una **VPN**.
- Utilizará el paquete **IPMI TOOLS** para que el servidor pueda ser encendido y apagado desde nuestras casas.
- Tendrá un **servidor de correo electrónico** (usando el DNS del servidor) para comunicarnos entre nosotros.

Porqué de su elección. Debido a la gran cantidad de tareas que se deben llevar a cabo por una sola máquina, pensamos que lo mejor era utilizar una máquina con una capacidad de cálculo y almacenamiento suficientes con los que garantizar una buena disponibilidad y rendimiento de los servicios.

Origen. Material de la sala Darwin.

2.32.15 Cliente Desktop (Darwin)

Descripción. El cliente del que disponemos en Darwin es un ordenador de sobremesa con un sistema operativo Ubuntu Desktop instalado.

Cometido. Este ordenador se encarga del encendido y apagado de nuestro servidor a través del BMC. La razón de tener que incorporar este intermediario es debido a que no es posible añadir al BMC cómo peer de nuestra VPN y, por ende, enviar comandos directamente al mismo con ipmi-tools. Con este cliente, accedemos a él por SSH y, cómo este se encuentra en la red Darwin, enviamos las señales al BMC con ipmi-tools. Una vez encendido el servidor, accedemos a él directamente con la VPN.

Porqué de su elección. Escogimos este ordenador, ya que es de bajo consumo y no realiza ningún ruido. Esto último es vital si queremos que se mantenga encendido siempre, pues, no molestaremos a aquellos que trabajen allí.

Origen. Material del centro

2.4 Definición de las funcionalidades

Citamos a continuación los diferentes casos de uso de nuestro proyecto,

2.41 Descarga de ROMs a través del lanzador de emuladores

A partir del sistema de menú-submenú de nuestro lanzador de emuladores, el usuario tendrá la posibilidad de acceder a un submenú, el cual contendrá una lista con todas las ROMs contenidas en el repositorio de nuestro servidor. El usuario verá una línea por cada ROM contenida y tendrá la opción de descargarla presionando uno de los botones laterales del gamepad. A la hora de la descarga, el script se encarga de almacenarlos en un directorio común. Estas ROMs podrán ser utilizadas posteriormente en el emulador.

2.42 Lanzado de emuladores contenidos en la RPI 5

Una de las opciones de nuestro script será la ejecución del emulador contenido en la propia RPI de forma local. Con ello conseguiremos que el cliente sea capaz de ejecutar los programas que guste sin necesidad de introducir líneas de texto necesarias en la CLI ni saber la ruta al programa.

2.43 Acceso a una web de tienda

Del lado de los servicios y nuestro servidor, tenemos nuestra tienda virtual montada a partir de un plugin de **wordpress**. El acceso al storefront debería ser realizado por un host que cumpla con las siguientes características

- **Con una IP de Darwin o de nuestra VPN**
- **DNS de nuestro dominio configurado cómo opción primaria o secundaria.**

Una vez hemos accedido, será posible

- Autorregistro cómo usuario ordinario
- Navegación sobre un catálogo de productos con precios normales y en oferta
- Añadido de productos a un carrito de compra para un, ficticio, proceso de facturación con métodos de pago cómo paypal.
- Descarga de información relacionada con los productos de la tienda.
- Cambio de idioma de la tienda

2.44 Envío de correos entre usuarios del dominio **FIXMaster**

Otra funcionalidad importante será la de nuestro servicio de correo. Los hosts que tengan un MUA con una dirección de correo electrónico de nuestra empresa podrán enviar correos a otros usuarios del mismo dominio. Cabe recalcar que los hosts en una red diferente de Darwin deberán ser obligatoriamente **peers** de nuestra VPN con el fin de poder acceder a su servidor IMAP (contenido en el supermicro de Darwin) y descargar los correos entrantes.

2.5 Gastos de nuestro proyecto

A continuación adjuntamos el enlace a la tabla de gastos de nuestro proyecto, donde se ve claramente todos los componentes, origen, costes... Y el porcentaje de aportación de cada miembro y los gastos asumidos por cada uno de ellos. [Enlace](#)

3. Procesos de desarrollo

3.1 Aula Darwin (Servidor Xeon y Ordenador)

Después de solucionar los problemas que tuvimos con el servidor, este es el contenido final del servidor XEON

3.11 Escoger servidor XEON

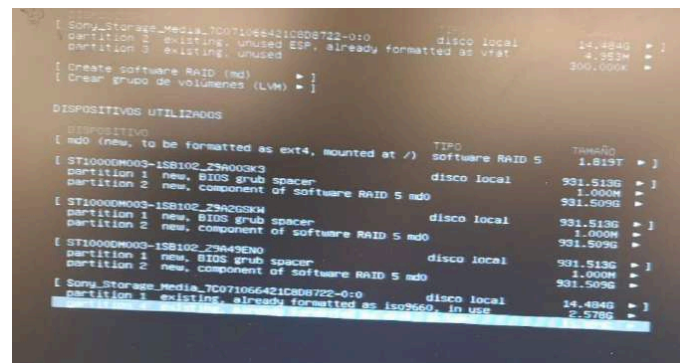
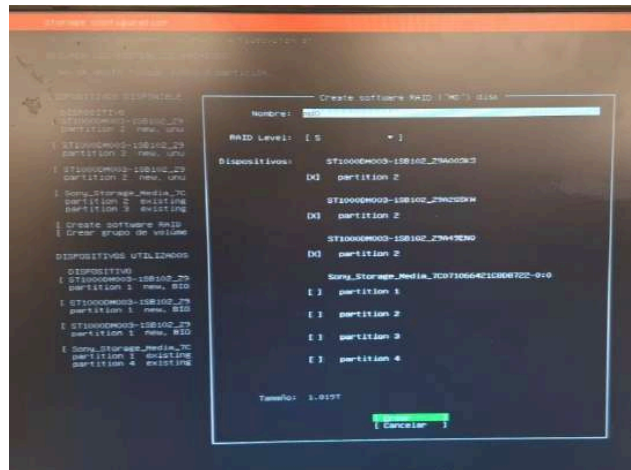


3.12 Discos

Escogimos 3 discos de 1TB cada uno solicitados a Gonzalo.

3.13 Instalación de Ubuntu Server 24.04

Ya que nos había dado problemas el hecho de hacer la Raid, hablamos tanto con José Camuñez como Víctor Carcelero y llegamos a la conclusión de que lo mejor era hacer el RAID desde la instalación, además de que la última vez que intentamos hacer la Raid 1, la hicimos sin problema, pero al querer añadir el tercer disco para tener todo, no sabíamos cómo hacerlo y, ya que íbamos a hacer un RAID con 3 discos, se nos abrió la posibilidad de hacer un RAID 5 en vez del 1.



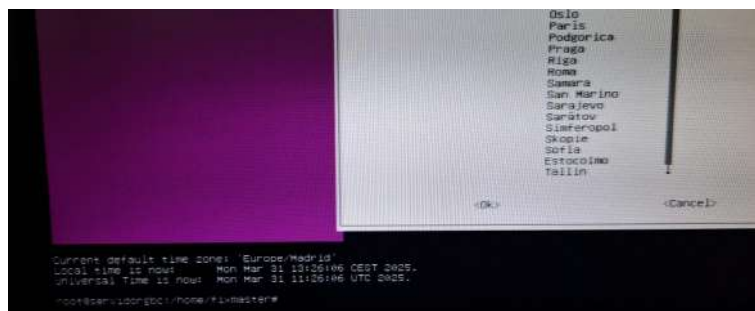
Comprobación de la instalación y la RAID 5 con el comando /proc/mdstat



3.14 Configuración básica del servidor

- a. Netplan:
 - i. IP (192.168.19.250)
 - ii. Máscara de red (255.255.255.0)
 - iii. MAC (00:25:90:77:a0:8e)
 - iv. GW (192.168.19.10)
 - v. DNS (192.168.19.10)
- b. Hostname / Host
 - i. supermicro
- c. Usuario
 - i. fixmaster
- d. Zona Horaria
 - i. Madrid

```
2: enp2s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:25:90:77:a0:8e brd ff:ff:ff:ff:ff:ff
    inet 192.168.19.250/24 brd 192.168.19.255 scope global enp2s0f0
        valid_lft forever preferred_lft forever
    inet6 fe80::225:90ff:fe77:a08e/64 scope link
        valid_lft forever preferred_lft forever
```



3.15 Wireguard

Rehicimos el Wireguard por los problemas posteriormente mencionados con el servidor, por lo que decidimos hacer varios cambios.

1ro: Script para VPN.

- Hicimos un script sencillo para introducir las VPNs en el fichero wg10.conf, este script te pregunta por la IP que quieres y por la clave privada del cliente y una vez respondido los pasa al fichero correspondiente.

```
#!/bin/bash
echo "Introduce el valor de la CLAVE vpn que has recibido (se parece a esto: yEg
AqVg3Z5J16nPowk3dwhAsRt7b+h9U+3GerT7QXQ=)"
read privkey
echo "Introduce el valor de la ip VPN que has recibid (se parece a esto: 10.8.10
1.38) -> ponla SIN mascara"
read ip
echo "[Interface]" > /etc/wireguard/wg10.conf
echo "PrivateKey = \"$privkey\" >> /etc/wireguard/wg10.conf
echo "Address = \"$ip/24\" >> /etc/wireguard/wg10.conf
echo "" >> /etc/wireguard/wg10.conf
echo "[Peer]" >> /etc/wireguard/wg10.conf
echo "Endpoint = wireguard.elputg.xeill.net:48002" >> /etc/wireguard/wg10.conf
echo "PublicKey = ax1RKE62Q8tkkcv8HfD2w4ckgu1wyHgGF0e6Lw44" >> /etc/wireguard/wg10.conf
echo "AllowedIPs = 10.8.102.0/24" >> /etc/wireguard/wg10.conf
echo "PersistentKeepalive = 25" >> /etc/wireguard/wg10.conf
systemctl enable wg-quick@wg10
systemctl restart wg-quick@wg10
ip -c a show wg10
```

2do: Utilizamos la red 10.0.102.0/24

- Ya que teníamos problemas con la anterior red y nos estaba frustrando, decidimos hacer un reset y empezar desde 0 el wireguard y probar con las claves que nos dio Juan Morote. Aquí la configuración de las máquinas:

PC usado para el IPMI

```
root@usuario-HP-Compaq-6000-Pro-SFF-PC: /home/usuario
root@servidorgbc: /home/fixmaster x root@usuario-HP-Com
[Interface]
PrivateKey = yN5o3aq8M0/c57oENEVp+MVG+qWe5JNMd+1NdEbyD3Y=
Address = 10.0.102.48/24

[Peer]
Endpoint = wireguard.elpuig.xeill.net:40002
PublicKey = ZmyzjyMinfDqvOzaIKf/WcAr9jjlaSKyFOksYxGY+FE=
AllowedIPs = 10.0.102.0/24
PersistentKeepalive = 25
```

Servidor Fixmaster

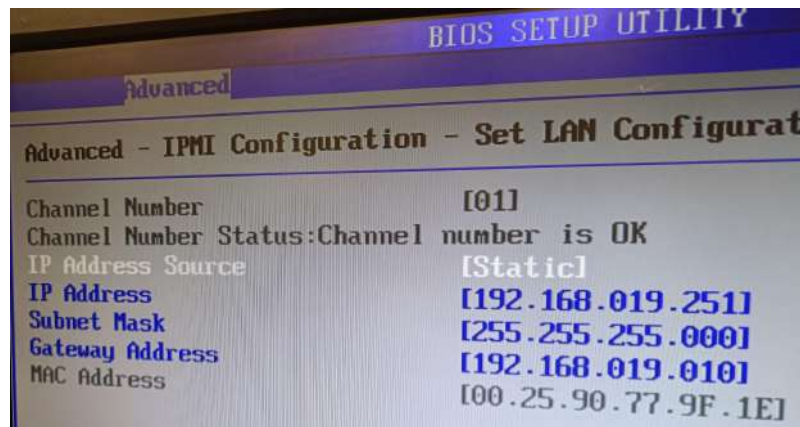
```
root@servidorgbc: /home/fixmaster x root@usuario-HP-Compa
[Interface]
PrivateKey = 6PEXaLzobjuSKbT4s0XK0P52XJoJ1WrxVlgqKl3vTVM=
Address = 10.0.102.43/24

[Peer]
Endpoint = wireguard.elpuig.xeill.net:40002
PublicKey = ZmyzjyMinfDqvOzaIKf/WcAr9jjlaSKyFOksYxGY+FE=
AllowedIPs = 10.0.102.0/24
PersistentKeepalive = 25
```

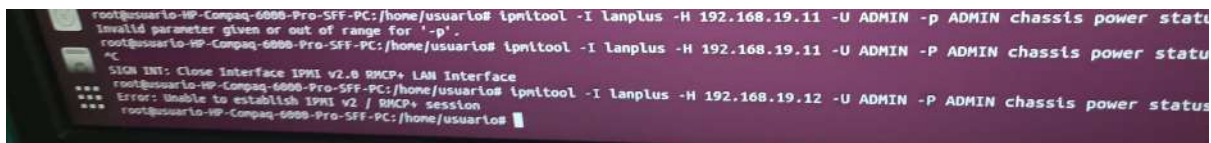
3.16 Instalación IPMI

Para no tener que bajar al servidor y encenderlo/apagarlo para poder usar el **wireguard**, decidimos instalar IPMI, con el que podríamos encender y apagar el servidor desde la clase con una máquina virtual si estamos en la misma red.

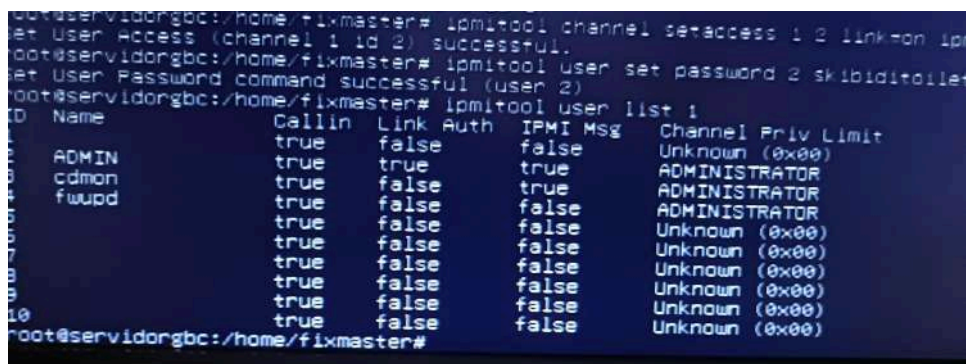
Primero fuimos a la BIOS para ponerle una IP estática al IPMI, en este caso le establecimos la IP 192.168.19.251/24, poniendo el gateway de darwin (192.168.19.10). Sin esto no sabríamos qué ip tiene el servicio.



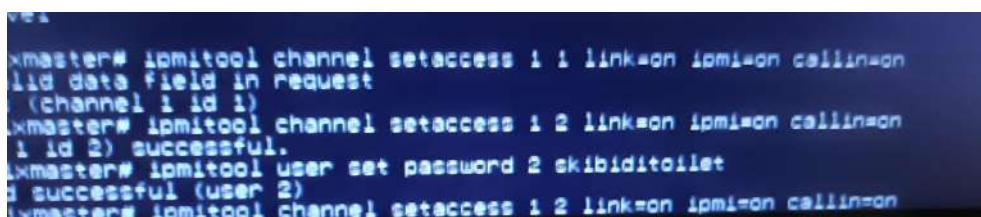
Al comprobar el comando para ver el estado del servidor vimos que no estaba funcionando, algo estaba mal, después de investigar un tiempo me di cuenta de que el IPMI tiene una configuración por defecto y eso era lo que estaba dando errores.



Esta es la configuración actual del usuario ADMIN del IPMI



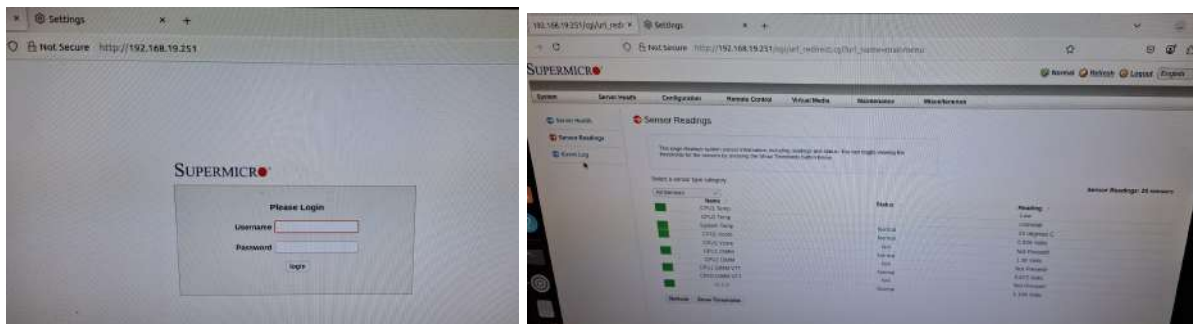
Pudimos ver que el error que estaba provocando que no funcionase eran los permisos del usuario ADMIN, ya que tenía el Callin y el Link auth en false. Sin el Callin en True no podemos iniciar sesión con el usuario y el link auth nos permite usar la contraseña establecida. Una vez activados, cambiamos la contraseña a una que pudiéramos recordar con facilidad, ya que según Internet la contraseña por defecto puesta por supermicro es ADMIN, pero no nos la quisimos jugar y establecimos una contraseña por nuestra cuenta.



Una vez ya hechos estos cambios, comprobamos el funcionamiento con power chassis status y vemos que funciona.

```
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# ipmitool -I lanplus -H 192.168.19.12 -U ADMIN -P ADMIN chassis power status
Error: Unable to establish IPMI v2 / RMCP+ session
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# ping 192.168.19.12
PING 192.168.19.12 (192.168.19.12) 56(84) bytes of data:
64 bytes from 192.168.19.12: icmp_seq=1 ttl=64 time=0.542 ms
64 bytes from 192.168.19.12: icmp_seq=2 ttl=64 time=0.371 ms
64 bytes from 192.168.19.12: icmp_seq=3 ttl=64 time=0.357 ms
64 bytes from 192.168.19.12: icmp_seq=4 ttl=64 time=0.370 ms
^C
--- 192.168.19.12 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3083ms
rtt min/avg/max/ndev = 0.357/0.418/0.542/0.076 ms
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# ping 192.168.19.12
PING 192.168.19.12 (192.168.19.12) 56(84) bytes of data:
64 bytes from 192.168.19.12: icmp_seq=1 ttl=64 time=0.459 ms
64 bytes from 192.168.19.12: icmp_seq=2 ttl=64 time=0.432 ms
64 bytes from 192.168.19.12: icmp_seq=3 ttl=64 time=0.386 ms
64 bytes from 192.168.19.12: icmp_seq=4 ttl=64 time=0.451 ms
64 bytes from 192.168.19.12: icmp_seq=5 ttl=64 time=0.493 ms
64 bytes from 192.168.19.12: icmp_seq=6 ttl=64 time=0.353 ms
64 bytes from 192.168.19.12: icmp_seq=7 ttl=64 time=0.309 ms
64 bytes from 192.168.19.12: icmp_seq=8 ttl=64 time=0.361 ms
64 bytes from 192.168.19.12: icmp_seq=9 ttl=64 time=0.368 ms
^C
--- 192.168.19.12 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8210ms
rtt min/avg/max/ndev = 0.309/0.401/0.493/0.037 ms
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# ipmitool -I lanplus -H 192.168.19.12 -U ADMIN -P ADMIN chassis power status
Error: Unable to establish IPMI v2 / RMCP+ session
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# ipmitool -I lanplus -H 192.168.19.12 -U ADMIN -P skibdttoilet chassis power status
Chassis Power is on
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario#
```

El servidor Xeon tiene un miniordenador encendido 24/7 (siempre y cuando esté conectado a la corriente), este miniordenador se le dice BMC, el cual tiene varias funciones interesantes, de hecho, tiene una función de encender o apagar el servidor, esta función ya la usamos antes, pero con comandos, ya que también se puede de manera gráfica entrado a la IP del BMC del servidor.



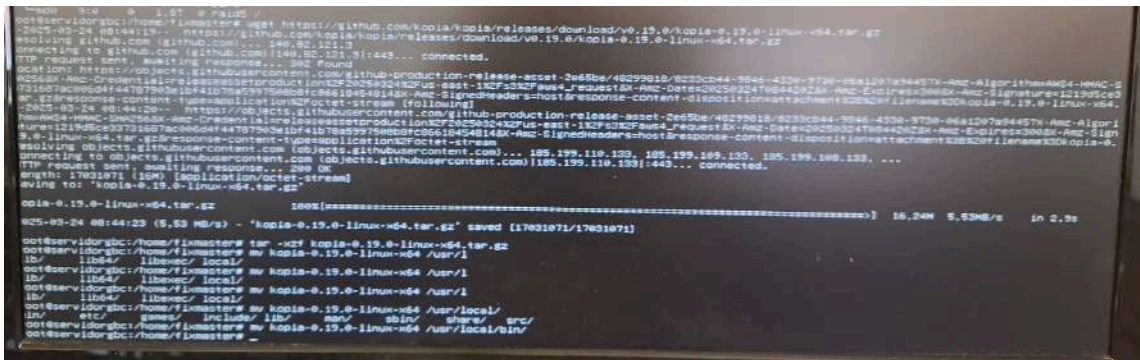
3.17 Instalación Kopia

Debido a la gran cantidad de problemas que habíamos tenido con el servidor, decidimos instalar “Kopia”, herramienta fundamental para crear snapshots y poder restaurar el servidor.

Para instalarlo, seguimos las instrucciones proporcionadas por Victor Carcelar en la [página web del centro](#) y en clase.

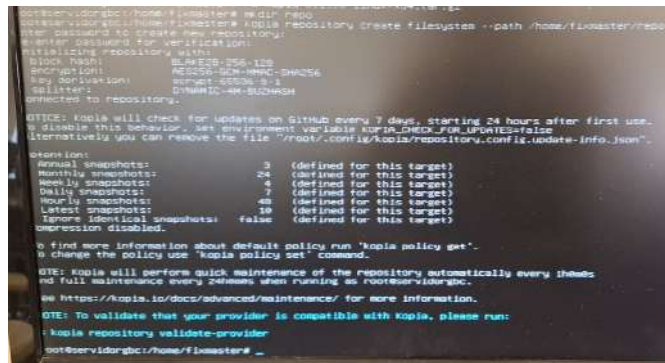
3.17.1 Descargar

Para descargar Kopia, cogimos la versión más reciente a la hora de instalarlo (0.19.0), expresado en [github](#) e instalándolo con el comando wget. Posteriormente, descomprimos el zip con el comando “tar -gz” y lo movemos al directorio /usr/local/bin para que esté en el PATH.



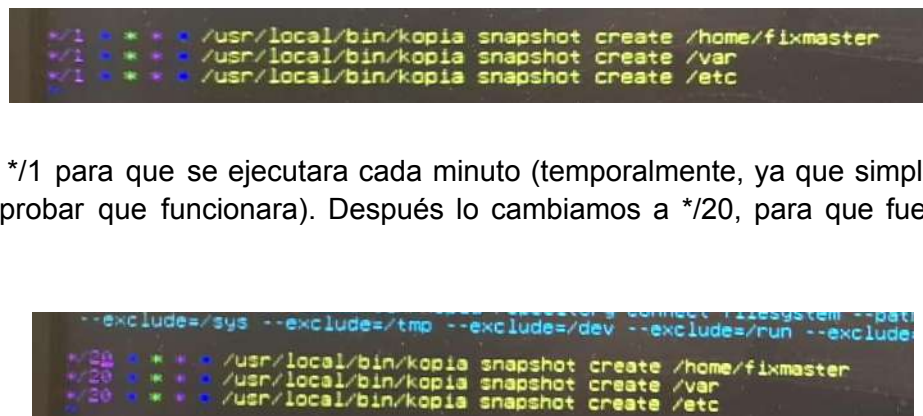
3.17.2 Crear el repositorio

Para crear el repositorio, primero hay que saber donde crearlo, decidimos hacerlo en `/home/fixmaster/repo`



3.17.3 Crear snapshot

Para crear una snapshot que se ejecute cada x tiempo, lo ponemos en el “crontab -e”. Escogimos hacer la snapshot del contenido de /var (ya que está la web y el archivo .hosts del bind9), el /etc (ya que es el archivo que guarda gran parte de la configuración del servidor (apache2, bind9 y wireguard primordialmente) y el /home/fixmaster para todo lo que tenga el usuario creado o cree.



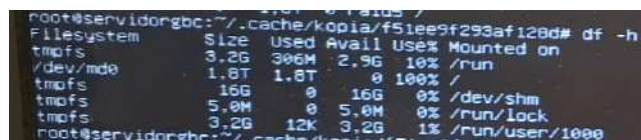
3.17.4 Comprobación

Con el comando “kopia snapshot list” puedes comprobar los snapshots que se han creado, y se puede ver que se estaban realizando cada minuto, por lo tanto, funcionaba. Pero nos dimos cuenta de algún problema.



3.17.5 Problemas encontrados

Debido a dejar el servidor encendido durante un par de días, al encender el servidor, nos dimos cuenta de que el espacio del raid estaba ocupado al 100%, esto se debe a que, el snapshot se estaba creando del propio repositorio, entonces era un bucle infinito de snapshots.



3.17.6 Solución

La solución tomada, gracias a preguntar a Victor Carcelar, fue crear el repositorio por sftp en una máquina externa (la elegida sería el ordenador utilizado para la consola IPMI). Esto beneficia en que, si petara o se incendiara el servidor, se rompiera un disco o cualquier otro motivo que haga dejar de funcionar el servidor, tener todo guardado en un ordenador externo, lo cual te permite restablecer todo en el mismo servidor u otro distinto.

3.17.61 Eliminar repositorio antiguo

Para empezar con este cambio, eliminamos todo el repositorio para liberar el espacio (con `rm -r`)



3.17.62 Crear repositorio por sftp

En el ordenador donde se creará el repositorio, creamos el directorio donde se guardarán las snapshots (mkdir repo), y después le cambiamos los permisos y los propietarios.

```

Music/                               Videos/
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# ls -ld /home/usuario/repos/
drwxr-xr-x 2 root root 4096 abr  9 10:52 /home/usuario/repos/
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# chown usuario:usuario /home/
e/usuario/repos/
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario# chmod 777 /home/usuario/re
po
root@usuario-HP-Compaq-6000-Pro-SFF-PC:/home/usuario#

```

En el servidor, crearemos el repositorio por sftp. Ya que al conectarse por sftp, es como conectarse por ssh, previamente a la creación del repositorio, haremos ssh a la máquina para que aparezca en el archivo “known_hosts”, una vez realizado el ssh, crearemos el repositorio sftp con el siguiente comando:

[illegible]

3.17.63 Crear snapshot

Para no tener muchas líneas ejecutándose en el “crontab -e”, decidimos hacer un archivo ejecutable donde pondremos todas las instrucciones y así solo tendremos que decir en el crontab que ejecute el archivo, el cual, obviamente, tiene que ser ejecutable, por eso le otorgamos permisos de ejecución.

```
#!/bin/bash

# Realizo un snapshot de los datos del usuario
/usr/local/bin/kopia snapshot create /home/fixmaster

# Realizo un snapshot de la web
/usr/local/bin/kopia snapshot create /var/www

# Realizo un snapshot del /var/cache para tener el archivo .hosts del bind9
/usr/local/bin/kopia snapshot create /var/cache/bind

# Realizo un snapshot del /etc/apache2 para tener toda la configuración de apache
/usr/local/bin/kopia snapshot create /etc/apache2

# Realizo un snapshot del /etc/wireguard para tener toda la configuración del wireguard
/usr/local/bin/kopia snapshot create /etc/wireguard

# Realizo un snapshot del /etc/bind para tener toda la configuración del bind9
/usr/local/bin/kopia snapshot create /etc/bind

# Todas tus tareas
#
```

```
# For example, you can run a backup
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups
#
# For more information see the man
# m h dom mon dow   command
#
# * * * * * /root/bin/backup
```

```
servidorgbc: /bin#  
  chmod +x backup
```

```
-rwxr-xr-x 1 root root 748 abr 9 12:34 backup*
```

3.17.64 Comprobación

Imagen de comprobación de los snapshots desde el servidor:

```
root@server:~# ls -la /var/lib/snapshots/
total 12
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
drwxr-xr-x 2 root root 4096 Apr 9 12:07
```

Imagen de comprobación del directorio que hace de repositorio sftp:

```
usuario@usuario-HP-Compaq-6000-Pro-SFF-PC: $ ll repo
total 44
drwxrwxrwx 7 usuario usuario 4096 abr 9 12:36 /
drwxr-x-- 20 usuario usuario 4096 abr 9 10:52 /
drwxrwxr-x 3 usuario usuario 4096 abr 9 12:07 /
-rw-rw-r-- 1 usuario usuario 30 abr 9 12:07 kopta.blobcfg.f
-rw-rw-r-- 1 usuario usuario 1209 abr 9 12:36 kopta.maintenance.f
-rw-rw-r-- 1 usuario usuario 1101 abr 9 12:07 kopta.repository.f
drwxrwxr-x 9 usuario usuario 4096 abr 9 12:38 /
drwxrwxr-x 29 usuario usuario 4096 abr 10 11:59 /
drwxrwxr-x 29 usuario usuario 4096 abr 10 11:59 /
-rw-rw-r-- 1 usuario usuario 43 abr 9 12:07 .shards
drwxrwxr-x 3 usuario usuario 4096 abr 9 12:07 /
```

3.18 Apache2

Al principio, dudábamos en instalar Apache o Nginx, pero determinamos que la configuración de apache era más sencilla, ya que hemos realizado varias actividades y lo conocemos mejor. Lo único malo de Apache es que no permite las conexiones http 3, cosa que Nginx sí, pero, al ser tan nuevo, aún están de pruebas y mejor directamente utilizar http 2.

3.18.1 Instalar Apache2

Hemos hecho “apt install apache2”. A continuación explicaremos la configuración llevada a cabo.

```
root@server:~# apt update && apt install apache2
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [671 kB]
Hit:6 http://es.archive.ubuntu.com/ubuntu noble/main Translation-es [325 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu noble-security/main amd64 Components [8,368 B]
```

3.18.2 Configuración Apache2

3.18.21 Archivo 000-default-ssl

De ServerName escogemos “fixmaster.puig.es” para poder utilizar el nombre en vez de la IP. Creamos nuestro correo y lo ponemos en ServerAdmin, para que en caso de problemas, nos puedan contactar. Hacemos la redirección para poder hacer uso del protocolo HTTPS, es decir, con seguridad.


