UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Instalación y configuración de un servidor para administrar varios sitios web

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Ibáñez Zarzo, Raul

Tutor/a: Pons Terol, Julio

CURSO ACADÉMICO: 2023/2024

Resum

El treball va consistir a crear i configurar un servidor per allotjar un conjunt de llocs web. L'objectiu va ser posteriorment poder instal·lar o migrar pàgines web que funcionin amb WordPress.

Es va instal·lar i configurar aquest servidor amb l'última versió d'Ubuntu LTS (Long Time Support) amb totes les característiques necessàries per al correcte funcionament dels llocs web. A continuació, es realitzà la migració d'un lloc WordPress que actualment estigui funcionant en un altre servidor.

Per a l'administració dels llocs a nivell del servidor, s'estudiaren els panells de control més utilitzats i es va procedir a la instal·lació d'un d'ells.

Posteriorment, es realitzà una anàlisi del manteniment necessari al servidor i als llocs web perquè tinguin un funcionament estable

Paraules clau: Servidor web; panell de control; instalació sistema operatiu; migració web; WordPress; Ubuntu LTS;

Resumen

El trabajo consiste en crear y configurar un servidor para albergar un conjunto de sitios web. El objetivo ha sido posteriormente poder instalar o migrar páginas web que funcionen con WordPress.

Se ha instalado y configurado este servidor con la última versión de Ubuntu LTS (Long Time Support) con todas las características necesarias para el correcto funcionamiento de los sitios web y luego se ha realizado la migración de un sitio WordPress que actualmente esté funcionando en otro servidor.

Para la administración de los sitios a nivel del servidor se han estudiado los paneles de control más utilizados y se ha procedido a la instalación de uno de ellos.

Posteriormente se realizó un análisis del mantenimiento necesario en el servidor y los sitios web para que tengan un funcionamiento estable.

Palabras clave: Servidor web; panel de control; instalación sistema operativo; migración web; WordPress; Ubuntu LTS;

Abstract

The task involves creating and configuring a server to host a set of websites. The objective has subsequently been to be able to install or migrate web pages that work with WordPress.

This server has been installed and configured with the latest version of Ubuntu LTS (Long Time Support) with all the necessary features for the correct functioning of the websites and then the migration of a WordPress site that is currently running on another server has been carried out.

For the administration of the sites at the server level, the most used control panels have been studied and one of them has been installed.

Subsequently, an analysis of the necessary maintenance was carried out on the server and the websites so that they have stable operation.

Key words: Web Server; Control panel; operating system installation; web migration; WordPress; Ubuntu LTS;

Índice general

Índice de figuras

CAPÍTULO 1
Introducción

Hoy en día, con la evolución constante del mundo de la informática, se aplica en todos ámbitos, y en particular la web es una de las aplicaciones más utilizadas. En el tema de las páginas web y en general Internet, incluso a la gente más familiarizada con este aspecto de la informática le parece un mundo abrumadoramente extenso, debido a su infinidad de posibilidades.

Las páginas web están a la orden del día en empresas, organizaciones o incluso en particulares que quieren tener su propia web por motivos varios. Crear, mantener y gestionar estas páginas web es fundamental para que estas lleguen al máximo rango de población objetivo posible y nos causen los más mínimos problemas.

Este trabajo se va a centrar en la creación y configuración de un servidor capaz de alojar un conjunto de páginas web, siempre con la vista a futuro de poder añadir más de estas, con la suficiente capacidad y con la configuración adecuada para que sean seguras. Nos vamos a enfocar en las web que funcionan mediante WordPress que a día de hoy es uno de los miles de métodos que existen para la creación de esta herramienta tan útil para el mundo actual

1.1 Motivación

Con la vista en el mundo laboral tras mi paso por la universidad y conociendo que uno de los problemas que sufren las empresas hoy en día son los ciberataques, me he decidido a hacer un proyecto para cubrir una de las aristas de este complicado mundo de la ciberseguridad, la seguridad web.

He aprovechado la oportunidad del trabajo de fin de grado para ponerme en la piel de una empresa la cual tiene unos sitios web albergados en un servidor totalmente desactualizado y vulnerable a ataques debido a que el mantenimiento de este no está en regla.

Mi finalidad es indagar en este sector y encontrar una solución para tener tu sitio web actualizado y en un estado correcto para garantizar su funcionamiento y mantenimiento.

1.2 Objetivos

El objetivo principal de este proyecto es crear un servidor robusto y seguro capaz gestionar páginas web y probarlo haciendo una migración de una web existente en un servidor desactualizado. Para llevarlo a cabo se plantean los siguientes objetivos:

1. Crear y configurar un servidor: Se instalará y configurará un servidor con la última versión de Ubuntu LTS (Long Time Support) que nos proporcionará actualizaciones a largo plazo para poder garantizar la estabilidad del mismo con el paso del tiempo, que es un aspecto clave en el funcionamiento continuo de un sitio web.

2. Estudio de Paneles de control: Haremos un estudio de los paneles de control que tenemos actualmente en el mercado y cuál se adapta de mejor forma a nuestras necesidades. Los paneles de control facilitan las tareas de gestión de dominios, monitorización e instalación de aplicaciones y plugins.

3. Migración de sitios web: Se va a llevar a cabo la migración de un WordPress existente desde otro servidor a este nuevo entorno que vamos a crear , con todo lo que conlleva, transferir bases de datos, archivos y configuraciones.

4. Seguridad: Poner medidas de seguridad para hacer que nuestro servidor y página web sean lo más seguros posibles.

5. Estabilidad y mantenimiento: Haremos un estudio de qué requisitos de mantenimiento son necesarios para poder asegurar el funcionamiento del sitio web con el paso del tiempo. Este análisis se va a centrar en las actualizaciones de software, seguridad, copias y optimización.

   Siempre teniendo en cuenta que la creación y configuración de servidores web es un campo en constante evolución.

1.3 Impacto esperado

El impacto esperado para este proyecto es mejorar la calidad de vida en el trabajo de la persona encargada de gestionar este tipo de sistemas, además de conseguir tener una parte tan importante como es nuestra web totalmente configurada y segura. Con la utilización de nuevas tecnologías de gestión como puede ser un panel de control y dejando todo documentado y con un plan de mantenimiento, ahorraremos mucho tiempo y salud mental en el ámbito laboral ya que esto nos va a facilitar muchas tareas y vamos a prever muchos problemas.

1.4 Estructura de la memoria

Contexto Tecnológico

En este capítulo vamos a abordar la tecnología existente en el área que incumbe al proyecto, se va a realizar una pequeña crítica de las soluciones actuales, además de presentarse la propuesta de trabajo, diciendo qué mejoras y diferencias tenemos respecto a las tecnologías y métodos existentes.

Análisis del Problema

Aquí vamos a profundizar en el problema que vamos a estar resolviendo, se va a analizar la seguridad y se contemplan varias soluciones posibles para resolver el problema, y finalmente explicamos la solución elegida y por qué la hemos elegido.

Diseño de la solución

En este capítulo nos vamos a centrar en la solución propuesta, aquí vamos a detallar la arquitectura del sistema, describiremos los componentes utilizados, como el panel de control, la tecnología LAMP, el gestor de máquinas virtuales y una explicación breve de la gran mayoría de las herramientas como MySQL, PHP, WordPress, etc.

Implementación

Haremos una breve explicación de los pasos a seguir en el caso de que hubiéramos que poner en producción el servidor web creado.

Pruebas

En este capítulo vamos a documentar las pruebas que realizaremos al proyecto para verificar que todo está en correcto funcionamiento, con pruebas de funcionalidad, seguridad y rendimiento.

Conclusiones

En este apartado vamos a resumir los resultados del proyecto, teniendo en cuenta los objetivos iniciales y una reflexión final, además de alguna sugerencia o directriz para trabajos futuros o mejoras adicionales.

Bibliografía

Se presentan todas las fuentes bibliográficas consultadas y utilizadas durante el proyecto, facilitando la consulta de la información adicional.

CAPÍTULO 2
Contexto tecnológico

En la actualidad es normal el uso de sistemas de gestión de contenido como en nuestro caso WordPress[1] en un entorno LAMP. LAMP es un acrónimo que se refiere a un conjunto de tecnologías de software de código abierto que se utilizan comúnmente para desarrollar y desplegar aplicaciones web. LAMP significa:

Linux: Es el sistema operativo que sirve como base del stack[2].

Apache: Es el servidor web que maneja las solicitudes HTTP[3].

MySQL: Es el sistema de gestión de bases de datos relacional[4]. MySQL permite almacenar y gestionar datos de manera eficiente.

PHP: Es el lenguaje de programación que se utiliza para desarrollar las aplicaciones web[5].

Este tipo de herramientas suelen ser fundamentales para el desarrollo y mantenimiento de sitios web.

En lo que respecta a las versiones instaladas de cada uno de los componentes de un servidor LAMP (Linux, Apache, MySQL, PHP), es importante revisar que siempre están en la versión estable más actualizada posible para poder garantizar su estabilidad, seguridad y rendimiento. El hacer este tipo de mantenimiento garantiza un funcionamiento óptimo del sistema.

Con el tiempo la demanda de todo tipo de contenido en línea ha crecido exponencialmente, los sitios web se han convertido en sitios más complejos.

Los servidores web comerciales surgieron en los años 90 cuando Apache y Nginx hicieron muchas mejoras de rendimiento y escalabilidad en este tipo de servidores. En esa época los servidores web no mostraban nada mucho más complejo que texto plano, pero con el paso del tiempo se tuvieron que adaptar para poder generar contenido dinámico con la popularización de tecnologías como PHP o Node.js.

Con todo esto la seguridad se convirtió en algo importante a tener en cuenta por lo que estos servidores tuvieron que incorporar medidas para evitar ataques.

La evolución hacia la virtualización y la computación en la nube permitió una escalabilidad horizontal mucho más amplia.

A día de hoy los servidores web están implementando tecnologías de microservicios, contenedores y arquitecturas sin servidor. Todo esto hace que se pueda ofrecer una mayor flexibilidad, escalabilidad y eficiencia a la hora de hacer un despliegue y gestión.

2.1 Crítica al estado del arte

En la gran mayoría de los trabajos que se han realizado como TFG en la UPV sobre temas similares, he visto que muchos desarrollan una página web utilizando el gestor de páginas web WordPress, o securizando un sitio web ya implementado o en el caso de migrar un servidor solo hacen la migración, un ejemplo de migración es el trabajo "Plan de migración de un servidor web basado en Windows Server 2008 e IIS" [6] donde se realiza una migración del servidor y se pone en funcionamiento.

En cambio, mi trabajo no se enfoca en la parte del diseño y de la implementación web como muchos trabajos que se centran en este aspecto como el trabajo "Desarrollo completo de un sitio web con buenas prácticas" [7], sino que una vez hecho este trabajo de diseño y creación de la web, poder alojarla en un sitio seguro sin ningún tipo de fallo y proporcionando las herramientas necesarias para que su gestión sea sencilla y buscar la minimización de los problemas en el estado de producción del servidor y en su migración.

2.2 Propuesta

Lo que presentamos es una solución que no es innovadora como tal porque muchas de las partes están hechas en otros trabajos, pero nuestro objetivo es unificar y hacer una guía definitiva para poder hacer esta gestión de creación, configuración y plan de mantenimiento de una forma eficiente y segura, analizando por qué deberíamos hacer unas cosas y por qué no, además de dar una opción a mejorar el servicio que proponemos utilizando software más avanzados que exigen de un pago.

CAPÍTULO 3
Análisis del problema

Nos encontramos con un servidor web antiguo creado con un sistema de creación de máquinas virtuales llamado Bitnami[8] que aloja un WordPress.

Bitnami lo que hace es que te proporciona una máquina virtual ya configurada con un LAMP y con una instalación de WordPress limpia para que hagas lo que quieras. Con el paso del tiempo esta máquina ha sido olvidada y no se le ha dado el mantenimiento adecuado, al punto de que hasta actualizarla es una tarea complicada y sin mucho resultado.

Desde el punto de vista de la ciberseguridad, tener un servidor sin ningún tipo de mantenimiento en un estado de producción es un grave error el cual hay que solucionar.

Al no poder tener acceso a ningún tipo de parche de seguridad o de mantenimiento, el servidor se ha convertido en una posible puerta de entrada para aquellos atacantes que quieran comprometer la integridad y confidencialidad de los datos alojados en el servidor.

Toda esta falta de actualizaciones ha dejado al servidor en gran parte estancado y con ciertas limitaciones, que limitan tanto las funcionalidades disponibles para el usuario y además dificultan la integración de nuevas características u opciones que podrían mejorar la seguridad o la experiencia de usuario. Todo esto ha llevado a que actualizarlo sea una tarea complicada por la falta de herramientas en la propia máquina y la falta de documentación actualizada.

3.1 Análisis de la seguridad

El tener un servidor desactualizado, desde el punto de vista de un administrador de sistemas, es una puerta de entrada en nuestro sistema para cualquier ciberdelincuente que nos quiera hacer daño, ya que al no tener actualizaciones de seguridad, te expones a que puedan utilizar exploits conocidos para acceder. Por eso, el objetivo general de este proyecto es solucionar este problema y reforzarlo añadiendo medidas de seguridad extra.

Siempre teniendo en cuenta que conseguir una seguridad completa en cualquier servicio de informática es imposible, debido a que siempre hay algún fallo o modo de poder acceder, por eso nuestro trabajo como administradores de sistemas en este caso es minimizar esas puertas abiertas que hay en todo sistema informático para que sea más difícil acceder o casi imposible. Para ello haremos uso de plugins y configuraciones que nos van a permitir cerrar estas puertas abiertas, entre ello tenemos dobles factores de seguridad, restricciones de IPs y puertos y demás configuraciones que nos ayudarán a mantener nuestro servidor y página web lo más seguros posible.

3.2 Identificación y análisis de soluciones posibles

Una vez hemos visto el problema que queremos solucionar, he decidido poner sobre la mesa las diferentes opciones que tengo para dar una solución a este proyecto de la mejor manera posible.

3.2.1. Actualizar el servidor Bitnami y WordPress

La primera solución que planteo es la de conservar el servidor que ya tenemos y simplemente buscar la forma de actualizarlo y dejarlo de la mejor manera posible. Bitnami proporciona soporte de actualizaciones a sus máquinas virtuales, pero esta fue creada hace mucho y está desfasada por lo que luego de ver que la propia entidad que proporcionó el sistema le ha dejado de dar soporte, esta opción ya casi la dejo descartada. Pero como he comentado en la descripción del problema, actualizar un sistema antiguo como este a veces es un poco difícil ya que el servidor está estancado y muchas veces tenemos conflictos de versiones al intentar actualizarlo o al intentar aplicar parches de seguridad que crean un montón de conflictos y problemas de compatibilidad que comprometen la estabilidad del sistema.

3.2.2. Reimplementar el servidor

Esta es la opción desde el punto de vista académico y profesional más llamativa, ya que nos da opción a aventurarnos a confeccionar nuestro propio servidor y de esta forma poder instalar la última versión estable de cada uno de los componentes a nuestro gusto y utilizando las herramientas con las que nos sintamos más cómodos, ya que al fin y al cabo no hay mejor herramienta que la que ya conocemos.

Esta opción nos permite hacer una máquina nueva, la cual siempre intentaremos buscar la versión con mayor soporte al largo del tiempo. En este caso, habría que buscar las versiones de Ubuntu LTS, nuestra decisión ha sido la 22.04 LTS (Long Term Support). La elección de esta versión de sistema operativo es por el compromiso que tienen de ofrecer un soporte a largo plazo y actualizaciones de seguridad durante 5 años, lo que nos garantiza una mayor seguridad para nuestro sitio web pensando en el futuro. Además de ello, haremos la instalación mediante un panel de control, que nos ayudará al control de versiones de cada uno de los componentes de nuestro servidor, además de facilitarnos la tarea de configuración del mismo.

Con todo esto, una vez esté todo correctamente configurado, contemplaremos la opción de la implementación de un firewall para restringir el tráfico y protegernos de ataques externos, la configuración de certificados SSL/TLS para garantizar la confidencialidad e integridad de los datos transmitidos, además de la instalación y configuración de ciertos plugins de WordPress que son muy llamativos desde el punto de vista de la seguridad que implementaremos. Como último, se creará un plan de gestión con recomendaciones de cómo deberíamos de actuar de ahora en adelante para que nuestro servidor no vuelva a caer en el olvido y tengamos la misma problemática que estamos teniendo ahora en unos años.

3.2.3. Migrar a un proveedor de alojamiento gestionado

Esta solución la pongo debido a que en el caso de ser una empresa o particular sin los recursos necesarios para poder encargarse de la creación y mantenimiento de un servidor, lo mejor sería contratar a una empresa especializada para que se encargue de ello y de esa

forma despreocuparse de esta tarea. Esta solución conlleva un coste y ese coste depende de la empresa y el servicio que contratemos.

Este tipo de servicios nos van a proporcionar más tiempo y recursos debido a que la tarea de gestión y mantenimiento se externaliza, además de asegurarnos unas medidas de seguridad robustas, ya que las empresas especializadas en el sector tienen herramientas muy potentes de firewall, monitoreo y de actualizaciones automáticas.

Esto nos lleva a una clara dependencia del proveedor para mantener nuestro sistema operativo por eso es importante negociar con la empresa unos acuerdos de nivel de servicio (SLA) claros para luego no tener ningún tipo de inconveniente.

Y por último, aunque estas empresas ofrecen una amplia gama de personalización, servicios y características, es posible que no cubran por completo todas nuestras necesidades, por eso sería conveniente en el caso de elegir esta opción deberíamos de evaluar los diferentes servicios que nos pueden proporcionar cada uno de los proveedores y elegir sabiamente cual se nos adapta de la mejor manera a nuestra demanda.

3.3 Solución Propuesta

La solución que propongo es la solución que hemos puesto en segundo lugar, reimplementar el servidor.

Esta solución va a consistir en crear el servidor desde cero y implementar todo lo necesario para que todo funcione correctamente.

Haremos una guía de cómo debemos de crear una nueva máquina y la configuración que le tenemos que dar y qué comandos vamos a utilizar para la instalación de los componentes necesarios.

Una vez implementados esos componentes, detallaré dónde acceder y cómo, dentro del panel, para poder poner en marcha un servidor web con un gestor de bases de datos.

Una vez tengamos todo en funcionamiento procederemos con la migración del WordPress.

Una vez el WordPress migrado y todo en funcionamiento haremos una serie de modificaciones para poder securizar el sitio web y dejar ciertas recomendaciones que seguir para que todo siga en correctas condiciones.

CAPÍTULO 4
Diseño de la solución

Una vez seleccionada una de las anteriores opciones, hemos decidido decantarnos por la opción de reimplementar el servidor y crear uno nuevo. Ahora debemos considerar cómo lo vamos a hacer y lo primero que deberíamos hacer sería elegir qué panel vamos a utilizar para la gestión del servidor ya que el panel de control va a ser nuestra piedra angular, desde donde vamos a realizar casi todo tipo de gestiones.

Para poder tener criterio a la hora de decidir qué panel de control vamos a instalar, he hecho una pequeña investigación sobre qué panel deberíamos elegir analizando los paneles de control más extendidos en el mercado.

Luego presentaré los diferentes componentes que componen nuestro servidor comparados con la versión antigua instalada.

4.1 Arquitectura del Sistema

4.1.1. Paneles de control

Después de hacer una investigación sobre los posibles paneles que podemos instalar en nuestra máquina, hemos encontrado que las mejores opciones son las siguientes.

cPanel / WHM

Este panel[9] es uno de los paneles más generalizados y populares en lo que se refiere al alojamiento web. Ofrece una interfaz intuitiva y buenas herramientas de gestión.

Las ventajas que hemos encontrado en este panel son las siguientes:

- Interfaz fácil de usar y bastante intuitiva.
- Amplia gama de funciones y herramientas de administración para facilitar su uso.
- Muy buen soporte técnico.

Las desventajas de este panel son las siguientes:

- Para poder hacer uso de este panel se necesita una licencia de pago, lo que aumenta los gastos.
- Requiere de conocimientos previos sobre este tipo de tecnologías para hacer un uso adecuado.

PLESK

PLESK es otro de los paneles de control más utilizados en el ámbito del alojamiento web[10], conocido por su robustez y flexibilidad. Ofrece una interfaz moderna y una variedad de herramientas avanzadas para la gestión de servidores y sitios web.

Las ventajas que hemos encontrado en este panel son las siguientes:

- PLESK cuenta con una interfaz gráfica de usuario (GUI) muy intuitiva que facilita la navegación y gestión de las funciones del servidor.
- Funciona tanto en sistemas operativos Linux como Windows, lo que lo hace una opción versátil para diferentes entornos de servidor.
- Permite la gestión centralizada de múltiples sitios web, dominios y correos electrónicos desde un único panel de control.