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

ServerAdmin fixmaster.eipug@gmail.com
DocumentRoot /var/www/wordpress
Redirect "/" "https://fixmaster.puig.es"

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

3.18.22 Archivo default-ssl.conf

Este archivo es el que configura los virtual hosts que llegan al puerto 443 (HTTPS), configurando los certificados de seguridad (en nuestro caso, autofirmados). También introducimos la línea de comando "Protocols" para indicar que se permite HTTP/2 y HTTP/1.1. Finalmente, añadimos la línea "Redirect", para que cualquier consulta que sea mediante la IP (tanto de la red LAN como de la red VPN), sean redirigidas al nombre de nuestra web.

```
<VirtualHost *:443>
ServerAdmin fixmaster.eipug@gmail.com
ServerName fixmaster.puig.es
DocumentRoot /var/www/wordpress
Redirect "https://192.168.19.258/" "https://fixmaster.puig.es"
Redirect "https://19.0.102.43/" "https://fixmaster.puig.es"
Protocols h2 http/1.1

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/key1-fixmaster.pem
SSLCertificateKeyFile /etc/ssl/private/key2-fixmaster.key
# Secure Certificate Chain:
```

Para activar estos sitios, hacemos uso de a2ensite "nombre del sitio", en este caso solo activamos 1 (el default-ssl), ya que, por defecto viene activado el 000-default, ya que es el archivo de configuración sin seguridad y no estás obligado (aunque sí, recomendado) a poner seguridad.

```
oot@servidorgbc:/etc/apache2# a2ensite default-ssl
abling site default-ssl.
o activate the new configuration, you need to run:
systemctl reload apache2
```

3.18.23 Certificados

Los certificados SSL deben ser otorgados por una CA (Certificate Authority), la cual revisa tu página web y te lo otorga si realmente es seguro, pero esto tiene un coste por ese certificado, entonces, existe la posibilidad de autocrearse tu certificado, pero, al cliente le notificara que no es un certificado válido de una CA. Para auto crear tu certificado, tienes que ejecutar el siguiente comando, y después en el archivo “default-ssl”, colocar las 2 siguientes líneas:

```

sites-available/default-ssl.conf 186L, 482B  escrito
root@servidor:~/etc/apache2# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/key2-fixmaster.key -out /etc/ssl/certs/key1-fixmaster.crt
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ES
State or Province Name (full name) [Some-State]:Barcelona
Locality Name (eg, city) [Some-Where]:Santa Coloma de Gramenet
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FIXMaster
Organizational Unit Name (eg, section) []:FIXMaster
Common Name (e.g. server FQDN or your name) []:FIXMaster
Email Address []:fixmaster@gmail.com
-----

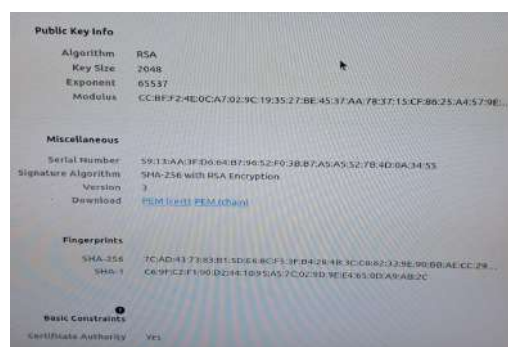
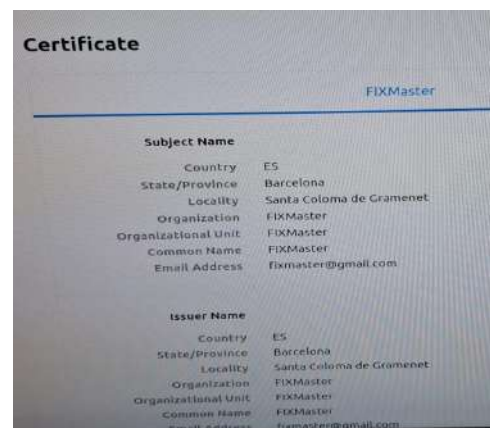
```

```

# If both key and certificate are stored in the same file, only
# SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/certs/key1-fixmaster.pem
SSLCertificateKeyFile    /etc/ssl/private/key2-fixmaster.key
#
# Server Certificate Chain:

```

Aquí tenemos la información que podrá ver el cliente sobre nuestro certificado:



3.18.24 Activar Módulos

Para poder hacer uso de https y la versión http2, tendremos que activar los siguientes módulos:

- SSL
- HTTP2

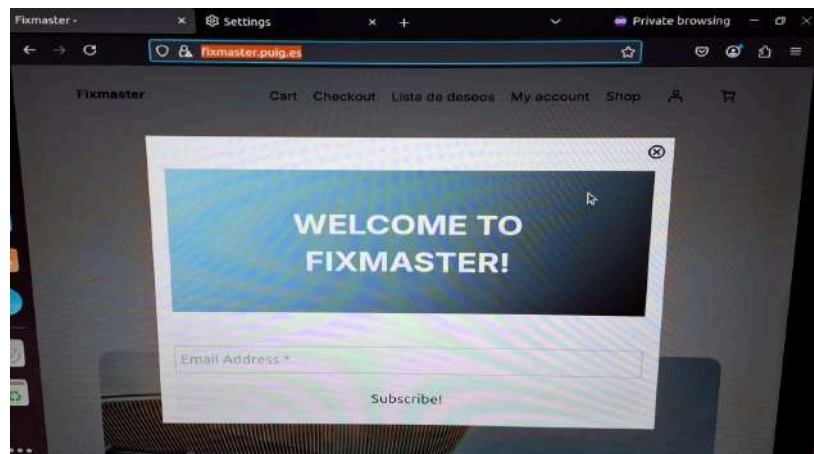
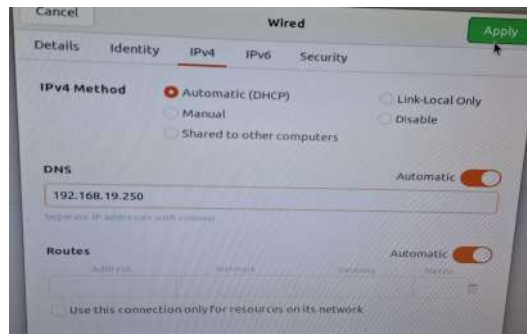
```
a2enmod ssl  
a2enmod http2
```

Una vez activados estos módulos y sitios, hay que reiniciar apache para activar la nueva configuración. Con “systemctl restart apache2”.

```
root@servidordngbc:/etc/apache2# systemctl restart apache2  
root@servidordngbc:/etc/apache2#
```

3.18.25 Comprobación

Para comprobar que funciona correctamente, cogimos un ordenador de la sala Darwin, pusimos de DNS, la IP de nuestro Servidor y buscamos tanto la IP como el nombre y ambos, te redirigen a nuestra página web (fixmaster.puig.es) con HTTPS.




```
ns fixmaster.gmail.com (
  1 : Serial
  10000 : Refresh
  3600 : Retry
  604800 : Expire
  60 : Minimum TTL
)
ns
192.168.19.250
fixmaster A 192.168.19.250
```

3.19.3 Añadir Servidor de Correo

En esta misma zona y archivo, añadiremos el servidor de correo de nuestra empresa, para ello, tendremos que declarar que fixmaster tiene además el sftp y el imap, ambos en el ordenador que utilizamos para el IPMI (Ordenador de la sala Darwin)

```
ns fixmaster.gmail.com (
  1 : Serial
  10000 : Refresh
  3600 : Retry
  604800 : Expire
  60 : Minimum TTL
)
ns
192.168.19.250
fixmaster A 192.168.19.250
fixmaster MX 10 smtp.fixmaster.puig.es.
smtp.fixmaster.puig.es. A 192.168.19.152
fixmaster MX 10 imap.fixmaster.puig.es.
imap.fixmaster.puig.es. A 192.168.19.152
```

3.19.4 Reiniciar el servicio

Cada vez que realizamos modificaciones en alguno de los archivos de configuración de bind9, tenemos que realizar “systemctl restart bind9” para reiniciar el servicio y se aplique la nueva configuración.

```
root@servidorgbc:/home/fixmaster# systemctl restart bind9
```

3.19.5 Comprobación

La comprobación es fácil de realizar, ya que puedes con el comando “host” hacer solicitudes a una ip en específico, es decir, me puedo preguntar a mí mismo por un nombre:

Web:


```
root@servidordbc:/home/fixmaster# host -a fixmaster.puig.es 127.0.0.1
Trying "fixmaster.puig.es"
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

;;->HEADER<- opcode: QUERY, status: NOERROR, id: 20990
;; flags: qr aa rd ra: QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
fixmaster.puig.es.                IN      ANY

;; ANSWER SECTION:
fixmaster.puig.es.                IN      A      192.168.19.250
fixmaster.puig.es.                IN      MX      10 smtp.fixmaster.puig.es.
fixmaster.puig.es.                IN      MX      10 imap.fixmaster.puig.es.

Received 93 bytes from 127.0.0.1#53 in 0 ms
```

Imap:

```
root@servidordbc:/home/fixmaster# host -a imap.fixmaster.puig.es 127.0.0.1
Trying "imap.fixmaster.puig.es"
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

;;->HEADER<- opcode: QUERY, status: NOERROR, id: 54712
;; flags: qr aa rd ra: QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
imap.fixmaster.puig.es.          IN      ANY

;; ANSWER SECTION:
imap.fixmaster.puig.es. 5      IN      A      192.168.19.152

Received 56 bytes from 127.0.0.1#53 in 0 ms
```

Smtp:

```
Received 93 bytes from 127.0.0.1#53 in 0 ms
root@servidordbc:/home/fixmaster# host -a smtp.fixmaster.puig.es 127.0.0.1
Trying "smtp.fixmaster.puig.es"
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

;;->HEADER<- opcode: QUERY, status: NOERROR, id: 12679
;; flags: qr aa rd ra: QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
smtp.fixmaster.puig.es.          IN      ANY

;; ANSWER SECTION:
smtp.fixmaster.puig.es. 5      IN      A      192.168.19.152

Received 56 bytes from 127.0.0.1#53 in 7 ms
root@servidordbc:/home/fixmaster#
```

3.19.6 Añadir DNS para VPN

Para añadir como DNS a la IP de la VPN, principalmente realizamos esto:

```
$TTL 5
@      IN      SOA      ns.fixmaster.gnail.com (
1      ; Serial
10800  ; Refresh
3600   ; Retry
604800 ; Expire
60     ; Minimum TTL
)

;
;
;
ns     IN      NS       ns
ns     IN      A        192.168.19.250
ns     IN      A        10.0.102.43

fixmaster IN A 192.168.19.250
fixmaster IN A 10.0.102.43
fixmaster IN MX 10 smtp.fixmaster.puig.es.
smtp.fixmaster.puig.es. IN A 192.168.19.152
fixmaster IN MX 10 imap.fixmaster.puig.es.
imap.fixmaster.puig.es. IN A 192.168.19.152
```

Posteriormente, hablando con Victor Carcelar, nos comentó que podíamos realizar un [DNS con vistas o Split Horizon](#).

3.19.62 DNS con vistas / Split Horizon

Esto nos servirá para dividir la red LAN de la VPN, es decir, si la consulta viene de la VPN, entregará una respuesta, y si viene de la red LAN, otorgará otra respuesta, ya que, lo que

estábamos realizando anteriormente, podía realizar alguna confusión, ya que entrega 2 IPs para la misma consulta y el navegador, intentaría las 2.

3.19.62.1 Archivo “/etc/bind/named.conf.local”

Para realizar el DNS con vistas, declararemos primero las 2 zonas (aunque sean las mismas).

- Trusted: Consultas realizadas desde la red 192.168.19.0/24, es decir, desde la red LAN.
- Guest: Consultas realizadas desde la red 10.0.102.0/24, es decir, desde la VPN

```
acl trusted { 192.168.19.0/24; localhost; };
acl guest { 10.0.102.0/24; };

view trusted {
    match-clients { trusted; };
    allow-recursion { any; };

    zone "puig.es" {
        type primary;
        file "trusted/puig.es.hosts";
    };
};

view guest {
    match-clients { guest; };
    allow-recursion { any; };

    zone "puig.es" {
        type primary;
        file "guest/puig.es.hosts";
    };
};
```

3.19.62.2 Archivo “guest/puig.es.hosts” (VPN)

Este archivo sería la respuesta para las consultas realizadas desde la VPN. También podríamos añadir el servidor de correo para este DNS, ya que el ordenador donde está almacenado el servidor de correo, también está en la VPN con la IP 10.0.102.48.

```
$TTL 5
@      IN      SOA     ns fixmaster.gmail.com (
1      ; Serial
10800  ; Refresh
3600   ; Retry
604800 ; Expire
60     ; Minimum TTL
)

@      IN      NS      ns
ns     IN      A       10.0.102.43
fixmaster IN A       10.0.102.43
```

3.19.62.3 Archivo “trusted/puig.es.hosts” (Red LAN)

Este archivo sería la respuesta para las consultas realizadas desde la red LAN.

```
TTL 5
      IN      SOA      ns fixmaster.gmail.com (
          1      ; Serial
        10800    ; Refresh
        3600     ; Retry
       604800    ; Expire
        60      ; Minimum TTL
      )

@      IN      NS      ns
ns     IN      A       192.168.19.250

fixmaster      IN      A       192.168.19.250
fixmaster      IN      MX      10      smtp.fixmaster.puig.es.
smtp.fixmaster.puig.es. IN      A       192.168.19.152
fixmaster      IN      MX      10      imap.fixmaster.puig.es.
imap.fixmaster.puig.es. IN      A       192.168.19.152
```

3.110 Página web

Para realizar nuestra página web teníamos varias posibilidades:

- Realizar una página web con HTML
- Tienda online (PrestaShop)
- Wordpress (Plugin Tienda + Blog)

Hemos escogido realizar nuestra página web con Wordpress porque es una opción “sencilla” y a la vez, práctica y compleja, es decir, gracias a los plugins se puede realizar una web muy currada y con una tienda, lo cual es perfecto para nuestra empresa.

3.110.1 Instalación Lamp Stack + Virtual Host

3.110.11 MySQL

Creamos la base de datos donde se almacenarán los datos del wordpress.
También creamos usuario y contraseña del administrador de la base de datos.

Database: fixmasterwordpress

User: fixmasterroot

Password: skibiditoilet

```
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 155 no se
root@servidorgbc:/home/fixmaster# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE fixmasterwordpress;
Query OK, 1 row affected (0.63 sec)

mysql> CREATE USER 'fixmasterroot'@'localhost' IDENTIFIED BY 'skibiditoilet';
Query OK, 0 rows affected (0.23 sec)

mysql> GRANT ALL ON fixmasterwordpress.* to 'fixmasterroot'@'localhost';
Query OK, 0 rows affected (0.21 sec)

mysql>
```


3.110.12 PHP

Instalamos PHP para poder crear nuestra web dinámica.

```
8 sudo apt install php php-fpm php-common php-mbstring php-xmlrpc php-soap  
php-gd php-xml php-intl php-mysql php-cli php-ldap php-zip php-curl  
Hecho  
sudo apt install libapache2-mod-php  
Hecho  
sudo apt-get install phpmyadmin  
Hecho
```

3.110.13 Virtual Host + Permisos

Creamos la carpeta “wordpress” con los permisos que vemos a continuación:

```
root@servidorgbc:/home/fixmaster# sudo mkdir /var/www/wordpress  
root@servidorgbc:/home/fixmaster# sudo chown -R usuario:www-data /var/www/wordpress  
chown: invalid user: 'usuario:www-data'  
root@servidorgbc:/home/fixmaster# sudo chown -R fixmaster:www-data /var/www/wordpress  
root@servidorgbc:/home/fixmaster# sudo chmod -R 775 /var/www/wordpress  
root@servidorgbc:/home/fixmaster# sudo a2enmod rewrite  
Enabling module rewrite.  
To activate the new configuration, you need to run:  
systemctl restart apache2  
root@servidorgbc:/home/fixmaster# vi /etc/apache2/apache2.conf  
root@servidorgbc:/home/fixmaster# systemctl restart apache2
```

En esta carpeta, es donde irá el Wordpress una vez lo instalemos y es la ruta que pondremos en la configuración de Apache, en el apartado “DocumentRoot”.

Activamos el AllowOverride:

```
<Directory /var/www/>  
Options Indexes FollowSymLinks  
AllowOverride All  
Require all granted  
</Directory>
```

3.110.4 Instalar Wordpress

Desde la web de Wordpress, podremos descargarlo, pero para hacerlo en el servidor, ya que no tenemos interfaz gráfica, deberemos copiar el link y hacer wget del enlace copiado.



```
root@servidorgbc:/home/fixmaster# systemctl restart wpcoms
root@servidorgbc:/home/fixmaster# wget https://es.wordpress.org/latest-es_ES.zip
--2025-04-17 18:58:37-- https://es.wordpress.org/latest-es_ES.zip
Resolving es.wordpress.org (es.wordpress.org)... 198.143.164.252
Connecting to es.wordpress.org (es.wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 35926313 (34M) [application/zip]
Saving to: 'latest-es_ES.zip'

latest-es_ES.zip                               100%[=====]
2025-04-17 18:58:40 (17.2 MB/s) - 'latest-es_ES.zip' saved [35926313/35926313]
```

En este momento, tenemos el wordpress instalado, pero hay que descomprimirlo, ya que está en .zip. Para ello, deberemos instalar el paquete “unzip”.

```
root@servidorgbc:/home/fixmaster# unzip latest-es_ES.zip
Command 'unzip' not found, but can be installed with:
apt install unzip
root@servidorgbc:/home/fixmaster# apt install unzip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Paquetes sugeridos:
zip
Se instalarán los siguientes paquetes NUEVOS:
unzip
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 139 no actualizados.
Se necesitan descargar 174 kB de archivos.
```

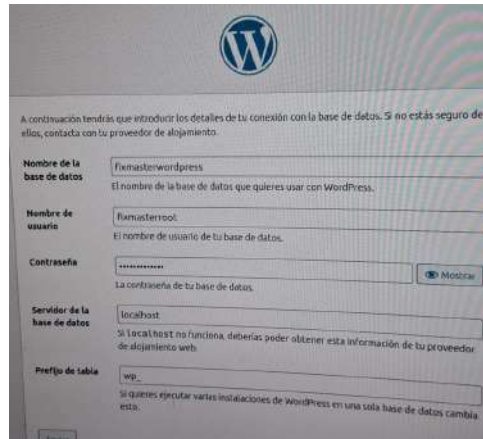
Descomprimos el .zip y movemos la carpeta al /var/www y le otorgamos los permisos (igual que en el virtual host), ya que hay que eliminar la creada previamente para poder poner esta carpeta, ya que así aparece directamente la web.

```
489 unzip latest-es_ES.zip
490 ll
491 mv wordpress/ /var/www/
492 ll /var/www/
493 cd /var/www/
494 chmod 775 wordpress/
495 ll
496 sudo chmod -R 775 /var/www/wordpress
497 ll
498 sudo chown -R fixmaster:www-data /var/www/wordpress
499 ll
```

Una vez descargado Wordpress y colocado en la carpeta correspondiente, al buscar la web “fixmaster.puig.es” o “192.168.19.250” o “10.0.102.43”, nos aparecerá para poder instalar el wordpress



Conectamos nuestro Wordpress con la base de datos que hemos creado previamente con MySQL.



WordPress logo at the top.

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

Nombre de la base de datos: El nombre de la base de datos que quieres usar con WordPress.

Nombre de usuario: El nombre de usuario de tu base de datos.

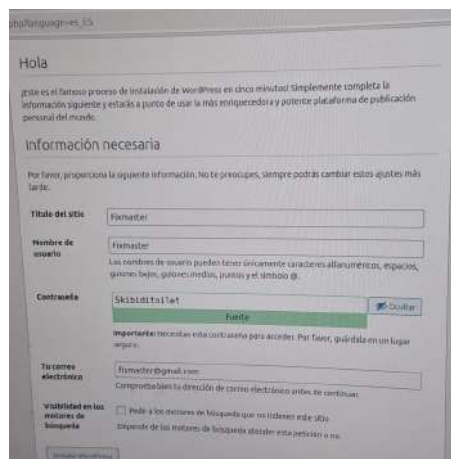
Contraseña: La contraseña de tu base de datos. [Mostrar](#)

Servidor de la base de datos: Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

Prefixo de tabla: Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Instalar](#)

Aquí configuramos la información de nuestro sitio y nuestra cuenta.



WordPress logo at the top.

Hola

grate es el famoso proceso de instalación de WordPress en cinco minutos! simplemente completa la información siguiente y estás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, proporcióna la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio:

Nombre de usuario: Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña: [Fuerte](#) [Ocultar](#) importante: necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Tu correo electrónico: Comprueba bien la dirección de correo electrónico antes de continuar.

Visibilidad en los motores de búsqueda: ☐ **Permitir a los motores de búsqueda que nos indexen este sitio.** Dependiendo de los motores de búsqueda, dárles esta permissão o no.

[Instalar WordPress](#)

Y ya estaría instalado:

































Ya podemos acceder al escritorio de Wordpress y empezar a configurar nuestra página web



3.110.5 Plugins Wordpress

Instalamos varios plugins:

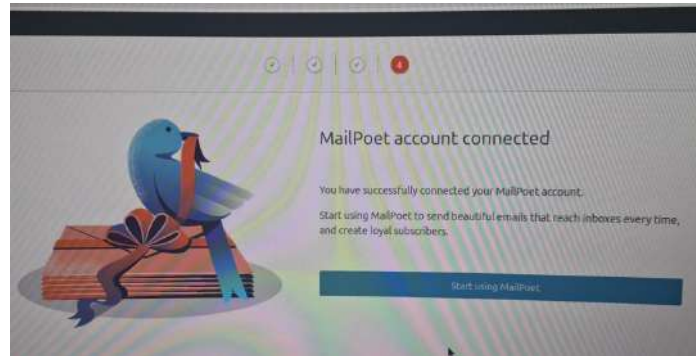
- Advanced Editor Tools: Facilitar el uso del editor de bloques Gutenberg
- Checkout Field Editor: Plugin para la finalización de compra de nuestra tienda (Especializado para WooCommerce)
- Contact Form 7: Para formularios de contacto
- Contact Form CFBD7: Base de datos para las respuestas al formulario de contacto (Sólo Contact Form 7)
- Cookie Notice & Compliance for GDPR/CCPA: Mostrar notificación del uso de cookies
- Facebook: Conecta tu tienda a Facebook, no utilizado por no ser sitio visible para todos (solo con nuestro DNS)
- Google: Conecta tu tienda a Google, no utilizado por no ser sitio visible para todos (solo con nuestro DNS)
- Joinchat: Conectar Wordpress a Whatsapp, permitiendo que puedan contactar con nosotros mediante un numero de telefono
- MailPoet: Enviar correos a usuarios de Wordpress o Tienda
- Pinterest: Conecta tu tienda a Pinterest, no funciona al 100% debido a que no es un sitio visible, pero hay alguna publicación
- Pixel Manager: Recopilar datos de los visitantes de tu tienda, no utilizado por no ser sitio visible para todos (solo con nuestro DNS)
- Polylang: Permitir tener tu página web en diferentes idiomas
- Site Kit: Para ofrecer una mejor experiencia al usuario
- Tiktok: Conecta tu tienda a Tiktok, no utilizado por no ser sitio visible para todos (solo con nuestro DNS)
- UnderConstruction: Para deshabilitar la web y que aparezca una página de “en mantenimiento”
- WooCommerce: Realizar una tienda
- WooCommerce Payments: Métodos de pago en la tienda
- WooCommerce Stripe Gateway: Método de pago en la tienda (Tarjeta de crédito con Stripe)
- WooCommerce Tax: Cálculo de Impuestos automatizados
- WooPayments: Otro método de pagar con tarjeta de crédito, en este caso, con Wordpress
- Wordfence Security: Antivirus, Cortafuegos y exploración de malware, ofreciendo seguridad a nuestra página web
- YITH WooCommerce Wishlist: Permite agregar los productos de la tienda a una “wishlist” o lista de deseos
- Yoast SEO: Es una herramienta para analizar las páginas / entradas de Wordpress y te otorga un feedback

	Advanced Editor Tools Ayudas / Desactivar	Amplía y mejora el editor de bloques (Gutenberg) y el editor clásico (TinyMCE). Versión 3.9.2 Por Automattic Ver detalles
	Checkout Field Editor for WooCommerce Ayudas / Desactivar / Obtener Pro	Personaliza los campos de pago de WooCommerce (añadir, editar, borrar y reorganizar los campos). Versión 2.1.4 Por Themehigh Ver detalles Únete a la comunidad (999/mes)
	Contact Form 7 Ayudas / Desactivar	Simplifica otros plugins de formularios de contacto. Servicio para flexible. Versión 6.0.6 Por Takayuki Miyoshi Ver detalles
	Contact Form CFDB7 Ayudas / Desactivar	Cuenta y optimiza los mensajes de Contact Form 7. Nunca pierdas los datos importantes. El plugin Contact Form CFDB7 es una extensión para el plugin Contact Form 7. Versión 1.3.0 Por Achil Ver detalles
	Cookie Notice & Compliance for GDPR / CCPA Ayudas / Desactivar / Actualización gratuita	Cookie Notice te permite informar automáticamente a los usuarios de que tu sitio usa cookies y te ayuda a cumplir con el RGPD, la CCPA y otras leyes de privacidad de datos. Versión 2.5.0 Por the mantas Ver detalles
	Facebook for WooCommerce Configurar / Documentación / Soporte / Valorar / Desactivar	¡Has creado tu negocio en Facebook! Usa este plugin oficial para ayudar a vender más productos usando Facebook. Después de completar la configuración, estarás listo para crear anuncios que promocionen tus productos y también puedes crear una sección de tienda en tu página donde los clientes pueden explorar tus productos en Facebook. Versión 3.4.7 Por Facebook Ver detalles Requiere: WooCommerce
 Hay disponible una nueva versión de Facebook for WooCommerce. Ver los detalles de la versión 3.4.10 o actualízalo ahora		
	Google for WooCommerce Integración previa / Documentación / Desactivar	Integración nativa con Google que permite a los vendedores mostrar fácilmente sus productos a través de la red de Google. Versión 2.9.12 Por WooCommerce Ver detalles Requiere: WooCommerce
 Hay disponible una nueva versión de Google for WooCommerce. Ver los detalles de la versión 2.9.12 o actualízalo ahora		
	Joinchat Ayudas / Desactivar / Premium	Conecta un chat de WordPress con WhatsApp. La mejor solución para marketing y soporte. Deja de perder clientes y aumenta tus ventas. Versión 5.2.4 Por Chrome Ver detalles Documentación / Soporte
		
	MailPoet Ayudas / Desactivar	Crea y envía boletines, notificaciones de publicaciones y correos electrónicos de bienvenida desde tu WordPress. Versión 3.11.0 Por MailPoet Ver detalles
 Hay disponible una nueva versión de MailPoet. Ver los detalles de la versión 3.11.0 o actualízalo ahora		
	Pinterest for WooCommerce Desactivar	Has creado tu negocio en Pinterest. Usa este complemento oficial para que los compradores puedan añadir Pins de productos mientras navegan por tu tienda, hacer un seguimiento de las conversiones y anunciarse en Pinterest. Versión 1.4.17 Por WooCommerce Ver detalles Requiere: WooCommerce
 Hay disponible una nueva versión de Pinterest for WooCommerce. Ver los detalles de la versión 1.4.18 o actualízalo ahora		
	Pixel Manager for WooCommerce Instalar / Desactivar / Settings	Seguimiento del valor de los visitantes y de la conversión para WooCommerce. Profundamente optimizado para tener precisión en los datos. Versión 1.47.0 Por SmartCode Ver detalles
 Hay disponible una nueva versión de Pixel Manager for WooCommerce. Ver los detalles de la versión 1.48.0 o actualízalo ahora		
	PolyLang Ayudas / Desactivar	Añade capacidad multilingüe a WordPress. Versión 3.7.1 Por WP PolyLang Ver detalles
 Hay disponible una nueva versión de PolyLang. Ver los detalles de la versión 3.7.2 o actualízalo ahora		
	Site Kit by Google Desactivar	Site Kit es una solución integral para que los usuarios de WordPress obtengan todo lo que Google tiene que ofrecer para que tengan éxito en la web. Versión 1.151.0 Por Google Ver detalles Valorar Site Kit / Soporte
	TikTok Desactivar	Con la integración de TikTok a WooCommerce, es más fácil que nunca distinguir innovadoras Funciones de comercio social para impulsar el tráfico y las ventas a una comunidad altamente comprometida. Con una configuración guiada y sencilla, puedes aumentar tu catálogo de productos WooCommerce y promocionarlo con anuncios personalizados sin salir de tu panel de control. Además, en todo el mundo y aumenta las ventas de comercio electrónico extendiendo a través de una de las aplicaciones más descargadas del mundo. Versión 1.3.5 Por TikTok Ver detalles
	Under Construction Ayudas / Irreversible a la versión PRO / Desactivar	Por tu sitio desde una genial página de aterrizaje, modo de mantenimiento, en construcción, o disponible pronto. Versión 6.02 Por Underlaying Ver detalles / Soporte / Obtener la versión PRO
	WooCommerce Ayudas / Desactivar	Una herramienta de comercio electrónico que te ayuda a vender cualquier cosa. Con elegancia. Versión 9.8.2 Por Automattic Ver detalles / Documentación / Documentación de la API / Soporte / Módulo de la comunidad Requiere: por: Facebook for WooCommerce, Google for WooCommerce, Pinterest for WooCommerce, WooCommerce PayPal Payments, WooCommerce Stripe Gateway, WooCommerce Tax, WooCommerce Twitter WooCommerce Wishlist. Nota: Este plugin no se puede desactivar o eliminar hasta que se desactiven o eliminen los plugins que lo requieren.
 Hay disponible una nueva versión de WooCommerce. Ver los detalles de la versión 9.8.2 o actualízalo ahora		
	WooCommerce PayPal Payments Ayudas / Desactivar	La última solución completa de procesamiento de pagos de PayPal. Acepta PayPal, eChecks más tarjetas, tarjetas de crédito/débito, tarjetas digitales alternativas, tipos de pago locales y cuentas bancarias. Activa todas las opciones de PayPal y promueve una sencilla experiencia de checkout de pago. Activa las transacciones guiadas con una amplia selección de diseños y temas. Versión 3.0.1 Por PayPal Ver detalles Documentación / Obtén ayuda / Solicita una característica / Ayudar de un fallo Requiere: WooCommerce
	WooCommerce Stripe Gateway Ayudas / Desactivar	Acepta pagos con tarjeta de crédito en tu tienda usando Stripe. Versión 9.4.1 Por Stripe Ver detalles / Documentación / Soporte Requiere: WooCommerce
 Hay disponible una nueva versión de WooCommerce Stripe Gateway. Ver los detalles de la versión 9.5.0 o actualízalo ahora		
	WooCommerce Tax Desactivar	Cálculo de impuestos automatizado para WooCommerce. Versión 3.0.0 Por WooCommerce Ver detalles Requiere: WooCommerce
		
	WooPayments Configuración / Desactivar	Acepta pagos con tarjeta de crédito. Gestiona las transacciones en WordPress. Versión 9.2.1 Por WooCommerce Ver detalles Requiere: WooCommerce

A continuación, mostraremos la instalación, configuración y la utilidad de algunos de los plugins más destacados.

3.110.51 MailPoet

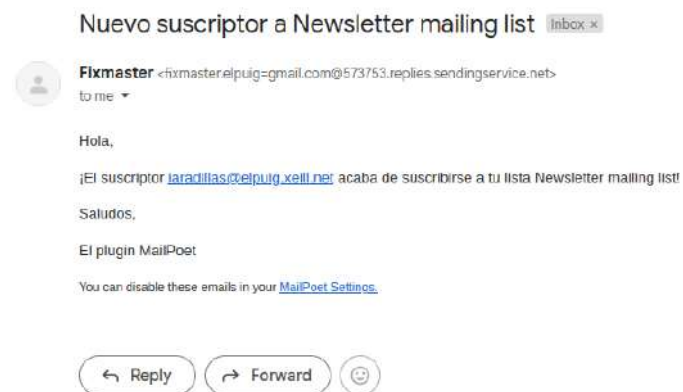
Conectamos la cuenta a MailPoet

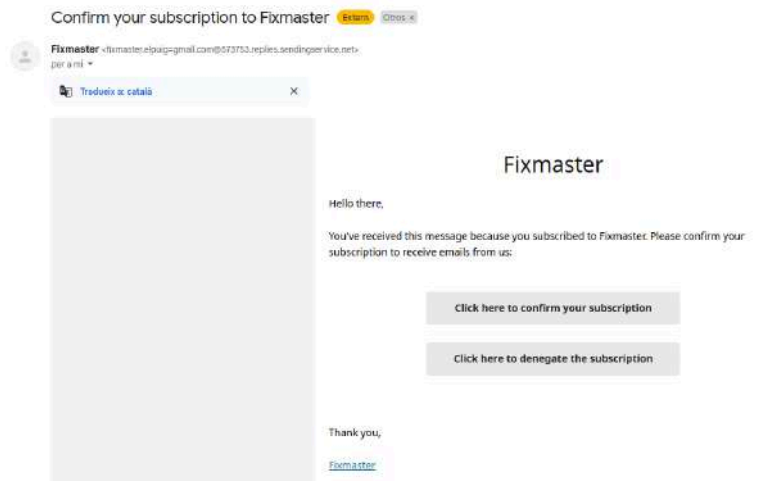


Creamos nuestro formulario:



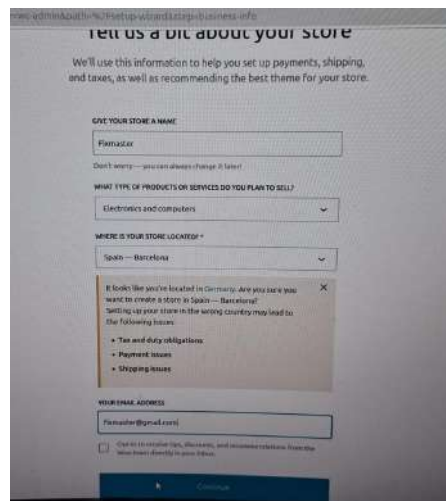
Comprobación de que funciona correctamente:



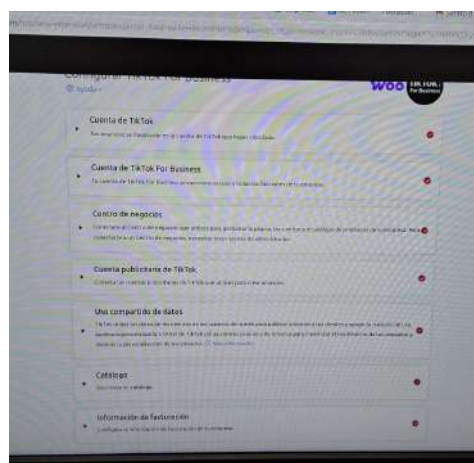


3.110.52 WooCommerce

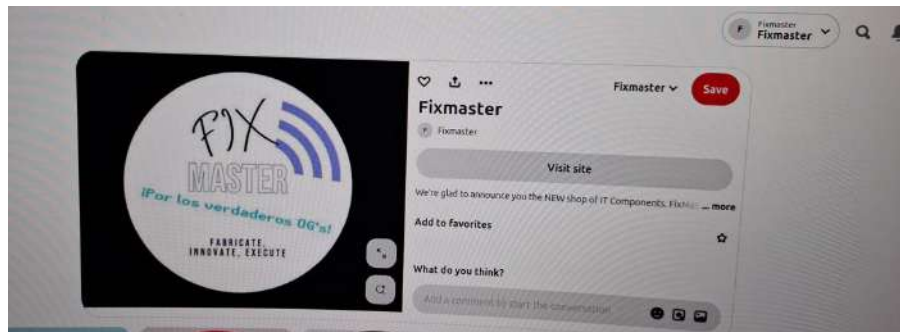
Información sobre nuestra tienda



Intentamos conectar nuestro WooCommerce a nuestra cuenta de Tik Tok, pero no se puede ya que nuestra tienda no es real y no es un sitio público.



También lo intentamos con Pinterest, y creamos nuestro primer anuncio:



Ajustes generales de la tienda:

Ajustes

Opciones generales

Ubicación(es) de venta:

Ubicación(es) de envío:

Ubicación del cliente por defecto:

Activar impuestos: ☒ Activar tasas de impuestos y sus cálculos.
Las tasas ser podrán configurar, y los impuestos se calcularán durante el pago.

Activar cupones: ☒ Activar el uso de códigos de cupones.
Pueden aplicarse cupones desde las páginas de carrito y de finalizar compra.
☐ Calcular descuentos de cupones secuencialmente.
Al aplicar múltiples cupones, se aplica el primer cupón al precio completo y el segundo cupón para el precio ya rebajado y así sucesivamente.

Opciones de moneda
Las siguientes opciones afectarán a cómo se muestran los precios en la tienda.

Moneda:

Ubicación de la moneda:







Separador de miles:

3.110.52.1 Categorías, Marcas y Productos

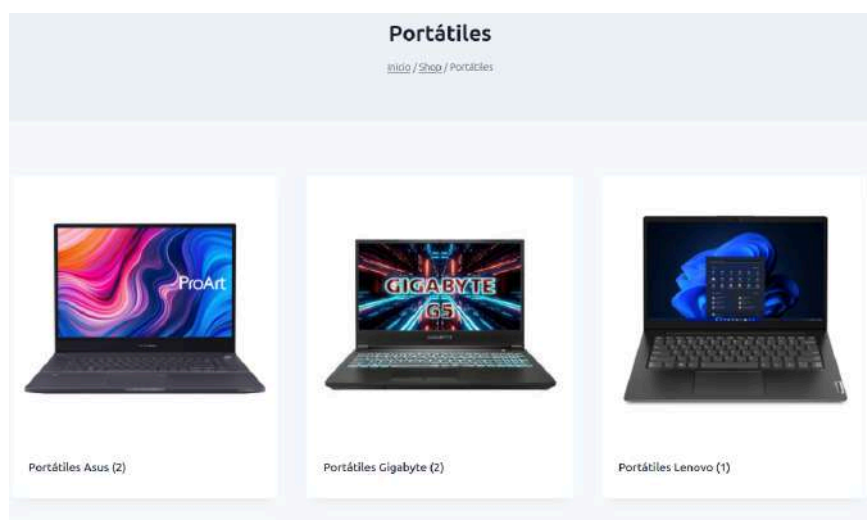
Categorías

Hemos seguido una estructura clara para todas las categorías de nuestra tienda:

- Categoría padre: Producto general
- SubCategorías: Producto con la marca
- Slug (Cómo será el enlace a la categoría): producto / producto-marca












Acciones en lote ▾		Aplicar		83 elementos		1 2 de 5	
<input type="checkbox"/>	Imagen	Nombre	Descripción	<input type="checkbox"/>	Slug	Cantidad	
<input type="checkbox"/>		Fuentes de Alimentación	—	<input type="checkbox"/>	Fuentes-de-alimentacion	0	≡
<input type="checkbox"/>		Fuentes de Alimentación MSI	—	<input type="checkbox"/>	Fuentes-de-alimentacion-msi	0	≡
<input type="checkbox"/>		Impresoras	—	<input type="checkbox"/>	impresoras	0	≡
<input type="checkbox"/>		Impresoras Brother	—	<input type="checkbox"/>	impresoras-brother	0	≡
<input type="checkbox"/>		Impresoras HP	—	<input type="checkbox"/>	impresoras-hp	0	≡
<input type="checkbox"/>		Memorias RAM	—	<input type="checkbox"/>	memorias-ram	0	≡

Si vamos a la categoría padre se veria algo así:



Marcas

















Creamos una sección en el apartado “marcas” para cada marca que nos vende sus productos para poder venderlos en nuestra tienda.

Acciones en lote ▾		Aplicar		33 elementos		1 de 2	
<input type="checkbox"/>	Imagen	Nombre	Descripción	<input type="checkbox"/>	Slug	Cantidad	
<input type="checkbox"/>		Acer	—	<input type="checkbox"/>	acer	0	≡
<input type="checkbox"/>		AMD	—	<input type="checkbox"/>	amd	0	≡
<input type="checkbox"/>		Antec	—	<input type="checkbox"/>	antec	0	≡
<input type="checkbox"/>		Apple	—	<input type="checkbox"/>	apple	0	≡
<input type="checkbox"/>		ASRock	—	<input type="checkbox"/>	asrock	0	≡
<input type="checkbox"/>		Asus	—	<input type="checkbox"/>	asus	2	≡
<input type="checkbox"/>		BenQ	—	<input type="checkbox"/>	benq	0	≡
<input type="checkbox"/>		Brother	—	<input type="checkbox"/>	brother	0	≡
<input type="checkbox"/>		Cooler Master	—	<input type="checkbox"/>	cooler-master	0	≡
<input type="checkbox"/>		Corsair	—	<input type="checkbox"/>	corsair	0	≡
<input type="checkbox"/>		Crucial	—	<input type="checkbox"/>	crucial	0	≡

Esto es útil para que el usuario o cliente pueda buscar directamente todos los productos de una marca en específico (ya sean portátiles o fuentes de alimentación o placas bases)

Productos

En cuanto a productos, según las categorías y las marcas previamente comentadas, es decir, tenemos productos de todos nuestros proveedores. Estos son algunos productos:

Acciones en lote		Aplicar	Todas las puntuaciones 5B	Todas las puntuaciones de	Selecciona una categoría	Filtrar por tipo de producto	Filtrar por el estado del inv	Filtrar por marca	Filtrar
<input type="checkbox"/>	Nombre	SKU	Inventario	Precio	Categorías	Etiquetas	★	Fecha	Marcas
<input type="checkbox"/>	 Portátil Lenovo Y15 G3 IAP 82T06KBP	PORLTH1	Se puede reservar (0)	449,99 € 399,99 €	Portátiles, Portátiles Lenovo	-	☆	Publicada 27/05/2025 a las 13:46	 Lenovo
<input type="checkbox"/>	 Portátil MSI Thin A15 B7VF-070KES	PORMSI3	Hay existencias (10)	1.299,99 €	Portátiles, Portátiles MSI	-	☆	Publicada 27/05/2025 a las 13:34	 MSI
<input type="checkbox"/>	 Portátil MSI Creator Pro 16 AI Studio A1VJG-208ES	PORMSI2	Hay existencias (3)	3.399,99 €	Portátiles, Portátiles MSI	-	☆	Publicada 27/05/2025 a las 13:25	 MSI
<input type="checkbox"/>	 Portátil MSI Modern 14-402XES	PORMSI1	Hay existencias (100)	485,99 €	Portátiles, Portátiles MSI	-	☆	Publicada 27/05/2025 a las 13:15	 MSI
<input type="checkbox"/>	 Portátil Gigabyte G5 MF5-52ES34SD	PORGIC2	Hay existencias (20)	999,99 €	Portátiles, Portátiles Gigabyte	-	☆	Publicada 27/05/2025 a las 13:01	 Gigabyte
<input type="checkbox"/>	 Portátil Asus ExpertBook B15	PORASU2	Hay existencias (15)	749,99 € 599,99 €	Portátiles, Portátiles Asus	-	☆	Publicada 27/05/2025 a las 12:10	 Asus
<input type="checkbox"/>	 Portátil Asus ExpertBook B1 B1502CVA-BQ083W	PORASU1	Hay existencias (30)	689,99 € 629,99 €	Portátiles, Portátiles Asus	-	☆	Publicada 27/05/2025 a las 10:51	 Asus
<input type="checkbox"/>	 Portátil GIGABYTE AORUS 16K AG-53ESCS45H	PORGIC1	Hay existencias (10)	2.999,99 €	Portátiles, Portátiles Gigabyte	-	★	Publicada 27/05/2025 a las 09:11	 Gigabyte

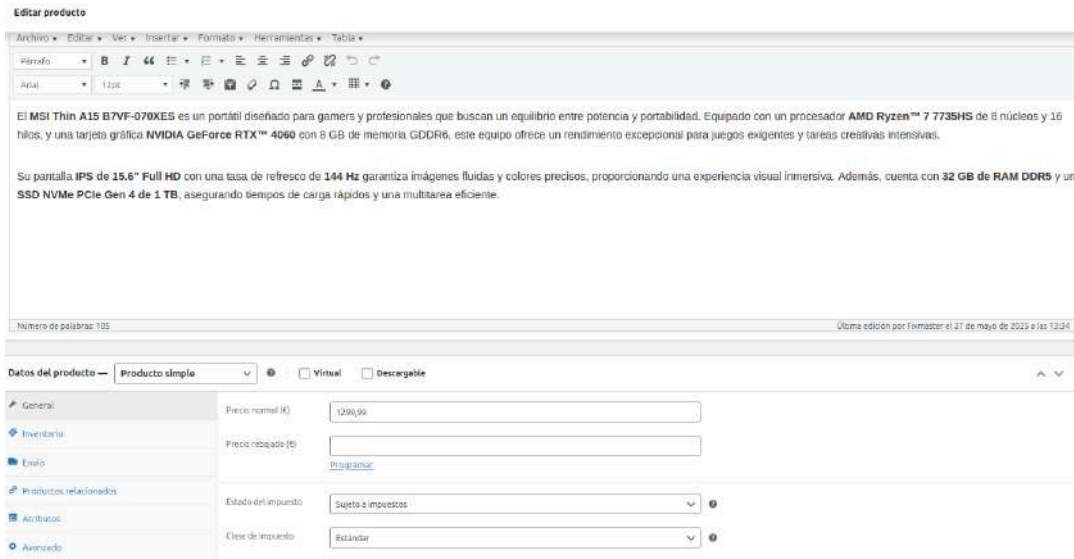
Estos productos contienen varias características:

- Imagen: Imágenes del producto
- Título: Nombre del producto
- SKU: Nombre que define al producto. Ejemplo:
 - PORMSI3. POR= Producto (Portátil) MSI= Marca (MSI) 3= Número cualquiera (en este caso, es el 3º portátil msi que introducimos en la tienda)
- Stock: Cantidad de existencias de un producto
- Precio: Precio del producto
- Categorías: Categoría que representa a ese producto
- Etiquetas: Otras maneras de encontrar a este producto (En nuestro caso, no hemos puesto)
- Destacado: Puedes destacar un producto para que aparezca en tu tienda principal
- Fecha de publicación: Cuando se ha publicado o modificado por última vez
- Marca: Marca oficial del producto

<input type="checkbox"/>	Portátil MSI Thin A15 B7VF-070KES	PORMSI3	Hay existencias (10)	1.299,99 €	Portátiles, Portátiles MSI	—	Publicada 27/05/2025 a las 13:34	MSI
--------------------------	-----------------------------------	---------	----------------------	------------	----------------------------	---	----------------------------------	-----

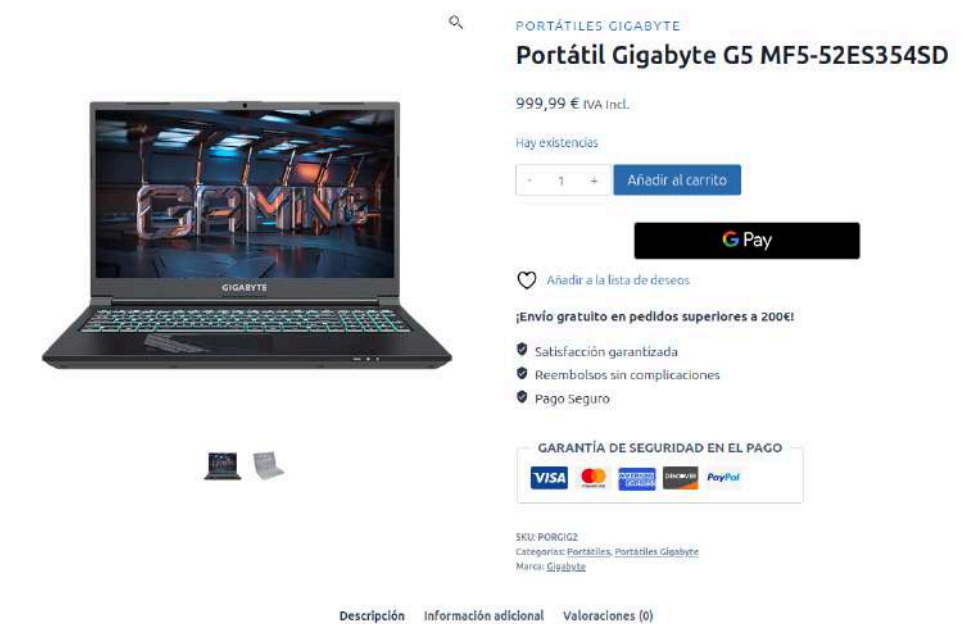
Si nos introducimos en la edición o creación de un producto, podremos observar otro tipo de características y/o ajustes:

- Descripción del producto
- Tipo de producto (Simple, variado, agrupado...)
- Ajustes generales (Precios e Impuestos)
- Ajustes de Inventario (SKU y control del stock)
- Ajustes de envío (Tamaño y peso, y añadirle la clase de envío)
- Productos relacionados (Añadir productos parecidos o recomendados)
- Atributos (Etiquetas)
- Avanzado (Nota de compra y valoraciones)



Si nos introducimos en la tienda y vamos a un producto, podemos observar la siguiente interfaz:

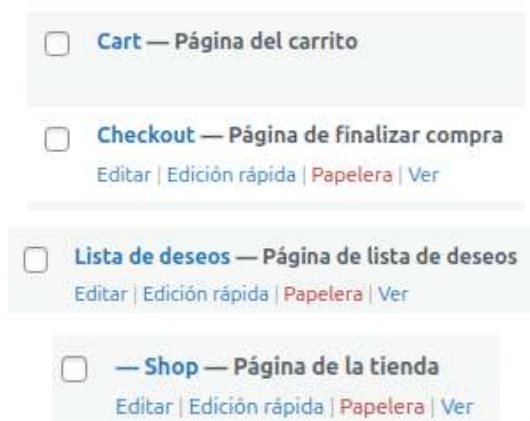
- Categoría del producto
- Título
- Precio
- Si hay existencias (En caso de ser 10 o menos, indica la cantidad)
- Añadir al carrito
- Google Pay
- Añadir a la Wishlist (gracias al plugin YITH WooCommerce Wishlist)
- Información para atraer clientes
- Información del producto (SKU, Categorías y Marcas)
- Descripción / Información adicional / Valoraciones



Descripción

3.110.52.2 Páginas Tienda

- Cart: Página donde se almacenan los productos que añades al carrito. Desde aquí puedes ir a la página de checkout para finalizar la compra o pagar con google pay. También puedes añadir cupones y añadir o eliminar productos.
- Checkout: Página donde se finaliza la compra o pagar con google pay, solicitando recogida en tienda o indicando el lugar donde se enviará el producto y la facturación. También puedes añadir cupones.
- Lista de deseos: Página donde se almacenan los productos que añades a la wishlist.
- Shop: Página principal de la tienda.



3.110.52.3 Configuraciones

Tarifas (IVA)

En una tienda es fundamental tener configurados las tarifas / tasas de impuestos. Nosotros hemos configurado los siguientes tipos de tarifas:



Para dividir las provincias, llevamos a cabo la siguiente tabla (cambiando el 21% respectivamente al que corresponda):

Opciones (de impuestos) Tarifas estándar Tarifas IVA Reducido (10%) Tarifas IVA Superreducido (4%) Tarifas Tasa Cero									
Tasas de impuestos «Estándar»									
Código de país	Código de provincia	Código postal / ZIP	Ciudad	Tarifa %	Nombre del impuesto	Prioridad	Compuesto	Envío	
ES	CE	*	*	0.0000	Exento de IVA	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ES	CC	*	*	0.0000	Exento de IVA	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ES	ML	*	*	0.0000	Exento de IVA	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ES	TF	*	*	0.0000	Exento de IVA	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ES	*	*	*	21.0000	21% IVA (Estándar)	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Insertar fila		Eliminar fila(s) seleccionada(s)					Importar CSV		Exportar en formato CSV

Aquí podemos comprobar que distribuimos las tarifas, dejando exentos de IVA a las Islas y a toda la península española, le añadimos el porcentaje que le toque respectivamente de la clase de impuesto.

Costes de envíos / Recogida en tienda

Para los envíos, hemos configurado que:

- Envíos a la península sean envíos gratuitos en caso de compras superiores a 200, en caso contrario, son 2 euros
- Envíos a islas españolas con un coste fijo de 5 euros
- Envíos a países extranjeros con un coste fijo de 7 euros
- En caso de que el paquete sea frágil o pesado, serán 10 euros fijos, sin influir el lugar de envío.
- La recogida en tienda será una opción en la que no se cobrarán gastos de envíos ni paquetes frágiles o pesados

The image displays three screenshots of a web application's shipping configuration interface, arranged in a grid. Each screenshot shows a modal window for configuring shipping options.

- Top Left: Configurar envío gratuito**
 - Nombre:** Envío gratuito
 - El envío gratuito requiere:** Una cantidad mínima de pedido (0 o un cupón)
 - Cantidad mínima de pedido:** 200,00 €
 - Cupones de descuento:** ☐ Aplicar la regla de pedido mínimo antes del descuento del cupón
 - Guardar** button
- Top Right: Configurar precio fijo**
 - Nombre:** Precio fijo
 - Estado del impuesto:** Sujeto a impuestos
 - Coste:** 2,00 €
 - Costes de la clase de envío:**
 - Coste de la clase de envío "Paquetes Frágiles / Pesados": 10,00 €
 - Coste sin clase de envío: N/D
 - Tipo de cálculo:**
 - Guardar** button
- Bottom Left: Configurar precio fijo**
 - Nombre:** Precio fijo
 - Estado del impuesto:** Sujeto a impuestos
 - Coste:** 5,00 €
 - Costes de la clase de envío:**
 - Coste de la clase de envío "Paquetes Frágiles / Pesados": 10,00 €
 - Coste sin clase de envío: N/D
 - Tipo de cálculo:**
 - Guardar** button
- Bottom Right: Configurar precio fijo**
 - Nombre:** Precio fijo
 - Estado del impuesto:** Sujeto a impuestos
 - Coste:** 7,00 €
 - Costes de la clase de envío:**
 - Coste de la clase de envío "Paquetes Frágiles / Pesados": 10,00 €
 - Coste sin clase de envío: N/D
 - Tipo de cálculo:**
 - Guardar** button

Recogida local

Generales

Activa o desactiva la recogida local en tu tienda, y define los costes. La recogida local solo está disponible en el pago mediante bloques.

[Ver página de finalizar compra](#)

Ubicaciones de recogida

Define ubicación de recogida entre las que puedan elegir tus clientes al finalizar la compra.

[Aprender más](#)

Zonas de envío

Activar zona

Una zona de envío consiste en la región o regiones a las que te gustaría enviar y el método o métodos de envío ofrecidos. Un comprador solo puede corresponder a una zona, y utilizaremos su dirección de envío para mostrarle los métodos disponibles en su zona. Para ofrecer recogida local, configura las ubicaciones de recogida en los [planes de recogida local](#).

Activar la recogida local

Cuando está activo, la recogida local aparecerá como opción en el pago basado en bloques.

Título

Recogida

Este es el nombre del método de envío mostrado a los clientes.

☐ Añade un precio para los clientes que elijan la recogida local

Por defecto, el método de envío de recogida local es gratuito.

Ubicación de recogida

Tienda Física FDXMaster av. Anselm del Riu 10, Santa Coloma de Gramenet, Barcelona, 08924, España

Activar

Editar

Añadir ubicación de recogida

Nombre de la zona	Región(es)	Método(s) de envío
Península	Zaragoza, Zamora, Vicuña, Valladolid, Valencia, Toledo, Teruel, Tarragona, Soria, Sevilla y 37 otras regiones	<div>Envío gratuito</div> <div>Precio fijo</div> <div>Editar Borrar</div>
Islas Españolas	Baleares, Las Palmas, Melilla, Santa Cruz de Tenerife, Ceuta	<div>Precio fijo</div> <div>Editar Borrar</div>
Resto del mundo	Una zona adicional que puedes usar para configurar método(s) de envío disponible(s) por cualquier región que no se haya listado arriba.	<div>Precio fijo</div> <div>Editar</div>

Cupones descuento

Hemos configurado algunos cupones descuento, especializados en nuestro instituto Puig Castellar. Estos consisten en que, al poner “ElPuig”, se te aplicará un 20% de descuento, pero si además, vienes del aula “Ada”, se te aplicará un 10% extra de descuento combinable con otros cupones como “ElPuig”.

Cupones

Cupones [Añadir nuevo cupón](#) Opciones de personalización

PayPal Payments está casi listo. Para empezar, conecta tu cuenta con el botón **Activa PayPal** en la [página de configuración de la tienda](#).


Action Scheduler: 37 [past due actions](#). Fixing something may be serious. [Read documentation](#).

Todas (2) 1 Publicadas (2)

Acciones en lista
Aplicar
Mostrar todos los tipos
Filtrar

<input type="checkbox"/>	Código	Tipo de cupón	Importe del cupón	Descripción	IDs de Producto	Uso / Límite	Fecha de caducidad
<input type="checkbox"/>	efvgig	Descuento en porcentaje	20	Si vienes de parte del Instituto Pung Costillar, tienes un 20% de descuento	—	0 / —	Junio 30, 2015
<input type="checkbox"/>	ada	Descuento en porcentaje	10	Este descuento es aplicable al 10% del Pung, pero antes es especial para la clase Acta (10%)	—	0 / —	Junio 30, 2015
<input type="checkbox"/>	Código	Tipo de cupón	Importe del cupón	Descripción	IDs de Producto	Uso / Límite	Fecha de caducidad

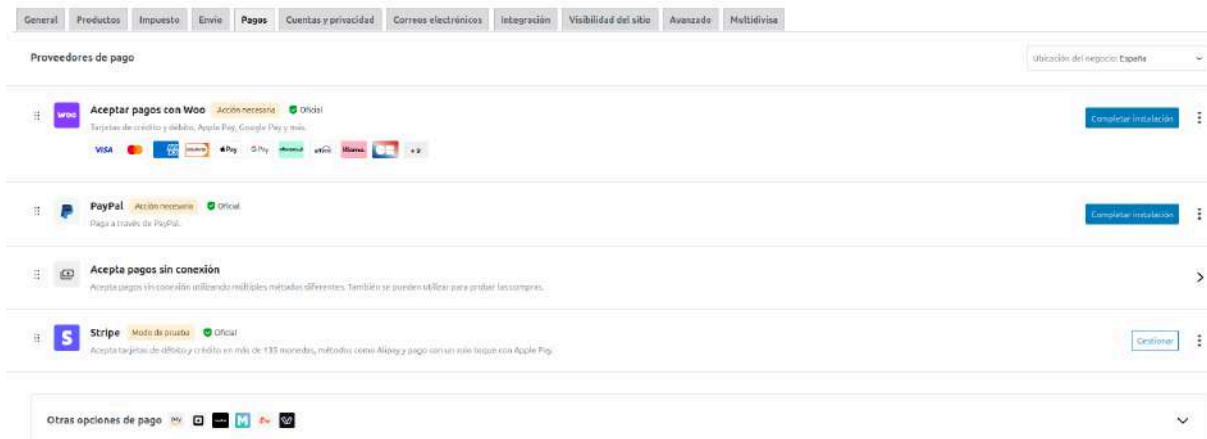
Comprobación de que funciona:

Añade un cupón 	
Subtotal	999,99 €
Descuento	-300,00 €

Métodos de pago

Los métodos de pago:

- Aceptar pagos con Woo (No deja completar la instalación debido a que no es una pagina publica y no se puede acceder desde Internet si no tienes nuestro servidor como DNS)
- PayPal (No hemos completado la instalación porque no tenemos pensado introducir ninguna cuenta de PayPal nuestra ni vamos a crear una específicamente para esto)
- Aceptar pagos sin conexión está permitido y configurado
- Stripe, también está permitido y configurado, por lo tanto, se tienen estas 2 opciones

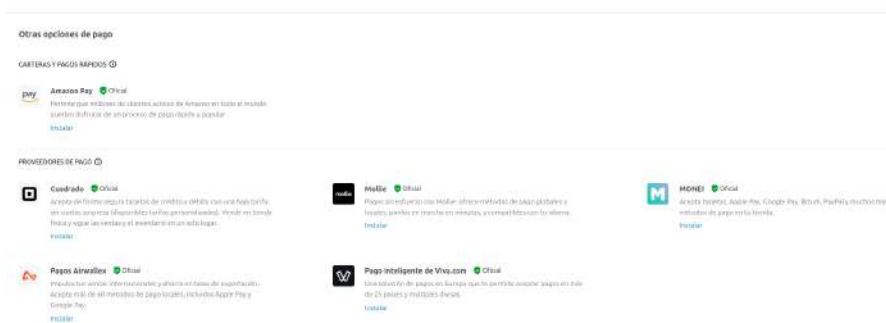


En pagos sin conexión, están estas 3 opciones:

- Transferencia bancaria directa
- Pagos por cheque
- Contra reembolso

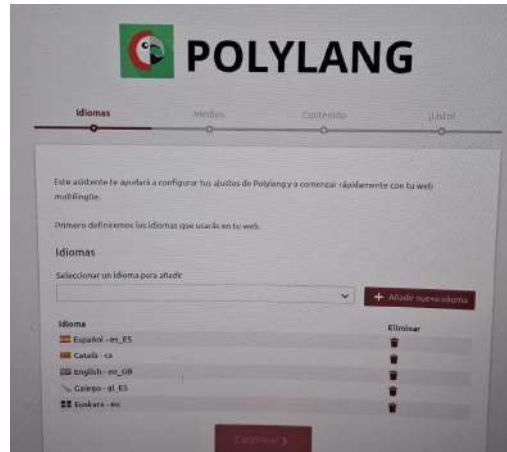


Otras opciones que podemos instalar son:



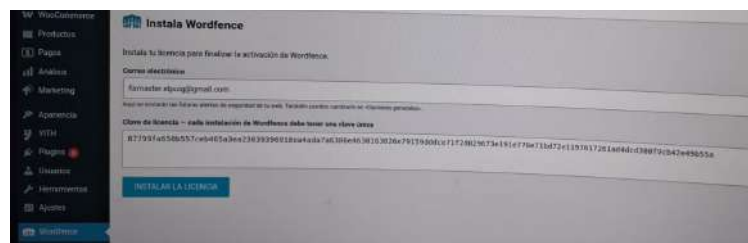
3.110.53 Polylang

También hemos descargado el plugin “Polylang”, para poder tener las entradas en varios idiomas, pero temporalmente, va a estar disponible solamente en español. Aquí podemos ver, a qué lenguas se traducirá próximamente nuestra página web:



3.110.54 WordFence

Este plugin sobre la seguridad que protegerá nuestra página de posibles ataques, amenazas, etc.