Las desventajas de este panel son las siguientes:

- Al igual que cPanel, PLESK requiere una licencia de pago, lo que puede aumentar los costos operativos.
- Aunque su interfaz es intuitiva, sacar el máximo provecho de todas sus funcionalidades requiere conocimientos técnicos previos.
- En algunos casos, PLESK puede ser un poco más pesado en cuanto a consumo de recursos del servidor en comparación con otros paneles más ligeros.

Webmin / Virtualmin

Es un panel de control[11] de código abierto que te da opción a una gran variedad de opciones administrativas. Virtualmin es una extensión de Webmin dedicada específicamente para la gestión de servidores web.

Las ventajas que hemos encontrado en este panel son las siguientes:

- Es totalmente gratuito.
- Tiene soporte en múltiples sistemas operativos.
- Tiene funciones avanzadas para su administración y de correo electrónico.
- Tiene mucha documentación en lo referente a su uso.

Las desventajas de este panel son las siguientes:

- Tiene una interfaz que puede llegar a ser poco intuitiva y necesitas un periodo de adaptación un poco más largo en comparación con otros paneles de control.
- Requiere de ciertos conocimientos técnicos para poder usar todo el potencial que ofrece.

VestaCP

Este panel de control[12] es otro panel de control de código abierto que te ofrece una interfaz muy sencilla y fácil de usar. Está diseñado para que una persona sin muchos conocimientos sea capaz de poder utilizarlo completamente, es una opción bastante extendida en servidores con recursos limitados.

Las ventajas que hemos encontrado en este panel son las siguientes:

- Interfaz sencilla y fácil de usar.
- Instalación fácil, sencilla y guiada.
- Utilización de recursos muy bajos, ideal para servidores con pocos recursos.
- Soporte para gestión de múltiples sitios.

Las desventajas de este panel son las siguientes:

- Menos funciones que otros paneles de control.
- Comunidad más pequeña, lo que hace que el soporte dado a este panel de código abierto sea menor.

ISPConfig

Es un panel de control[13] de código abierto que ofrece una gama bastante amplia de funciones para la administración y gestión del servidor. Es flexible y escalable, por lo que se adapta a servidores de cualquier tamaño.

Las ventajas que hemos encontrado en este panel son las siguientes:

- Totalmente gratuito.
- Interfaz fácil de usar y con bastantes opciones.
- Soporte para gestión de múltiples sitios web.
- Amplia documentación y comunidad.

Las desventajas de este panel son las siguientes:

- Necesitas hacer una configuración inicial un poco más complicada si tenemos en cuenta la configuración inicial de otros paneles.
- Puede llegar a demandar más recursos que otras opciones.

4.1.2. LAMP

Nuestro servidor va a ser un sistema LAMP, un sistema LAMP está compuesto de unas herramientas concretas que le dan ese nombre: L de Linux, que va a ser el sistema operativo que va a tener nuestra máquina; A de Apache, que va a ser nuestro servidor web; M de MySQL, que va a ser nuestro gestor de base de datos para nuestro sitio web; y por último la P de PHP, que es un lenguaje de programación dinámico muy utilizado para el desarrollo de sitios web. Además, es al mismo tiempo un servidor PHP que nos va a permitir la ejecución de scripts de este tipo.

4.1.3. Gestor de máquinas virtuales

Esta es una herramienta de virtualización[14], que nos va a permitir crear y ejecutar múltiples máquinas virtuales en un único ordenador o hardware. Al fin y al cabo, son entornos que emulan el comportamiento de un ordenador físico. Algunas de las herramientas más famosas dentro de este ámbito son las siguientes: VMware, Microsoft Hyper-V y Oracle VirtualBox.

4.2 Tecnología Utilizada

4.2.1. MySql

El primer componente instalado, el cual vamos a revisar, es MySQL.

MySQL es un sistema de gestión de base de datos de código abierto que tiene un modelo de uso cliente-servidor que nos permitirá gestionar las bases de datos. En este caso, tenemos instalada la versión 8.0.19, mientras que en el nuevo servidor tenemos instalada una MariaDB con la versión 10.6.16.

```
bitnami@webs_internas:/$ mysql -V
/opt/bitnami/mysql/bin/mysql.bin  Ver 8.0.18 for linux-glibc2.12 on x86_64 (MySQL Community Server - GPL)
bitnami@webs_internas:/$
```

Figura 4.1: Versión MySql servidor antiguo

4.2.2. Php

PHP es un lenguaje de programación, el cual es utilizado sobre todo en el ámbito de las aplicaciones web y sitios web. Es un lenguaje a modo de script que se lanza en el servidor y que genera contenido que más tarde se envía al usuario. La versión de PHP actualmente en el servidor antiguo es la versión 7.3.16, una versión bastante más antigua que la que vamos a utilizar ahora, la versión 8.3.7, que está considerablemente más actualizada.

```
bitnami@webs_internas:/$ php -v
PHP 7.3.16 (cli) (built: Mar 21 2020 09:58:19) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.16, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.16, Copyright (c) 1999-2018, by Zend Technologies
bitnami@webs_internas:/$
```

Figura 4.2: Versión PHP servidor antiguo

4.2.3. WordPress

WordPress es una plataforma para gestionar el contenido de páginas web. Facilita la tarea de creación y administración de sitios web y blogs. La versión de WordPress es una de las cosas que estaba más desactualizada en el servidor anterior, teniendo instalada la versión 6.0.2. Sin embargo, vamos a actualizarlo a la versión 6.5.3, que es la versión más nueva de WordPress.

```php
<?php
/**
 * WordPress Version
 *
 * Contains version information for the current WordPress release.
 *
 * @package WordPress
 * @since 1.2.0
 */

/**
 * The WordPress version string.
 *
 * Holds the current version number for WordPress core. Used to bust caches
 * and to enable development mode for scripts when running from the /src directory.
 *
 * @global string $wp_version
 */
$wp_version = '6.0.1';

/**
 * Holds the WordPress DB revision, increments when changes are made to the WordPress DB schema.
 *
 * @global int $wp_db_version
 */
$wp_db_version = 53496;

/**
 * Holds the TinyMCE version.
 *
 * @global string $tinymce_version
 */
$tinymce_version = '49110-20201110';

/**
 * Holds the required PHP version.
 *
 * @global string $required_php_version
 */
$required_php_version = '5.6.20';

/**
 * Holds the required MySQL version.
 *
 * @global string $required_mysql_version
 */
$required_mysql_version = '5.0';

$wp_local_package = 'es_ES';
~
```

Figura 4.3: Versión WordPress servidor antiguo

4.2.4. Apache

Apache es un servidor web, uno de los más popularizados. Nos proporcionará un servicio estable y fiable, además de ser escalable, pudiendo dar servicio a HTTP y HTTPS. La versión que teníamos instalada en el servidor antiguo es la versión 2.4.41, mientras que en el nuevo servidor tendremos la versión 2.4.52.

```
Server version: Apache/2.4.41 (Unix)
Server built:   Mar 21 2020 09:46:15
bitnami@webs_internas:/$ ^C
bitnami@webs_internas:/$ ^C
bitnami@webs_internas:/$ httpd -v
Server version: Apache/2.4.41 (Unix)
Server built:   Mar 21 2020 09:46:15
bitnami@webs_internas:/$
```

Figura 4.4: Versión Apache servidor antiguo

4.2.5. Linux

Linux es un sistema operativo de código abierto basado en UNIX. Es conocido por su estabilidad, seguridad y su capacidad de personalización. Una de las características principales de Linux es que permite a los usuarios tener libre elección sobre la configuración de los componentes del sistema operativo, pudiéndose adaptar en la medida de lo posible a las peticiones del administrador del servidor.

Linux se puede instalar en diferentes versiones, y cada una de ellas tiene un software y unas herramientas preinstaladas diferentes. Algunas de estas versiones son Ubuntu, Fedora, Debian y CentOS.

En el servidor antiguo, teníamos instalado un Debian 9.13 preconfigurado con una herramienta que crea las máquinas virtuales, llamada Bitnami. Ahora vamos a utilizar una distribución de Linux, Ubuntu 22.04 LTS. Hemos escogido esta versión porque proporciona un soporte mayor que las demás versiones.

```
login as: bitnami
bitnami@192.168.0.19's password:
Linux webs_internas 4.9.0-19-amd64 #1 SMP Debian 4.9.320-2 (2022-06-30) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
      ___ _ _                   _
     | _ |_) |_ _ _  __ _ _ __ (_)
     | _ \ |  _| ' \/ _` | '  \| |
     |___/_|\__|_|_|\__,_|_|_|_|_|

  *** Welcome to the Bitnami LAMP 7.3.16-0
    ***
  *** Documentation:  https://docs.bitnami.com/virtual-machine/infrastructure/la
mp/ ***
  ***                 https://docs.bitnami.com/virtual-machine/
    ***
  *** Bitnami Forums: https://community.bitnami.com/
    ***

########################################################
###    For frequently used commands, please run:    ###
###         sudo /opt/bitnami/bnhelper-tool         ###
########################################################

Last login: Thu Nov  9 14:42:47 2023 from 192.168.1.140
bitnami@webs_internas:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 9.13 (stretch)
Release:        9.13
Codename:       stretch
bitnami@webs_internas:~$
```

Figura 4.5: Versión Linux antiguo servidor

4.2.6. Panel de control

Luego de hacer un análisis de unos cuantos paneles de control, totalmente válidos cada uno de ellos, me he decidido por el panel de control Webmin/Virtualmin. Los argumentos de por qué he decidido utilizar este panel son los siguientes:

- Virtualmin es un panel de control gratuito el cual nos ofrece una amplia gama de recursos para poder hacer todo lo necesario con nuestro servidor, lo cual descarta las opciones de Cpanel y PLESK, ya que aunque son las mejores opciones en general, es necesario pagar una licencia para su uso y en nuestro caso tenemos todo lo que queremos en un panel gratuito, por lo que podemos tener un ahorro por esta parte.
- Descarto la opción del VestaCp debido a que es el panel más limitado de todos los que hemos analizado y se nos queda un poco corto a la hora de ciertas funciones.
- Al tener que elegir entre Virtualmin e ISPConfig es difícil debido a que los dos son muy buenas opciones y totalmente válidos para el uso que le vamos a dar, pero nos hemos decantado por Virtualmin debido a que hemos tenido formación previa en el uso de este panel en nuestro trayecto por la universidad, por lo que hemos elegido la opción que ya conocemos.

4.2.7. Gestor de máquinas virtuales

Vamos a utilizar VirtualBox ya que es una aplicación gratuita, además de ser la que más familiarizada tengo, debido a que gran parte de las prácticas de la universidad se han desarrollado en entornos virtuales creados con VirtualBox.

Como he dicho anteriormente, VirtualBox nos va a permitir crear y ejecutar máquinas virtuales desde nuestro ordenador, asignándole los recursos que queramos dedicar a dicha máquina. Además, también tiene un sistema de snapshots que nos va a permitir hacer guardados de la máquina antes de hacer modificaciones importantes.

CAPÍTULO 5
Desarrollo de la solución

5.1 Creación máquina virtual

Primero de todo, para poder crear una máquina virtual y hacer uso de la misma, deberemos de tener un software para poder crearla. Este software, como he detallado en el capítulo anterior, va a ser VirtualBox. Deberemos de ir a la página oficial de VirtualBox, para poder descargar la versión adecuada al sistema operativo de la máquina en la cual vamos a trabajar. Una vez tenemos VirtualBox descargado e instalado, lo podremos abrir para poder importar la imagen del sistema operativo que deseemos.

En nuestro caso, la imagen del sistema operativo ha sido la de Ubuntu 22.04 (LTS), la cual también tendremos que descargar desde la página oficial del proveedor de Linux.

Una vez todo descargado, deberemos de importar la imagen del sistema operativo desde la opción de VirtualBox. Arriba a la izquierda donde pone "Máquina", clicamos y damos en nueva, y elegiremos la ubicación de la imagen que nos acabamos de descargar desde el panel desplegable donde pone imagen ISO.

Figura 5.1: Creación máquina virtual 1

Deberemos de seleccionar el nombre que le queremos dar a la máquina virtual y le damos a siguiente. Deberemos de poner cuál queremos que sea el nombre de usuario y contraseña de nuestro usuario de la máquina virtual, además de especificar el dominio, en caso de querer un dominio en específico.

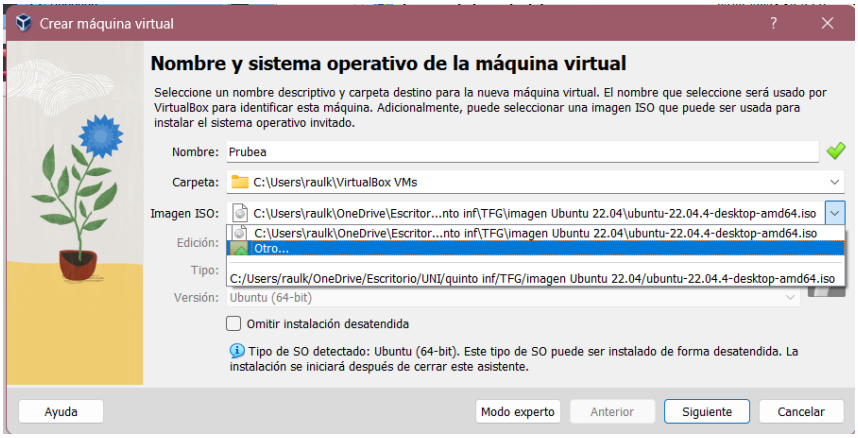

Figura 5.2: Creación máquina virtual 2

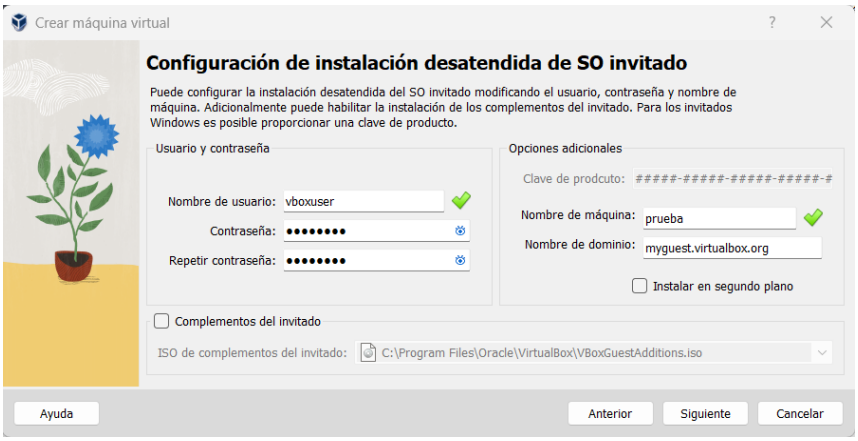

Figura 5.3: Creación máquina virtual 3

Luego deberemos de asignar los recursos que queremos asignar a nuestra máquina virtual en lo que se refiere a memoria RAM y núcleos de procesador, además de asignarle una partición de disco para poder almacenar ahí todos los datos relacionados con la máquina. Una vez creada la máquina virtual, la iniciamos y empezará a realizar la configuración inicial que hace de forma automática, esto puede tardar unos minutos.

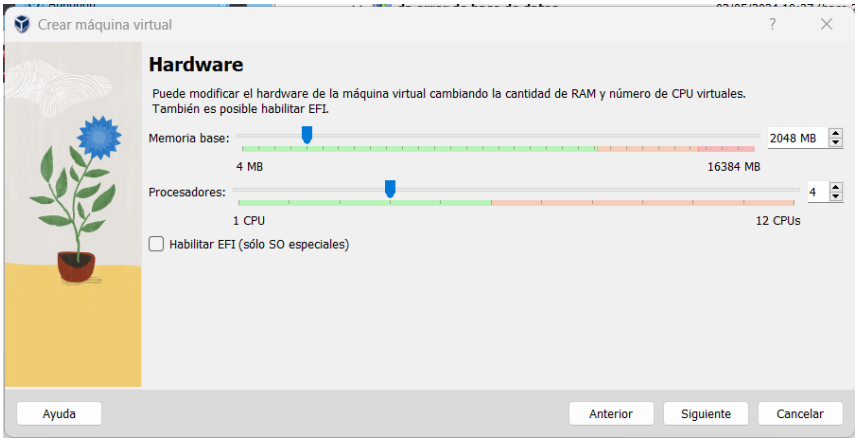

Figura 5.4: Creación máquina virtual 4

5.2 Configuración máquina virtual

Una vez está todo inicializado, nos vamos a dar cuenta de que el teclado no corresponde con el teclado estándar español, además de que no vamos a poder abrir una terminal de forma normal. Esto nos impide trabajar de una forma normal, así que nuestra prioridad ahora mismo es solucionar esto para poder seguir adelante sin ningún tipo de molestia.

5.2.1. Permiso administrador a usuario

Para solucionar esto, lo que vamos a hacer es, primero de todo, darle al usuario que hemos creado el poder de administrador. Para ello, en la parte de la configuración, nos vamos a la sección de usuarios y le damos el privilegio de administrador a nuestro perfil desde la opción que aparece y que podemos ver en la figura 5.5.

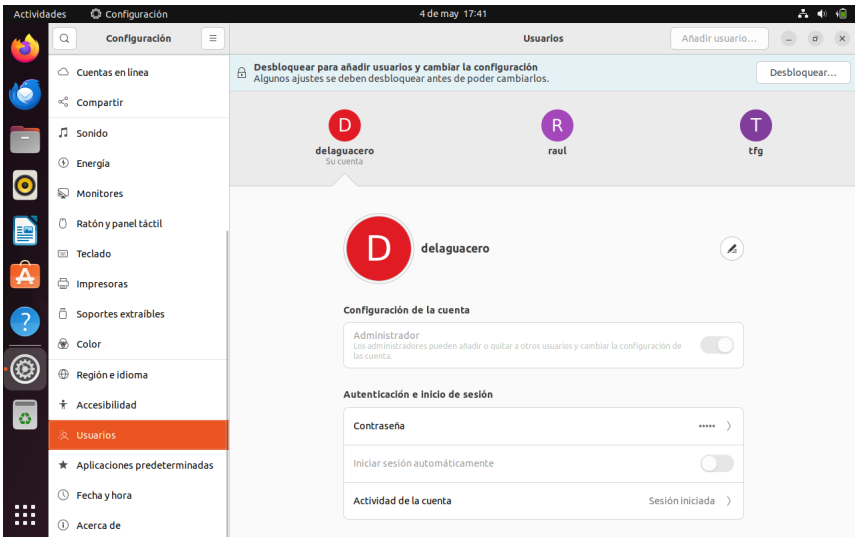

Figura 5.5: Configuración Máquina virtual 1

5.2.2. Idioma y teclado

Una vez ya tenemos el usuario que queremos como administrador, vamos a cambiar el teclado y el idioma del sistema al español. Para ello, desde la misma pestaña de configuración en la que estábamos para hacer al usuario administrador, buscamos en la lista "teclado" y dejamos el Español como preferencia en el idioma del teclado. De la misma forma, nos dirigimos a la pestaña de región y idioma y hacemos lo mismo, poniendo español como idioma principal. En las figuras 5.6 y 5.7 podemos ver las configuraciones correspondientes como deberían de quedar.