3.110.55 UnderConstruction

Este plugin nos ayudará a poder trabajar con calma en solucionar los problemas o hacer las modificaciones sin que la gente pueda acceder a ella. Este plugin será utilizado en casos concretos.



Estos son algunos de los plugins más destacados, en la presentación y demostración, enseñaremos el resto de plugins.

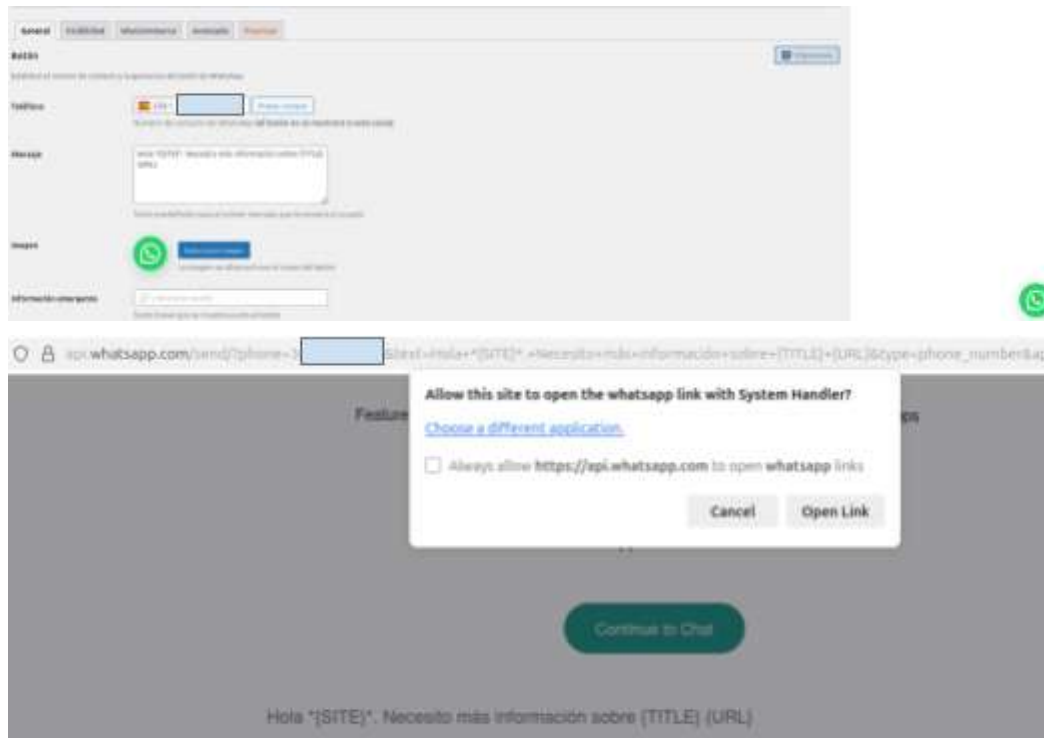
3.110.56 Cookie Notice & Compliance for GDPR / CCPA

Este plugin nos permitirá avisar al usuario del uso de cookies, permitiéndole aceptar o rechazar.



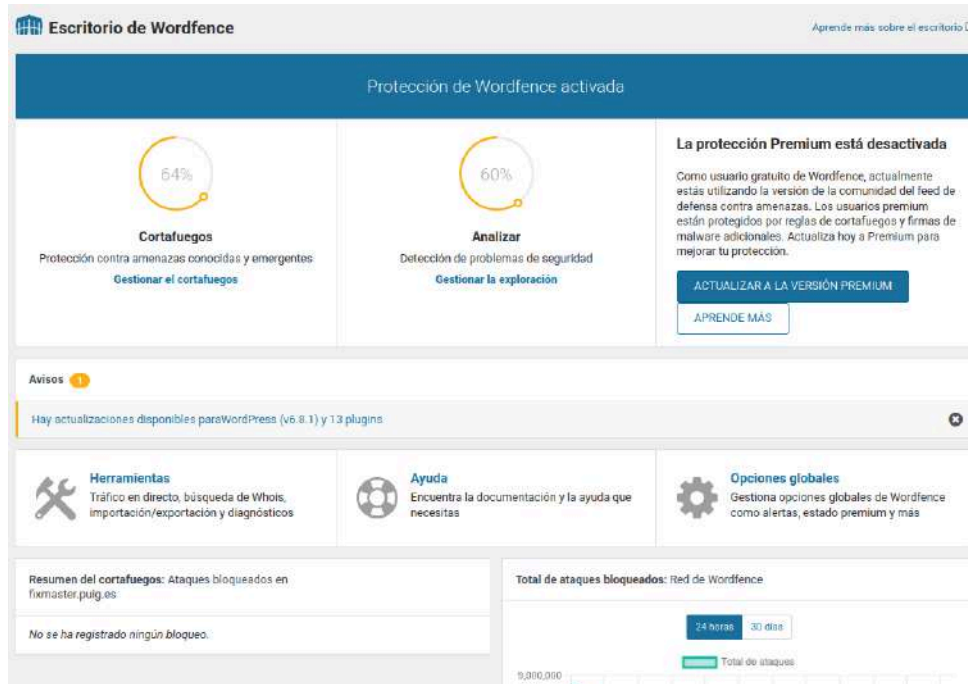
3.110.57 JoinChat

Este plugin permite enviar un mensaje o tener un contacto más directo con nosotros.



3.110.57 WordFence Security

Este plugin nos permite mantener nuestra página web protegida de ataques de malware, ya que funciona como cortafuegos. Aquí podemos ver el escritorio y la información que podemos observar:



3.110.58 Yoast SEO

Este plugin realiza un análisis de las páginas, entradas y productos y nos otorga una puntuación para saber si se puede mejorar la legibilidad u otros aspectos.

Aquí vemos un resumen del escritorio de Yoast SEO:



3.110.59 Contact Form 7

Este plugin es un formulario de contacto el cual permite al usuario comunicarse con nosotros.

The screenshot displays the Contact Form 7 plugin interface. At the top, it shows the title 'Formulario de contacto 1' and a message: 'Copia este shortcode y pégalo en el contenido de una entrada, página o widget de texto:'. Below this, the shortcode is provided: `[contact-form-7 id="87cc040" title="Formulario de contacto 1"]`. The interface includes tabs for 'Formulario', 'Correo', 'Mensajes', and 'Ajustes adicionales'. The 'Formulario' tab is active, showing a message: 'Puedes editar aquí la plantilla del formulario. Para más detalles, consulta [Editar la plantilla del formulario](#).' Below this, there are buttons for 'texto', 'correo electrónico', 'URL', 'teléfono', 'número', 'fecha', 'área de texto', 'menú desplegable', and 'c'. The form template is displayed as follows:

```
<label> Tu nombre
[text* your-name autocomplete:name] </label>

<label> Tu correo electrónico
[email* your-email autocomplete:email] </label>

<label> Asunto
[text* your-subject] </label>

<label> Tu mensaje (opcional)
[textarea your-message] </label>

[submit "Enviar"]
```

Este formulario está introducido en la página de “Contacta con nosotros”

Tu nombre

Tu correo electrónico

Asunto

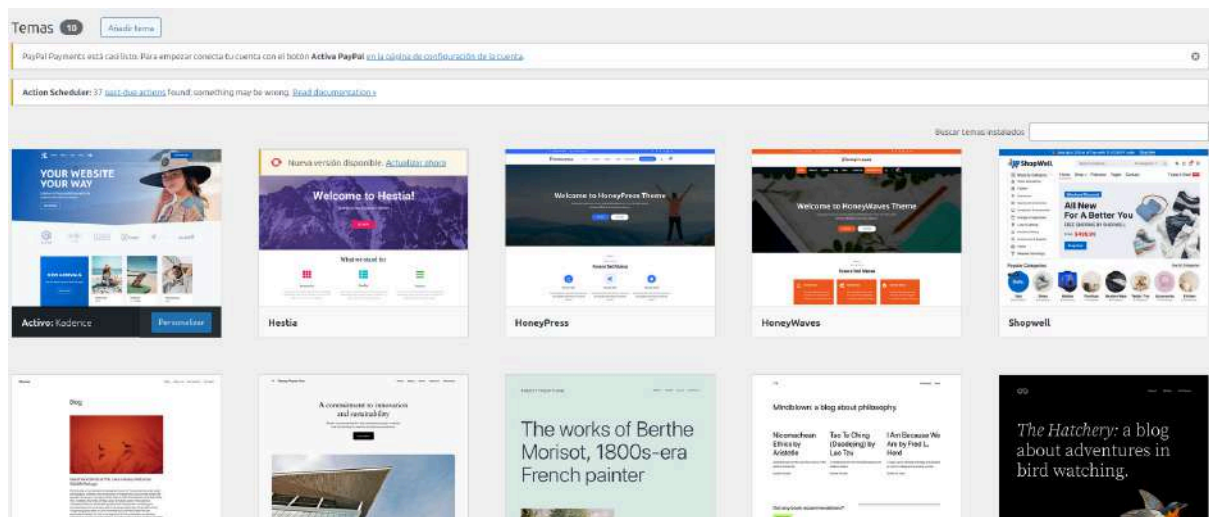
Tu mensaje (opcional)

Al rellenar este formulario, deberá guardarse la información en una base de datos para formularios de contact form 7, esto es con ayuda del plugin Contact Form 7 CFBD7.

3.110.6 Apariencia / Temas y Menus

3.110.61 Apariencia / Temas

Hemos buscado entre los posibles temas a instalar, y hemos acabado escogiendo “Kadence”, un Tema bastante interesante y que te permite realizar una gran cantidad de modificaciones y personalizaciones.

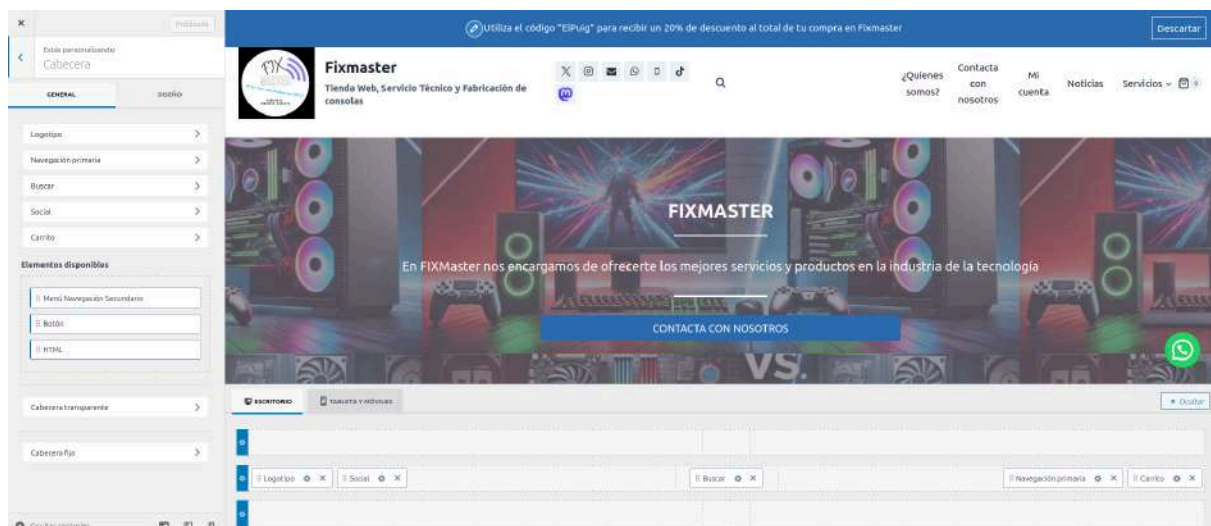


Una vez activamos este tema, podemos personalizar, aquí podemos ver la gran cantidad de posibilidades a modificar:



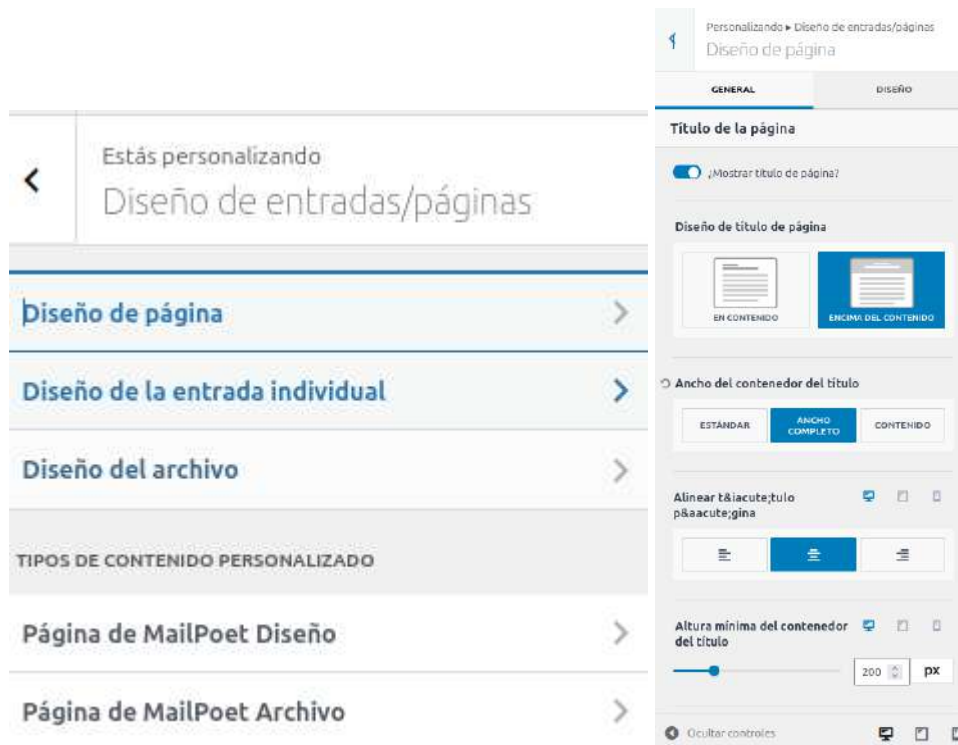
3.110.61.1 Cabecera / Pie de página

La manera de personalizar el pie de cabeza y la cabecera es la misma. Ejemplo de la configuración de la cabecera:



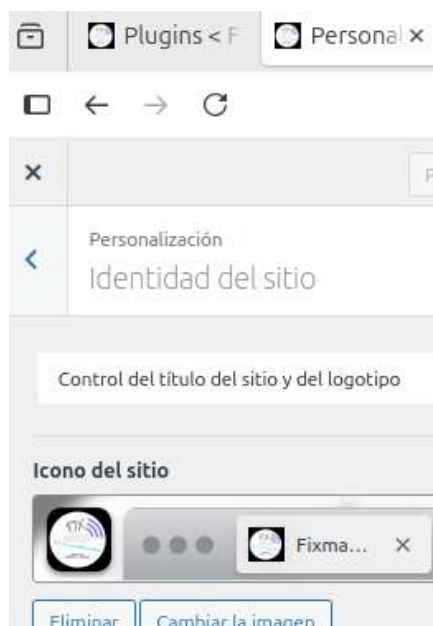
3.110.61.2 Diseño de páginas / entradas

Esta sería la parte de configuración y diseño de las páginas y las entradas (por defecto, ya que cada página / entrada puede ser personalizada individualmente).



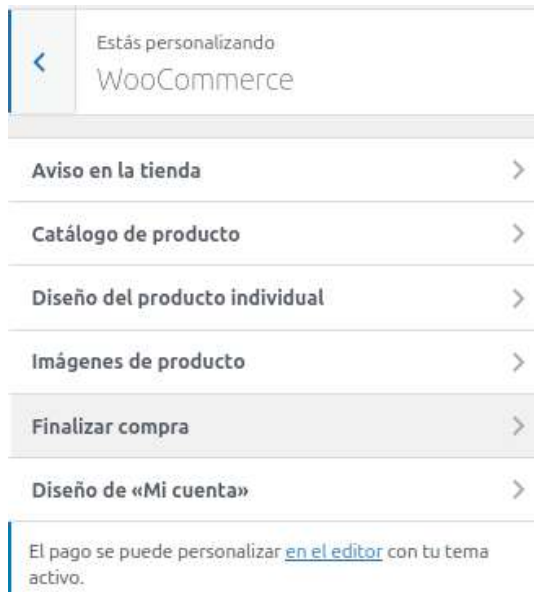
3.110.61.3 Identidad de la web

Aquí podemos personalizar como se verá la pestaña en el buscador. Nosotros hemos puesto la imagen de nuestro logo y el nombre de la página que visita.



3.110.61.4 WooCommerce

Aquí se puede configurar el diseño de la tienda. Hemos realizado varios cambios que se verán en la presentación y demostración final.



3.110.62 Menús

Los menús se pueden modificar y configurar desde 2 sitios diferentes:

3.110.62.1 Personalización del tema Kadence

Los menús se pueden configurar y personalizar desde el tema "Kadence"



3.110.62.2 Ajustes de apariencia

Desde los ajustes de configuración de apariencia, en el apartado “Menús”

3.110.62.3 Ubicaciones de los menús

Menú Principal

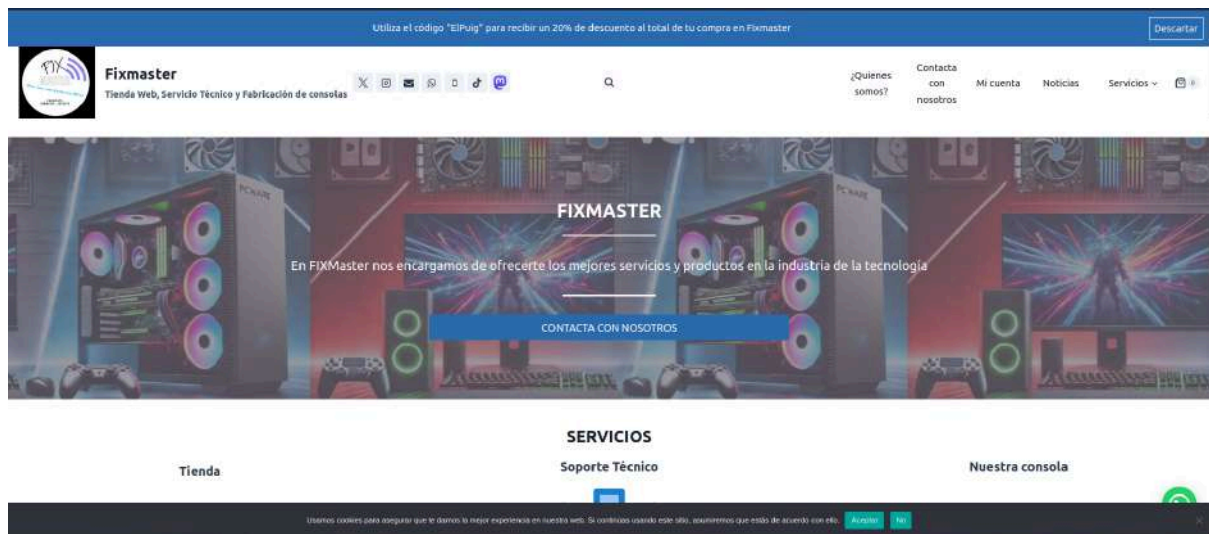
¿Quiénes somos? Contacta con nosotros Mi cuenta Noticias Servicios ▾

Menú Secundario

Aviso legal Política de privacidad
Política de Cookies Contacta con nosotros
FAQ

3.110.63 Resultado Final Portada

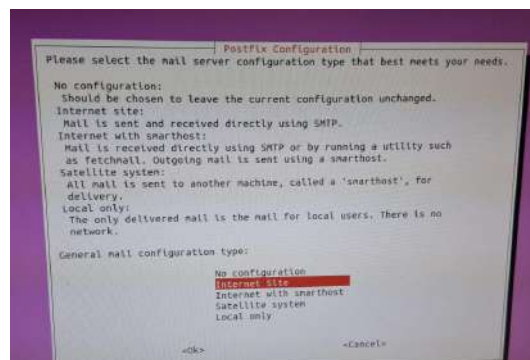
La portada de nuestra página web es:



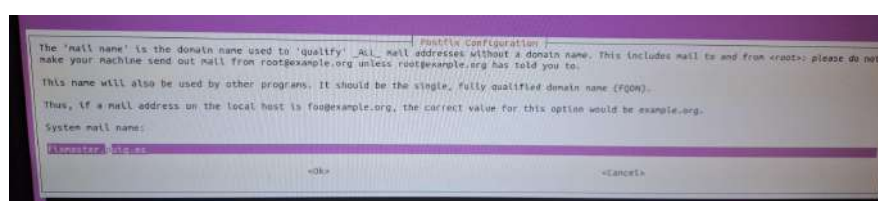
3.111 Servidor de Correo

Quisimos montar un servidor de correo para que nuestros empleados se puedan comunicar entre sí o enviar un comunicado a todos los empleados en caso de emergencia o cambio de posiciones, etc.

Para empezar instalamos postfix y mailutils, nos pregunta sobre la configuración de postfix, a lo que seleccionamos Internet Site para que los mails se envíen y reciban usando SMTP.



Seguidamente, nos pregunta por el nombre del dominio, nuestra empresa se llama fixmaster por lo que pondremos fixmaster.puig.es.

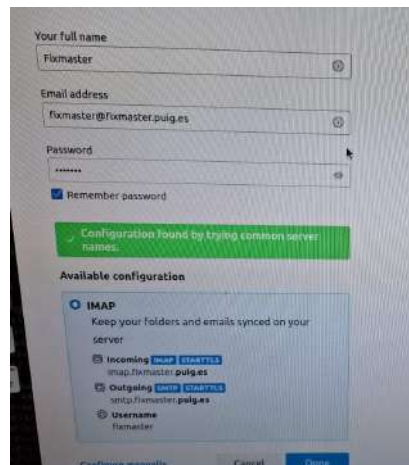


A continuación descargamos el MDA dovecot para que el correo almacenado sea IMAP4, de esta manera los correos recibidos pueden ser vistos desde diferentes dispositivos, ya que no se eliminan.

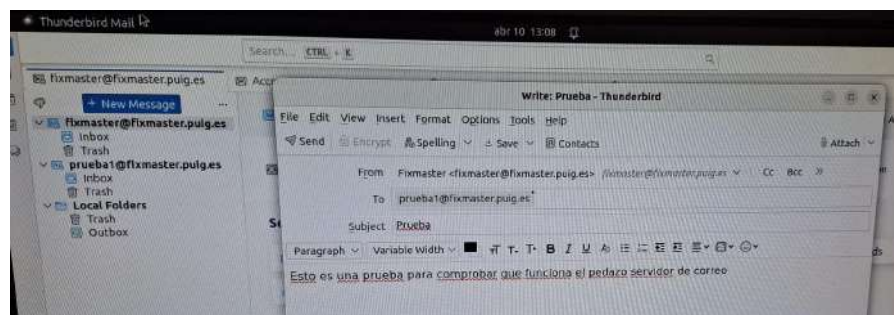
```
dovecot-core
Suggested packages:
dovecot-gssapi dovecot-ldap dovecot-imtp dovecot-lucene dovecot-managesieved dovecot-mysql dovecot-pgsql dovecot-pop3d dovecot-sieve dovecot-solr
dovecot-sqlite dovecot-submissiond ntp
The following NEW packages will be installed:
dovecot-core dovecot-imapd
0 upgraded, 2 newly installed, 0 to remove and 238 not upgraded.
Need to get 3.512 kB of archives.
After this operation, 11,3 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dovecot-core amd64 1:2.3.16+dfsg1-3ubuntu2.4 [3.328 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dovecot-imapd amd64 1:2.3.16+dfsg1-3ubuntu2.4 [193 kB]
Fetched 3.512 kB in 1s (5.762 kB/s)
Selecting previously unselected package dovecot-core.
(Reading database ... 207801 files and directories currently installed.)
Preparing to unpack .../dovecot-core_1:2.3.16+dfsg1-3ubuntu2.4_amd64.deb ...
Unpacking dovecot-core (1:2.3.16+dfsg1-3ubuntu2.4) ...
```

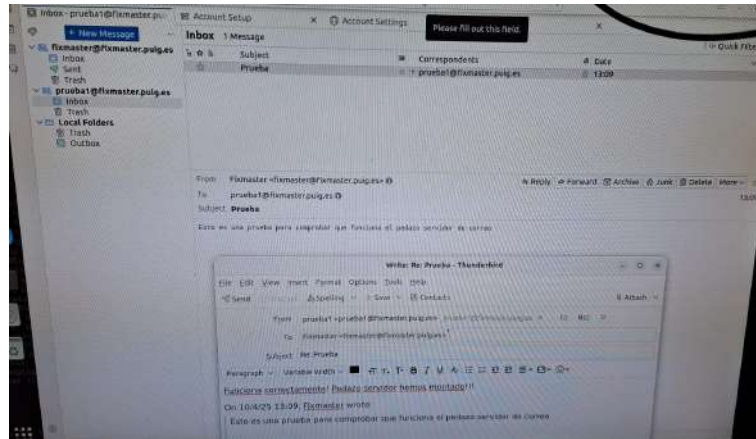
Luego creamos algunos usuarios para testear el servidor de correo e instalamos flatpak para descargar el MUA Thunderbird para tener un entorno gráfico para que sea más cómodo tanto para nosotros como para los empleados.

```
root@angara:/home/usuario# adduser fixmaster
info: Añadiendo el usuario 'fixmaster' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Añadiendo el nuevo grupo 'fixmaster' (1001) ...
info: Adding new user 'fixmaster' (1001) with group 'fixmaster (1001)' ...
info: Creando el directorio personal '/home/fixmaster' ...
info: Copiando los ficheros desde '/etc/skel' ...
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para fixmaster
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []:
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n]
```



Iniciamos la prueba enviando un mail a “prueba1”.





Una vez comprobamos que funciona decidimos hacer un relay para enviar los correos desde fixmaster.elpuig@gmail.com. De esta manera los clientes recibirán los mails desde ese correo, tenemos que hacer esto, ya que la lista de DNS de google no conoce nuestro dominio, por lo que las notificaciones que enviamos a los clientes no les llegarán.

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/postfix/cacert.pem
smtp_use_tls = yes
-- INSERT --
```

```
[smtp.gmail.com]:587 fixmaster.elpuig@gmail.com:skibiditoilet
```

```
root@usuario-fixmaster:/home/usuario# postmap /etc/postfix/sasl_passwd
```

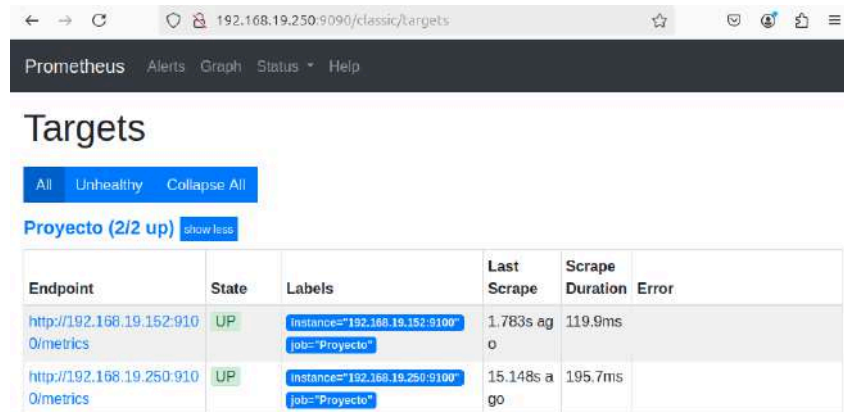
Desgraciadamente, esta idea no siguió adelante por la falta de tiempo y la [guía](#) que seguíamos estaba desactualizada (2021 y para Ubuntu Server 20.04). Nos tuvimos que conformar con un plug-in desde la página web.

3.112 Prometheus, Grafana y Loki

Para asegurarnos del correcto funcionamiento del servidor, decidimos instalar prometheus, que sirve para monitorizar dispositivos, este lo que hace es pillar métricas del ordenador, por ejemplo, la memoria libre, la usada, el número de cores, etc. Estas métricas se guardan en <ip>:9090 para después instalar un programa que pueda graficar dichas métricas para que sea visualmente más cómodo.

Aquí ponemos el grupo al cual queremos monitorizar, en este caso, he puesto el nombre "Proyecto" y que pille las métricas de los siguientes ordenadores.


```
- job_name: Proyecto
static_configs:
- targets:
  - 192.168.19.250:9100
  - 192.168.19.152:9100
"/etc/prometheus/prometheus.yml"
```



Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.19.152:9100/metrics	UP	<code>instance="192.168.19.152:9100"</code> <code>job="Proyecto"</code>	1.763s ago	119.9ms	
http://192.168.19.250:9100/metrics	UP	<code>instance="192.168.19.250:9100"</code> <code>job="Proyecto"</code>	15.148s ago	195.7ms	

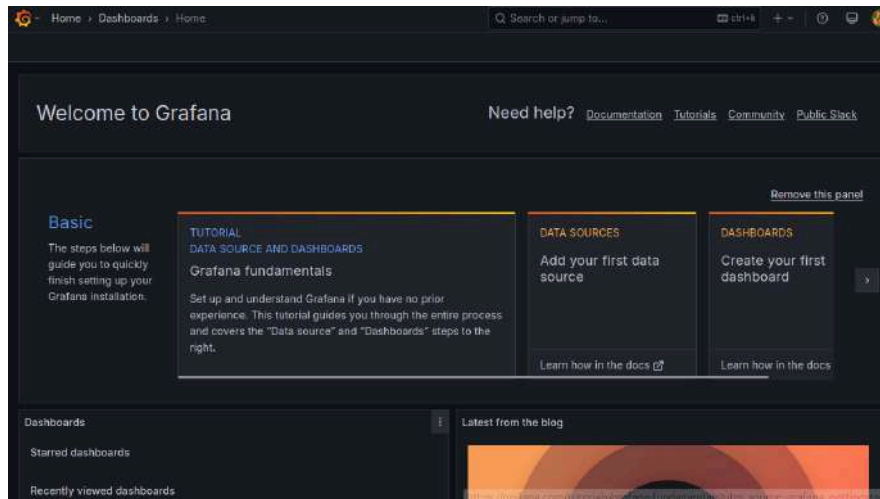
Métricas PC ipmi



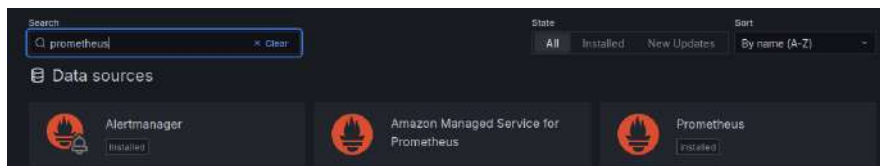
En el puerto 9100 encontramos las métricas del servidor.



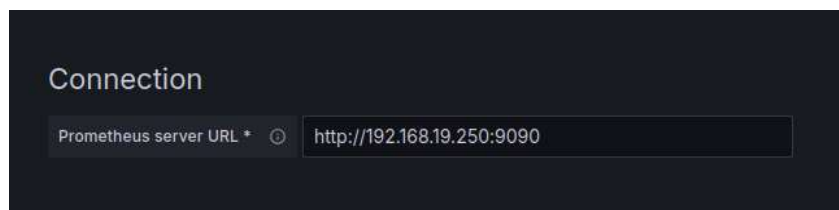
Instalamos el Grafana y lo abrimos en un navegador con el puerto 3000 (por defecto).



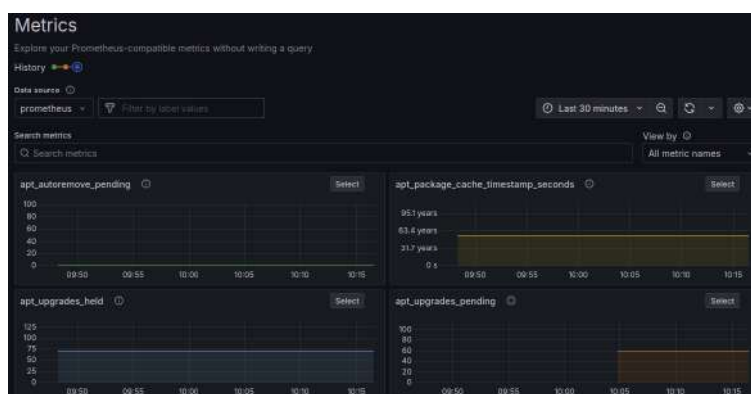
Seleccionamos Prometheus y lo añadimos



Le indicamos que el servidor es el Servidor Xeon, ya que este tiene los targets que necesitamos, estos targets están en el Job anteriormente creado.



Una vez Grafana se ha conectado con el servidor, podremos ver todas las métricas y crear dashboards de manera libre.



También quisimos introducir Loki en Grafana, ya que el punto fuerte de Loki son los logs, de esta manera podremos ver de mejor manera los eventos del servidor.

Para empezar, tenemos que instalar el paquete debian de loki, para no instalar una versión antigua decidimos ir a la página del creador y descargar la última versión, seguidamente instalamos el paquete y al hacer un status del servicio vemos que hay algo mal.

```
● loki.service - Loki service
   Loaded: loaded (/etc/systemd/system/loki.service; enabled; preset: enabled)
   Active: activating (auto-restart) (Result: exit-code) since Mon 2025-05-12 09:39:12 CEST; 6min ago
     Process: 5485 ExecStart=/usr/bin/loki -config.file /etc/loki/config.yml
    Main PID: 5485 (code=exited, status=1/FAILURE)
      CPU: 65ms
```

Al ver los logs del journal, vimos que el error estaba en un archivo .yaml que indica la configuración de loki, para ser más específico, el error estaba en la línea 51, comentamos esa línea y ya funciona.

```
● loki.service - Loki service
   Loaded: loaded (/etc/systemd/system/loki.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-12 09:39:12 CEST; 6min ago
     Main PID: 1044 (loki)
        Tasks: 21 (limit: 38375)
      Memory: 185.6M (peak: 186.9M)
         CPU: 7.008s
       CGroup: /system.slice/loki.service
              └─1044 /usr/bin/loki -config.file /etc/loki/config.yml
```

Seguidamente, instalamos promtail, el cual es quien recoge los logs y los envía a Loki.

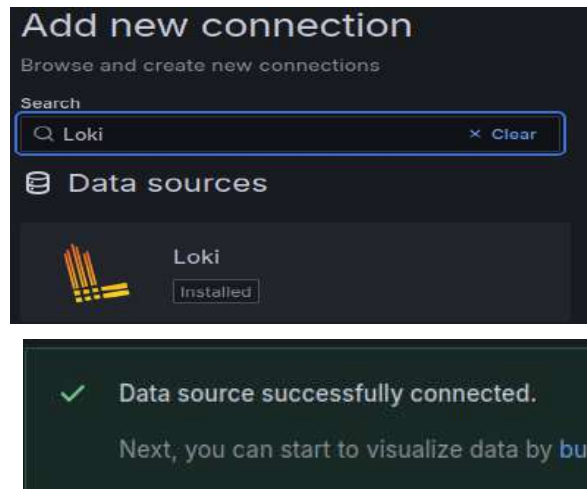
Para instalarlo hacemos lo mismo que Loki, vamos a la página de Grafana, al apartado promtail y descargamos el paquete .deb para luego instalarlo dentro del S.O. Como podemos ver, promtail funciona ya nada más descargarlo, pero vamos a añadirle otra configuración grabada en la [página del puig](#).

```
root@servidorgbc:/home/fixnaster# systemctl status promtail
● promtail.service - Promtail service
   Loaded: loaded (/etc/systemd/system/promtail.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-12 09:45:15 CEST; 20min ago
     Main PID: 2914 (promtail)
        Tasks: 12 (limit: 38375)
      Memory: 16.1M (peak: 17.2M)
         CPU: 6.714s
       CGroup: /system.slice/promtail.service
              └─2914 /usr/bin/promtail -config.file /etc/promtail/config.yml
```

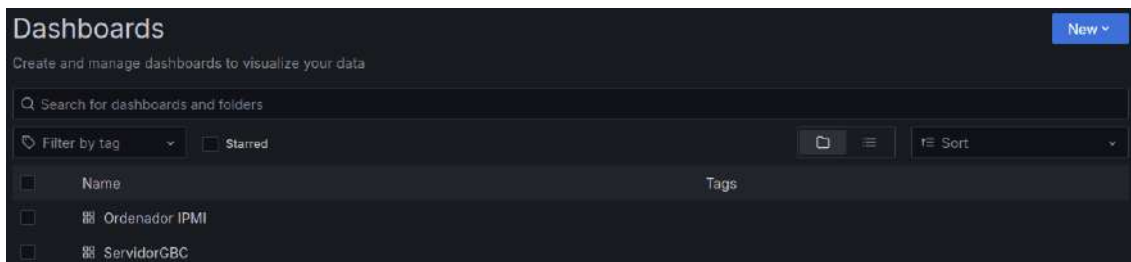
Una vez ya hemos cambiado la configuración de promtail, debemos añadir al usuario promtail al grupo adm, este para que tenga permisos y pueda ser capaz de leer los logs.

```
root@servidorgbc:~# cat /etc/passwd
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,fixnaster,promtail
```

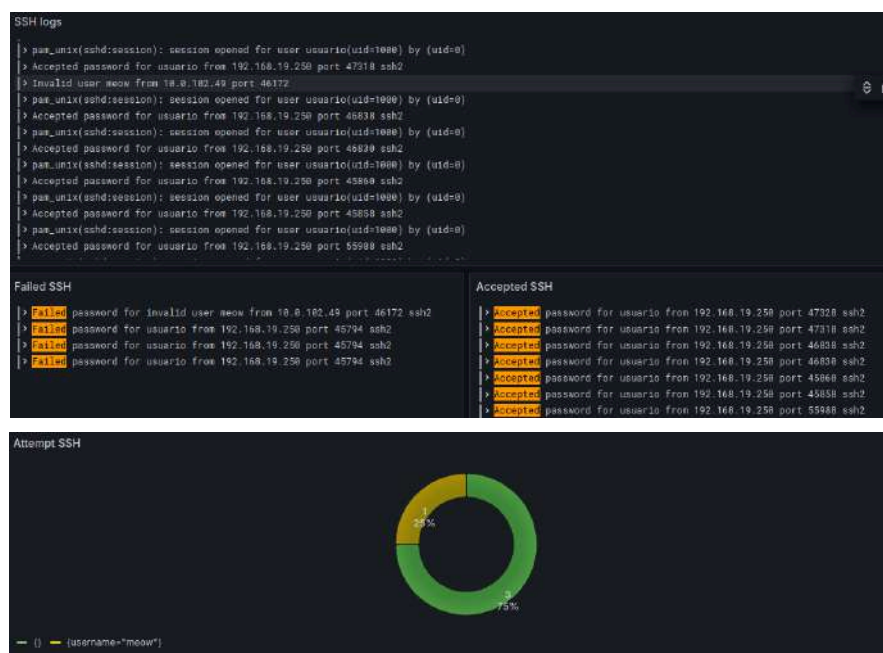
Ahora que ya hemos terminado, vamos al Grafana y añadimos una nueva conexión de Loki.



Ya añadido, podemos crear los dashboards que queramos con los logs para monitorizar.



En este caso, hemos añadido los logs del ssh.service, para ver si alguien intenta entrar al servidor desde ssh o algo parecido, para esto hemos filtrado por ssh rechazado y aceptado, aparte que hemos puesto intentos de usuarios, es decir, si lo intentan con un usuario que no existe, lo marcará.



3.112 Problemas obtenidos

3.112.1 Primer Problema

El día 10 de febrero, teníamos planeado instalar apache y configurarlo para poder hacer que sea un Servidor Web. Al iniciar el servidor, nos encontramos con que se ponía en modo emergencia y no dejaba iniciar. Con el paso de las horas, pudimos detectar que el error era en uno de los discos gracias al UUID de dichos discos, concretamente el que fallaba era el que tenía la información del sistema, y por ello, hemos tenido que volver a empezar con la instalación completa.

Lo hablamos con el Fede y nos dijo que el disco no era de fiar, por lo que lo marcamos como no fiable con rotulador.

```

root@kali:~# cat /etc/default/grub | grep --color=always 'GRUB_CMDLINE_LINUX="type=journalistic -x" to view
system logs, systemctl reboot to reboot, or exit"
^C continue bootup.
Press Enter for maintenance
or press Control-D to continue):
oot@servidorqbc:~#
oot@servidorqbc:~#
oot@servidorqbc:~#
oot@servidorqbc:~#
oot@servidorqbc:~#
oot@servidorqbc:~#
oot@servidorqbc:~#
oot@servidorqbc:~# exit
^C
xit
eloading system manager configuration.
default.target is not inactive. Please review the default.target setting.
allback to the single-user shell.
ress Enter for maintenance
or press Control-D to continue):
eloading system manager configuration.
default.target is not inactive. Please review the default.target setting.
allback to the single-user shell.
ress Enter for maintenance
or press Control-D to continue):
oot@servidorqbc:~# _

```

[illegible]

```

parted /dev/sda: 465,76 GiB. 500017862016 bytes, 976773168 sectors
lsk model: WDC WD5000AAKX-0
nits: sectors of 1 * 512 = 512 bytes
sector size (logical/physical): 512 bytes / 512 bytes
/0 size (minimum/optimal): 512 bytes / 512 bytes
isk label type: gpt
isk identifier: SC015747-A77A-457B-8C64-44CFC5FAE396

evlve
dev/sda1      3048      4096      2048      1K      BIOS boot
dev/sda2      4096      4139394      4139394      20      Linux filesystem
dev/sda3      4198400      976771071      972572572      463,6B      Linux filesystem

lsk /dev/sdb: 465,76 GiB. 500017862016 bytes, 976773168 sectors
lsk model: ST35000410A0
nits: sectors of 1 * 512 = 512 bytes
sector size (logical/physical): 512 bytes / 512 bytes
/0 size (minimum/optimal): 512 bytes / 512 bytes

lsk /dev/sdc: 465,76 GiB. 500017862016 bytes, 976773168 sectors
lsk model: ST35000410A0
nits: sectors of 1 * 512 = 512 bytes
sector size (logical/physical): 512 bytes / 512 bytes
/0 size (minimum/optimal): 512 bytes / 512 bytes

lsk /dev/mapper/ubuntu--vg-ubuntu--lv: 100 GiB. 107374182400 bytes, 209715200 sectors
nits: sectors of 1 * 512 = 512 bytes
sector size (logical/physical): 512 bytes / 512 bytes
/0 size (minimum/optimal): 512 bytes / 512 bytes

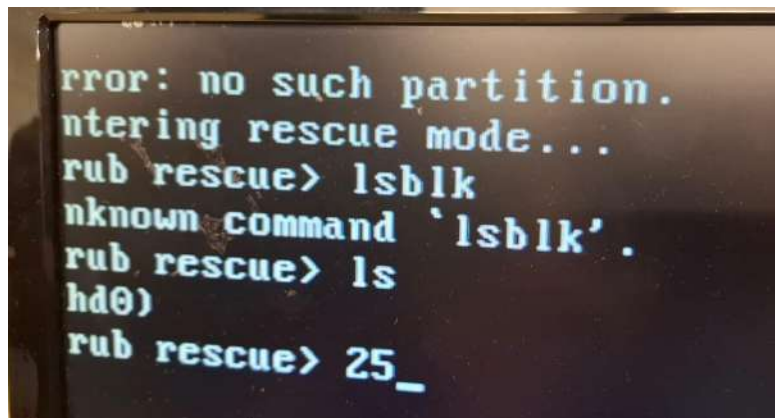
```

En casa, instalamos el Ubuntu 24.04 Server de nuevo en un USB para poder solucionarlo, el día 19, con la mala suerte de que la estética del USB escogido, era demasiado grande y no podíamos conectarlo al puerto del servidor. Con suerte de que habíamos traído uno más

pequeño, pero sin nada instalado, tendremos que volver a descargar la imagen en clase para poder empezar con la reinstalación.

3.112.2 Segundo Problema

Otro problema que hemos tenido con el servidor es que, al ir a hacer la RAID, no sabíamos donde estaba el Ubuntu instalado y desmontamos un disco que tenía ya varias particiones, que casualmente, era el del Ubuntu, por lo tanto, nos salió el modo “grub rescue” y tuvimos que volver a instalarlo de 0 (habiendo hecho ya la configuración de Apache, DNS, Wireguard).

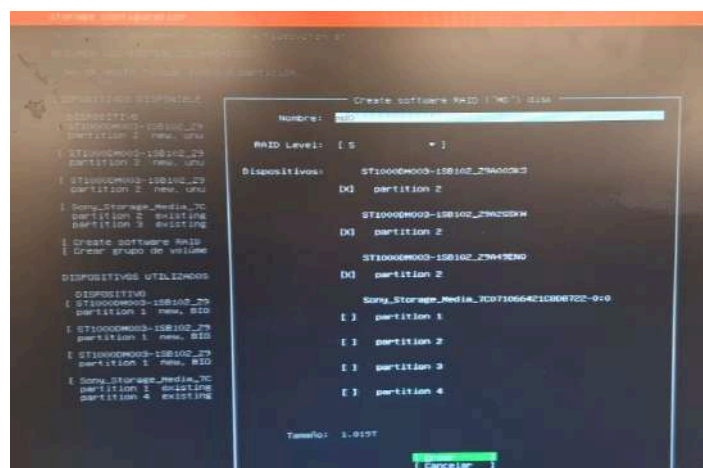


El principal motivo ha sido que no teníamos ni idea de cuál era cada disco, ya que los detecta en sentido contrario, es decir, el disco que está más arriba es sdc y él más abajo es sda.

Debido a esto, estuvimos investigando durante un par de días sobre como hacer el raid y porque nos salía el “grub rescue”.

Solución Final:

- Hablamos tanto con José Camuñez como Victor Carceler y llegamos a la conclusión de que lo mejor era hacer el RAID desde la instalación, y ya que íbamos a hacer un RAID con 3 discos, se nos abría la posibilidad de hacer un RAID 5 en vez del 1.



```
[ Sony_Storage_Media_7C071066421C8D8722-0:0
partition 1 existing, unused ESP, already formatted as vfat 14.484G > ]
partition 3 existing, unused 4.983M > ]
[ Create software RAID (md) > ]
[ Crear grupo de volúmenes (LVM) > ]

DISPOSITIVOS UTILIZADOS

DISPOSITIVO TIPO TAMAÑO
[ md0 (new, to be formatted as ext4, mounted at /) software RAID 5 1.819T > ]
[ ST10000M003-1S8102_Z3A003K3 disco local 931.513G > ]
partition 1 new, BIOS grub spacer 1.000M > ]
partition 2 new, component of software RAID 5 md0 931.509G > ]
[ ST10000M003-1S8102_Z3A26SKW disco local 931.513G > ]
partition 1 new, BIOS grub spacer 1.000M > ]
partition 2 new, component of software RAID 5 md0 931.509G > ]
[ ST10000M003-1S8102_Z3A43EN0 disco local 931.513G > ]
partition 1 new, BIOS grub spacer 1.000M > ]
partition 2 new, component of software RAID 5 md0 931.509G > ]
[ Sony_Storage_Media_7C071066421C8D8722-0:0 disco local 14.484G > ]
partition 1 existing, already formatted as iso9660, in use 2.578G > ]
```


3.2 Consola de sobremesa

3.2.1 Conexiones eléctricas de la consola

Vamos a hacer ahora un recuento de todas las conexiones a realizar entre la Raspberry Pi y sus periféricos, dando más detalles técnicos en algunas de ellas.

RPI 5 → Monitor BenQ. La conexión se realizará mediante nuestro cable HDMI con conector estándar en un lado y conector microHDMI en el otro (gracias al adaptador).

RPI 5 → Teclado y Ratón. La conexión se realizará mediante el propio cable USB tipo A de los periféricos a los puertos del mismo tipo en la RPI.

RPI 5 → Amplificador MAX98357A.

El siguiente recuento está basado en la documentación oficial del vendedor Adafruit sobre el amplificador **MAX98357A**. Será el controlador (driver) proporcionado por esta empresa el cual usaremos, más adelante, para configurar nuestro amplificador y que este pueda ser utilizado como periférico de salida.

- **Amp Vin to RPI 3.3V Power.** Alimentamos nuestro módulo a partir del pin de 3.3V de nuestra RPI que se conecta al pin de entrada de voltaje de nuestro amplificador. Este módulo admite entradas de hasta 5.5V y, por ende, también sería posible alimentarlo con el pin de 5V de la RPI. Pese a esto, como se indicó en la descripción de componentes, lo alimentamos con 3.3V para reducir el consumo de corriente del amplificador y que este no trabaje ni se caliente tanto. Además, reduce la potencia de salida del altavoz, reduciendo el riesgo de quemadura.
- **Amp GND to RPI GND.** Al igual que hemos conectado la línea positiva, tendremos que cerrar el circuito conectando una de las líneas negativas (GND) de la RPI con GND del amplificador.
- **Amp DIN to RPI GPIO 21 (PCM_DOUT).** El pin PCM_DOUT de la RPI es por donde se emiten las señales de audio de salida digital que son luego recibidas por el convertor DAC mediante el pin DIN.
- **Amp BCLK to RPI GPIO 18 (PCM_CLK).** El pin PCM_CLK de la RPI es por donde se emiten los pulsos que le hacen saber al convertor cuando llegará nueva información. Dichos pulsos son recibidos por el pin BCLK.
- **Amp LRC to RPI GPIO 19 (PCM_FS).** Por el pin PCM_FS de la RPI se envían los pulsos que le hacen saber al convertor para que canal suyo (izquierdo, derecho, ambos) será el audio digital que se está recibiendo por DIN.

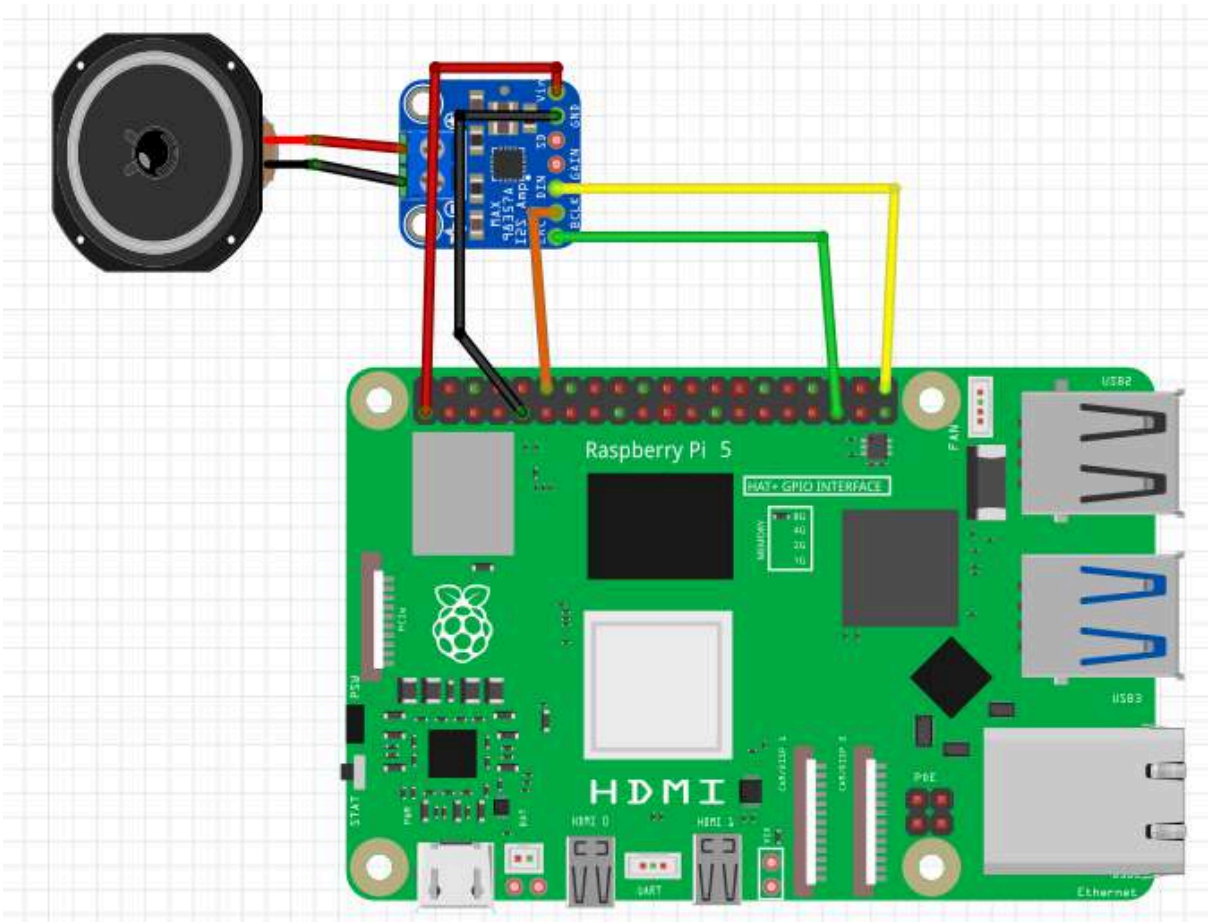
MAX98357A → Altavoz 2W 8Ω. La conexión será realizada uniendo los polos positivo/negativo del altavoz de la forma correcta en la salida del módulo amplificador.

RPI 5 → PS 27W. La conexión será realizada mediante el conector USB-C de la PS con el puerto para el mismo integrado en la placa.

RPI 5 → DS4. La conexión será realizada mediante un cable microUSB a USB tipo A.

RPI 5 → microSDXC 64GB. Conexión mediante el slot de tarjetas microSD.

Hecho el recuento, decidimos hacer un diagrama de las conexiones más complejas, cómo lo son las del MAX98357 con la RPI, con el programa Fritzing. Este programa libre y gratuito se utiliza mucho para la realización de esquemas eléctricos profesionales, diagramas eléctricos (cómo el de a continuación) y diseño de PCB. Este programa dispone de una serie de librerías con una gran diversidad de componentes que pueden ser utilizados en los diagramas. Si hay algún componente no existente, los usuarios tienen la capacidad de crearlo y añadirlo cómo un archivo **.fzpz** el cual puede ser importado en la librería *mine* (mis partes).



RPI 5	Amplificador
GPIO21 (PCM_DOUT)	DIN
GPIO19 (PCM_FS)	LRC
GPIO18 (PCM_CLK)	BCLK
3.3V	V _{in}
GND	GND

Para el anterior diagrama usamos partes internas de fritzing más una de la comunidad para la RPI 5. [Enlace aquí.](#)

3.22 Componentes y funcionalidades del lanzador de emuladores

3.22.1 Componentes

1 Emulador (Mesen2)

Es el núcleo o una de las partes más importantes de nuestro programa, será este el que proporcione al usuario la capacidad de leer las ROMs descargadas y que estas puedan ser ejecutadas por la arquitectura de dicho ordenador (en nuestro caso una RPI con ARM64).

2 Videojuegos (ROMs)

Otra de las principales fuentes de acción de nuestro emulador y deberán tener el formato adecuado para el emulador, en este caso, la extensión **.nes**. Cabe recalcar que solo descargaremos ROMs o demos de las mismas que sean totalmente gratuitas.

3. Macros

Las usaremos para que el usuario con *gamepad* pueda realizar las acciones que podría realizar un usuario con teclado y ratón, esto es debido a que muchas de las opciones del emulador Mesen2 son accesibles mediante teclado y ratón. Esto lo haremos mediante el uso de un programa de macros en la CLI que nos permita crear un script para cada macro necesaria, estas luego serían lanzadas a demanda del jugador mediante otro script con un menú iterativo (hecho por nosotros). Las macros serían ejecutadas de la siguiente manera,

1. El jugador, mediante un mapeado configurado con Input Remapper, hará aparecer la terminal con el script de *macros* con la lista de macros disponibles.
2. El usuario navega por este menú haciendo uso de otro mapeo realizado con Input Remapper y selecciona la macro correspondiente según el tipo de acción que quiera realizar

Para poder interactuar con el *gameplay* del propio Mesen2, haremos lo siguiente:

1. Habilitar el modo de juego con teclado en Mesen2
2. Realizar un mapeado en InputRemapper de los botones correspondientes a las teclas correspondientes que usamos para jugar en Mesen2 si lo hiciéramos directamente con teclado y ratón.

Nota.: Más adelante, en la conclusión sobre el análisis de diferentes programas para macros en la CLI, se realiza un cambio sobre el hecho de cómo vamos a implementar dicho componente

4. Programa para hacer *focus* en la UI

Para que las teclas inyectadas con las macro tengan efecto en el emulador, será necesario otro programa que ponga la atención o el *focus* en dicho programa. Esto sería básicamente cuando, por ejemplo, hacemos clic en la ventana de la terminal para que las teclas que pulsemos se introduzcan en el prompt. Mientras no hagamos clic en otro sitio, las teclas solo introducen caracteres en la terminal y no en otros sitios. Por ello de la necesidad de este programa, el cual se ejecutará siempre en nuestro script de macros justo antes del lanzamiento de una macro.

5. Mapper de botones

Este segundo programa tendrá las siguientes funciones

- Mapeado de botones del gamepad real para
 - Utilización del gamepad virtual (NES)
 - Abrir y/o cerrar la ventana del programa macros con las acciones disponibles
 - Navegación por la ventana con las macros

Pese a que este programa también permite configurar Macros, hemos decidido incorporar otro programa para cumplir esta función ya que sino, con Input Remapper, el usuario tendría que recordar demasiadas combinaciones de botones debido a la gran cantidad de acciones posibles en el emulador Mesen2.

6. Servicio para el copiado de ficheros por red

Este servicio será esencial para poder transferir los juegos almacenados en el repositorio del servidor *supermicro*.

5. Script lanzador

Será este script hecho con **shell-script** el que nos permita automatizar todo el procedimiento necesario para el *gameplay* de una forma intuitiva. Los procesos a automatizar serían

- Ejecución del emulador Mesen2
- Descarga y eliminación de ROMs
- Configuración del gamepad real y virtual
- Selección de opciones en la UI del emulador

3.22.2 Funciones/casos de uso del script lanzador

1. Ejecución local del emulador Mesen2. Dicho emulador será contenido dentro de un directorio fijo en la Raspberry Pi 5. Luego, dentro de nuestro script lanzador de emuladores dispondremos de una opción que permitirá ejecutar y poner en pantalla completa el emulador con solo una pulsación de botones.

2. Repositorio de ROMs en el servidor *supermicro*. En el servidor dispondremos de un repositorio con las ROMs que hemos escogido de la comunidad de desarrolladores de juegos para NES. Mediante nuestro script y la herramienta scp, el cliente sería capaz de

- Acceder a una lista de juegos que se actualiza dinámicamente
- Visualizar que juegos ya tiene o no tiene en su máquina
- Escoger una ROM y descargarla a un directorio local fijo con scp.

Desde este menú también sería posible desinstalar las ROMs que veamos que ya tenemos (tienen una marca). De igual manera, puede ser que el cliente descargue juegos que luego pueden desaparecer del repositorio, impidiendo su desinstalación. Para ello introducimos la siguiente función.

3. Borrado de ROMs descargadas. En caso de que el cliente tenga ROMs descargadas que ya no están presentes en el repositorio del servidor, ponemos a disposición del usuario una opción que muestra solo los juegos instalados. Desde aquí podrá desinstalar las ROM presentes o no en el repositorio.

3.23 Valoración de servicios para las funciones del script lanzador

3.23.1 Servicio para la descarga de ficheros vía red

1. Requisitos

1. Sin autenticación de usuario para una mayor velocidad de descarga. Los datos que contenemos en el directorio de origen no serán confidenciales y, por ende, no queremos limitar el acceso para así poder disfrutar de una mayor velocidad de acceso y descarga.

2. Servicio moderno y con cifrado. Pese a que se contradice un poco con el requisito anterior, establecemos este requisito, ya que, seguramente, no se dará soporte a herramientas sin cifrado en internet. Será mucho más fácil encontrar una buena documentación en servicios modernos y con cifrado, que en servicios antiguos y sin cifrado (cómo es el caso de FTP).

3. En caso de haber un programa para el cliente y otro para el servidor...

- **Versión de cliente compilada para ARM64**
- **Versión de servidor compilada para x86**

4. Que permita la ejecución de comandos en una sola línea

2. Opciones

2.1 VSFTPD

1. Sin autenticación de usuario para una mayor velocidad de descarga.

2. Servicio moderno y con cifrado.

3. En caso de haber un programa para el cliente y otro para el servidor...

- **Versión de cliente compilada para ARM64.** Los repositorios de Debian 12 incluyen un paquete debian de vsftpd compilado para ARM64.
- **Versión de servidor compilada para x86.** Los repositorios de Ubuntu 24.04 de servidor incluyen un paquete debian de vsftpd compilado para x86 (AMD64).

4. Que permita la ejecución de comandos en una sola línea. Es un servicio de carácter interactivo, el cual no permite ejecutar varios comandos “de golpe” (qué sería aquello que haríamos en un script).

2.1 scp

1. Sin autenticación de usuario para una mayor velocidad de descarga. Esta herramienta funciona sobre ssh, por lo cual podemos aplicar la autenticación por clave pública para no tener que introducir un password.

2. Servicio moderno y con cifrado.

3. En caso de haber un programa para el cliente y otro para el servidor...

- **Versión de cliente compilada para ARM64.** Existen versiones compiladas para ARM64 en los repositorios de Debian 12
- **Versión de servidor compilada para x86.** Existen versiones compiladas para AMD64 en los repositorios de Ubuntu 24.04 de servidor

4. Que permita la ejecución de comandos en una sola línea

3.23.2 Descripción técnica de la herramienta scp

1. Protocolo scp

Este protocolo es básicamente el comando **cp (copy)** de *bash* el cual funciona sobre el protocolo SSH con el fin de poder copiar ficheros de máquina local a máquina remota y viceversa todo de forma cifrada. Será muy útil para nuestro script, ya que es un programa que no requiere mucha interacción, cosa que lo hace fácil de implementar en un script al poder utilizar sus funciones con una sola línea de comandos.

3.23.3 Pruebas del servicio scp

Para ello vamos a crear un escritorio Ubuntu Desktop 24.04 con Isard VDI el cual vamos a configurar para que forme parte de nuestra VPN. Con esto, podremos acceder a nuestro servidor *supermicro* y ver si el protocolo scp funciona correctamente entre estas dos máquinas aún estando en redes diferentes.

Isard con Ubuntu 24.04 LTS

1. Configuración inicial

Lo que hicimos fue crear primero un escritorio IsardVDI Ubuntu Desktop 24.04 LTS con la red **puigcastellar1**. Una vez dentro...

- Cambiamos el hostname
- Instalamos SSH

Ahora, un paso fundamental para poder acceder a las máquinas que tenemos montadas en el centro es configurar la VPN, para ello seguimos los siguientes pasos.

Nota.: La guía de wire guard del centro (Autor: Victor Carceller) nos fue de mucha ayuda.

2. Configuración de la VPN

1. Editar el fichero `/etc/wireguard/wg0.conf` e introducir:

```
[Interface]
# Clave privada del cliente
PrivateKey = ULNhMsNHictlnoqBd0pGkLuS5ccNfflLg6mnZywsIw=
# IP para el cliente en la VPN
Address = 10.0.102.40
[Peer]
# Clave pública del servidor
PublicKey = ZmyzjyMinfDqvOzaIKf/WcAr9jjlaSKyFOksYxGY+FE=
# Tráfico que se aceptará (toda la red /24)
AllowedIPs = 10.0.102.0/24
# IP y puerto UDP en el que escucha el servidor
Endpoint = 10.0.102.43:40002
# Si no hay tráfico transmitir un paquete cada 30 segundos
PersistentKeepalive = 30
```

Habilitar e iniciar la interfaz VPN con:

- `systemctl enable wg-quick@wg0`
- `systemctl start wg-quick@wg0`

Si lo hemos hecho bien, veremos el estado **active** en el servicio wire guard (mediante la unidad de systemd).