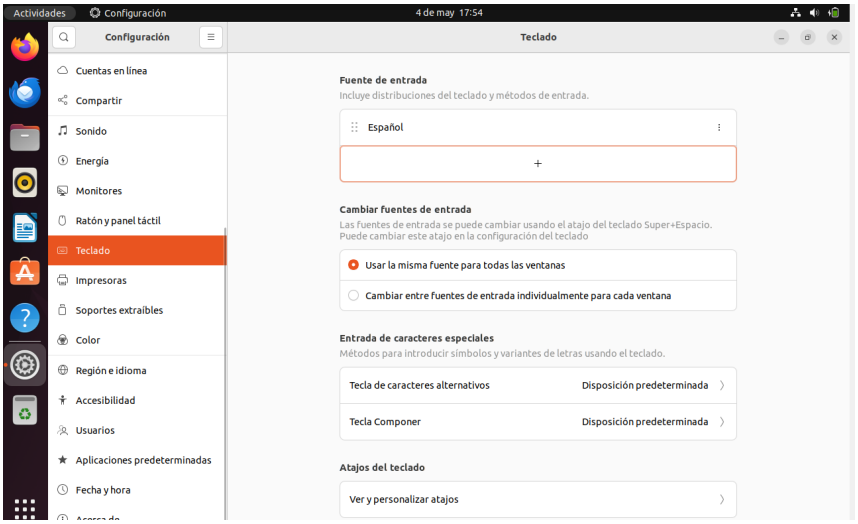

Figura 5.6: Configuración Máquina virtual 2

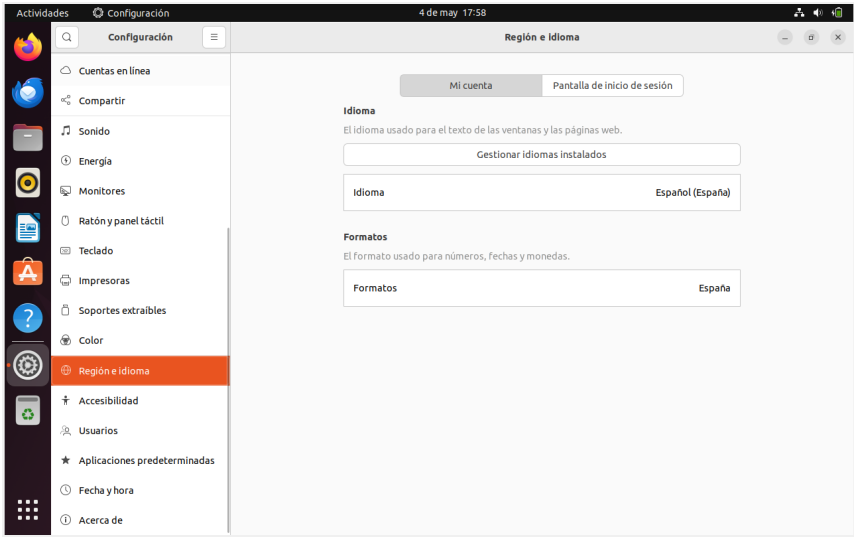

Figura 5.7: Configuración Maquina virtual 3

5.2.3. Actualización del sistema

Una vez tenemos todo lo anterior hecho, podremos abrir una terminal de forma normal, ya que si no lo hubiésemos hecho, la única forma de abrir una terminal era teniendo que teclear las teclas CTRL+ALT+F3 para abrir una terminal, pero seguiríamos teniendo el problema de la distribución del teclado que nos impediría escribir con normalidad.

Ahora abrimos la terminal y ponemos el siguiente comando:

sudo apt-get update -y

Este comando actualiza la lista de paquetes disponibles en el sistema, verificando si hay paquetes disponibles para descargar. Además, añadimos la opción -y para que acepte todas las confirmaciones de instalación que normalmente solicitaría al sistema.

Una vez lanzado este comando, debemos de lanzar seguidamente el siguiente comando:

sudo apt-get upgrade -y

Este comando busca y actualiza todos los paquetes instalados en el sistema operativo a las versiones más actuales. Esto incluye actualizar los paquetes disponibles que se han obtenido en la ejecución del comando anterior. Recuerda lanzarlo con la opción -y para que no tengamos que aceptar cada una de las actualizaciones.

5.3 Instalación virtualmin

Para poder instalar Virtualmin, debemos de descargar el script de instalación del mismo. Este lo podemos descargar desde la página oficial de Virtualmin. Una vez descargado el script, nos vamos a dar cuenta de que no podemos ejecutarlo por tema de permisos, por lo que vamos a tener que darle permisos al archivo del script para poder ejecutarlo.

Para otorgar permisos de ejecución al script, lanzamos el siguiente comando:

Chmod +x virtualmin-install.sh

Este comando, chmod +x virtualmin-install.sh, se utiliza para darle permisos de ejecución al archivo en cuestión, en este caso, el script de instalación de Virtualmin.

chmod nos permite cambiar los permisos de un archivo o carpeta, por lo que es justo lo que necesitamos para poder cambiar los permisos del script.

+x lo que hace es darle al script el permiso de ejecución, que es lo que queremos hacer con este archivo.

virtualmin-install.sh es el nombre del archivo que queremos ejecutar.

Ahora, abrimos una terminal en la carpeta donde esté ubicado el archivo y ejecutamos el script con el siguiente comando:

sudo ./virtualmin-install.sh

Una vez el script está en ejecución, este nos va a guiar en la instalación de Virtualmin. Una vez ha verificado que todas las configuraciones son correctas, procede a la instalación de Virtualmin en nuestra máquina. Puede que dé un fallo a la hora de comprobar las configuraciones, para solucionarlo bastará con reiniciar la máquina.

Una vez que la ejecución del script ha finalizado, nos saltará un mensaje en la terminal que nos indica que podemos acceder a nuestro panel de Virtualmin mediante un navegador web buscando la IP de nuestra máquina y el puerto en el que se ha configurado. Esta sería la forma de acceder:

192.168.0.x:10000

Una vez hemos accedido a la web, nos va a pedir que pongamos usuario y contraseña para acceder a nuestro panel de administración y vamos a empezar a seguir la guía de instalación que nos aparece la primera vez que accedemos a nuestro panel de control.

En esta guía de instalación, nos va a aparecer primero la opción de si queremos precargar las librerías de Virtualmin, lo cual nos va a consumir más recursos, pero a la vez nos va a dar un poco más de velocidad al navegar por la interfaz de nuestro panel. Esto depende de los recursos que puedas destinar a tu máquina. En mi caso, debido a que tampoco es una cantidad de recursos muy grandes y me lo puedo permitir, lo acepté. En el caso de la segunda opción, es en el caso de que vayas a hacer uso de la función de correo electrónico que nos ofrece Virtualmin. Yo la dejé marcada, pero no voy a hacer uso de ella. En caso de ir ajustado de recursos, recomiendo desmarcarla ya que te ahorras un poco.

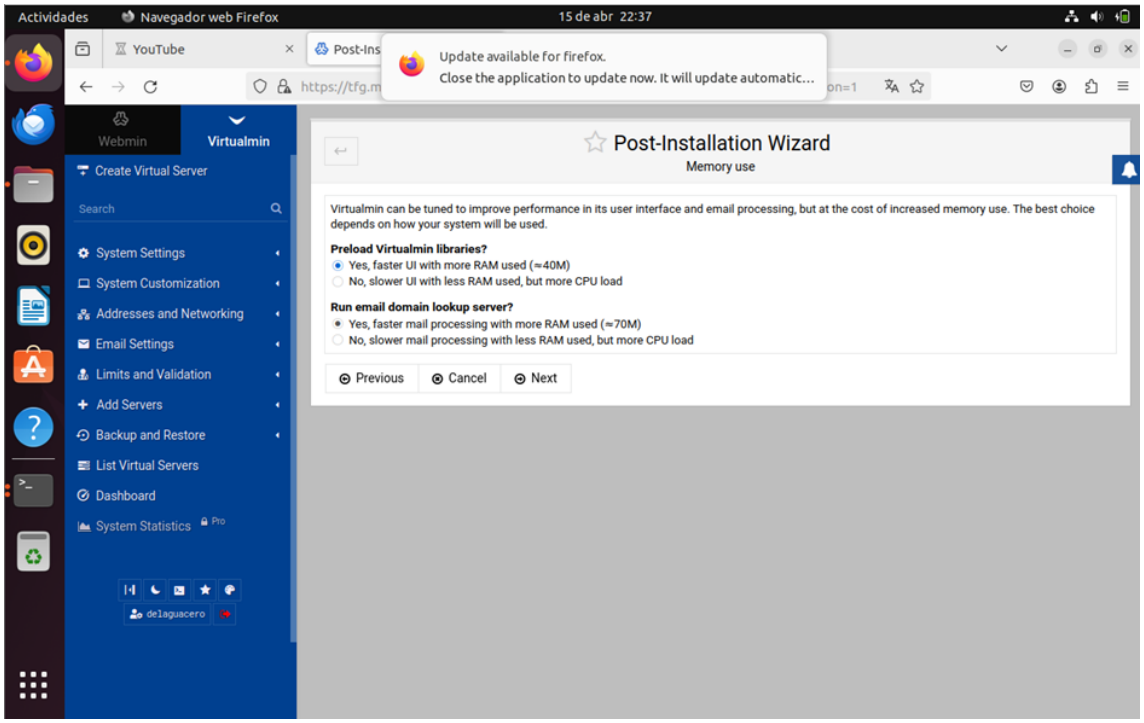

Figura 5.8: Instalación virtualmin

Al darle a "Next", nos va a aparecer qué tipo de servidor de base de datos queremos que lance para cuando el servidor esté activo, si queremos MariaDB o PostgreSQL, que son los dos servidores de base de datos que soporta Virtualmin.

En mi caso, marco únicamente la opción de MariaDB, ya que no es necesario el uso del otro servidor de base de datos.

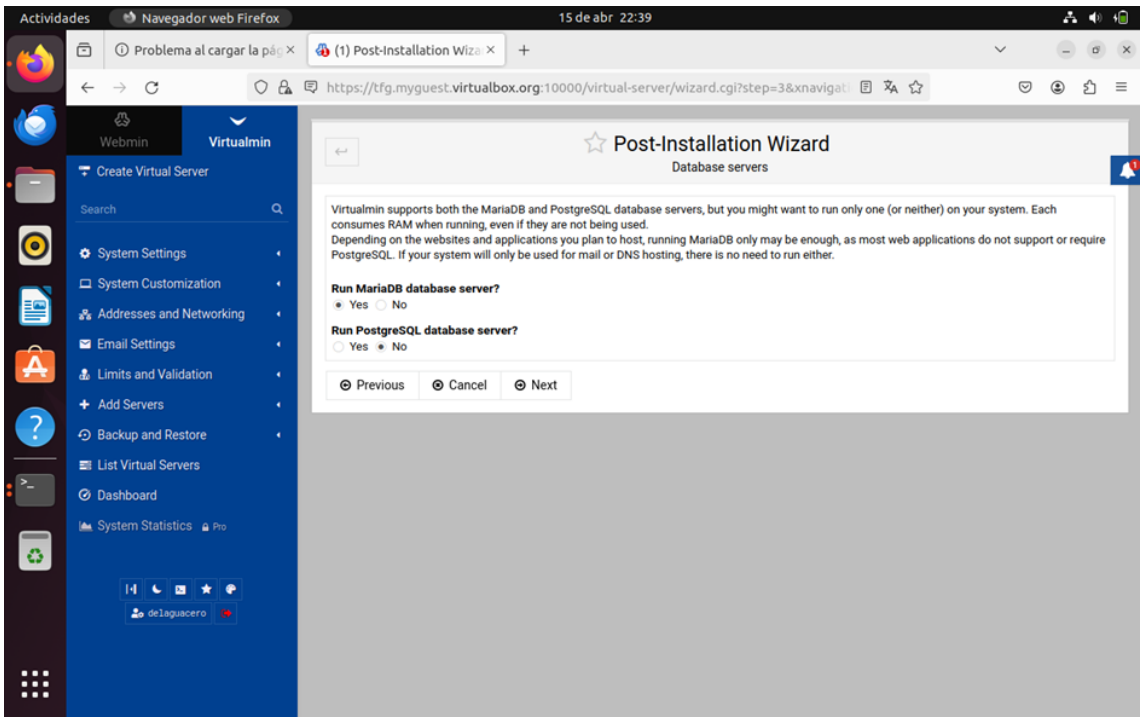

Figura 5.9: Instalación virtualmin 2

Seguidamente a este paso, nos va a pedir que decidamos una contraseña para el servidor de base de datos o que dejemos la autenticación con el usuario root. Para una mayor seguridad y control, ponemos una contraseña generada aleatoriamente la cual deberíamos de almacenar en algún sitio ya sea virtual o físicamente.

Ahora nos va a pedir el modo de almacenamiento de las contraseñas. En este caso, decidimos que solo almacene las contraseñas hasheadas. Como te explica el propio Virtualmin, las contraseñas de texto plano son más convenientes pero el problema de estas es que son menos seguras. De esta forma, como bien indica, todos los nuevos servidores creados, crearán una contraseña diferente, totalmente aleatorizada.

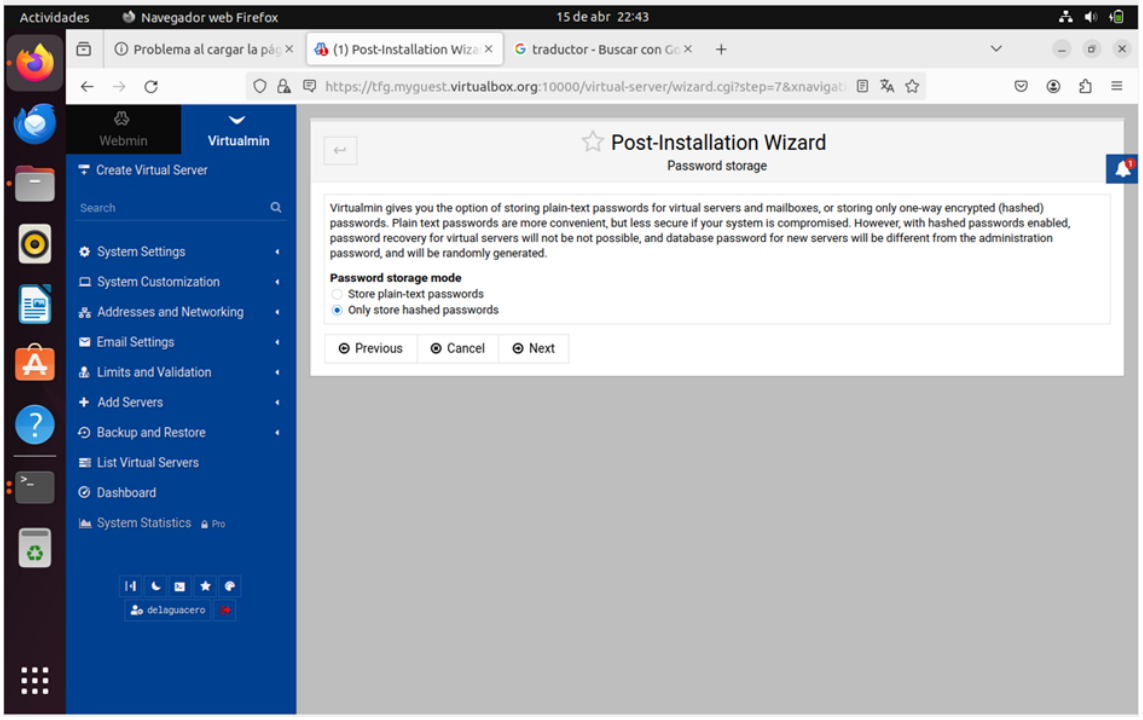

Figura 5.10: Instalación virtualmin 3

Ahora, para finalizar, nos va a pedir que elijamos los recursos que queremos destinar al servidor de base de datos, que lo dejamos predeterminado, además de pedirnos en caso de querer añadir un certificado en qué ruta queremos que se ubiquen, que también lo dejamos como viene predeterminado.

Una vez terminados todos estos pasos, todo lo necesario para una máquina LAMP estará instalado, debido a que Virtualmin ya instala todos los componentes necesarios para su funcionamiento de forma automática en sus últimas versiones.

5.4 Configuración de Virtualmin

Ahora vamos a crear un servidor virtual en el cual vamos a hacer la instalación de WordPress. Para ello, recomiendo que desde los ajustes de la máquina virtual vayas a la configuración de red y cambies la asignación automática de IP mediante DHCP a asignación manual. De esta forma, no tendremos problemas con cambios de IP que hagan inaccesibles nuestros servidores y tengamos una IP fija. Ten en cuenta que si cambiamos de la red en la que habitualmente trabajamos, debemos hacer cambios ya que el punto de acceso puede cambiar.

Para realizar este cambio, nos dirigimos arriba a la derecha de nuestra interfaz de la máquina virtual y damos en configuración de red, como se ve en la Figura 5.11.

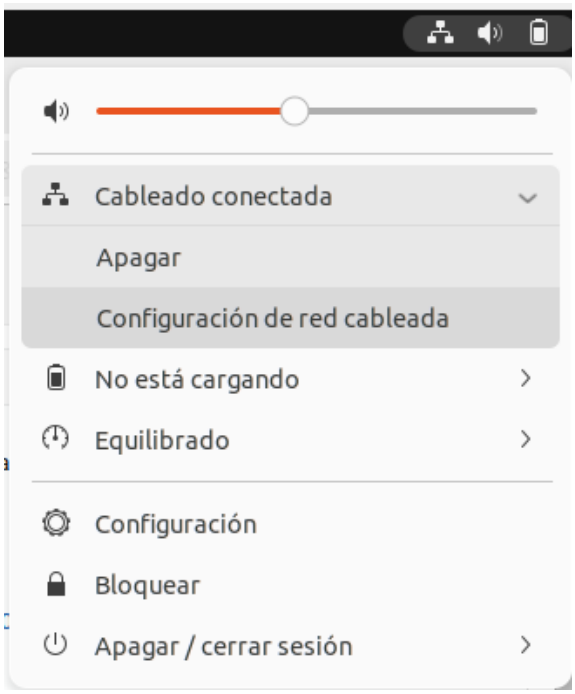

Figura 5.11: Configuración IP fija

Ahora, desde dentro de la configuración de la red a la que estamos conectados, debemos irnos a la sección IPv4 y cambiar la selección predeterminada que suele ser "Automático (DHCP)" por una configuración manual. En esta configuración manual, debemos poner la IP que queremos tener. En mi caso, he puesto la que ya se me había asignado automáticamente, la IP 192.168.0.25, y he puesto como puerta de enlace el router de mi casa, que es el que me proporciona internet (esto es lo que deberíamos cambiar en el caso de que cambiemos de proveedor de internet). Podemos ver el resultado de esta configuración en la figura 5.12.

Figura 5.12: Configuración IP fija 2

Una vez dentro de Virtualmin, nos vamos a la opción de "Create Virtual Server" donde crearemos el servidor web en el que vamos a alojar nuestras páginas web. Aquí sim-

plemente debemos poner el nombre de usuario y contraseña de este servidor, además del dominio del mismo.

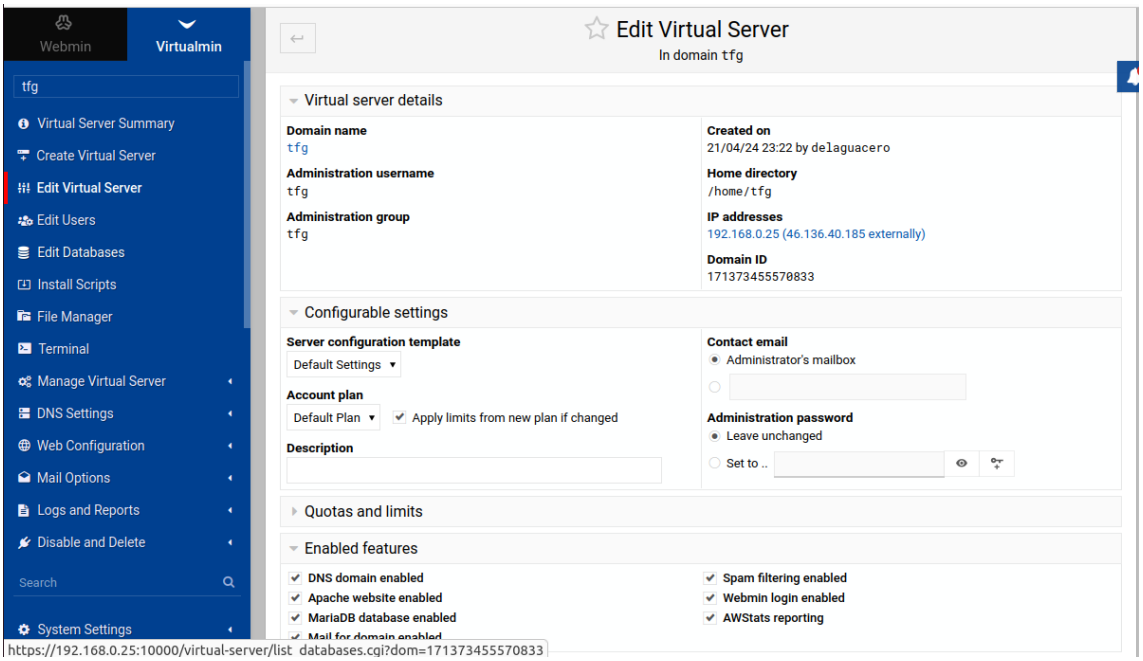

Figura 5.13: Configuración del servidor

Ahora, como última comprobación nos vamos a la pestaña Webmin ->System ->Software package updates y revisamos que todo está correctamente actualizado. En mi caso, me encontré con que habían bastantes paquetes que aún faltaban por actualizar. Simplemente darle a actualizar y ya está, es esperar a que se instalen y lo tendríamos todo correctamente configurado.