```
root@rpi-ficticia:/home/usuario# systemctl status wg-quick@wg0
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
   Loaded: loaded (/usr/lib/systemd/system/wg-quick@.service; enabled; preset: enabled)
   Active: active (exited) since Thu 2025-05-15 20:44:04 CEST; 23s ago
     Docs: man:wg-quick(8)
```

Cómo era de esperar, tenemos ya nuestra interfaz activa con la IP 10.0.102.40.

```
4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420
    link/none
    inet 10.0.102.40/24 scope global wg0
        valid_lft forever preferred_lft forever
root@rpi-ficticia:/home/usuario#
```

3. Prueba de acceso a las máquinas de Darwin

Primero probamos a acceder al cliente de Darwin. Cómo se observa en la terminal de Isard, estamos en la sesión del mismo, ya que se aprecia su VPN (10.0.102.48) y su IP en Darwin (red 192.168.19.0/24).

```

2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 00:23:24:0e:0e:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.19.152/24 brd 192.168.19.255 scope global dynamic noprefixroute
        valid_lft 509sec preferred_lft 509sec
    inet6 fe80::2324:0e0e:f5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wls2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 70:f1:a1:9f:f0:51 brd ff:ff:ff:ff:ff:ff
    altname wlp48s0
4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default
    link/none
    inet 10.0.102.217/24 scope global wg0
        valid_lft forever preferred_lft forever
5: wg10: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default
    link/none
    inet 10.0.102.48/24 scope global wg10
        valid_lft forever preferred_lft forever

```

Si probamos de acceder a nuestro servidor, también tendremos éxito (tiene la VPN 10.0.102.43).

```

root@supermicro:/home/fixmaster# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp2s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
    link/ether 00:25:90:77:a0:8e brd ff:ff:ff:ff:ff:ff
    inet 192.168.19.250/24 brd 192.168.19.255 scope global enp2s0f0
        valid_lft forever preferred_lft forever
    inet6 fe80::2590:77:a0:8e/64 scope link
        valid_lft forever preferred_lft forever
3: enp2s0f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
    link/ether 00:25:90:77:a0:8f brd ff:ff:ff:ff:ff:ff
4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default
    link/none
    inet 10.0.102.43/24 scope global wg0
        valid_lft forever preferred_lft forever

```

4. Autenticación por clave pública en las máquinas darwin

Para ello vamos a generar un par de claves en nuestra máquina isard.

```

usuario@rpi-ficticia:~/.ssh$ ls -lh
total 16K
-rw-r----- 1 usuario usuario  0 jun 24  2024 authorized_keys
-rw-r----- 1 usuario usuario 411 may 18 09:36 id_ed25519
-rw-r----- 1 usuario usuario 102 may 18 09:36 id_ed25519.pub
-rw-r----- 1 usuario usuario 2,0K may 18 09:27 known_hosts
-rw-r----- 1 usuario usuario 1,1K may 15 21:24 known_hosts.old
usuario@rpi-ficticia:~/.ssh$

```

Hecho esto, moveremos con **scp** la clave pública al **home** del usuario de cada máquina con la cual queremos acceder directamente. Es decir

- Movemos la clave pública de nuestro Isard (creada con usuario) al home de “usuario” en el cliente Darwin
- Movemos la clave pública de nuestro Isard al home de “fixmaster” en el servidor

Hecho lo anterior, hemos de incluir el *hash* de dicha clave pública al fichero **/.ssh/authorized_keys** de cada usuario (podemos hacerlo con cat). Obviamente, debemos

acceder a dichas máquinas con SSH (mediante password) para hacer lo anterior. Si lo hemos hecho bien, podremos acceder como usuario en Isard a:

- usuario "usuario" en el cliente (**Fig. 1**)
- usuario "fixmaster" en el servidor(**Fig. 2**)

```
usuario@rpi-ficticia:~/.ssh$ ssh fixmaster@10.0.102.43
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-60-generic x86_64)
```

Fig. 2

```
usuario@rpi-ficticia:~/.ssh$ ssh usuario@10.0.102.48
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 6.8.0-59-generic x86_64)
```

Fig. 1

3.24 Instalación y configuración de RPI OS

3.24.1 Instalación de RPI OS

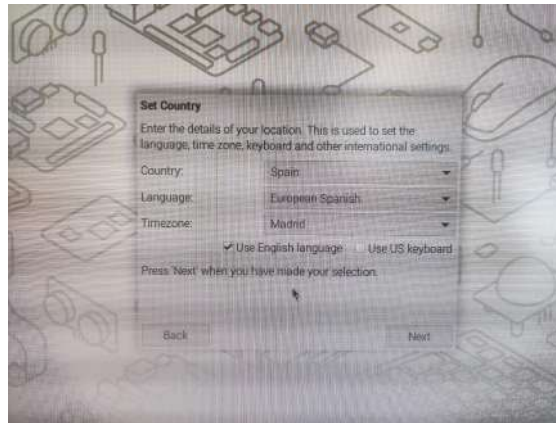
1er intento

Podemos preparar la imagen de Raspberry Pi OS Desktop 64 con el uso de la versión *latest* de RPI Imager. En este seleccionamos el dispositivo, la versión de sistema operativo y el almacenamiento donde dejar nuestra imagen. En nuestro caso escogimos **RPI 5 > Raspberry PI OS Desktop 64 > Almacenamiento USB**

Hecho esto, conectamos nuestro USB a uno de los puertos USB y, además, también nuestra microSD que será el almacenamiento final.



Si encendemos la RPI, veremos que automáticamente se detectará el entorno live y comenzará el proceso de instalación. Uno de los primeros pasos será escoger la localización y el layout de nuestro teclado.



El siguiente paso será definir un usuario en la RPI.



Hecho lo anterior, tendremos que confirmar algunas actualizaciones antes de que finalice la instalación. Acabadas estas, ya se iniciará nuestro SO.



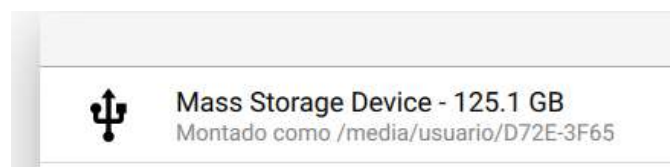
De igual forma, al cabo de investigar los dispositivos de almacenamiento activos, nos dimos cuenta de que el SO se había instalado en el USB y no en la microSDXC. Esto seguramente es debido a la naturaleza del instalador, el cual no te deja escoger el medio de almacenamiento final y escoger aquel que contiene la imagen. Debido a esto, tendremos que volver a repetir el proceso anterior pero introduciendo la imagen, al principio, directamente en la microSD.

2o intento

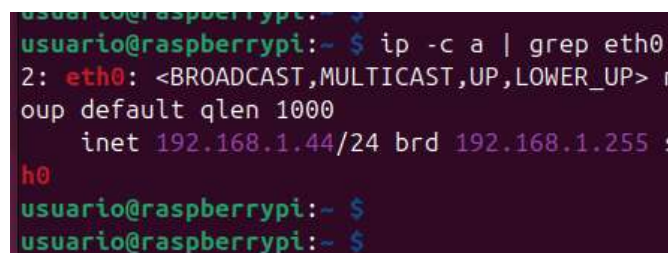
Debemos iniciar RPI Imager y, ahora sí, escoge **RPI 5 > Raspberry PI OS Desktop 64 > tarjeta microSD**.



A la hora de hacer lo anterior, veremos un error... “No se ha podido realizar la escritura”. Este se repitió un par de veces y por ende decidimos utilizar otra tarjeta microSDXC de 128GB. Esta tarjeta microSDXC utilizada proviene del centro, así como el adaptador utilizado para conectarla por USB.



Finalmente, lo único que nos queda es introducir dicha tarjeta microSD con la imagen e instalar el sistema operativo. Hecho esto, lo más importante fue iniciar y habilitar el servicio **ssh** para la administración remota. Cómo se ve en pantalla, hemos podido acceder a la misma mediante ssh.



3.24.2 Configuración de RPI OS

Configuración de **wireguard-tools**

El procedimiento será el mismo el cual se siguió para la configuración de la VPN con la máquina Isard en el apartado **3.224**. Para que nuestra RPI formará parte de dicha VPN,

escogimos una de las IPs libres de las proporcionadas por Juan Morote y configuramos los aspectos correspondientes

- Clave privada del cliente
- IP de la VPN
- Clave pública del servidor
- Tráfico que se aceptará
- Endpoint (servidor de la VPN)
- PersistentKeepalive (para no perder conexión con el servidor de la VPN debido a la desconexión automática).

```
[Interface]
# Clave privada del cliente
PrivateKey = CG31K6QuXKBzNGzdVUUPIMpeCzpMfQaBeUjssWr0H1g=

# IP para el cliente en la VPN
Address = 10.0.102.42/24

[Peer]
# Clave pública del servidor
PublicKey = ZmyzjyMinfdQvOzaIKf/WcAr9jjlaSKyFOksYxGY+FE=

# Tráfico que se aceptará (toda la red /24)
AllowedIPs = 10.0.102.0/24

# IP y puerto UDP en el que escucha el servidor
Endpoint = wireguard.elpuig.xeill.net:40002

# Si no hay tráfico transmitir un paquete cada 30 segundos
PersistentKeepalive = 30
```

Hecho lo anterior, lo único que nos queda es reiniciar y habilitar el servicio **wg-quick** en la interfaz **wg0**. Con ello veremos dicha interfaz activa en nuestra RPI.

```
valid_lft forever preferred_lft forever
4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420
    link/none
    inet 10.0.102.42/24 scope global wg0
        valid_lft forever preferred_lft forever
```

3.24.3 Autenticación por clave pública en las máquinas de darwin

Para poder acceder más fácilmente a las máquinas que tenemos en Darwin, vamos a activar la autenticación por clave pública de la misma forma que lo hicimos con la máquina Isard del apartado **3.224**. Para ello generamos, en la RPI, un par de claves con **ssh-keygen** y copiamos la clave pública al usuario de cada una de las máquinas.


```
usuario@raspberrypi:~/.ssh $  
usuario@raspberrypi:~/.ssh $ scp ./id_rsa.pub fixmaster@10.0.102.43:/home/fixmaster/  
fixmaster@10.0.102.43's password:  
id_rsa.pub 100% 573 5.3KB/s 00:00  
usuario@raspberrypi:~/.ssh $  
usuario@raspberrypi:~/.ssh $ scp ./id_rsa.pub usuario@10.0.102.48:/home/usuario/  
usuario@10.0.102.48's password:  
id_rsa.pub 100% 573 6.0KB/s 00:00  
usuario@raspberrypi:~/.ssh $
```

En el servidor, copiamos el contenido de dicha clave recibida al fichero **/home/fixmaster/.ssh/authorized_keys**. Con ello podremos acceder directamente al servidor (**Figura 1**). Lo mismo haremos en el cliente para ver el mismo resultado (**Figura 2**).

```
usuario@raspberrypi:~/.ssh $  
usuario@raspberrypi:~/.ssh $ ssh fixmaster@10.0.102.43  
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-60-generic x86_64)
```

Figura 1

```
usuario@raspberrypi:~/.ssh $  
usuario@raspberrypi:~/.ssh $ ssh usuario@10.0.102.48  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 6.8.0-59-generic x86_64)
```

Figura 2

3.24. bv4 Visor de escritorio remoto para la RPI 5: RPI-Connect

A la hora de trabajar con un sistema adicional, nos es conveniente poder tener una ventana con el sistema a estudiar y/o desarrollar más el propio sistema que usamos para hacer el trabajo y su seguimiento. Por ello creemos que sería una buena idea instalar una aplicación que abra sesiones que emitan la visión del escritorio.

Para poder acceder al escritorio de la Raspberry Pi 5 de forma remota hemos decidido utilizar la herramienta Raspberry Pi Connect. Es una alternativa oficial muy fiable y que se integra muy bien con el sistema operativo que hemos instalado. Este visor permite abrir sesiones cifradas que van directamente desde la RPI hasta el navegador del cliente (peer-to-peer), o bien se introduce un *relay* entre los dos si no se ha podido establecer la conexión en un inicio. Otro punto a favor es que las sesiones pueden ser abiertas desde la CLI (Command Line Interface), permitiendo a los usuarios con solo acceso remoto vía SSH configurar dicha aplicación y acceder al escritorio desde un navegador externo.

Para poder operar esta aplicación vamos a seguir los siguientes pasos.

RPI 5

1. Instalamos el programa con:

```
apt update && apt install rpi-connect --yes
```

```
usuario@raspberrypi:~$ rpi-connect --version
rpi-connect 2.5.2 (revision 0691027f16f3ddb6417b0030e1ebf50b2e74790) [arm64]
usuario@raspberrypi:~$
```

2. Iniciar una sesión con el usuario actual con → **rpi-connect on**

```
usuario@raspberrypi:~$ rpi-connect on
✓ Raspberry Pi Connect started
usuario@raspberrypi:~$
```

3. Habilitar inicio de sesión sin depender de la sesión del usuario que activó rpi-connect con **loginctl enable-linger**

Con esto, cómo indica en el propio punto, podemos iniciar sesiones remotas al escritorio de la RPI sin la necesidad de que el usuario que activó la conexión con el comando **rpi-connect** deba tener la sesión abierta.

4. Asociar una cuenta de Raspberry Pi (RPI ID) con RPI-Connect desde la CLI

Para ello debemos crear primero una cuenta de Raspberry Pi. Acto seguido, lanzamos el comando **rpi-connect signin**

Esto generará un enlace que podrá ser visitado desde cualquier dispositivo, y que permitirá iniciar sesión con dicha cuenta de RPI creada para enlazar la cuenta con el número de serie de nuestra RPI 5.

```
usuario@raspberrypi:~$ rpi-connect signin
Complete sign in by visiting https://connect.raspberrypi.com/verify/65M4-K3PH

::: Waiting for a response...
::: Waiting for a response...
```

Si visitamos la URL, podremos iniciar sesión con nuestra cuenta y vincular la RPI con la misma.

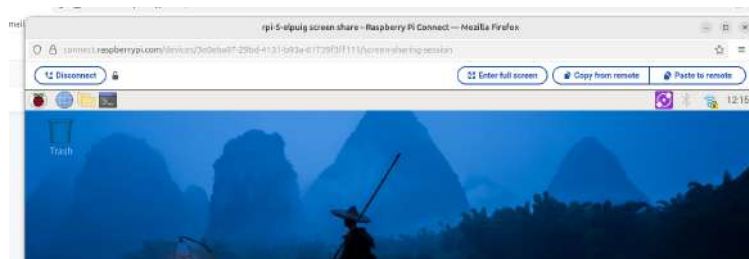
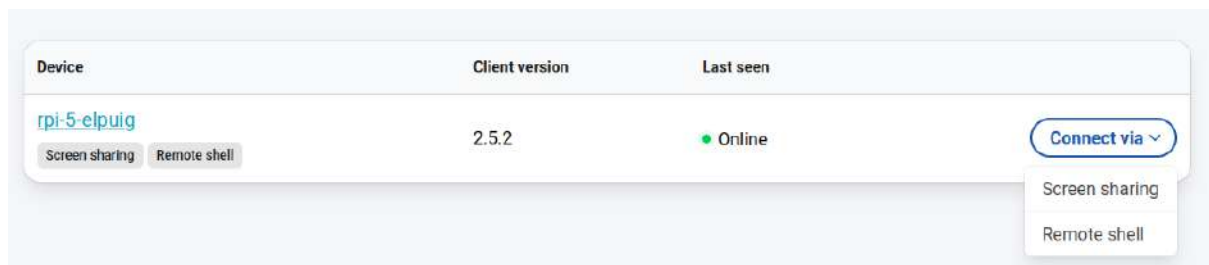


Hecho lo anterior, le damos un nombre a nuestra RPI.

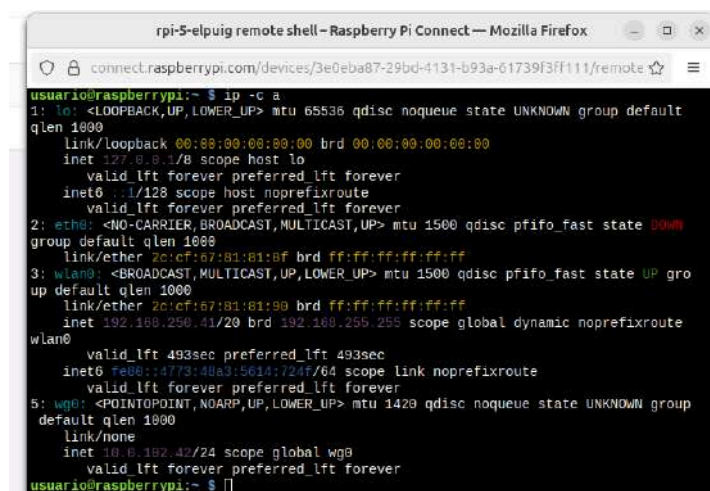


Navegador local

Ahora, para acceder al escritorio de nuestra RPI, debemos visitar la página connect.raspberrypi.com en cualquier navegador de cualquier cliente e iniciar sesión con nuestra cuenta de RPI. Hecho esto, veremos nuestro dispositivo y las diferentes formas de acceder a él.



Acceso al escritorio remoto



Acceso remoto a la shell

A la hora de conectar desde un ordenador que no forma parte de la misma red que la RPI, se hará uso de un relay para la conexión y eso podría ralentizar el establecimiento de la sesión. Si hay algún medio que los una directamente (LAN, VPN...) la conexión será directa entre los dos y la sesión se establecerá más rápidamente.

Otras opciones

GUI

En la GUI de RPI 5 podemos habilitar o deshabilitar este servicio. También es posible activar/desactivar los dos métodos de compartición disponibles. Por último, también encontramos la opción de desvincular este dispositivo de nuestra cuenta de RPI ID creada anteriormente. Hay un acceso que nos permite hacer todo esto en la parte superior derecha a la izquierda del icono de *bluetooth*.

CLI

Habilitar/deshabilitar servicio → **rpi-connect on/off**

Activar/desactivar compartición de pantalla → **rpi-connect vnc on/off**

Activar/desactivar acceso de shell → **rpi-connect shell on/off**

Eliminar vínculo del dispositivo con la cuenta de RPI ID → **rpi-connect signout**

Habilitar acceso remoto siempre → **loginctl enable linger**. Esta opción permite que podamos acceder a través de este visor a la RPI 5 independientemente de si esté la sesión iniciada o no del usuario con el que activamos rpi-connect. Sin esta opción, si el usuario que activó rpi-connect no tiene la sesión abierta, no podríamos acceder.

3.25 Instalación del emulador Mesen2 y pruebas de rendimiento

3.25.1 Instalación del emulador en la RPI 5

Para ARM64 solo disponemos, en el repositorio oficial de github, de las build con **NET8**. Para que este ejecutable funcione tendremos que instalar en la RPI 5 (RPI OS Desktop 64):

- **NET8**
- **SDL2**

1. Instalación del binario (Mesen2)

Para ello nos bajamos el zip correspondiente y lo descomprimos en un directorio.

```
usuario@raspberrypi:~/mesen2 $
usuario@raspberrypi:~/mesen2 $ ls -lh
total 15M
-rw-r--r-- 1 usuario usuario 15M May 16 16:24 'Mesen (Linux - ubuntu-22.04-arm - clang).zip'
usuario@raspberrypi:~/mesen2 $
usuario@raspberrypi:~/mesen2 $ unzip 'Mesen (Linux - ubuntu-22.04-arm - clang).zip' && rm 'Mesen (Linux - ubuntu-22.04-arm - clang).zip'
Archive:  Mesen (Linux - ubuntu-22.04-arm - clang).zip
  inflating: Mesen
usuario@raspberrypi:~/mesen2 $
usuario@raspberrypi:~/mesen2 $ ls -lh
total 27M
-rwxr-xr-x 1 usuario usuario 27M May 16 14:24 Mesen
```

2 Instalación de SDL2

Para ello ejecutamos, **sudo apt-get install libsdl2-2.0-0**.

3 Instalación de NET8

Para ello vamos a realizar la instalación manual del binario para **linux** arm64. Acto seguido, deberemos instalar las dependencias requeridas por la librería con *apt*.

3.1 Binario

El binario que nos vamos a descargar será la versión para desarrolladores ya que también incluye la librería para tiempos de ejecución de programas hechos con la misma. Para ello nos descargamos el archivo **tarball** para **linux arm64** de la página web de Microsoft y lo descomprimos en un directorio.

```
usuario@raspberrypi: ~/NET8 $  
usuario@raspberrypi:~/NET8 $ tar xzf dotnet-sdk-8.0.409-linux-arm64.tar.gz && rm dotnet-sdk-8.0.409-linux-arm64.tar.gz  
usuario@raspberrypi:~/NET8 $  
usuario@raspberrypi:~/NET8 $ ls -lh  
total 102K  
-rwxr-xr-x 1 usuario usuario  67K Apr 15 19:28 dotnet  
drwxr-xr-x 3 usuario usuario  4.0K Apr 15 19:38 host  
-rw-r--r-- 1 usuario usuario  1.1K Apr 15 19:14 LICENSE.txt  
drwxr-xr-x 6 usuario usuario  4.0K Apr 18 18:42 packs  
drwxr-xr-x 3 usuario usuario  4.0K Apr 18 18:42 sdk  
drwxr-xr-x 3 usuario usuario  4.0K Apr 18 18:42 sdk-manifests  
drwxr-xr-x 4 usuario usuario  4.0K Apr 18 18:42 shared  
drwxr-xr-x 3 usuario usuario  4.0K Apr 18 18:42 templates  
-rw-r--r-- 1 usuario usuario   93K Apr 15 19:14 ThirdPartyNotices.txt  
usuario@raspberrypi:~/NET8 $
```

Además de esto, también será necesario añadir al PATH la ruta a este directorio con el fin de poder utilizar la herramienta **dotnet**. Esto último es opcional, pero si será importante exportar una nueva variable llamada **DOTNET_ROOT** con la ruta a este mismo directorio. Será esta variable la que usen los programas, que utilizan esta librería, con el fin de encontrarla (p ej.: Mesen2).

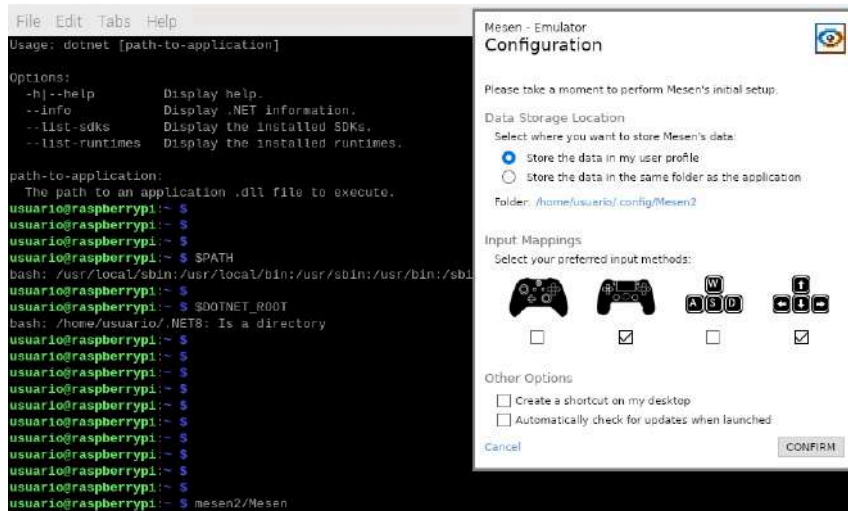
```
export DOTNET_ROOT=/home/usuario/.NET8  
export PATH=$PATH:$DOTNET_ROOT
```

3.2 Librerías

Otro paso importante será instalar las siguientes librerías, necesarias para ejecutar NET8 en sistemas basados en Debian 12.

```
root@raspberrypi:/home/usuario/.dotnet#  
root@raspberrypi:/home/usuario/.dotnet# apt install libc6 libgcc-s1 libgssapi-krb5-2 libicu72 libssl3 libstdc++6 zlib1g --yes
```

Si todo ha ido bien, ya podremos ejecutar el emulador Mesen2.



3.25.2 ROMs a usar para las pruebas

Vamos a crear un directorio en el servidor que contendrá una serie de ROMs que poder utilizar para las pruebas más adelante. Utilizamos las recomendaciones del [siguiente foro](#) para la obtención de ROMs redistribuidas de manera legal o bien que han sido desarrolladas por desarrolladores o comunidades dan permiso para la descarga gratuita de una versión total o *demo* de su videojuego para NES.

- [D-pad hero 1 y 2](#). Son una serie de juegos de ritmo basados en el popular **Guitar-Hero**.
- [Lizard \(demo\)](#). Lizard es un juego de plataformas el cual ofrece hasta seis personajes (lagartos) distintos cada uno con habilidades diferentes. Según los diferentes entornos que encontramos durante la aventura, debemos escoger uno u otro.
- [Faux Game Co.](#) Juego de plataformas hard-core
- [The Wit.nes](#). The Wit.nes nos ofrece un personaje con el cual nos podemos mover libremente por un entorno en el que podemos realizar hasta 22 puzzles diferentes.
- [Tapeworm disco puzzle \(demo\)](#). Tapeworm disco puzzle es un juego de puzzles *grid-based* (rejillas).
- [Street Fighter II Nostalgic Edition \(demo\)](#) Este juego es una adaptación a NES del clásico juego **Street Fighter II** para la SNES.
- [Alwa's Awakening \(demo\)](#)
- [Super Tilt Bro](#). Super Tilt Bro es un juego basado en la saga de Super Smash Bros.
- [From Below](#). From Below es un juego basado en el famoso juego originado para GB llamado Tetris.
- [Light From Within \(demo\)](#). Light From Within es un juego de acción y aventura que está basado en el juego The Legend of Zelda para NES.
- [Steel Legion \(demo\)](#). Es un juego de plataformas 2D que junta el género de los juegos Metroid y Castlevania (Metroidvania).
- [Old Towers](#). En este juego, disponemos de una jugabilidad lineal donde debemos escalar una serie de torres mediante una jugabilidad simple pero limitada.

3.25.3 Creación de directorio temporal con las ROMs del supermicro en la RPI

Para las pruebas del emulador vamos a necesitar las ROM del servidor, por lo tanto vamos a copiarlas con **scp** a un directorio local.