5.5 Instalación WordPress

Para instalar Worpress lo vamos a hacer desde el panel de control, ya que Virtualmin nos ofrece una herramienta de ejecución de scripts la cual nos va a facilitar enormemente la instalación de este software en nuestra máquina.

Ahora, para poder hacer la instalación de WordPress vamos a necesitar crear una base de datos. Nos vamos a la pestaña de “Edit databases” y creamos una base de datos con el nombre que nosotros queramos, en este caso WordPress.

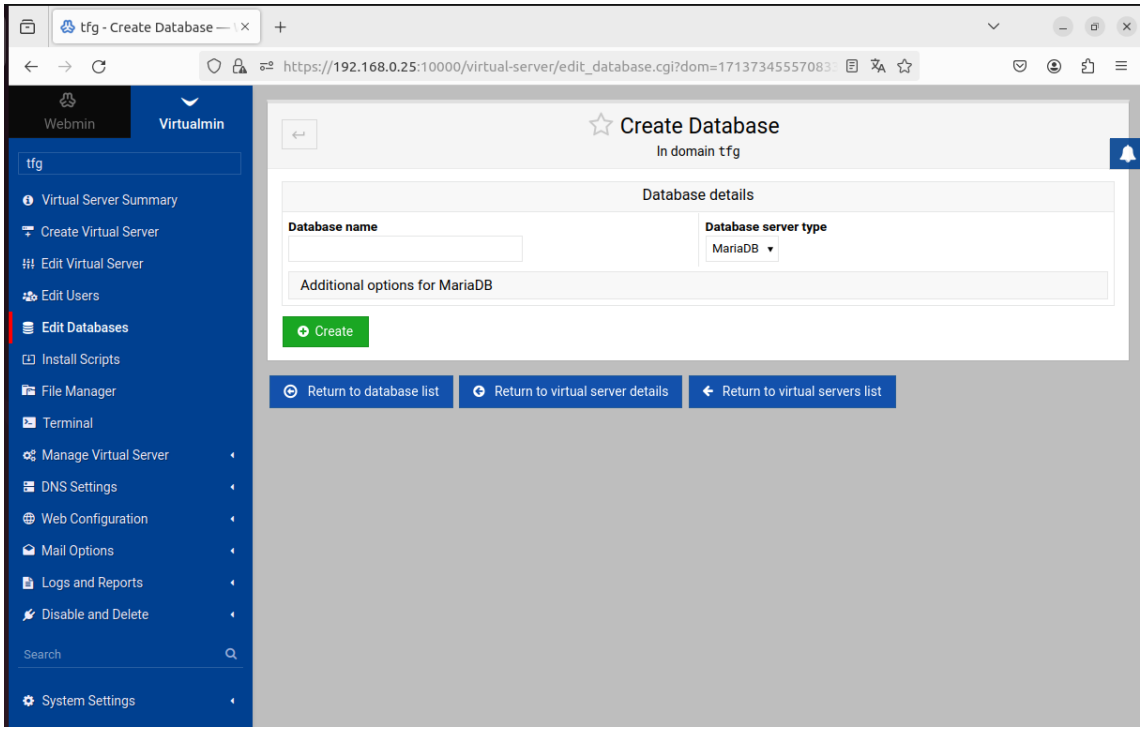

Figura 5.14: Creación de la base de datos

Nos dirigimos al display de nuestro servidor acabado de crear, buscamos la sección de "install scripts" y luego "available scripts". Una vez en esta pestaña, buscamos el script de instalación de WordPress. Una vez seleccionado, bajamos hasta abajo y le damos en "install scripts". En la Figura 5.15 podemos ver la ubicación anteriormente mencionada.

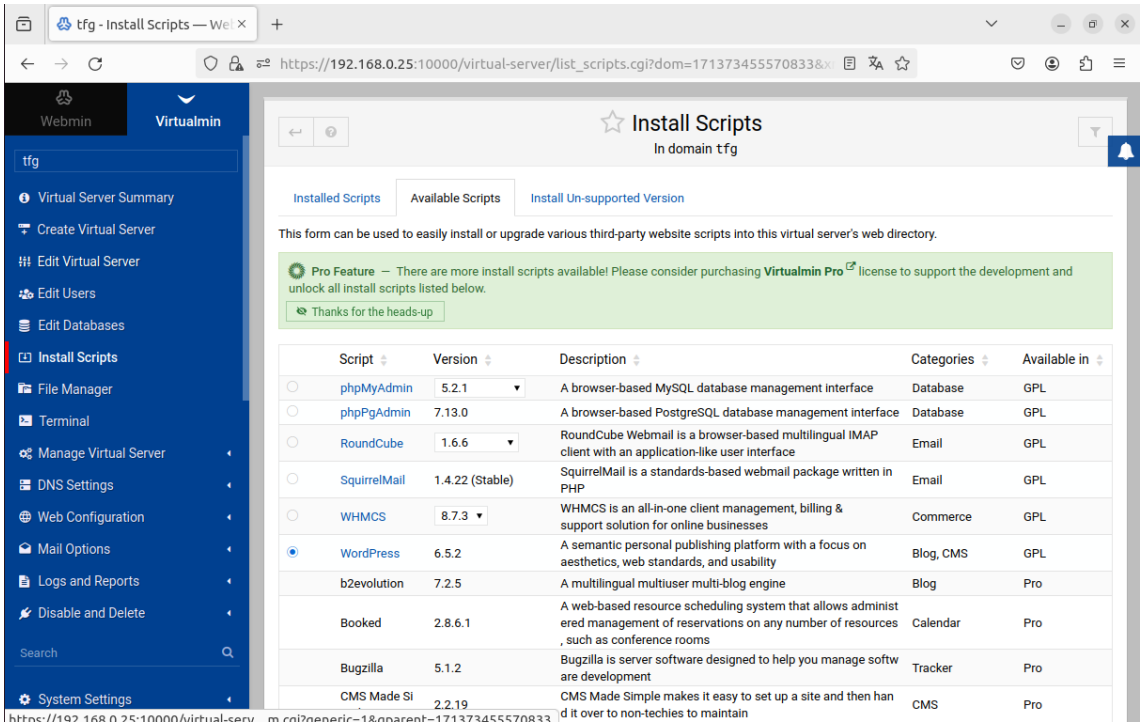

Figura 5.15: Instalación script WordPress

Cuando le demos a instalar el script, nos va a pedir que pongamos la configuración con la cual vamos a querer que se instale nuestro WordPress. Primero de todo elegimos

la base de datos la cual acabamos de crear, además de indicar el nombre de WordPress y la ubicación que vamos a querer que tenga en la web.

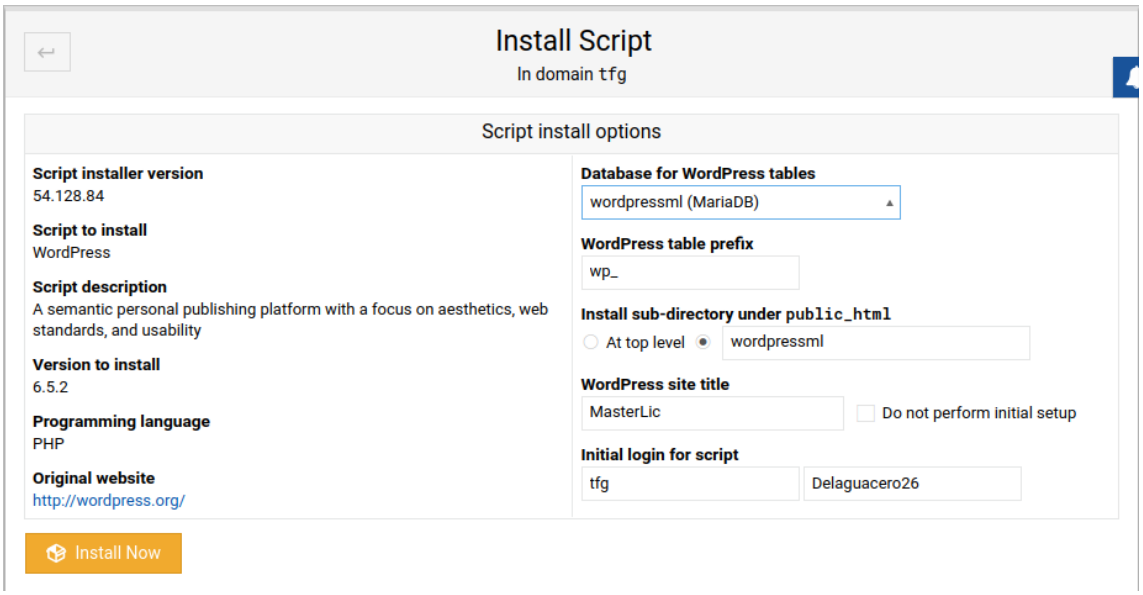

Figura 5.16: Opciones del Script de WordPress

Ahora nos volvemos a ir a la sección de "Available scripts", como hemos hecho anteriormente con la instalación de WordPress. Buscamos phpMyAdmin y lo instalamos también. Al fin y al cabo, phpMyAdmin es un gestor de bases de datos mucho más intuitivo que el que nos proporciona el propio Virtualmin. Una vez instalado, nos dirigimos a un buscador y accedemos a nuestro gestor de base de datos phpMyAdmin poniendo en el buscador 192.168.0.25.

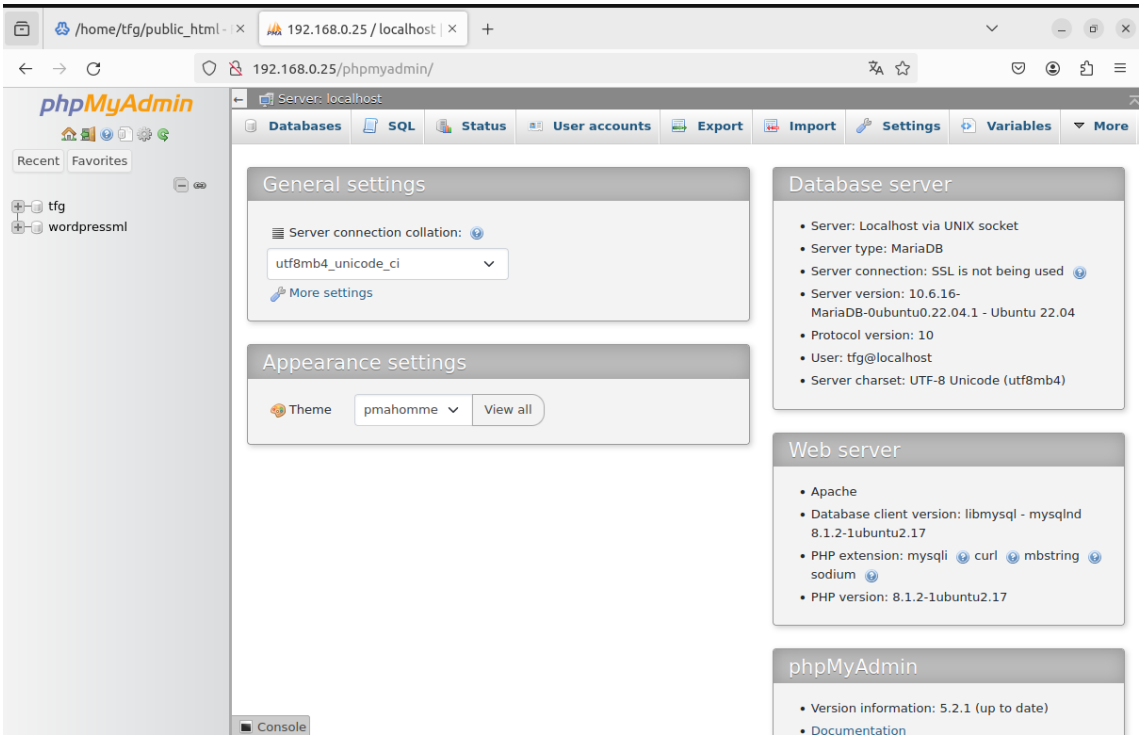

Figura 5.17: Acceso a WordPress

Una vez hecho todo esto, podemos acceder a nuestro WordPress poniendo en el buscador nuestra IP seguido del nombre que le hemos dado a nuestro WordPress.

192.168.0.25

Al acceder por primera vez, el propio WordPress nos va a guiar mediante una instalación básica. Nada más accedemos a la URL que he puesto anteriormente, primero de todo deberemos seleccionar el idioma en el que vamos a querer instalar el WordPress.

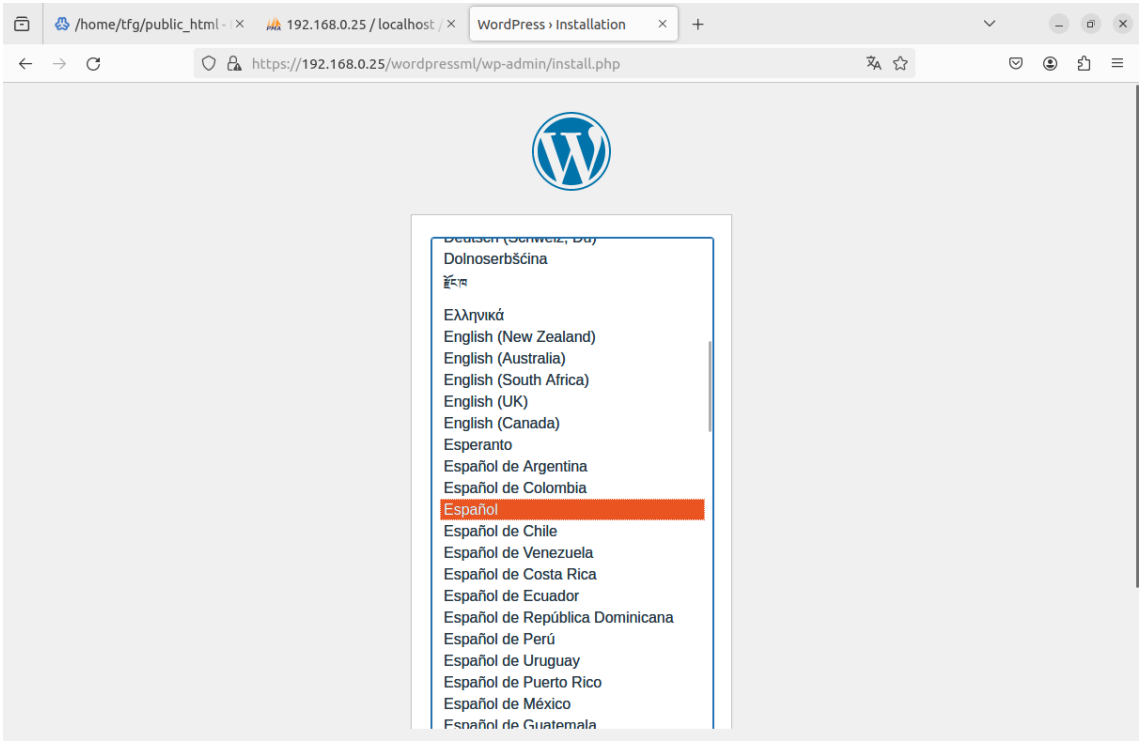

Figura 5.18: Primer acceso a WordPress

Luego nos va a pedir rellenar ciertos datos para completar la configuración de nuestro sitio web. En la figura 5.19 podemos ver los datos a rellenar.

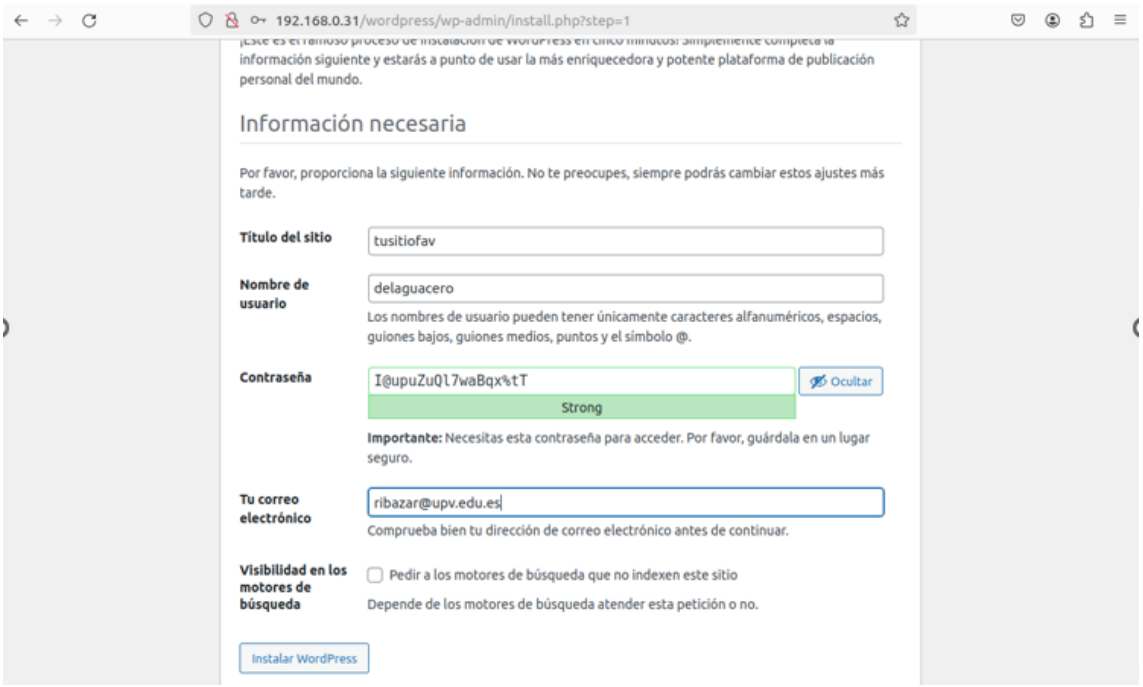

Figura 5.19: Primer acceso a WordPress 2

Ahora podríamos configurar nuestro sitio web desde cero, creando nuestra página y configurándola a nuestro gusto. Sin embargo, este no es nuestro caso. Nosotros vamos a migrar un sitio web de WordPress alojado en un host diferente el cual estaba desactualizado y bajo ningún mantenimiento. Por eso, lo primero que vamos a hacer es irnos a nuestro anterior host y debemos irnos a nuestro phpMyAdmin y buscar la base de datos que pertenece a nuestra anterior página web. Deberemos exportar esta base de datos para después poder importarla en el nuevo host. Además de todo esto, deberemos buscar la carpeta donde se encuentran todos los archivos correspondientes a nuestro sitio web y comprimirlos en un archivo ZIP para poder descomprimirlo en nuestro nuevo servidor.

Una vez hemos hecho esto, ya podemos ir a nuestro servidor actual y tenemos dos opciones: utilizar el sitio web que acabamos de crear desde cero o hacer una nueva instalación de un WordPress con la misma IP o dominio. Esto hace que el servidor, dependiendo de si ponemos /wordpressml o /otrositioweb, nos redirija a ese mismo, con la ventaja de que desde un mismo servidor virtual podemos tener diferentes WordPress alojados.

En nuestro caso, vamos a utilizar esa misma instalación de WordPress la cual ya está instalada.

5.6 Migración de base de datos y WordPress

Una vez la base de datos está exportada y la carpeta del WordPress comprimida en un ZIP, nos disponemos a hacer la migración. Primero de todo, vamos a hacer, como hemos hecho anteriormente, una instalación de WordPress normal y crear una base de datos normal desde cero.

Nos vamos a dar cuenta de que si accedemos a phpMyAdmin y nos vamos al apartado de privilegios, nos vamos a dar cuenta de que el usuario que hemos creado no tiene privilegios sobre nada, como vemos en la Figura 5.20.

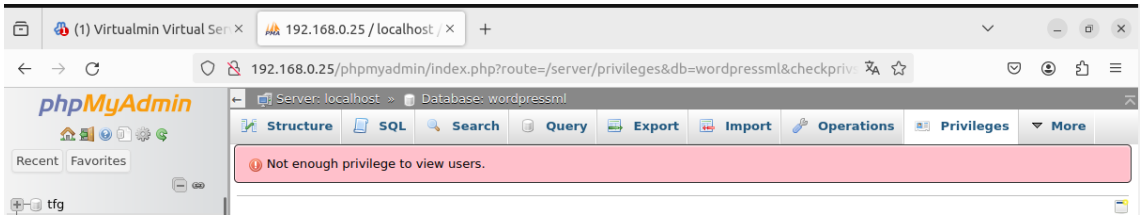

Figura 5.20: Sin privilegios de PhpMyAdmin

Para darle permisos a nuestro usuario, nos vamos a ir a la interfaz de Virtualmin. Damos en Webmin y cuando nos salgan las diferentes opciones, damos en el desplegable de "Servers" donde podremos ver "MariaDB Database server". Al hacer clic ahí, podremos ver las diferentes opciones de configuración que tenemos para el servidor de base de datos.

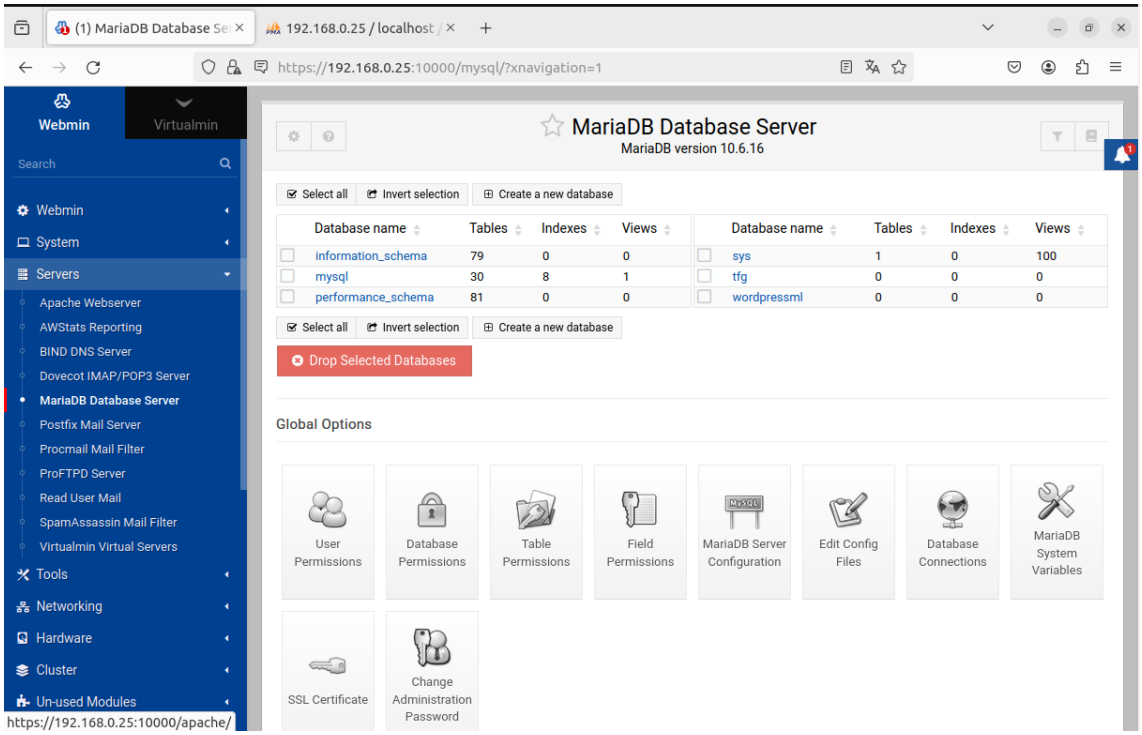

Figura 5.21: Opciones del servidor de base de datos

Ahora, entre todas las opciones que vemos, nos vamos a la opción que pone "User Permissions". Una vez accedamos, tendremos que seleccionar el usuario al que le queremos dar los permisos. En nuestro caso, es el usuario que creamos con el nombre de "tfg". Una vez dentro del usuario, deberemos concederle todos los permisos que tenemos en la lista de permisos que nos aparecen, como se puede ver en la figura 5.22, y darle a "Save" para que se guarde la selección de permisos.

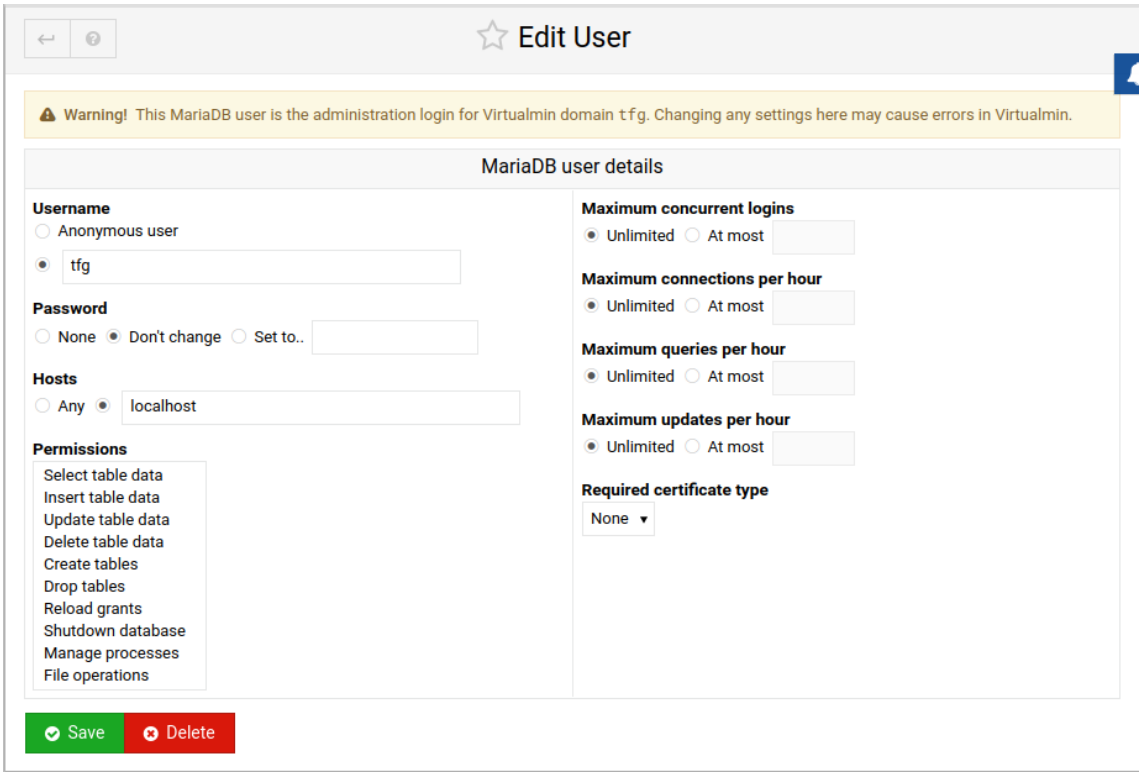

Figura 5.22: Permisos Base de datos

Ahora, desde la propia pestaña de Virtualmin, deberíamos ver que el usuario tiene en la columna de privilegios "all". Pero para confirmar que es cierto, nos dirigimos al phpMyAdmin donde hemos accedido anteriormente y ahora sí que tendremos los privilegios para poder verlo todo.

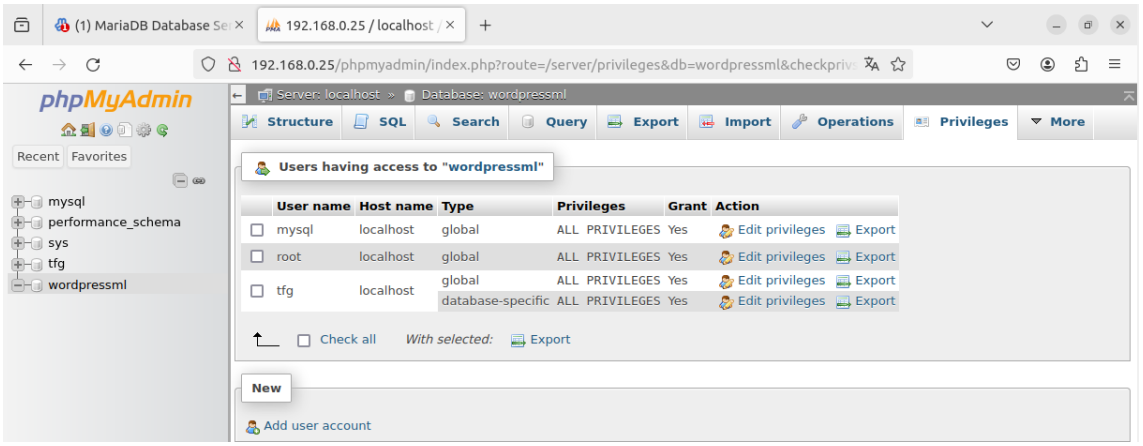

Figura 5.23: Revisar permisos de usuario en base de datos

Una vez que tenemos los privilegios otorgados, nos dirigimos a la base de datos que vamos a utilizar para migrar el WordPress y borramos todo tipo de contenido que pueda tener la base de datos previamente. Para ello, nos vamos en el panel de Virtualmin a la sección de nuestro servidor, y seleccionamos "Edit Databases". Una vez allí, seleccionamos la base de datos que queremos borrar y le damos en "Manage". Dentro de esta opción, podemos borrar todas las tablas o contenido que tengan. Alternativamente, podemos seleccionar "Drop Database" para borrar por completo la base de datos y luego podemos crear una nueva sin nada dentro.

Ahora, lo que vamos a hacer es copiar los archivos del antiguo WordPress en donde hemos hecho la instalación del WordPress completamente nuevo. Para ello, primero deberemos borrar el directorio actual en el cual se encuentra el WordPress vacío almacenado. Este se encuentra en la ruta /home/tfg/public-html. Para borrarlo, tenemos dos opciones: desde la interfaz gráfica que nos proporciona Virtualmin, desde la opción "File Manager", buscamos el directorio y con clic derecho nos sale la opción de eliminar. Otra opción es mediante el uso de un comando en la terminal, el cual es el siguiente:

sudo rm -R /home/tfg/public-html/wordpressml

Con este comando, con la opción -R, borramos todo el directorio con sus subcarpetas incluidas.

Una vez borrado, nos disponemos a copiar el archivo con el mismo nombre "wordpressml" a la ruta de donde lo acabamos de borrar. Para ello, descomprimimos el ZIP donde tenemos almacenado todo con el comando:

sudo unzip wordpressml.zip

Una vez que todo esté descomprimido, le damos los permisos a la carpeta para que no haya ningún tipo de error con el tema de permisos al intentar acceder a los recursos de la misma. Para ello, primero vamos a cambiarle el grupo y el propietario al mismo que si lo hubiésemos creado nosotros, además de darle todos los permisos para poder hacer con el archivo lo que queramos.

sudo chown -R delaguacero:delaguacero /home/Downloads/wordpress

sudo chmod -R 777 /home/Downloads/wordpress

Una vez lanzados estos comandos, el primero de ellos para hacer que el archivo sea propiedad de "delaguacero" y del grupo "delaguacero", que es el usuario administrador que yo mismo creé, y el segundo de los comandos para darle todo tipo de permisos sobre esta carpeta. Ahora, para poder copiarla en el directorio que queremos, lanzaremos el siguiente comando:

sudo cp -R /home/Downloads/wordpressml /home/tfg/public-html

Ahora ya tendremos todos los archivos necesarios para que nuestro WordPress funcione. Pero antes de que funcione completamente, debemos importar sobre la base de datos vacía que hemos creado anteriormente la base de datos que va relacionada con los archivos que acabamos de copiar.

Para ello, nos dirigimos al phpMyAdmin y seleccionamos la base de datos objetivo. Una vez seleccionada, debemos dar en "Import" y seleccionar el archivo que hemos creado anteriormente en la fase de exportación de la antigua base de datos. El archivo es un archivo con la extensión .sql, y lo podemos seleccionar desde donde pone "Examinar". En la Figura 5.24 se muestra tanto donde tenemos la pestaña de "Import" como donde tenemos el botón de "Examinar".

Figura 5.24: Importación de la base de datos

Una vez seleccionado el archivo de importación deseado, confirmamos la importación desde la parte inferior con un botón que pone "Import".

Ahora vamos a hacer los cambios necesarios para que nuestro WordPress sea funcional en este nuevo host. Para ello, desde phpMyAdmin nos dirigiremos a la tabla de "wp-options", donde debemos cambiar el enlace al que hace referencia nuestro WordPress. Actualmente, al importarlo, aún está el enlace que hace referencia al antiguo host. En nuestro caso, vamos a tener que sustituirlo por:

http://192.168.0.25/wordpressml

En la Figura 5.25 podemos ver de forma más clara a qué me refiero exactamente.

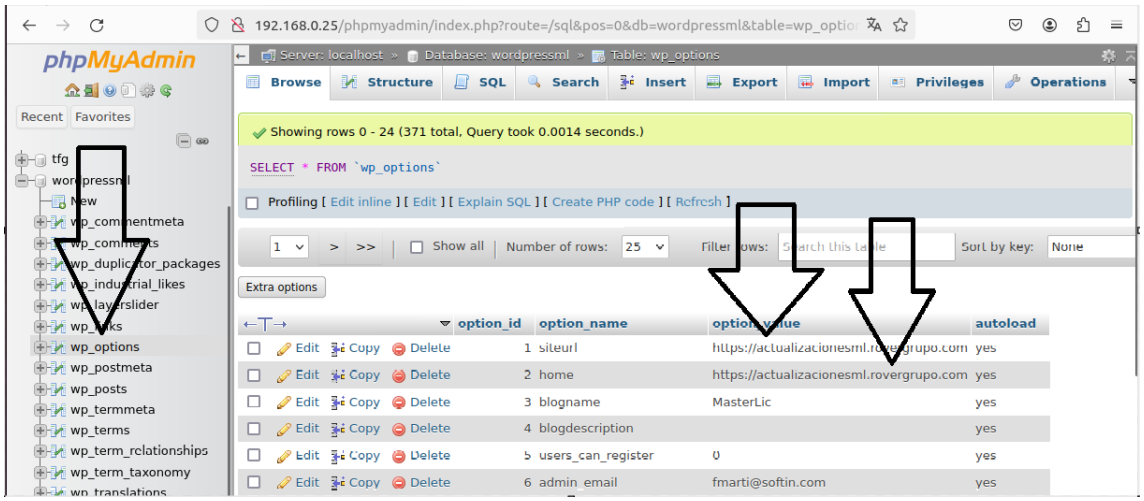

Figura 5.25: Que Cambiar en la base de datos

En las dos flechas de la derecha, deberemos darle en "Edit" y poner en ellas la URL del nuevo host, que es la que he indicado anteriormente.

Ahora, como último paso, deberíamos ir al archivo de configuración de WordPress. Podemos acceder a este desde el File Manager de Virtualmin. Nos dirigimos a Virtualmin, seleccionamos la opción de "File Manage", accedemos al directorio "wordpressml". Una vez dentro, buscamos el archivo wp-config.php, al cual le damos clic derecho y seleccionamos "Edit".

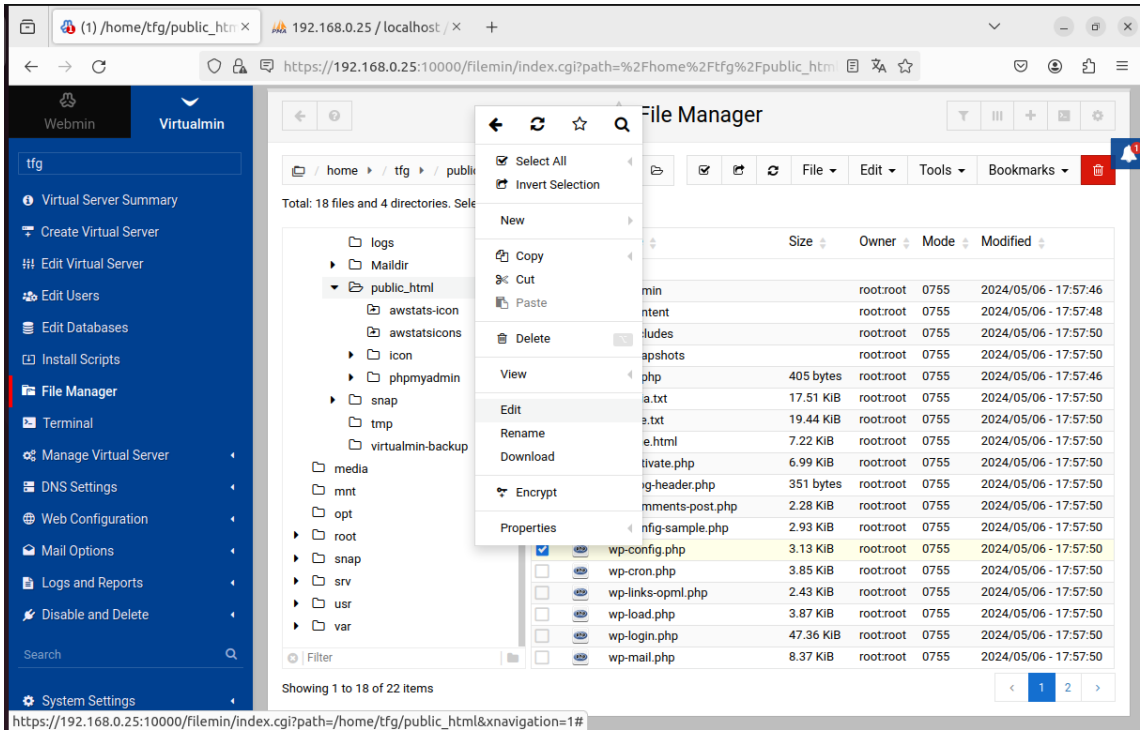

Figura 5.26: Como editar wp-config.php

Al editar el archivo, deberemos poner en el apartado del nombre de la base de datos el nombre de nuestra base de datos actual. Donde pone usuario y contraseña de MySQL, lo podemos encontrar fácilmente en el propio Virtualmin, en la sección “Edit Database”, dándole en “User” y en “Password”, como se puede ver en las Figuras 5.27 y 5.28.

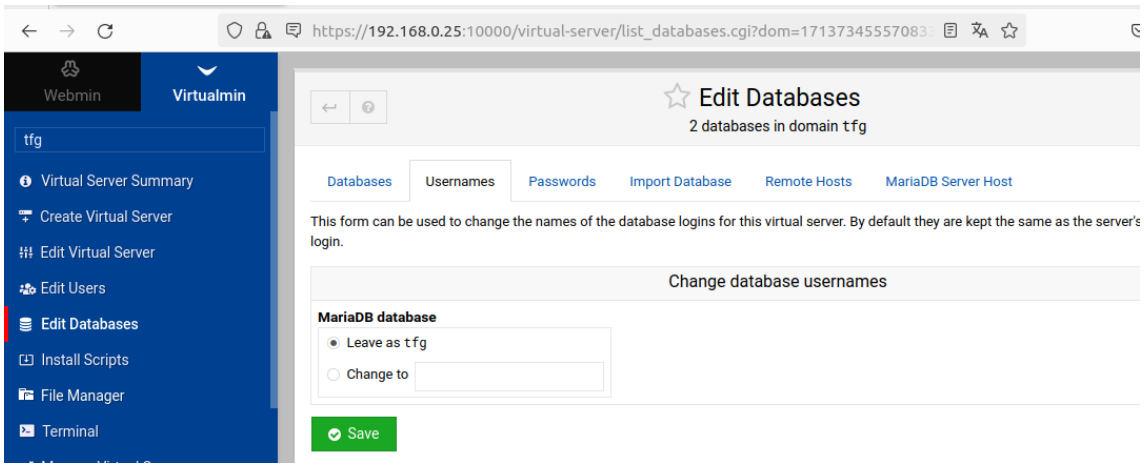

Figura 5.27: Usuario base de datos

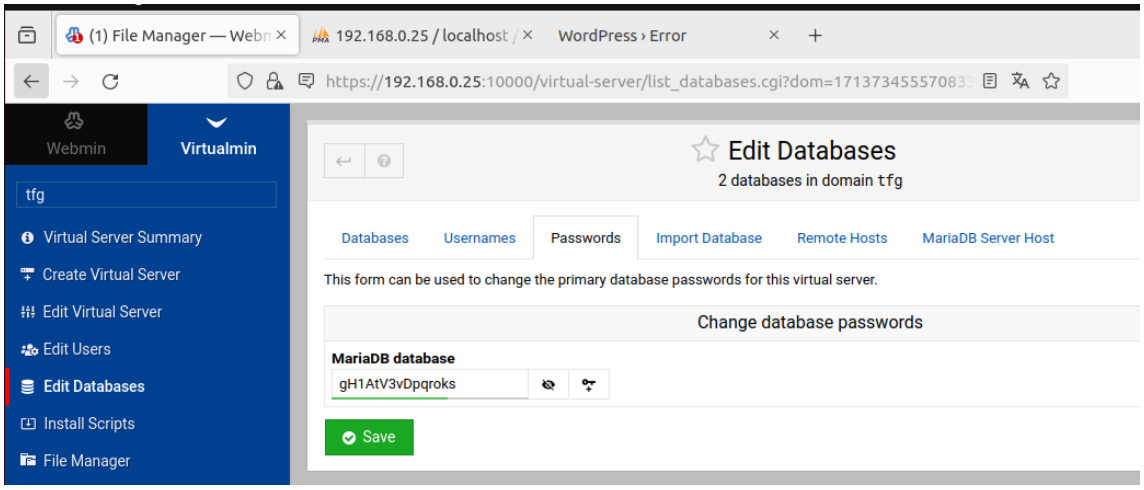

Figura 5.28: Contraseña base de datos

Una vez tenemos esta información la sustituimos tal y como podemos ver en la Figura 5.29.

```
18 /** El nombre de tu base de datos de WordPress */
19 define('DB_NAME', 'wordpressml');
20
21 /** Tu nombre de usuario de MySQL */
22 define('DB_USER', 'tfg');
23
24 /** Tu contraseña de MySQL */
25 define('DB_PASSWORD', 'gH1AtV3vDpqroks');
26
27 /** Host de MySQL (es muy probable que no necesites cambiarlo) */
28 define('DB_HOST', 'localhost');
29
30 /** Codificación de caracteres para la base de datos. */
31 define('DB_CHARSET', 'utf8mb4');
32
33 /** Cotejamiento de la base de datos. No lo modifiques si tienes dudas. */
34 define('DB_COLLATE', '');
35
```

Figura 5.29: Wp-config.php

Ahora, por fin, podremos acceder a nuestro nuevo sitio web migrado a otro host. Pero cuando intentamos acceder, nos vamos a llevar la sorpresa de que no podemos acceder debido a que hay algún error, el cual no sabemos de dónde proviene. El error lo podemos ver en la Figura 5.30 .