```

usuario@raspberrypi:~/mesen2-roms $ scp fixmaster@19.8.192.43:/home/fixmaster/mesen2/roms/* ./
AlwaysAwakening_demo.nes          100% 512KB 697.8KB/s
Light_From_Within_Demo_V4.8B.nes  100% 512KB 1.5MB/s
SteelLegion_Demo_1.3.nes          100% 517KB 1.8MB/s
StreetFighterII_NE_(DEMO_by_Parisoft).nes 100% 1024KB 2.3MB/s
Super_Tilt_Bro_(E).nes            100% 512KB 1.8MB/s
TapewormDemo8.nes                 100% 64KB 306.8KB/s
Upadhero.nes                      100% 256KB 818.8KB/s
Upadhero2.nes                     100% 256KB 811.5KB/s
From_Below_2020_09_16_v_1.8.8.nes 100% 40KB 193.1KB/s
lizard_demo.nes                   100% 512KB 1.8MB/s
oldTowers_1.1.nes                 100% 256KB 808.6KB/s
streamerz-v02.nes                 100% 128KB 606.1KB/s
thwilt.nes                         100% 40KB 192.8KB/s
usuario@raspberrypi:~/mesen2-roms $
usuario@raspberrypi:~/mesen2-roms $ ls
AlwaysAwakening_demo.nes  From_Below_2020_09_16_v_1.8.8.nes  oldTowers_1.1.nes  'Street Fighter II NE (DEMO by Parisoft).nes'  thwilt.nes
Upadhero2.nes             Light_From_Within_Demo_V4.8B.nes  SteelLegion_Demo_1.3.nes  'Super Tilt Bro_(E).nes'
Upadhero.nes              lizard_demo.nes                    streamerz-v02.nes      TapewormDemo8.nes

```

3.25.4 Ejecución de ROMs y apunte de problemas encontrados

A la hora de ejecutar las ROM del catálogo de la imagen anterior, el único que dio problemas fue **The Wit.nes**. Durante su ejecución se avistaron un par de problemas gráficos y, por ello, creemos que es mejor no incluirlo en el repositorio final. Todos los demás juegos funcionaron perfectamente a 50-60 FPS con el emulador Mesen2.

3.26 Montaje del amplificador MAX98357A y configuración del software de Adafruit

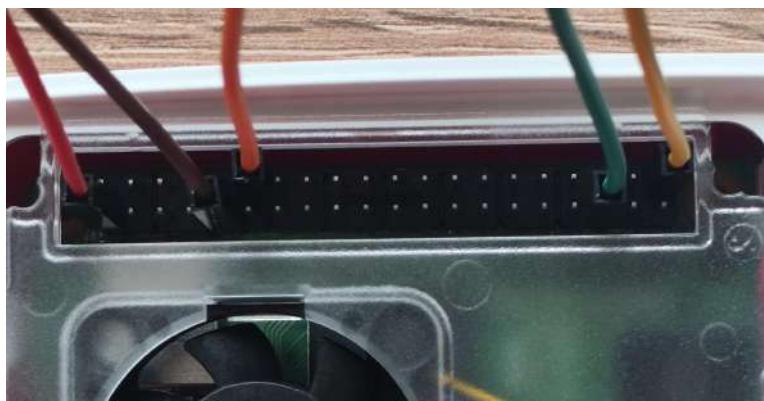
3.26.1 Preparación del hardware

Antes de comenzar a instalar el software necesario, vamos a montar y a hacer las conexiones eléctricas necesarias según el diagrama realizado en el apartado **3.21**. Este diagrama lo realizamos basándonos en esta misma guía, y lo incluimos mucho antes en la memoria, ya que creemos que es un aspecto importante para la realización del diseño 3D posterior a dicho apartado.

De igual manera, las conexiones de audio para la salida por la interfaz I2S en la RPI serán siempre las mismas al disponer solo de un único conjunto de pines I2S. Obviamente, las de alimentación pueden variar, ya que se dispone de varios pines distribuidos en el *header* para este propósito.



Amplificador MAX98357



Header GPIO de la Raspberry Pi 5

3.26.2 Instalación y configuración del software

1. Entorno virtual

Para ello, el primer paso será crear un entorno virtual con **python3** y acceder al mismo, comenzando por la creación de dicho entorno.

```
root@raspberrypi:/home/usuario#  
root@raspberrypi:/home/usuario# python -m venv env --system-site-packages  
root@raspberrypi:/home/usuario#
```

El comando anterior nos ha creado el directorio **env** en el directorio actual. Para acceder a él ejecutamos **source** seguido de la ruta, dentro de env, al ejecutable **activate**.

```
root@raspberrypi:/home/usuario#  
root@raspberrypi:/home/usuario# source env/bin/activate  
(env) root@raspberrypi:/home/usuario#  
(env) root@raspberrypi:/home/usuario#  
(env) root@raspberrypi:/home/usuario#
```

2. Preparación del script instalador

Para ello tendremos que instalar el paquete **adafruit-python-shell** usando el gestor de paquetes de python, **pip3**. (Comando: **pip3 install adafruit-python-shell**). Hecho esto, nos descargamos el script de instalación de la [siguiente URL](#) y lo ejecutamos con **sudo -E env PATH=\$PATH python3 i2samp.py**

```
(env) root@raspberrypi:/home/usuario# sudo -E env PATH=$PATH python3 i2samp.py
```

3. Instalación y configuración del regulador de volumen

Durante la ejecución, decidimos continuar en el primer prompt para realizar la instalación. Hecho esto, se nos da la opción de activar el *playback sound*. Con esto se lanzará un

segundo script en cada arranque que reproduce un *white noise* casi imperceptible que permite así que el dispositivo no se desacople cómo periférico de salida del sistema operativo RPI OS. Sin esto, posiblemente se perdería la conexión con el conversor DAC. Finalmente, reiniciamos.

```

Always be careful when running scripts and commands
copied from the internet. Ensure they are from a
trusted source.

Do you wish to continue? [y/n] y

Checking hardware requirements...

Adding Device Tree Entry to /boot/firmware/config.txt

Commenting out Blacklist entry in /etc/modprobe.d/raspi-blacklist.conf
Configuring sound output
Installing aplay system unit

You can optionally activate '/dev/zero' playback in
the background at boot. This will remove all
popping/clicking but does use some processor time.

Activate '/dev/zero' playback in background? [RECOMMENDED]
[Y/n] y
Created symlink /etc/systemd/system/multi-user.target.wants/aplay.service - /etc/sy
stemd/system/aplay.service.

All done!

Enjoy your new $productname!
REBOOT NOW? [Y/n] █
Activate Windows

```

Después del reinicio, volvemos a entrar en nuestro entorno virtual y lanzamos el script de nuevo. En esta segunda iteración realizaremos una prueba de audio, cuyo resultado fue erróneo. De igual manera, el amplificador ya debería estar configurado. Para comprobar esto, vamos a reiniciar una segunda vez para así poder habilitar el regulador de volumen de la GUI y de la CLI (alsamixer).

```

Set your speakers at a low volume if possible!
Do you wish to test your system now? [y/n] y
Testing...
XDG_RUNTIME_DIR (/run/user/1000) is not owned by us (uid 0), but by uid 1000! (This
[could e.g. happen if you try to connect to a non-root PulseAudio as a root user, o
ver the native protocol. Don't do that.])
ALSA lib pcm_dmix.c:999:(snd_pcm_dmix_open) unable to open slave
speaker-test 1.2.8

Playback device is default
Stream parameters are 48000Hz, S16_LE, 2 channels
WAV file(s)
Playback open error: -524,Unknown error 524

All done!

Enjoy your new $productname!
REBOOT NOW? [Y/n] █
Activate Windows

```

Hecho el reinicio, podremos usar el regulador de volumen del entorno gráfico PIXEL (Figura 3) y también el regulador de volumen de la CLI llamado **alsamixer** (Figura 4).

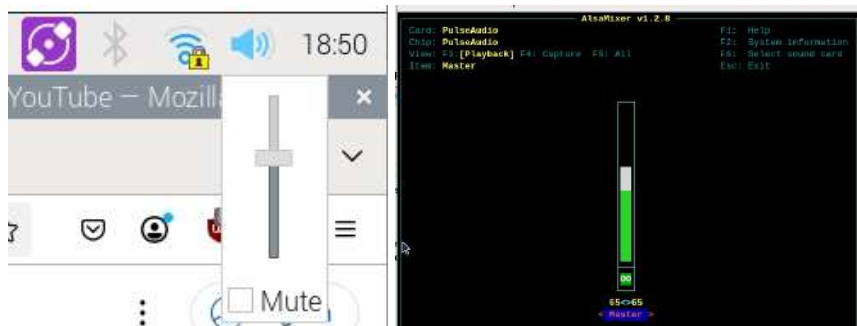


Figura 3

Figura 4

Cabe recalcar, de cara al uso de este periférico, que la interfaz I2S de la RPI solo soporta ficheros de audio **STEREO**. Si algún audio no se escucha en algún momento dado, una posibilidad sería que ese fichero sea de tipo **MONO**.

Si se llega a escuchar audio, pero en malas condiciones... Se podrían probar las siguientes soluciones de la [guía oficial de Adafruit](#).

3.27 Instalación y pruebas del programa Input-Remapper

3.27.1 Instalación

Para ello podemos usar el paquete *deb*ian *latest* de la última release del repositorio oficial e instalar dicho paquete con APT.

```
usuario@raspberrypi:~$ wget https://github.com/sezanzeb/input-remapper/releases/download/2.1.1/input-remapper-2.1.1.deb && sudo apt install ./deb --yes && rm input-remapper-2.1.1.deb
```

```
usuario@raspberrypi:~$ apt search remapper
Sorting... Done
Full Text Search... Done
input-remapper/now 2.1.1 all [installed,local]
  A tool to change the mapping of your input device buttons
```

3.27.2 Formas de uso

1. Ejecución de la GUI

Podemos hacer esto con el comando **input-remapper-gtk**. Al hacerlo se nos pide autenticación por parte del usuario *root* ya que se debe acceder al directorio */dev/input*. Para evitar esto también podríamos usar **sudo**.

2 Crear maps en la GUI

En input-remapper, cada dispositivo de entrada puede tener diferentes presets. Cada preset puede tener diferentes entradas o combinaciones de entradas que proporcionan una cierta salida (maps).

1. Para ello, abrimos el programa con **input-remapper-gtk** en la CLI.
2. En la sección **Devices** seleccionamos nuestro dispositivo y creamos un nuevo **preset**.
3. En la sección *input*, podremos crear diferentes *maps* y asignarles su correspondiente input/inputs y la salida (output) que tendrán.

Nota.: Para asignar un input o un output en un *map* podremos utilizar *record* y pulsar dicha tecla/botón o bien podemos escribir la palabra clave de dicha tecla/botón. Para saber estas palabras clave, simplemente podemos pulsar dicha tecla/botón y nos saldrá el nombre de ese input en un recuadro inferior derecho del menú.

3 Iniciar/detener los mapeados de un dispositivo en la GUI

Dentro del *preset* del dispositivo que queramos, hacemos clic en **Apply** para *inyectar* los diferentes mapeos realizados. Podemos detener la inyección con **Stop**.

4 Ubicación de los archivos de configuración generados por la GUI para un cierto dispositivo

Los presets creados para un dispositivo se encuentran en

```
~/.config/input-remapper-2/presets/<device name>/<preset name>.json
```

Podemos encontrar el `<device-name>` con el comando `sudo input-remapper-control --list-devices`

2.3 Cómo activar dichos maps desde la CLI sin abrir la GUI de Input-Remapper

Para operar este programa desde la interfaz en modo texto disponemos de un servicio llamado **input-remapper** el cual dispone de una unidad de **systemd**.

Para hacer uso de los diferentes comandos tenemos que iniciar el servicio con

```
systemctl restart input-remapper
```

Inyectar todos los presets de todos los dispositivos configurados:
`input-remapper-control --command autoload`

Inyectar un preset de un dispositivo específico:

```
input-remapper-control --command start --device <dispositivo>  
--preset <preset-sin-extensión>
```

Detener la inyección de un dispositivo específico:

```
input-remapper-control --command stop --device <dispositivo>
```

Detener la inyección de todos los dispositivos:

```
systemctl stop input-remapper
```

Nota.: Iniciada la inyección desde la CLI de algún preset de algún dispositivo, esta prevalece hasta que

- Reiniciamos
- Detenemos la inyección manualmente
- Detenemos el servicio con `systemctl`

3.28 Desarrollo del lanzador de emuladores

3.28.1 Descarga de ROMs a utilizar

Vamos a crear un directorio en el servidor que contendrá una serie de ROMs que poder utilizar para el funcionamiento de nuestro producto/script final. Estas serán movidas más adelante a un repositorio en el servidor que almacenará dichas ROMs para luego poder ser descargadas por la RPI. Al ser un emulador de NES el escogido, se apreciará que las ROMs siguientes tienen la extensión **.nes**.

Utilizamos las recomendaciones del [siguiente foro](#) para la obtención de ROMs re-distribuidas de manera legal o bien que han sido desarrolladas por desarrolladores o comunidades dan permiso para la descarga gratuita de una versión total o *demo* de su videojuego para sistemas NES.

- [D-pad hero 1 y 2](#). Son una serie de juegos de ritmo basados en el popular **Guitar-Hero**.
- [Lizard \(demo\)](#). Lizard es un juego de plataformas el cual ofrece hasta seis personajes (lagartos) distintos cada uno con habilidades diferentes. Según los diferentes entornos que encontramos durante la aventura, debemos escoger uno u otro.
- [Faux Game Co.](#) Juego de plataformas hard-core
- [The Wit.nes](#). The Wit.nes nos ofrece un personaje con el cual nos podemos mover libremente por un entorno en el que podemos realizar hasta 22 puzzles diferentes.
- [Tapeworm disco puzzle \(demo\)](#). Tapeworm disco puzzle es un juego de puzzles *grid-based* (rejillas).
- [Street Fighter II Nostalgic Edition \(demo\)](#) Este juego es una adaptación a NES del clásico juego **Street Fighter II** para la SNES.
- [Alwa's Awakening \(demo\)](#)
- [Super Tilt Bro](#). Super Tilt Bro es un juego basado en la saga de Super Smash Bros.
- [From Below](#). From Below es un juego basado en el famoso juego originado para GB llamado Tetris.
- [Light From Within \(demo\)](#). Light From Within es un juego de acción y aventura que está basado en el juego The Legend of Zelda para NES.
- [Steel Legion \(demo\)](#). Es un juego de plataformas 2D que junta el género de los juegos Metroid y Castlevania (Metroidvania).
- [Old Towers](#). En este juego, disponemos de una jugabilidad lineal donde debemos escalar una serie de torres mediante una jugabilidad simple pero limitada.

3.28.2 Programa de creación y ejecución de *macros*

3.28.21 Requisitos para el programa de macros a usar

Para escoger dicho programa analizamos los siguientes requisitos,

Que tenga documentación

Operable desde la consola (modo texto)

Disponible para **linux arm64**

- Repos
- Paquete deb suelto
- Código fuente para compilar
- Otros...

Que sea instalable y ejecutable en RPI OS Desktop 64 (Debian 12)

Que las macros se puedan componer de diferentes comandos de dicho programa

Que disponga de un ejemplo de macro que, al lanzarse, inyecta

- Una única tecla
- Una combinación de teclas

Que lo anterior aplique a la práctica en RPI OS Desktop

3.28.22 Programas analizados

Los cuatro programas disponibles para sistemas basados en Debian 12 para ARM que analizamos mediante estos requisitos fueron

- **x-macro**
- **SikuliX**
- **xdotool**
- **Xnee**

Después de analizar cada una de estas herramientas, vimos que ninguna de ellas nos ofrecía la funcionalidad que esperamos o bien no se ejecutaba correctamente. **x-macro** fue descartado debido a que carecía de documentación, **sikuli** fue descartado ya que este no era capaz de ejecutarse. Finalmente, probamos las herramientas **xdotool** y **Xnee** o “**cnee**”, ambas para la CLI, las cuales dieron el mismo resultado. Este resultado fue el efecto de que, a veces, se ejecutaban las macros creadas en las ventanas seleccionadas, pero no siempre. Al tener un funcionamiento bastante intermitente, decidimos descartar ambas opciones.

3.28.23 Programa escogido

Debido a que no hemos encontrado un programa funcional, al final hemos decidido utilizar la función de macros integrada con el programa Input Remapper. Con esto conseguiremos el mismo resultado solo con la diferencia de que el usuario deberá recordar las combinaciones de botones que realizan dichas acciones que hayamos implementado.

3.28.3 Configuración de Input Remapper

Este programa debe servirnos para configurar dos *presets* (conjunto de maps) para nuestro *gamepad*. Tendremos un *preset* a usar durante la ejecución del lanzador y otro *preset* a utilizar durante la ejecución del propio lanzador.

3.28.31 Mapeos y macros teórico

Dentro del emulador Mesen2 (Preset 1 del *gamepad*)

Acción	Mesen2	DS4 (DualShock 4)
Selección de archivo	Ctrl + o	L1 + R1
Pausar el juego / Salir de selección de archivo	Esc	R1 + Izquierda
Recargar la ROM	Ctrl + Shift + R	J _L + R1
Salir de la ROM	Ctrl + F	J _R + L1
Salir del emulador	Ctrl + °	Botón DS4
Mute	Ctrl + A	R1 + Derecha
Subir / bajar volumen	Ctrl + + / Ctrl + -	R1 + Arriba / Abajo
Seleccionar archivo de guardado	1 - 8	L1 + Cruceta (4) / Botones (4)
Guardar estado de la ROM	S	R1 + X
Cargar estado de la ROM	L	R1 + Triangulo
Gamepad		
Cruceta	Flechas	Cruceta
Select	N	Share
Start	M	Options
B	Ñ	X
A	Enter	O

Dentro del menú del script (Preset 2 del *gamepad*)

Acción	SO	DS4
Situarse en la opción de derecha / izquierda	Flecha izquierda / derecha	Cruceta izquierda / derecha

Situarse en la opción superior / inferior	Flecha arriba / abajo	Cruceta arriba / abajo
Seleccionar una opción	Enter	O

La razón de tener dos mapeados, pese a que el primero podría servir para lo segundo, es para limitar al usuario a solo las acciones previstas durante la ejecución del script. Dejar el primer mapeado supondría lo mismo que permite el segundo más muchas otras combinaciones que inyectarán una serie de teclas que podrían interrumpir el funcionamiento habitual (por ejemplo el L1 que inyecta un *escape*).

3.28.32 Configuración de ambos presets

Para conocer los códigos que hacen referencia a una tecla en este programa, podemos usar el comando `input-remapper-control --symbol-names`. Esto nos mostrará todos los símbolos disponibles y su nombre.

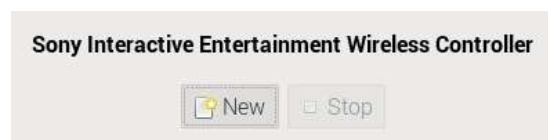
Preset 1 (Mesen2)

Vamos a mostrar cual es el proceso que seguimos para crear un *map*. Pues, todos los demás presentes fueron configurados de la misma manera.

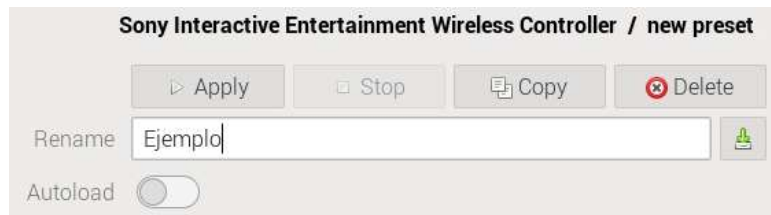
1. Iniciamos el programa con privilegios de *root* para que el programa pueda acceder al directorio de los periféricos de entrada (**/dev/input**)
2. Seleccionamos nuestro dispositivo, en nuestro caso *Sony Interactive Entertainment Wireless Controller*



3. Creamos un nuevo *preset* con el botón *New*



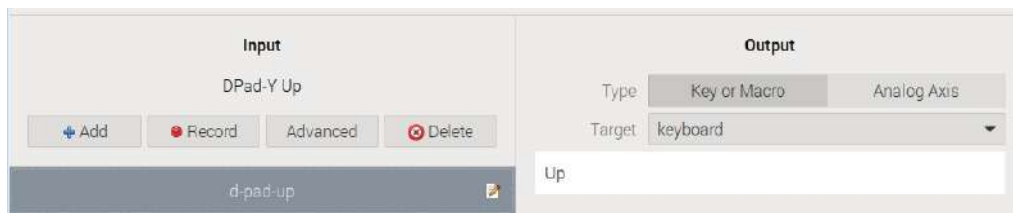
4. Dentro de dicho *preset*, le asignamos un nombre en la caja *rename*



5. Podremos crear un nuevo *map* en la *input* con el botón *add*



6. Lo único que nos quedará será asignar un nombre, grabar una entrada e introducir cuál será la tecla de salida



Los códigos de cada tecla pueden ser vistos en un cuadro de texto dinámico en la parte inferior derecha de dicha sección (**Figura X**), o bien con el comando en la terminal **input-remapper-control --symbol-names** (muestra toda la lista de *keycodes* disponible) (**Figura Y**).

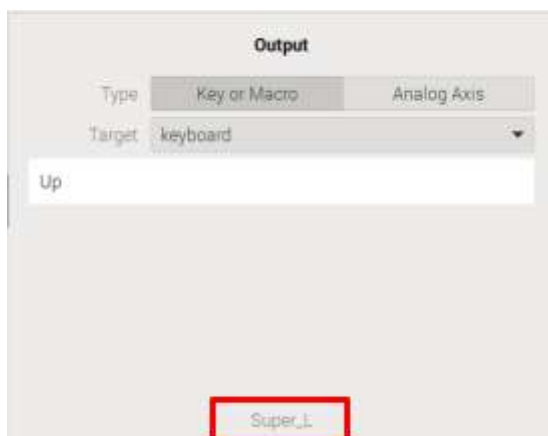


Figura X

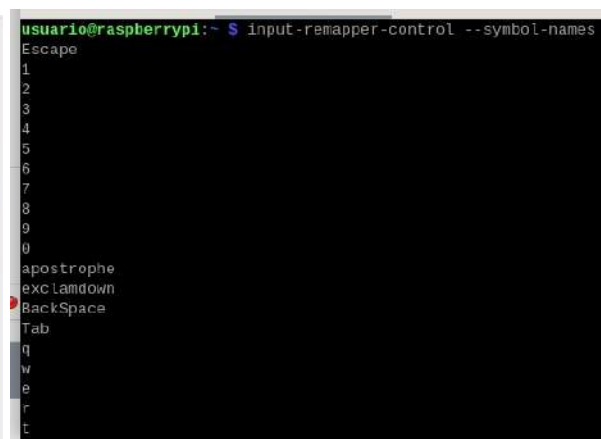


Figura Y

Preset 2 (Lanzador)

Para el lanzador creamos un segundo *preset*, conjunto de *mapeos*, en el mismo *gamepad*. Como ya se mencionó anteriormente, la razón de esto es debido a que tendremos una configuración en el entorno de nuestro script y otra configuración durante la ejecución del emulador.

3.28.4 Programa de gestión de ventanas desde la CLI

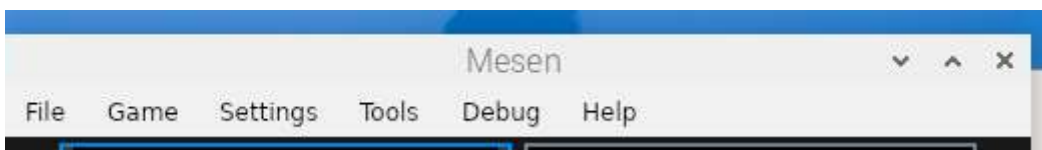
3.28.41 Requisitos a analizar

- Que disponga de documentación
- Instalable en sistemas **linux arm64**
 - Repositorios
 - Paquete debian suelto
 - Código fuente para compilar
 - Otros...
- **Que el programa sea capaz de identificar ventanas activas en la GUI**
- **Que el programa sea capaz de hacer *focus* a dichas ventanas con un comando en la terminal**

Con estos dos últimos requisitos nos aseguramos que dicho programa nos permite centrar la atención en una ventana de un programa en la UI del SO y realizar acciones sobre dicha ventana con comandos desde la CLI.

3.28.42 Alternativa escogida

A raíz del análisis de los requisitos anteriores, creemos que la mejor opción es el programa **wmctrl**. Pues, este dispone de varias opciones que le permiten identificar una ventana en el entorno gráfico. Por ejemplo, podemos usar el comando **wmctrl -a string** para dicho propósito. Este comando pondrá el *focus* en la ventana que contenga **string** en su título. El título de una ventana hace referencia a lo siguiente.



Donde "Mesen" es el título de la misma.

3.28.5 Funciones específicas de los scripts

3.28.51 Script lanzador

En nuestro caso, las funciones específicas serían una versión todavía más abstracta de lo que sería un pseudo-código. Este último, el pseudo-código el cual adjuntamos en el anexo, sería una versión menos abstracta de lo que vamos a citar ahora y más aproximada al código real.

1 Menú principal con desplazamientos

- Menú que consiste de diferentes entradas por las cuales te puedes desplazar con las flechas de arriba / abajo
- Se puede proceder a dichas entradas presionando la tecla enter
- Se ofrece la opción de cancelar la cual permite volver al menú padre del que estamos en ese momento

1.1 Opción 1: Lanzar el emulador

- Cuando lanzamos el emulador,
 - Se asigna la configuración correcta de Input Remapper para Mesen 2. Es decir, cambio del *preset* del lanzador al de Mesen2
 - Se ejecuta Mesen2 cómo un proceso independiente
 - Se lanza un script (script auxiliar 1) en segundo plano que espera a que el proceso principal del emulador termine
 - Se pone el focus en el emulador Mesen2
 - Se pone en *fullscreen* el emulador Mesen2
 - Se termina la ejecución del script lanzador
- Una vez cerramos el emulador, el script en segundo plano (auxiliar) lo detecta y...
 - Intercambia los mapeados de Input Remapper (Mesen2 → Lanzador)
 - Inicia la ejecución del script lanzador
 - Lo pone en *focus*
 - Lo pone en pantalla completa
 - Se termina la ejecución de este script en segundo plano (auxiliar)

1.2 Opción 2: Descarga de ROMs del repositorio del *supermicro*

- El usuario accede a una lista con las ROMs disponibles en el momento de entrar a dicha opción mediante la generación dinámica de dichas opciones del menú (en base a lo existente en el repositorio del servidor).
- El usuario se desplaza y...
 - Selecciona un juego
 - Si está descargado
 - Menú de confirmación
 - y o no (independientemente)→ Redirección a un segundo menú con un mensaje de error → Tecla para continuar al menú anterior (lista con las ROM)
 - Sino está descargado
 - Menú de confirmación

- y: Se descarga mediante scp del repositorio del servidor → Se introduce en un directorio fijado por código para las ROMs → Se genera de nuevo la lista de ROMs u opciones del menú
- n: Tecla para continuar al menú anterior (lista de ROM)
- Selecciona la opción de volver atrás (cancelar) → Vuelve al menú principal

Cabe recalcar que, al principio, decidimos implementar la sección de descarga de forma estática. Es decir, unas entradas en el menú que se añaden a mano y sin ningún tipo de dinamismo. Decidimos cambiar de idea por el trabajo que supone el hecho de añadir más juegos al servidor, pues, se tendría que modificar dicha lista también dentro del código. Hacemos adjunto en el anexo una primera y segunda versión del código del lanzador (siendo la segunda la que contiene el efecto dinámico).

1.3 Opción 3: Acceso al directorio de ROMs local para la revisión y eliminación

- Se accede a una lista con una serie de entradas generadas dinámicamente según la cantidad de ficheros (ROMs) contenidas en el directorio (el número de entradas es igual al múltiplo de ficheros en este directorio)
- El usuario se desplaza por el menú
 - Selecciona una ROM que quiere borrar → Submenú de confirmación
 - Quieres borrar?
 - y → Se borra → Se genera de nuevo el listado el listado anterior
 - n → Se genera de nuevo el listado anterior
 - Selecciona botón de volver atrás (cancelar) → Se vuelve al menú principal

Miscelánea

- Durante el arranque de la máquina (.bashrc) , se ejecuta un script (script auxiliar 2) que hace lo siguiente:
 - Se inicia el script lanzador
 - Se pone en pantalla completa
 - Se le hace *focus*
 - Se inicia el mapeado con Input Remapper para este entorno

3.28.52 Script auxiliar 1: Control del proceso del emulador Mesen2

- muestra la lista de procesos de todos los escritorios del sistema (comando **ps -e**)
- mientras la salida de **ps -e | grep Mesen** de algo mayor a 1
 - do
 - espera un segundo para la siguiente iteración
 - Verifica otra vez si **Mesen** sigue vivo.
 - done
- intercambia mapeado activo con input remapper
 - detener *device* DS4
 - volver a lanzar el *device* DS4 con el *preset* "Lanzador"

- ejecutar script lanzador
- poner en focus y en pantalla completa con **wmctrl**
- terminar la ejecución del script actual

Lo que ocurre después del **done** es posible cuando salimos del bucle. Para salir del bucle, se debe cumplir la condición de que el emulador ya no tenga un proceso vivo (ha terminado la ejecución y el usuario estaría viendo el escritorio).

3.28.53 Script auxiliar 2: Arranque al iniciar la sesión en la terminal

- ejecutar script lanzador
- poner en focus y en pantalla completa con **wmctrl**
- activar mapeado del lanzador con input remapper
 - iniciar servicio de input-remapper
 - lanzar el *device* DS4 con el *preset* "Lanzador" (configuración del *mapper* para dicho entorno)

3.28.6 Explicación de algunos bloques del código fuente final del lanzador

A continuación explicamos algunos bloques en el código del lanzador que, a simple vista e incluso con los comentarios, por naturaleza requieren de una explicación un poco más detallada que una simple línea de texto.

En la siguiente figura se aprecia el bloque de código que utilizamos para generar el menú principal de nuestro lanzador. Esta estructura será en la que se basan los otros menús de los cuales disponemos. A nivel resumido, esta sección define una serie de variables que podrán ser usadas como valores para los parámetros del comando **dialog**, este comando te permite componer menús desplazables fácilmente. **El array OPTIONS** contiene las opciones en un formato específico que le permite a **dialog** disponer las opciones en una columna. Ya por último, el comando **dialog** utiliza estos parámetros para definir el menú como se aprecia en la figura adyacente. Cabe recalcar que el array **options** se ha llamado con **@** para hacer referencia a todos los índices que contenga un elemento (en este caso solo son el 0, 1 y 2).