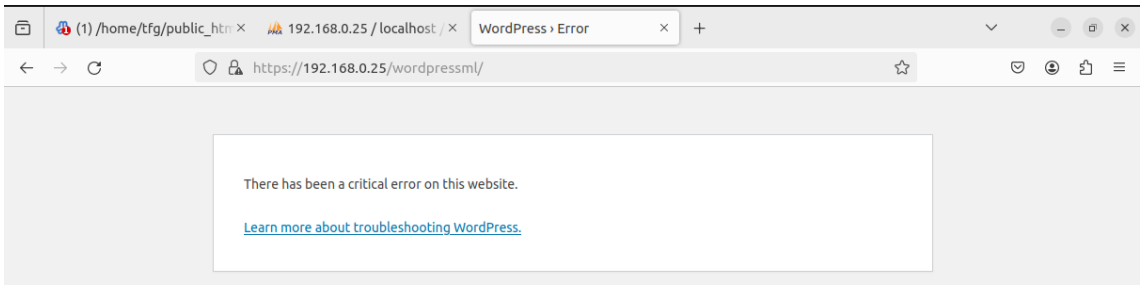

Figura 5.30: Error Migración WordPress

Este error es muy amplio y, luego de buscar por foros y otros medios, encontré que en muchos sitios, incluida la propia página de Virtualmin, nos indicaban que la mejor forma de encontrar la fuente de este error es activando la herramienta de debug de WordPress. Para ello, en el wp-config que hemos estado editando anteriormente, deberemos cambiar cierta línea del código, en la cual tenemos el parámetro de debug a false. Es tan sencillo como cambiar ese false por true y tendremos la herramienta de debug activada.

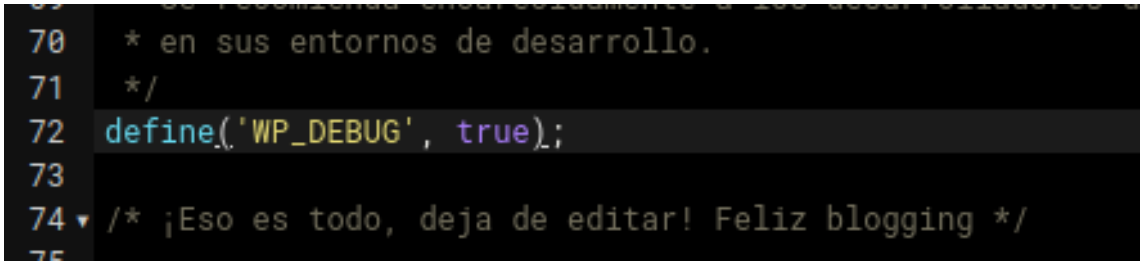

Figura 5.31: Donde activar herramienta de debug

Ahora, si refrescamos la página de WordPress, veremos los errores de la Figura 5.32:

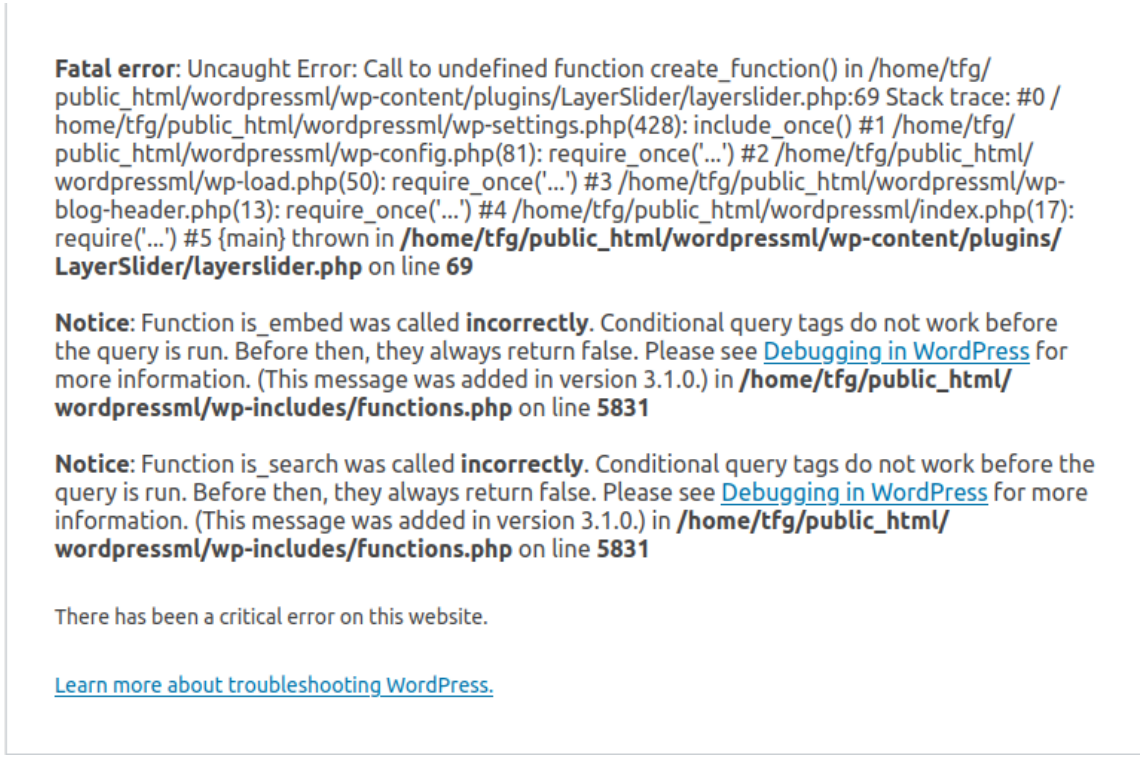

Figura 5.32: Errores de WordPress

Como podemos ver en la Figura 5.32, los errores que estamos teniendo son debidos a problemas con algunos plugins y, en concreto, con archivos PHP. Tras buscar información, puede ser causado por que el salto de versión de PHP es demasiado grande y no son compatibles ciertos plugins que había instalados en el servidor antiguo. Por lo tanto, voy a la documentación oficial del fabricante para ver si es posible trabajar con diferentes versiones de PHP, y efectivamente, sí se puede. Mediante el uso de 2 comandos, que nos instalarán la versión de PHP que nosotros queremos tener, en nuestro caso la versión 7.3 de PHP, que es la que había instalada antiguamente en el otro servidor, podremos tener acceso a esa versión.

Figura 5.33: Comandos para instalar otra versión de php

En la figura 5.33, podremos ver ambos comandos, además de ciertas recomendaciones que te da el fabricante como verificar que la versión que vas a instalar es correcta.

Una vez instalada, dirigiéndonos a la siguiente ubicación mostrada en la Figura 5.34 dentro del panel de administración de Virtualmin, podremos cambiar la versión de PHP.

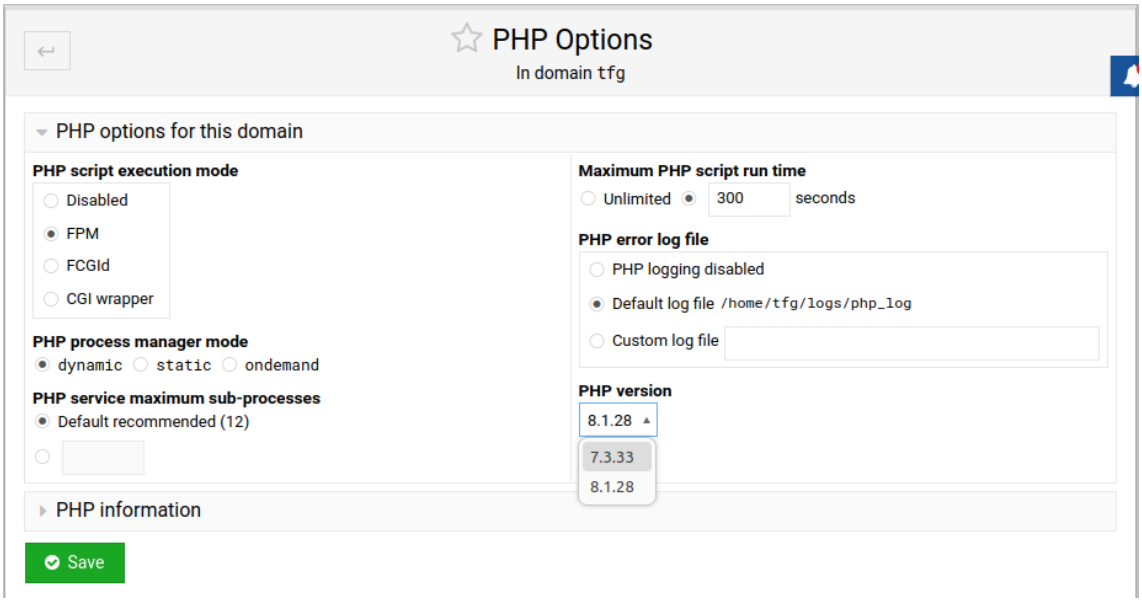

Figura 5.34: Cambiar versión php 1

Al cambiar la versión de PHP, se van a reiniciar todos los módulos y Virtualmin va a hacer todos los cambios necesarios para que todo funcione correctamente.

Ahora, si nos dirigimos a la página de WordPress y refrescamos, podremos ver cómo efectivamente la página ya no da error y podemos acceder de forma completamente normal al sitio web.

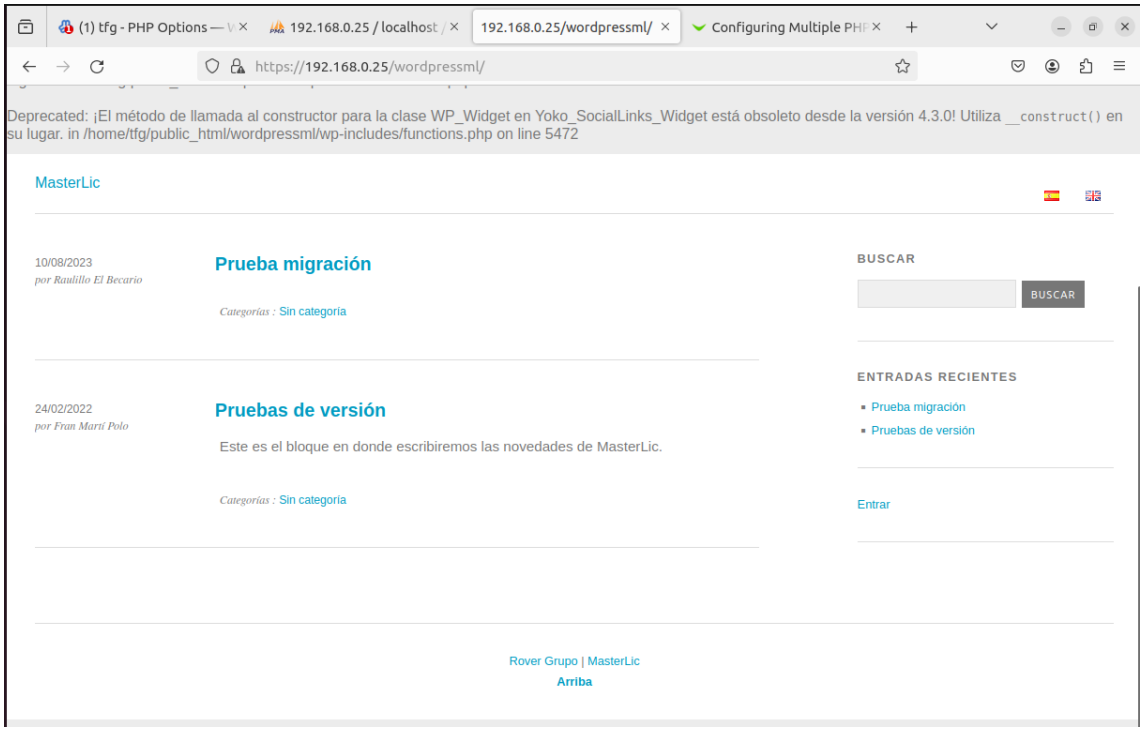

Figura 5.35: Acceso al WordPress migrado

Como podemos ver en la parte superior de la Figura 5.35, nos sigue apareciendo la herramienta de debug. Ahora que ya no nos hace falta, podríais ir a las líneas de código anteriormente mencionadas y volver a dejar el parámetro en false.

Nada más acceder, nos vamos a dar cuenta de que si clicamos en los enlaces, no van a funcionar y nos van a dar errores de que no se encuentra esa página. Esto se debe a que debemos actualizar los enlaces. Para ello, desde el propio panel de administración de WordPress, podemos acceder a él poniendo en el navegador la URL completa seguida de wp-admin, como se indica a continuación:

http://192.168.0.25/wordpressml/wp-admin

Una vez aquí, nos identificamos con nuestro usuario y contraseña de WordPress y nos dirigimos a la sección de “Ajustes” ->“Enlaces permanentes”.

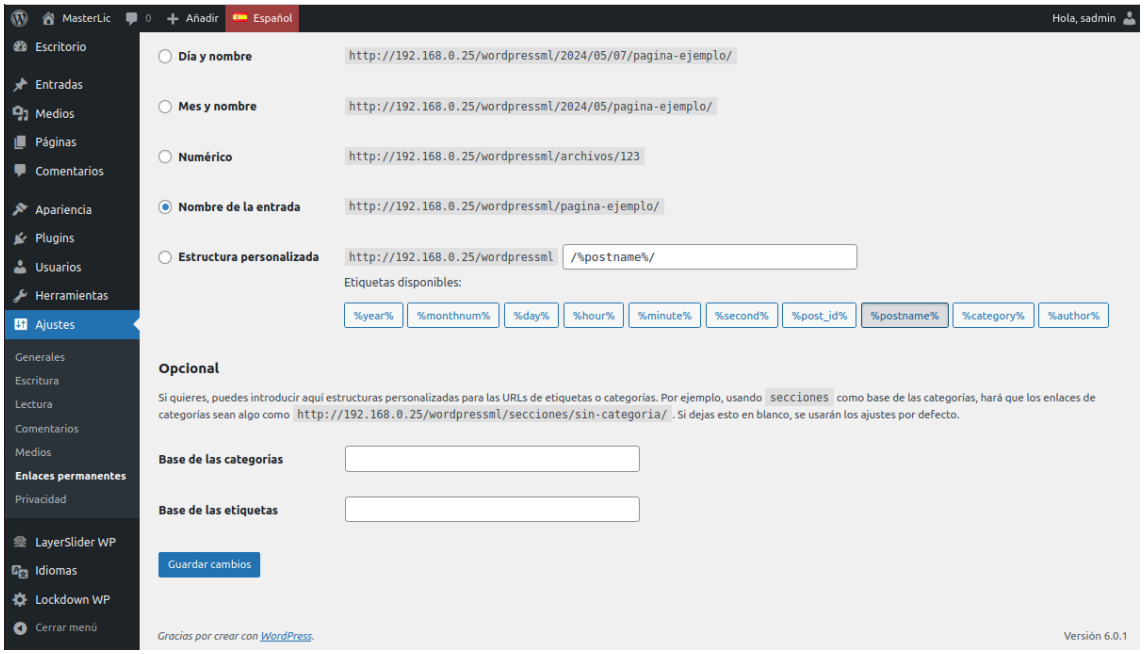

Figura 5.36: Corregir urls de WordPress

Deberemos seleccionar la opción “Nombre de la entrada”, como en la figura 5.36, y una vez marcada, deberemos darle a “Guardar cambios”. Ahora podremos comprobar que efectivamente las URL de nuestro sitio web ya están en funcionamiento.

Ahora vamos a solucionar todos los problemas con los plugins del sitio WordPress. Una vez tengamos estos problemas solucionados, lo que vamos a hacer es actualizar el PHP, ya que para que funcione el sitio de WordPress hemos tenido que usar la versión antigua de PHP. Además, el propio WordPress nos avisa de que tenemos una versión de PHP vulnerable.

Figura 5.37: Aviso de WordPress sobre php

Tras echar un vistazo a los plugins instalados, notamos que fueron instalados hace mucho tiempo, por lo que las versiones son muy antiguas o incluso algunos plugins ya no tienen una versión actualizada y han sido reemplazados por otros. Los plugins que estaban instalados en este WordPress son principalmente de aspecto visual, por lo que no tendrán un gran impacto. Ahora voy a la interfaz de administrador de WordPress y selecciono la parte donde dice “Plugins” ->“Plugins instalados”.

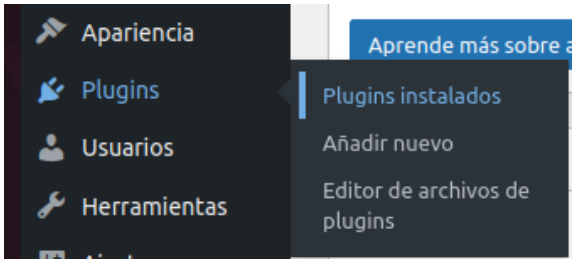

Figura 5.38: Plugins Instalados

Confirmamos la compatibilidad de los plugins y notamos que ninguno es compatible con la versión de PHP. Esta información la podemos encontrar en los detalles de cada plugin. Procedemos entonces a desinstalarlos, pero antes tomamos nota de ellos para poder reinstalar los equivalentes. Para eliminarlos, primero los desactivamos y luego los borramos. Este proceso se puede llevar a cabo desde el menú que nos proporciona WordPress.

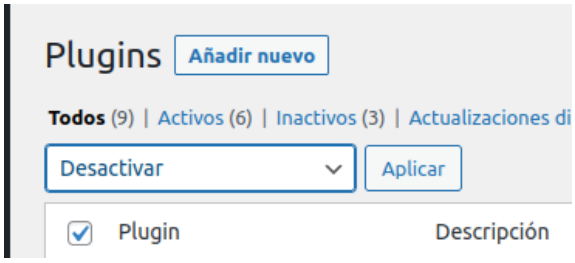

Figura 5.39: Desactivar plugins

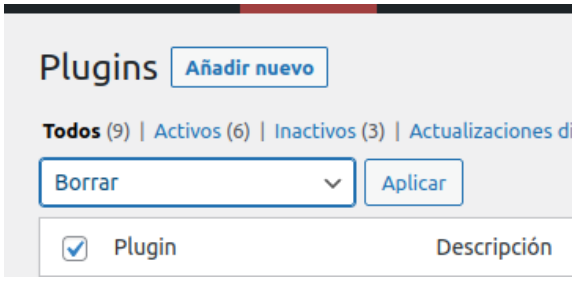

Figura 5.40: Borrar plugins

Actualizamos todo lo posible en nuestro WordPress y los temas desde la pestaña de "Actualizaciones". Simplemente hacemos clic en el botón de actualizar y el sistema se encarga del resto..

Figura 5.41: Actualizar WordPress

Figura 5.42: Todo actualizado

Ahora, sin ningún componente en WordPress que pueda causar alguna incompatibilidad con PHP, nos dirigimos de nuevo a Virtualmin, a la sección de “Virtualmin” ->“Configuración web” ->“Opciones de PHP”. Dentro de esta sección, elegimos la versión más reciente de PHP que tengamos disponible y hacemos clic en “Guardar”.

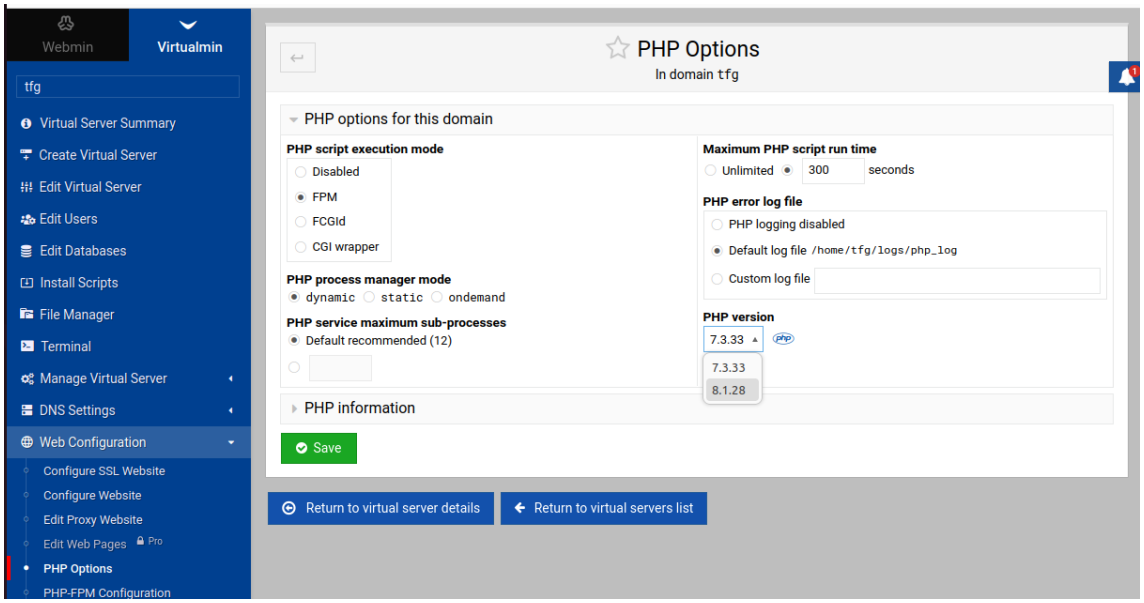

Figura 5.43: Cambio versión php

Ahora, después de esto, vamos a revisar si tenemos algún paquete por actualizar desde la pestaña “Webmin” ->“System” ->“Actualizaciones de paquetes de software”. Es posible que tengamos algunas actualizaciones pendientes. En mi caso, tengo algunas actualizaciones de PHP para obtener la última versión. Seleccionamos todo y hacemos clic en “Actualizar”.

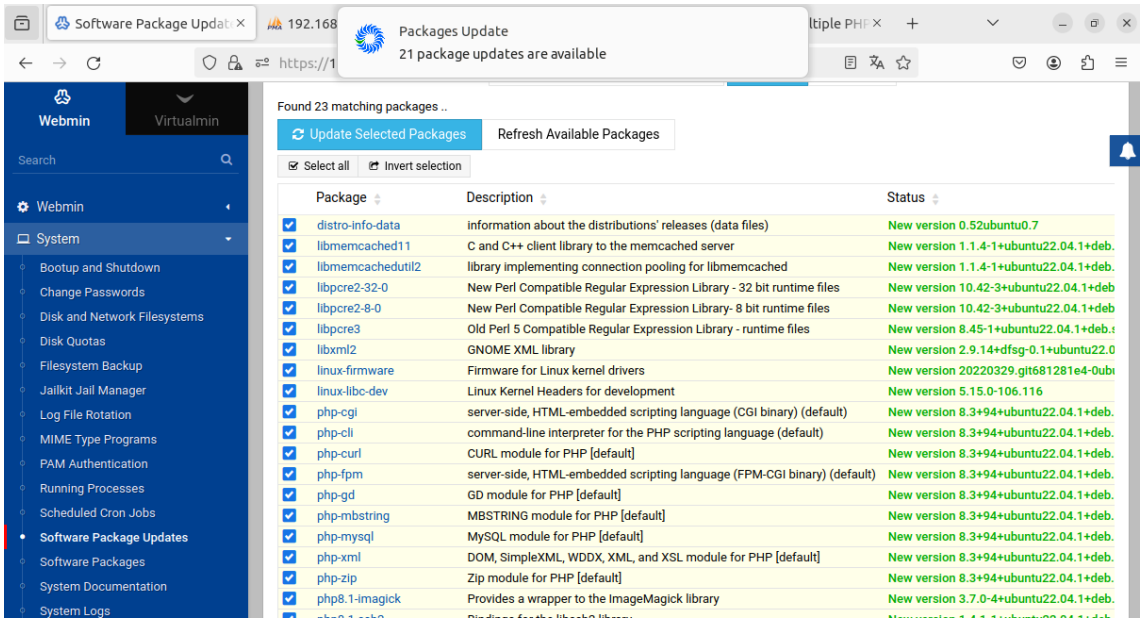

Figura 5.44: Revisar actualizaciones

Repetimos el proceso que vemos en la Figura 5.44 y seleccionamos la última versión, que acabamos de actualizar.

Ahora vamos a WordPress y revisamos que podemos seguir accediendo correctamente a nuestro sitio de WordPress de manera totalmente normal.

Con todo actualizado a las últimas versiones y el WordPress funcionando con normalidad, ahora vamos a reinstalar los plugins estéticos que anteriormente hemos quitado, utilizando las versiones actuales o sus equivalentes más recientes.

5.7 Configuración del firewall en Virtualmin y otros aspectos importantes

Para añadir medidas de seguridad sólidas, vamos a hacer uso de un firewall. Virtualmin nos permite utilizar firewalls integrados en el propio panel. En este caso, podemos utilizar el firewall que viene con Virtualmin, llamado "Firewalld", consultando algunos tutoriales de DigitalOcean [15] que la verdad son muy útiles.

Para activarlo, vamos a entrar en Virtualmin y dirigirnos a la pestaña "Webmin" ->"Networking" ->"FirewallD". Si no está activado, hacemos clic en "Start" para ponerlo en funcionamiento. Una vez activado, veremos una serie de reglas predeterminadas. Sin embargo, no es necesario dejarlas tal como están, ya que esto podría dejar muchos puertos abiertos que no estamos utilizando. Nuestro objetivo es dejar abiertos solo los puertos necesarios y restringir el acceso desde direcciones IP específicas.

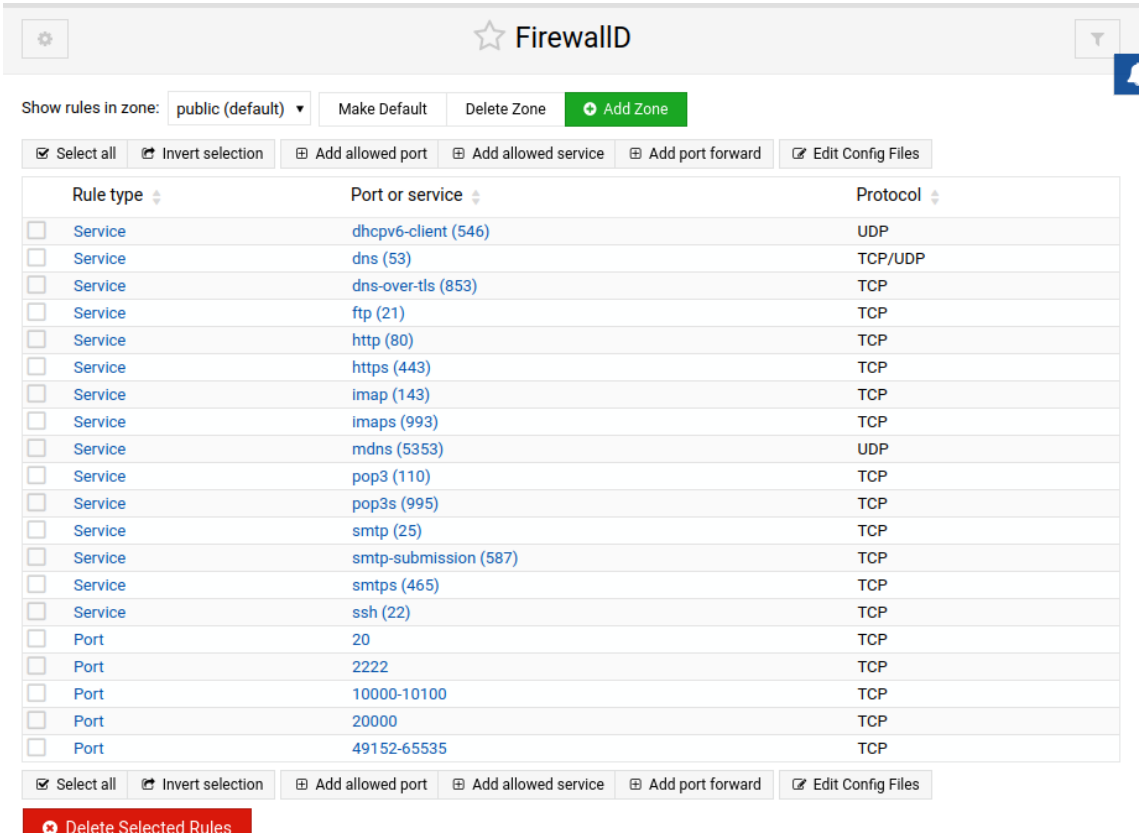

Figura 5.45: Configuración Firewall 1

Ahora lo que vamos a hacer es coger todas las reglas que tenemos puestas de más y las vamos a eliminar y solo vamos a dejar abiertos los puertos necesarios, los puertos 80 y 443 ya que son los puertos que se hacen cargo de http y https que son esenciales para un servidor web, y vamos a cerrar todos aquellos puertos que no estamos utilizando que van a ser todos aquellos que están marcado en la figura 5.46 para eliminarlos.

	Rule type	Port or service	Protocol
☑	Service	dhcpv6-client (546)	UDP
☐	Service	dns (53)	TCP/UDP
☐	Service	dns-over-tls (853)	TCP
☑	Service	ftp (21)	TCP
☐	Service	http (80)	TCP
☐	Service	https (443)	TCP
☑	Service	imap (143)	TCP
☑	Service	imaps (993)	TCP
☑	Service	mdns (5353)	UDP
☑	Service	pop3 (110)	TCP
☑	Service	pop3s (995)	TCP
☑	Service	smtp (25)	TCP
☑	Service	smtp-submission (587)	TCP
☑	Service	smtps (465)	TCP
☐	Service	ssh (22)	TCP
☑	Port	20	TCP
☑	Port	2222	TCP
☐	Port	10000-10100	TCP
☑	Port	20000	TCP
☑	Port	49152-65535	TCP

Figura 5.46: Configuración Firewall 2

Ahora nos vamos a dirigir a la regla que deja abierto el puerto 10000, ya que este es el puerto al que nos conectamos para poder gestionar el panel de control. Entonces, lo que vamos a hacer es restringir las IPs desde las cuales nos podemos conectar a este puerto. Vamos a restringir todas las IPs excepto la mía, que es la única desde la cual quiero tener acceso a controlar el panel de control de Virtualmin. En caso de poner este servidor en producción en cualquier tipo de empresa o negocio, solo deberíamos dejar acceso a este puerto a las IPs relacionadas con el departamento de informática o TI, que serían los encargados de gestionar este servidor.

Aunque desde la interfaz de Virtualmin tenemos una forma más fácil de gestionar el acceso de ciertas IPs, y es desde la propia configuración de Virtualmin. Tenemos un apartado específico para el control de las IPs que acceden a él. Para ello, debemos dirigirnos a “Webmin” ->“Webmin Configuration” ->“IP Access Control”.

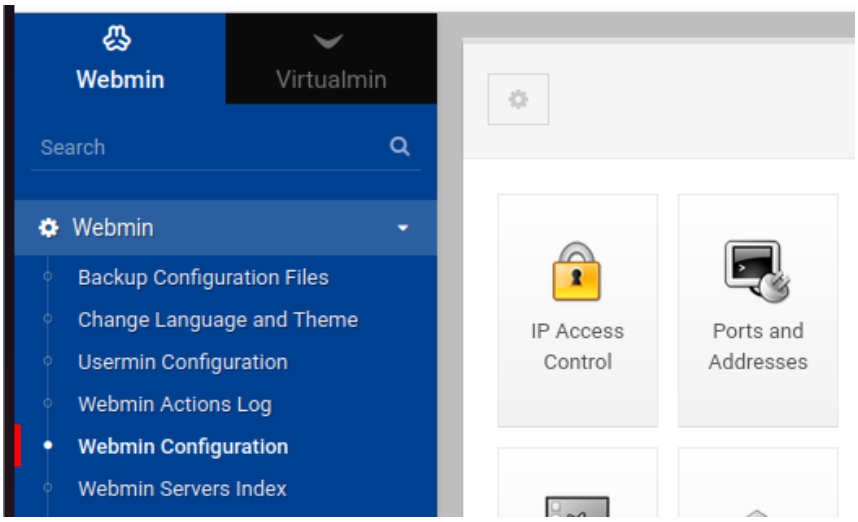

Figura 5.47: IPs con Acceso a Virtualmin

Una vez dentro, nos encontramos con la opción de administrar qué IPs queremos permitir que accedan al panel propio. Esta es una forma mucho más gráfica y sencilla que modificar desde la propia configuración del firewall. Además, con este método podemos añadir y quitar IPs mucho más fácilmente. Sin embargo, debemos tener mucho cuidado al gestionar este tipo de reglas, ya que una mala configuración podría hacer que dejemos de tener acceso a nuestro panel de control.

Por ello, siempre es recomendable tener una snapshot de la máquina antes de hacer este tipo de cambios. O, en el caso de poner la máquina en producción, hacer una copia de seguridad de la misma como respaldo y realizar los cambios en la máquina secundaria. Una vez comprobado que todo está en correcto funcionamiento, podemos aplicar esos cambios sobre la máquina principal o poner directamente en producción la máquina secundaria.

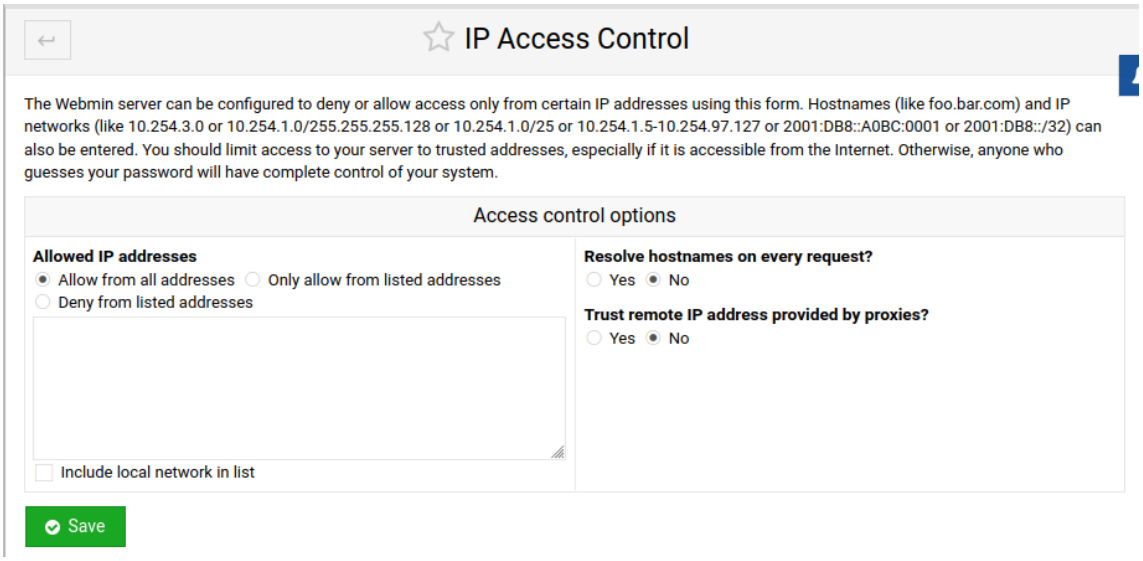

Figura 5.48: IPs con acceso a VirtualMin

Ahora nos dirigimos a la otra pestaña de configuración que vemos en la figura 5.48, donde se indica “Ports and Addresses”. Una vez dentro, observamos que las conexiones desde IPv6 están permitidas. Como no utilizamos IPv6, vamos a marcar que no se pueda acceder con IPv6, de esta forma restringimos aún más el acceso a nuestro servidor.

Además, vamos a definir un tiempo máximo de vida del proceso de Virtualmin para prevenir ataques de fuerza bruta. Marcaremos 600 segundos para este parámetro. También, para prevenir ataques de denegación de servicio (DoS), estableceremos un número máximo de conexiones concurrentes. En mi caso, pondré 50, aunque este número debe adaptarse al tráfico del servidor. Es recomendable comenzar siempre con un número limitado de conexiones e ir aumentándolo según la demanda.

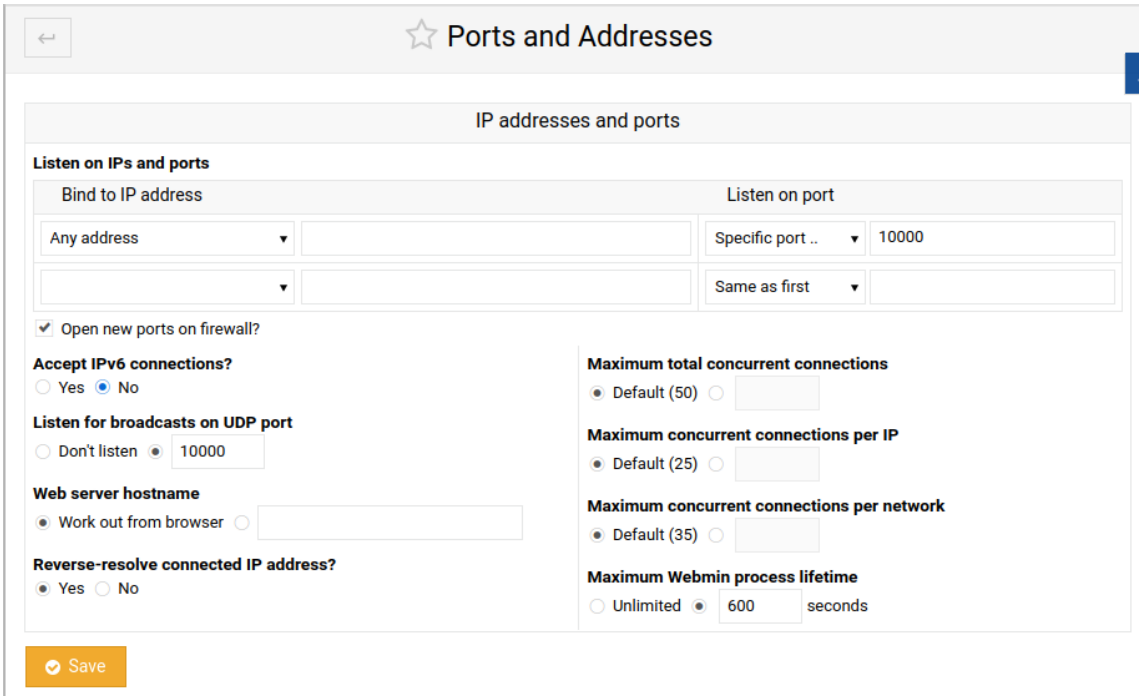

Figura 5.49: Limitar Acceso al servidor

Desde la misma ubicación que en la Figura 5.48, podemos encontrar una de las opciones que nos permite configurar un doble factor de autenticación. Es recomendable activarlo para evitar que, con el robo de las credenciales de acceso, puedan acceder a nuestro panel de control. Con esta medida, además de las credenciales, necesitarán tener acceso a alguno de nuestros dispositivos.

En mi caso, he configurado Google Authenticator en mi teléfono móvil.

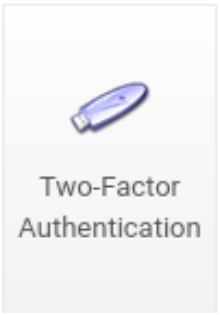

Figura 5.50: Doble factor de autenticación

Tener un certificado SSL (Secure Sockets Layer) es crucial para un servidor web y prácticamente obligatorio para todas las páginas web en la actualidad.