```

1 # Variables estándar para todos los menus
2 HEIGHT=15
3 WIDTH=40
4 CHOICE_HEIGHT=4
5 BACKTITLE="Mesen2 Launcher"
6 MENU="Escoge una de las siguientes opciones:"
7
8 # Variable a cambiar durante la ejecución del script
9 TITLE="Menú principal"
10
11 OPTIONS=(1 "Lanzar Mesen2"
12          2 "Descarga de ROMs"
13          3 "Borrado de ROMs")
14
15 # Con el comando dialog, se genera un menu con los titulos y dimensiones
16 # de las variables anteriores mas las opciones del array OPTIONS
17 CHOICE=$(dialog --clear \
18               --backtitle "$BACKTITLE" \
19               --title "$TITLE" \
20               --menu "$MENU" \
21               $HEIGHT $WIDTH $CHOICE_HEIGHT \
22               "${OPTIONS[@]}" \
23               2>&1 >/dev/tty)

```

Figura A



Figura B

Este bloque de código (**Figura C**) pertenece a la sección de descarga de ROMs del servidor. Esta bloque se encarga primero de almacenar en una variable la lista de ROMs en el directorio remoto del servidor. Para ello se accede al servidor con su dirección igualada a la variable `$srv` y se realiza un comando entre comillas. Hacer esto último supone

- Entrar a la máquina
- Ejecutar un comando
- Salir de la máquina

Con esto nos basta para obtener la stdout del directorio con ls en formato de 1 columna y almacenarla en una `$list`.

Después de ello, realizamos un proceso similar pero para el almacenaje en una variable del recuento de roms en el directorio remoto. Para ello hacemos exactamente lo mismo pero transmitimos la salida estándar a `wc` para que cuente las líneas con la opción `-l`.

Estas dos variables nos sirven más adelante para lo mostrado a continuación (**Figura D**)

```
2) # ***** Descarga de ROMs del repositorio del servidor *****
while true
do

    # Salida con solo los nombres de las ROMs en el dir. remoto en una sola columna
    list=$(ssh $srv "ls --format=single-column $ruta_dir_remoto")

    # Recuento de los ficheros del directorio remoto con las ROM
    nlines=$(ssh $srv "ls --format=single-column $ruta_dir_remoto" | wc -l)
```

Figura C

Con las dos variables anteriores, nos es posible crear un array que contenga cómo elementos las ROMs disponibles en un cierto momento en el servidor. Para ello, se inicializa un for que itera tantas veces cómo ROMs haya en el directorio. Por cada iteración una variable auxiliar, que se sobrescribe en cada iteración, almacena la ROM correspondiente mediante la extracción de la línea N con **sed** de listado de ROMs en formato mono-columna. Es este valor de **aux** luego introducido cómo elemento en el índice N (según la iteración) dentro del array. Cómo incrementamos N en cada iteración, nunca vamos a sobrescribir un elemento del array y, por lo tanto, los estaremos almacenando.

```
# ----- Array con solo las ROMs disponibles en el directorio remoto (servidor) -----
roms_d=()
for i in `seq 1 1 $nlines` # Iteramos N veces el número de ROMs que haya
do
    # Var que se sobrescribe en cada iteración y extrae la ROM correspondiente
    # al número de iteración de la lista en formato de una columna
    aux=$(echo "$list" | sed -n "$i p")

    # Se igual la ROM correspondiente cómo elemento N del array de ROMs. En la
    # siguiente iteración, cambiamos el índice y mantenemos valores anteriormente asignados
    roms_d[$i]=$aux
done
```

Figura D

Para generar el menú con las opciones dinámicas, debemos generar una variable que contiene, en una única línea, el nombre de cada ROM con el formato adecuado para **dialog**. Podemos hacer esto con un bucle for, iterando a través de los índices que contengan elementos en el array (para ello especificamos la ! antes del array).

Para dar este formato esperado por dialog a cada ROM impresa con **echo**, aprovechamos la variable inicializada para generar algo similar a **N ROM**. La variable **\$i** se inicializará en **1** ya que el primer índice del array que contiene elementos es el 1.

```
# ----- Variables para el 1er dialog de selección de ROM -----
TITLE="Descarga de ROMs del repositorio"

# Iteramos según los índices del array roms_d que contengan elementos e
# imprimos en cada iteración el formato esperado por dialog para cada opción
# del menú
LISTADO_ROMS=$(for i in ${!roms_d[@]}; do echo "$i ${roms_d[$i]}"; done)

# Primer dialog; Selección de ROM
CHOICE=$(dialog --clear \
    --backtitle "$BACKTITLE" \
    --title "$TITLE" \
    --menu "$MENU" \
    $HEIGHT $WIDTH $CHOICE_HEIGHT \
    $LISTADO_ROMS 2>&1 >/dev/tty)
```

Figura E

El siguiente bloque de código es el que nos permite salir del bucle que ejecuta la sección de descarga. Esto lo hacemos mediante la verificación de la variable que almacena nuestra elección en el dialog. Si esta es igual a "" (**nada**), entonces significa que hemos escogido el botón "**Cancel**" del menú y que, por lo tanto, queremos volver atrás. En ese caso, se corta el bucle, en caso contrario, no se hace nada (se sigue).

```
# Revisar si el usuario ha cancelado en el 1er dialog
if [ "$CHOICE" == "" ] # El botón cancelar en dialog devuelve un "" (nada)
then
    break
fi
```

Figura F

Con la selección de un juego en la **Figura E**, se nos llevará a un segundo menú que nos pide confirmación (**Figura F**). Según la opción escogida en este segundo menú, se hará una cosa u otra según el condicional "**case**" siguiente.

```
# Segundo dialog: Confirmación de la descarga
case $YN in
    1)
        # Se llama el nombre de la ROM seleccionada mediante su número de index, obtenido
        # con dialog, en el array y se verifica su existencia con grep
        existe=$(ls --format=single-column $ruta_dir_local | grep ${roms_d[$CHOICE]})
        if [[ $existe ]] # Si la variable contiene texto... (ha habido salida)
        then
            clear
            read -p "Ya está instalado, continuar..."
        else
            clear
            scp $srv:$ruta_dir_remoto/${roms_d[$CHOICE]} $ruta_dir_local
            read -p "Se ha instalado ${roms_d[$CHOICE]}, continuar..."
        fi
        ;;
    2)
        clear
        read -p "Continuar..."
        ;;
    *)
        clear
        read -p "Continuar..."
        ;;
esac
```

Figura F

Si queremos borrar, valor 1, primero se verifica si el juego existe con un grep del nombre de la ROM seleccionada en el directorio local de ROMs de la RPI 5. Pues, será este donde almacenemos los juegos y evitaremos, en la medida de lo posible, repetir datos.

Si el juego ya existe, se nos da la opción de continuar y de volver a iterar en el *while* de la sección de descarga (escoger otro juego). Si el juego NO existe (no está instalado), entonces se procede a la descarga del mismo desde el repositorio remoto al directorio local. Hecho esto, se ejecuta un read que nos imprime un mensaje y espera a que el cliente lea el resultado de su acción y quiera continuar.

En caso de haber escogido el valor 2 (No) u cualquier otro (esto sería el “Cancel” de **dialog**), se nos da la opción de seguir en el bucle y volver al menú de selección de ROMs (el cual se volvería a generar según el estado actual en ese mismo momento).

[3.28.7 Backups del código fuente del lanzador](#)

[rsync](#)

Para ello, hemos creado un script (almacenado en **/home/usuario/Backups**) que contiene la siguiente línea

```
rsync -av /home/usuario/Scripts/*  
fixmaster@10.0.102.43:/home/fixmaster/backup-lanzador/
```

Esto sincroniza el contenido del directorio con el código fuente en un directorio de resguardo de nuestro servidor. Acto seguido, introducimos una línea en el crontab de **usuario** que lanza este script cada 30 minutos. Con esto sincronizamos el contenido frecuentemente y lo mantenemos actualizado en el directorio del servidor.

[Deja-dup](#)

Deja-dup lo instalamos mediante **flat hub** a nivel de usuario con **flatpak install <URL> --user**.

Este otro programa está configurado para que los backups, automáticos o no, del directorio del código fuente **/home/usuario/Scripts** se dejen en el directorio **/home/usuario/Backups/backup-lanzador-deja_dup**

[3.28.8 Automatización del arranque del lanzador](#)

[3.28.81 Ejecución en cada *log in* de la terminal](#)

En el fichero **~/.bashrc** introducimos al final del mismo

```
/home/usuario/Scripts/aux/aux2-arranque.sh
```

[3.28.82 Ejecución de la terminal en cada arranque](#)

Lo anterior contado solo es válido al lanzar la terminal manualmente. Si queremos que el lanzador se ejecute con solo encender la RPI 5, entonces tendremos que ejecutar la terminal en el arranque. Para ello se realizaron dos intentos, ninguno de ellos con éxito.

Primera forma: directorio **/etc/init.d/**

1. Creamos un script que contenga una línea la cual ejecuta la **terminal**

- Creamos [terminal.sh](#) que contenga...
 - línea de ejecución del binario de la terminal en RPI OS en segundo plano (&)

Para conocer el ejecutable binario de un programa que solo lanzamos desde la UI

En **/usr/share/applications** encontramos las referencias de la UI. Dentro de estas encontramos **lxterminal.desktop**. Si hacemos un `cat` de este fichero y filtramos por la líneas que contengan “Exec”, encontramos una línea, variable, que muestra el string que introducimos en la terminal para ejecutar este programa.

Si hacemos `which` de este string, **which lxterminal**, vemos que el programa está en **/usr/bin/lxterminal**.

2. Copiamos el script al directorio **/etc/init.d**

3. Damos permisos de ejecución al script

4. Modificamos el script y añadimos los siguientes comentarios al inicio del script

Estos comentarios añaden unas dependencias necesarias para identificar el orden de arranque de este script en conjunto con otros presentes en **/etc/init.d**.

```
# /etc/init.d/terminal.sh
### BEGIN INIT INFO
# Provides:          terminal.sh
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start daemon at boot time
# Description:       Enable service provided by daemon.
### END INIT INFO
```

4. Aplicamos cambios con **sudo update-rc.d terminal.sh defaults**

5. Reiniciamos

Esto no surtió efecto, por ende probamos otra forma.

Segunda forma: unidad de systemd

De esta forma, podemos utilizar el proceso estándar de **systemd** para el control de programas que se ejecutan durante el arranque. Con la unidad a configurar, vamos a conseguir lanzar un programa justo después de que el SO haya cargado todo lo demás.

1. Crear una unidad de systemd

- Crear **/lib/systemd/system/lxterminal.service**
- Añadir el siguiente texto dentro del fichero

[Unit]

Description=Inicio de la terminal una vez cargado el SO

After=multi-user.target

[Service]

Type=idle

ExecStart=/usr/bin/lxterminal > /home/usuario/logs/arranque-terminal.log 2>&1

[Install]

WantedBy=multi-user.target

Con **After=multi-user.target** y **Type=idle**→ Significa que solo se cargará el programa en “**ExecStart**” cuando el entorno de multi-usuario esté disponible (una vez haya cargado todo). La línea **ExecStart** muestra la ruta al ejecutable que queremos lanzar y la redirección siguiente (>) permite que systemd redirija la **stdout** del script a un fichero log.

Nota.: “**2>&1**” se encargará de combinar la salida estándar y la salida de errores en una sola. Básicamente, conseguimos que ambas salidas acaben en nuestro log.

- Asignar permisos **644** a **lxterminal.service**

2. Configurar systemd

- Ejecutamos
 - **systemctl daemon-reload**
 - **systemctl enable lxterminal.service**

La ventaja de usar unidades de systemd es que, con esto, podremos controlar cuando queremos que se arranque la terminal en el inicio con solo habilitar o deshabilitar dicho servicio. Podríamos, también, verificar esto con **systemctl status** (cómo haríamos con cualquier otro servicio que tenga configurada la unidad de systemd).

3. Verificar efecto después de un init 6

Cómo resultado no obtuvimos lo esperado. Para tratar de solucionarlo, lo que hicimos fue crear un script que ejecuta el binario **/usr/bin/lxterminal** y que sea ese script el que lanza el servicio configurado. Pese a esto, no conseguimos hacer que funcione.

3.28.9 Pruebas del lanzador a nivel de usuario

3.28.91 Pruebas

Las pruebas que realizamos, con éxito, fueron las siguientes:

- Abrimos la terminal → Se ejecuta el lanzador en pantalla completa
- Podemos navegar por el lanzador con el *gamepad* con las *macros* para este entorno
- Lanzar emulador
 - Revisar
 - Fullscreen
 - Revisar efecto de cada una de las macros configuradas
 - Selección de archivo
 - Pausa
 - Recargar rom
 - Salir rom
 - Salir emulador
 - Mute
 - Subir/bajar volumen
 - Seleccionar archivo de guardado
 - Guardar estado de la rom
 - Cargar estado de la rom
 - Gamepad
 - Cruceta
 - Select
 - Start
 - B
 - A
- Que al cerrar el emulador con la macro correspondiente
 - Se abre el lanzador en el menú principal
 - Que hemos cambiado al preset de lanzador (para ello tocar alguna macro propia del emulador y revisar que no ocurre nada)
- Borrado de ROMs
 - Borrar todas
 - Revisar mensaje de error cuando no hay más roms
- Instalación de ROMS
 - Instalar todas una a una
 - Revisar que se anuncia para cada ROM instalada que ya se encuentra instalada
- Borrado de ROMs
 - Revisar que vuelven a aparecer aquí

3.28.92 Problemas encontrados durante las pruebas

Problema: El emulador no se pone en pantalla completa

Motivo → La nueva terminal que ejecuta Mesen2 también recibe el mismo título que la ventana del emulador. Por ende, **wmctrl** se confunde a la hora de hacer focus a la ventana “Mesen” (pues, tenemos dos ventanas con mismo título).

Solución 1:

Para ello tuvimos que encontrar un elemento diferenciador para estas dos ventanas sin contar el título. Con el comando **wmctrl -r Mesen -l** se muestran las instancias (ventanas) que se asocian con dicho título. Al hacer esto durante la ejecución de ambas ventanas, vemos dos registros con mismo título pero diferente dirección hexadecimal de la ventana. Es esta dirección la que podemos utilizar con **wmctrl** para poner focus de la siguiente manera

wmctrl -i -a 0x<dirección_hexadecimal>

Para saber qué dirección es la que se corresponde con la de Mesen2, podemos simplemente lanzar de manera manual dicho programa y revisar la salida del comando **wmctrl -r Mesen -l**. En este caso, nos sale un único registro con una dirección el cual nos dice que ha salido lanzado por la máquina **N/A**. Si hacemos esto mismo pero también con la ventana de xterm abierta, veremos que la ventana xterm con nombre ‘Mesen’ ha sido lanzado por la máquina **raspberrypi**. Podremos usar esto para detectar en el script ventana cuál es la correcta y hacerle focus mediante referencias numéricas con la opción **-i** (por defecto las referencias son con strings).

La aproximación realizada es (dentro de la opción 1)

// Se almacena la dirección hex de la ventana del emulador

```
mesenw_hexd=$(wmctrl -r Mesen -l | grep N/A | cut -d " " -f 1)
```

// Se hace referencia a la ventana con dicha dirección y se le hace focus y pantalla completa

```
wmctrl -i -a $mesenw_hexd; wmctrl -i -r $mesenw_hexd -b toggle,fullscreen
```

Resultado:

Cómo resultado obtuvimos el mismo efecto, el efecto de pantalla completa seguía alternando entre dichos programas. Por ende, probamos otra cosa.

Solución 2:

Esta consiste en que se lance de igual manera la terminal xterm que ejecuta mesen2 pero que la ejecución sea realizada mediante un script con un nombre diferente a “Mesen”. Esto hará que dicha ventana de xterm adquiera el nombre del script y, por lo tanto, se difiera de la ventana original (gracias al título). He aquí una aproximación de la solución.

```
// Lanzar el emulador mediante el script aux3, importante el & para liberar la terminal
actual y que ejecute las instrucciones siguientes
xterm -e /ruta/a/script-aux-3 &
sleep 2 // Damos tiempo al inicio de la ejecución
wmctrl -a Mesen // Focus
wmctrl -r Mesen -b toggle,fullscreen
```

Resultado: El efecto sigue siendo el mismo. Por ende, vamos a probar otra cosa más.

Solución 3:

En este caso, vamos a utilizar el comando **nohup**. Este nos permite lanzar un programa desde la terminal y que, después de haber cerrado la terminal, este permanezca abierto. He aquí el código

```
// Lanzar el emulador mediante el comando nohup. Con “&> /dev/null &” elimina la salida
del comando y, además, libera el comando en la terminal
nohup /ruta/a/Mesen &> /dev/null &
```

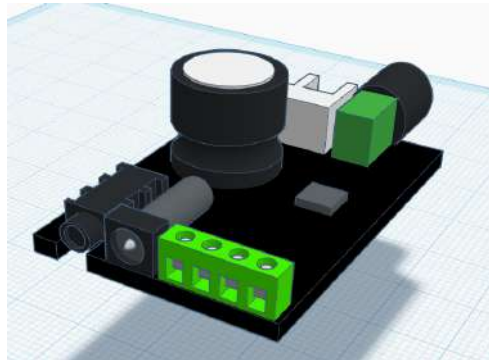
Resultado: Con esto, ya hemos conseguido centrar la atención en solo la pantalla correcta al prescindir de una terminal nueva para ejecutar dicho programa. (conseguido).

3.3 Diseño 3D de la consola

Para no tener que enviar la caja de raspberry al entregar el producto, decidimos hacer nuestra propia carcasa, para esto utilizamos la herramienta Tinkercad, la cual usaremos para crear los modelos 3D necesarios.

Ya que no teníamos mucha idea de esta herramienta, tuvimos que mirarnos un cursillo gratuito que encontramos en youtube, una vez ya supimos más o menos como manejar Tinkercad, comenzamos realizando pruebas de componentes que descartamos anteriormente, para así pillar más práctica.

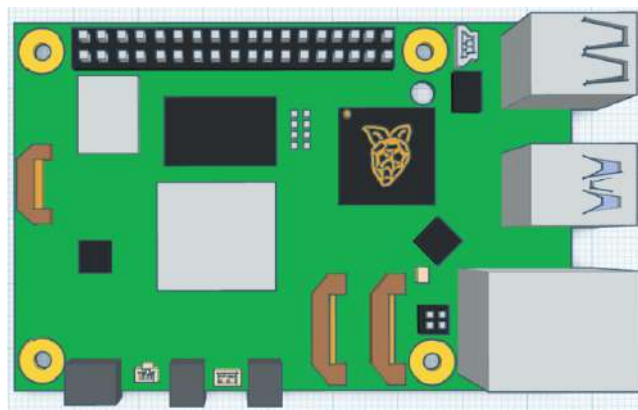
3.31 Práctica Tinkercad



Una vez ya consideramos que hemos practicado lo suficiente, empezamos modelando los componentes que utilizaremos, de esta manera, una vez ya todos modelados, será fácil adaptar el tamaño de la carcasa, ya que todo tiene que estar con una medida real, utilizamos milímetros para hacer el modelo.

3.32 Componentes de la Consola en Tinkercad

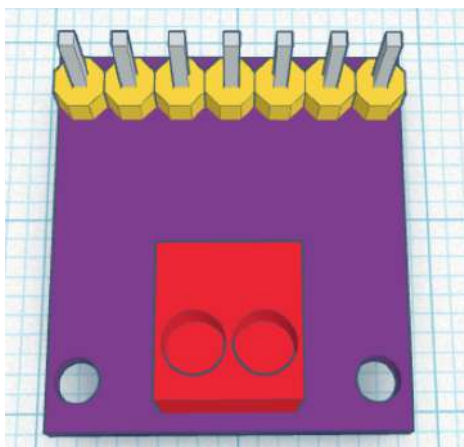
Raspberry Pi 5



Modelo 3d Raspberry Pi 5

85mm x 56mm. (Figura 1)

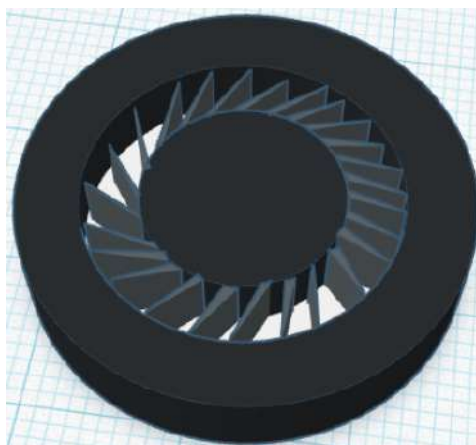
Amplificador



Amplificador MAX98357A

17,84mm x 19,50mm (**Figura 2**)

Altavoz



Altavoz "Weewooday"

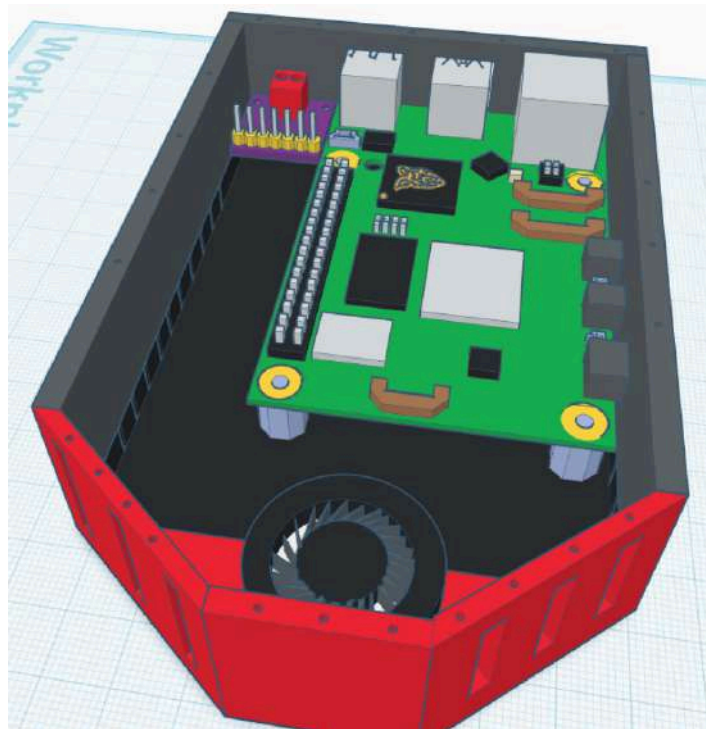
28mm de diametro (**Figura 3**)

3.33 Carcasa de la consola a base

Ahora que ya hemos realizado los modelos, realizaremos el modelo de la carcasa basándonos en **Figura 1**, **Figura 2** y **Figura 3**.

Usamos estos para no equivocarnos con el tamaño de la carcasa.

Carcasa FIX Master

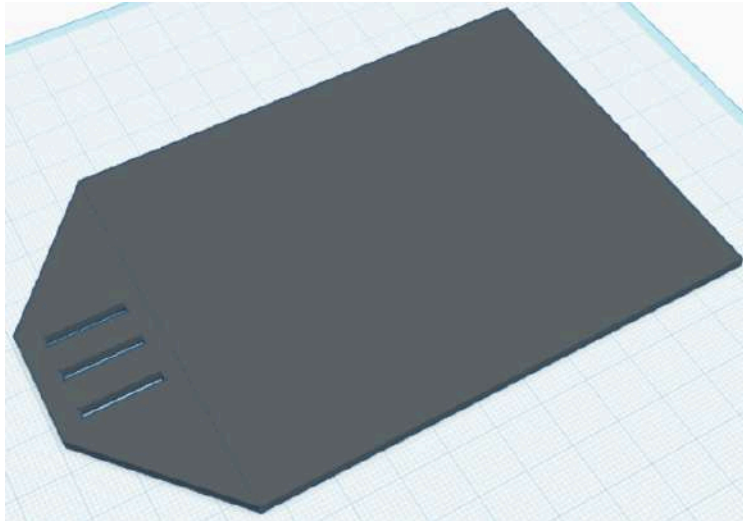


132,28mm x 84,40mm

Funciones de la carcasa:

- Contiene agujeros en el modelo para que salga el aire caliente.
- Contiene espacio para imanes, de esta forma podremos abrir y cerrar la carcasa.
- Contiene espacio para los puertos
- Contiene piezas de plástico debajo de donde colocamos los tornillos. Fundiremos estas piezas e introduciremos los tornillos para perforar el plástico para introducir las tuercas, de esta forma los tornillos se quedarán fijos, aparte, las piezas también sirven como soporte para los componentes.

Carcasa (parte de arriba)

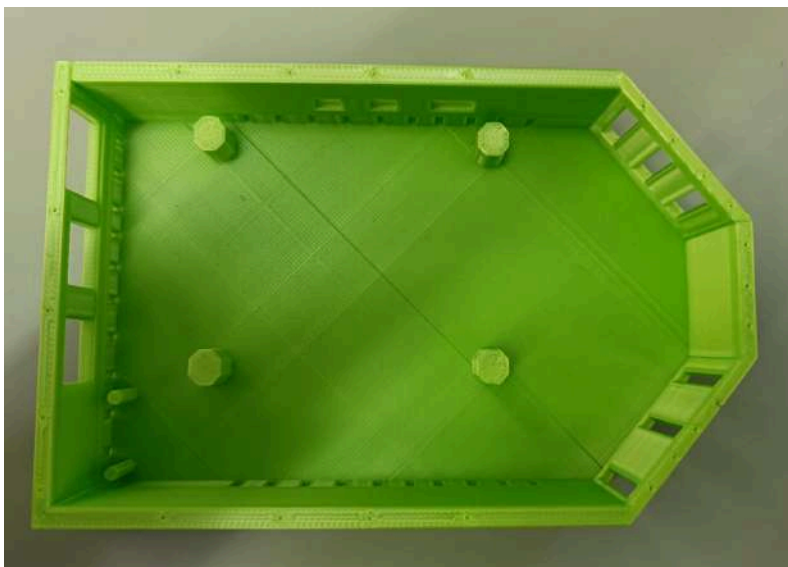


132,28mm x 84,40mm

3.34 Imprimir modelo 3D

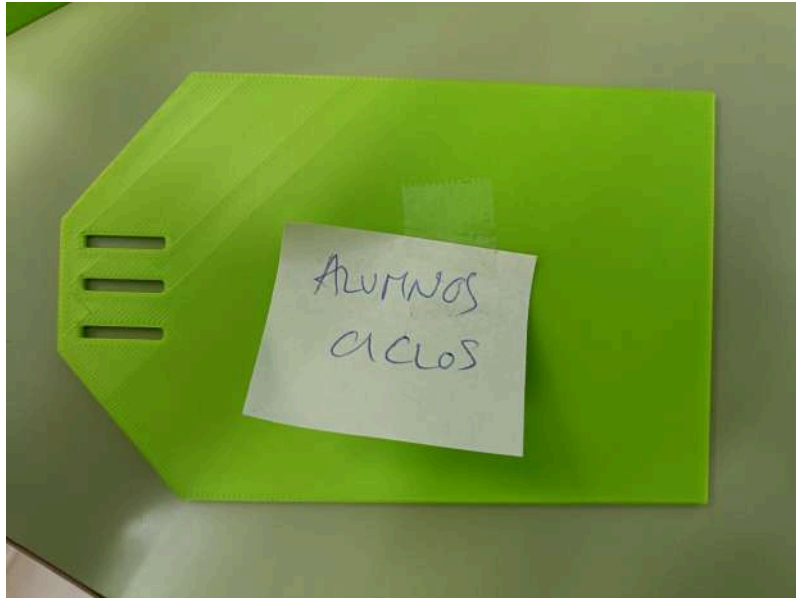
Ahora que ya tenemos la carcasa para la raspberry pi, mandaremos estos 2 últimos modelos a imprimir en 3d. Hablamos con el Jaime Morcillo para que nos preste sus impresoras 3d, él accedió y tuvimos que esperar alrededor de 2 horas para obtener el resultado.

Imagen del modelo en la vida real:



Ya que no teníamos tiempo, pedimos a Jaime que dejará el modelo en la recepción. Al día siguiente lo recogimos.

Parte superior de la carcasa:



Desgraciadamente, debido al poco tiempo que nos quedaba y nuestra falta de experiencia, tuvimos que hacer las medidas algo deprisa y al parecer hay partes que no encajaron muy bien a pesar de que en el visualizador encaja con sobresaliente.

Además, necesitábamos tornillos métrica 2.5, los cuales son bastante inusuales y poco vendidos, por lo que encontrarlos iba a ser bastante difícil.

Ya que tomamos mal las medidas, las salidas de los cables no encajaban, al parecer los agujeros eran demasiado pequeños, pasa lo mismo con el hueco de los imanes, fueron más pequeños de lo que esperábamos y no se pudo aprovechar.

Por descuido nuestro, los soportes del amplificador eran demasiado pequeños, por lo que apenas resisten un dedo de fuerza, son muy delicados y se pueden romper muy fácilmente.

4. Conclusión y cierre del proyecto

4.1 Conclusiones generales de proyecto

Este proyecto nos ha ayudado a ver el alcance real de nuestros objetivos, permitiéndonos aplicar en un entorno práctico los conocimientos adquiridos durante nuestra formación. Gracias al desarrollo de este trabajo, hemos podido enfrentarnos a situaciones reales que nos han obligado a buscar soluciones efectivas, trabajar en equipo y gestionar los recursos de manera eficiente.

Desde un punto de vista académico, el proyecto ha reforzado nuestra capacidad de análisis, síntesis y planificación. También ha favorecido la integración de conceptos teóricos en la práctica, lo que ha hecho que el aprendizaje sea mucho más significativo.

4.2 Consecución de los objetivos

La gran mayoría de los objetivos se han podido cumplir, desgraciadamente no todos, como por ejemplo el manual o la carcasa de la consola, sin embargo, estamos contentos de los logros que hemos conseguido en este proyecto y que el cambio de idea inicial no nos haya afectado demasiado.

4.3 Valoración de la metodología y planificación

Durante el desarrollo del proyecto hemos seguido bastante bien la planificación inicial, aunque ha sido necesario realizar algunos cambios por imprevistos. La metodología que usamos fue la **waterfall** la cual creemos que fue la adecuada, pero se tuvo que adaptar en algunos momentos para resolver problemas. Estos ajustes permitieron mantener el rumbo y asegurar el éxito del proyecto.

4.4 Visión de futuro

Nos hubiera gustado trabajar en la idea de la consola portátil, la cual no pudimos por falta de tiempo y experiencia, si tuviéramos que continuar el proyecto, creemos que la consola portátil sería el siguiente paso, junto a una investigación de emulación en hardware más profunda.

5. Bibliografía

En este documento encontraremos enlaces a páginas web o videos que nos han ayudado a hacer posible este proyecto → [Enlaces de intereses.](#)

6. Anexo

- Enlace a la primera entrega de la memoria → [1ra Versión](#)
- Justificación al cambio de la idea principal → [Cambio de Rumbo](#)
- Algunas etapas del diagrama de gantt → [Gantt1](#), [Gantt2](#),
- Programa lanzador de emuladores
 - [Pseudocódigo](#) (utilizado para el desarrollo del código fuente, puede contener incoherencias respecto al código real)
 - Código fuente → [Enlace a la carpeta de Drive](#)
 - Script principal
 - Primera versión → **lanzador.sh** (Descarga de ROMs estática)
 - Segunda versión → **v2-lanzador.sh** (Descarga de ROMs dinámica)
 - Scripts auxiliares
 - aux1 → Control del proceso Mesen2
 - aux2 → Arranque del lanzador
- Cuaderno de Bitácora → [Cuaderno de Bitácora](#)