La principal razón de usar un certificado SSL es cifrar la comunicación entre el navegador y el servidor web. Esto garantiza que cualquier dato transferido, como contraseñas o información de tarjetas de crédito, esté protegido contra atacantes que intenten interceptar la información utilizando un ataque “man-in-the-middle”. Sin SSL, esta información estaría expuesta en texto plano.

Al hacer que la pagina tenga ese “https” tan famosos a día de hoy hace que los usuarios confíen más en tu página y tengan esa sensación de seguridad al acceder a tu web,

además que nos va ayudar a darle más visibilidad a nuestra página web , ya que los motores de búsqueda dan mejor posicionamiento a las páginas con certificado.

En nuestro caso vamos a utilizar un certificado autofirmado para hacer las pruebas y ver que efectivamente todo funciona bien cuando utilizamos conexiones https, esto no es lo recomendable para un sitio web público, pero es lo que vamos a utilizar para nuestro entorno de pruebas, más tarde en el apartado de implementación doy una recomendación más concreta sobre que certificado utilizar. Este se generó la primera vez que accedimos a virtualmin en la configuración inicial de virtualmin, por lo tanto ya tenemos un certificado generado, lo lógico aplicado a un servidor web puesto en producción, es tener un certificado de una empresa certificadora de confianza, la cual te genere un certificado valido, este tipo de certificados tiene un precio y se debe de contemplar.

5.8 Securización sitio web

Luego de haber securizado nuestro servidor web desde la interfaz de virtualmin, de una manera eficaz, ahora nos dirigimos a securizar nuestra página web pero desde WordPress, mediante la propia configuración de WordPress y la utilización de algunos plugins que nos ayudaran también a desarrollar esta tarea, para ello hemos consultado diferentes webs y foros como WordPress Codex[16], Sucuri[17] o WPBeginer[18], donde se da ayuda para este tipo de tareas, que son muy interesantes y que nos proporcionan muchas herramientas, .

Primero de todo deberemos de acceder al panel de administración de WordPress (ip/wordpressml/wp-admin) y luego de autenticarnos ya tendremos acceso total.

Para poder securizar nuestro sitio web debemos de tener WordPress junto con sus temas y sus plugins totalmente actualizados, ya que estos vienen acompañados de actualizaciones de seguridad que cubren vulnerabilidades de versiones antiguas.

5.8.1. Restringir acceso de IPs

Vamos a restringir el acceso al panel de administración de WordPress solo desde ciertas IPs, para ello debemos de modificar el archivo .htacces en el directorio en el que se encuentra el WordPress instalado, es posible que este archivo no lo tengamos creado para ello creamos un archivo en blanco con ese nombre, lo editamos con un editor de textos y debemos de poner dentro de ese archivo el contenido que podemos ver en la figura 5.51.

```
<Files wp-login.php>
    Order Deny,Allow
    Deny from all
    Allow from 192.168.0.25
</Files>
```

Figura 5.51: Restringir acceso a Wp-admin

Estas lineas harán que solo se pueda acceder a loguearse al panel de administración desde las IPs que hemos puesto en el documento, en nuestro caso solo he puesto la IP nuestra, en el caso de encontrarse en un entorno real, deberíamos de darle acceso a las IPs de los administradores del sistema.

5.8.2. Plugins

Ahora vamos a instalar unos plugins que nos van a ayudar en la securización de nuestro sitio web, la lista de plugins para ayudarnos en la seguridad es infinita, pero vamos a utilizar los más relevantes y útiles.

Plugin BackUp

el primero de todos es el BackUpWordPress, este plugin es una herramienta para WordPress muy fácil de usar, que no requiere de una configuración, que nos va a ayudar a generar copias de seguridad de nuestra base de datos accediendo a "Herramientas" ->"copias de seguridad" esta nos va a crear diariamente una copia de seguridad de nuestra base de datos diariamente a la hora que nosotros le indiquemos, en este caso a las 23:00, esto va a almacenar la copia de seguridad de los últimos 7 días, esto lo podemos ajustar desde el botón de "ajustes" que tenemos, además podemos ejecutar la copia de seguridad en el momento que nosotros queramos de forma manual.

Figura 5.52: Plugin backUp

Plugin Anti-Spam

Otro de los plugins es el Akismet Anti-spam en una herramienta de antispam más fiables para WordPress, este plugin revisa los comentarios y envíos de formulario contra la base de datos, para prevenir la publicación de contenido malicioso, lo hace de forma automática y filtra los que parecen potencialmente spam, este plugin nos permite ver un log de cuantos spam se han detectado y cuales pueden ser spam pero no se han eliminados.

Cuando lo instalamos, debemos de dirigirnos al panel de los plugins que lo veremos en las opciones de la izquierda y seguimos las instrucciones del plugin para configurarlo, nos aparecerá un mensaje en grande que nos advierte de ello.

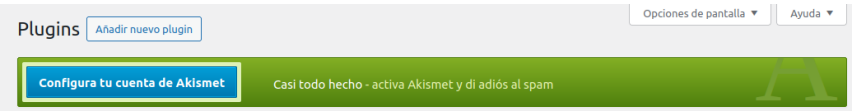

Figura 5.53: Advertencia de configuración del plugin

Clickamos en configurar y nos va a llevar a la página del fabricante, en ella se nos da a elegir un plan de pago, pero el primero de ellos es el que buscamos en mi caso, que es el básico el cual es gratuito y te ofrece el servicio básico anti-spam, los otros planes nos dan más servicios pero hay que pagar por ellos y no entra en nuestro presupuesto el comprar un plugin.

Otro plugin Anti-Spam

Otro plugin para evitar el spam es utilizar un plugin de captcha, el cual añade un captcha muy simple, pero que ya es suficiente para salvarte de muchos spam, el captcha al fin y al cabo el propósito principal de un CAPTCHA es evitar que los bots o cualquier inteligencia artificial automatizada interactúe con los sitios web y asegurarse de que todas las acciones emprendidas en una página sean humanas.

Plugin Doble Factor de autenticación

Ahora vamos a configurar un doble factor de autenticación que para ello debemos de descargar el plugin de google autenticator, esto nos va a ayudar aun más a securizar nuestro panel de WordPress, debido a que aunque nos roben las credenciales, tenemos segundo factor para poder evitar los ataques de robo de credenciales, ya sea mediante phishing o cualquier otro método.

Una vez instalado desde el panel de la izquierda, nos dirigimos a “ajuste” ->“google authenticator” y configuramos el plugin para que nos pida el doble factor a los roles que nosotros elijamos, en mi caso he seleccionado los roles de administrador y editor, para que tengan que configurar este doble factor.

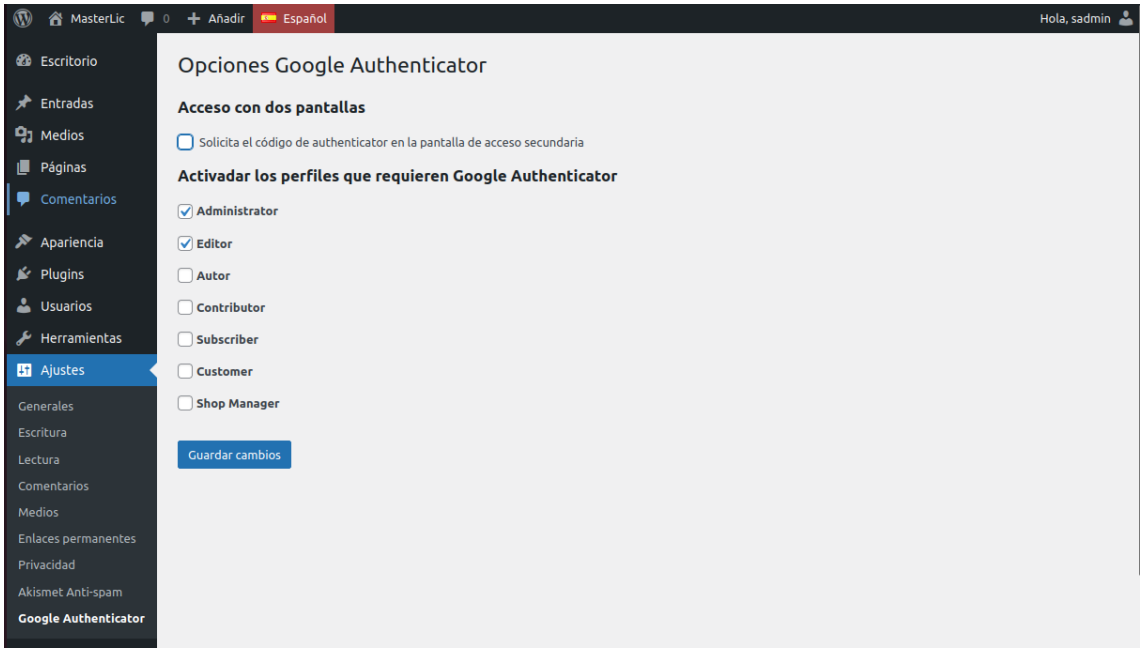

Figura 5.54: Configurar doble factor WordPress

Plugin para evitar ataques de fuerza bruta

Ahora vamos a poner solución a un problema que tiene WordPress, este problema es que a la hora de loguearte como un usuario de nuestro sitio web puedes probar usuarios

y contraseñas de forma infinita, lo que da paso a que seamos atacados con un ataque de fuerza bruta con un software o bot que este probando sin parar todo tipo de contraseñas, si somos conscientes de ello y ponemos una contraseña muy fuerte no deberíamos de preocuparnos de ello, pero para evitar esto aun habiendo puesto una contraseña segura , vamos a instalar un plugin que limita las veces que intentamos acceder, este plugin se llama "Limit Login Attempts Reloaded", una vez instalado veremos que nos sale en las opciones de la izquierda una opción con el mismo nombre que el plugin, nos dirigimos a esta pestaña y podremos configurar para que nos envíen un correo en caso de accesos no autorizados.

Figura 5.55: Configuración límite de intentos de inicio de sesión

Si continuamos llegaremos al panel para administrar el plugin y con la opción "Ajustes generales" , podremos configurar los parámetros para controlar cuantos inicios de sesión fallidos vamos a permitir, los parámetros más interesantes son los siguientes.

En primer lugar cada cuantos intentos de sesión fallidos queremos que se nos notifique con un correo.

Figura 5.56: Notificación por correo

En segundo lugar, podemos configurar los intentos máximos que tiene un usuario antes de que se le bloquee el usuario y durante cuanto tiempo queremos que se le bloquee, en mi caso he puesto en 4 veces los intentos antes de que se bloquee y 20 minutos de bloqueo hasta poder volver a intentarlo.

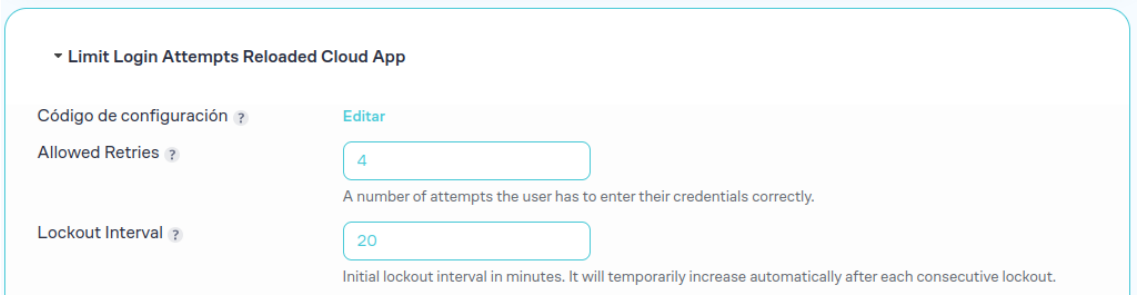

Figura 5.57: Configuración límite de intentos de inicio de sesión 2

Otro Plugin para evitar ataques de fuerza bruta

Otra forma de protegerte de los ataques de fuerza bruta es ocultar esa url en la cual se pueden hacer los inicios de sesión ya que es conocimiento general que WordPress tiene una url predeterminada para todos los sitios web configurados con WordPress, se puede acceder al login con "dominio/pagina/wp-admin", esta url la podemos cambiar a placer con un plugin que se llama "WPS Hide login", una vez instalado, en la parte de "ajustes" tendremos una opción con el nombre del plugin el cual nos va a permitir cambiar el nombre al login de forma muy sencilla como podemos observar en la figura 5.58.

Figura 5.58: Cambio url del login

Una vez cambiado esto, al acceder a las paginas de wp-admin o wp-login va a dar un error en cambio si accedemos a "pepito" nos aparecerá para poder loguearnos en nuestro WordPress.

Activar actualizaciones automáticas

Una configuración muy básica pero que es necesaria si queremos tener nuestro sitio web actualizado y seguro. es tener los plugins, temas y el propio WordPress al orden del día, es decir actualizados y para ello vamos a activar las actualizaciones automáticas de los mismos, esto lo podemos activar desde la sección principal del escritorio veremos una parte donde pone actualizaciones y tendremos una linea resaltada donde podemos activar y desactivar las actualizaciones automáticas del propio WordPress, las de los plugins las activamos desde el menú de la izquierda donde pone plugins y una vez en esa pestaña a la derecha de todo podemos activar las actualizaciones automáticas de los plugins y por último activamos las de los temas, nos dirigimos a "apariencia" ->"Temas" y seleccionamos el tema que utilicemos en nuestro sitio y ahí veremos la opción, en las

figuras 5.59, 5.60 y 5.61 podemos ver donde se encuentra cada una de las opciones que he mencionado.

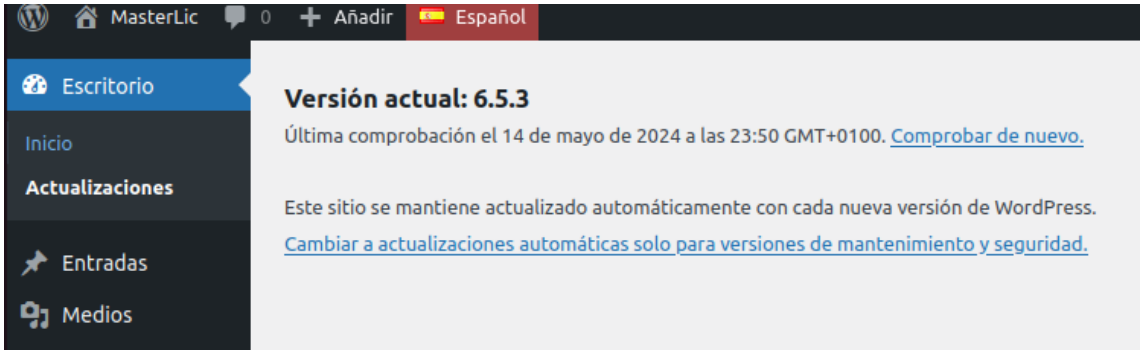

Figura 5.59: Activar actualizaciones automáticas WordPress

Figura 5.60: Activar actualizaciones automáticas plugins

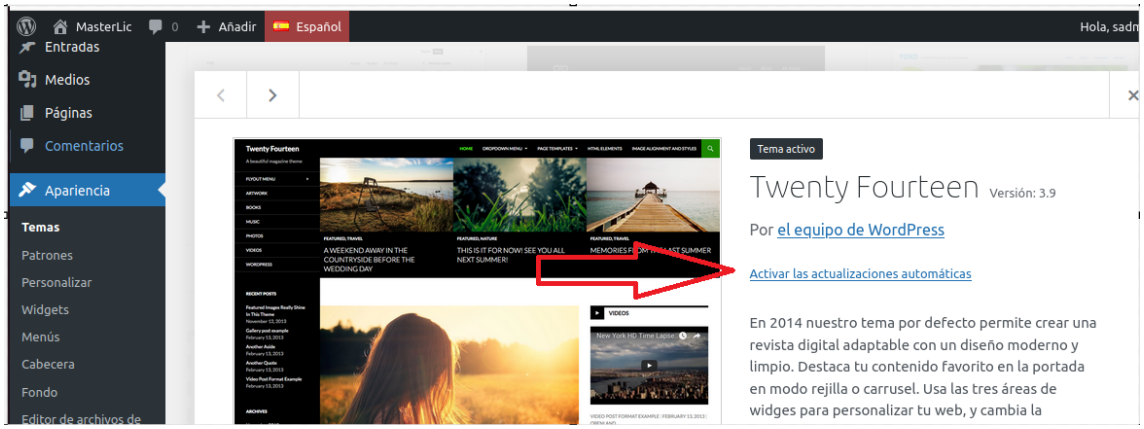

Figura 5.61: Activar actualizaciones automáticas temas

Activar las actualizaciones automáticas nos puede provocar algún error a largo plazo ya que alguno de los plugins o temas deje de ser compatible la nueva versión con la versión de php o WordPress y dejar nuestro sitio web inutilizado, pero para ello hemos configurado un plugin que nos ayuda en la generación de copias de seguridad para poder volver atrás y solucionar el error.

5.9 Plan de realización de copias de seguridad

Para implementar un plan de copias de seguridad, es importante garantizar que los datos estén seguros y accesibles en caso de desastre que afecte al servidor principal. A continuación vamos a dar unas directrices a seguir:

Para ello vamos a tener que programar copias de seguridad desde el panel de control y también desde WordPress, con el plugin que hemos instalado. Para programar las copias de Virtualmin, debemos de acceder al panel de control y en la búsqueda ponemos "Scheduled Backups". Una vez en la pestaña para configurar los backups, tenemos que seleccionar de que virtual server queremos programar las copias de seguridad y elegimos el virtual server deseado. Una vez hecho esto tenemos que decidir sobre que queremos hacer la copia, en nuestro caso lo dejamos como viene por defecto, que es una copia completa. Ahora tendremos que elegir el formato del archivo generado y la ubicación, el formato, he elegido que genere un archivo .zip y para la ubicación, he seleccionado una carpeta que he creado en /home que se llama "BackupVM" y una política de retención de los archivos de 30 días. Por último debemos de seleccionar cada cuanto queremos hacer estas copias, he seleccionado que haga copias todos los días a las 00:00.

Por lo que respecta a las copias de WordPress, en la configuración del propio plugin están detallados los pasos a seguir para configurarlo en el apartado "5.8.2 ->Plugin BackUp".

Con las copias de seguridad hechas en ambos lados, ahora tendremos que además de tener las copias de seguridad en el propio servidor, sacarlas fuera de el. Para ello, podemos subir los archivos generados a un sistema de almacenamiento en la nube como puede ser dropbox, Google Drive, One Drive o Amazon S3, etiquetándolos por semanas, de modo que la primera semana guardaríamos una copia de seguridad en cualquiera de las herramientas que hemos seleccionando antes, en mi caso OneDrive, etiquetando esta primera semana con la etiqueta "S1", de esta forma pasarían las semanas hasta la "S8", en ese punto sobrescribiríamos la "S1", teniendo copias de seguridad de 8 semanas, para llevar un control de ello, creamos en one note o cualquier otra herramienta donde podamos anotar, que semana del calendario está en cada una de las etiquetas, por ejemplo: S1 semana del 15 al 21 de abril 2024. Esto se debe actualizar cada semana hasta la semana 8 y sobrescribir la semana 1 en la semana 9, de esta forma llevamos un control de que copias hay en cada una de las carpetas con la etiqueta "S" y en que fecha se realizaron.

Para mayor seguridad podríamos copiar estos BackUp en un disco duro externo.

CAPÍTULO 6
Implementación

Una hemos tenemos todo el desarrollo hecho y todo en correcto funcionamiento en un ámbito de pruebas, ahora os daremos las directrices para poder poner este servidor web en producción en cualquier lugar. Esto es un trabajo el cual proponemos una solución general la cual vamos a poder incorporar en cualquier sistema, la idea no es ponerlo en producción en la realización del propio proyecto si no que el proyecto debe servir como guía para alguien que quiera implementar este servicio.

6.1 Obtener Dominio y Registro DNS

Primero de todo debemos de tener un dominio adquirido en algún proveedor de dominios, algunos de los más famoso son Namecheap o Google Domains, tiene que ser un nombre de dominio disponible el cual debemos de adquirir.

Una vez tengamos el dominio adquirido deberíamos de crear un registro en el DNS publico en el cual vinculemos el nombre de dominio elegido con la IP publica de nuestro servidor.

Ahora que tenemos el registro en la DNS pública deberíamos de apuntar dentro de nuestra DNS apuntar la IP al nombre del dominio.

Ahora tendremos que tener paciencia ya que estos cambios de DNS pueden tardar un tiempo en hacerse efectivos.

6.2 Cambio en Firewall y archivos de configuración

Para que nuestro servidor sea capaz de aceptar esas peticiones https, deberíamos de tener abierto el puerto 443 y hacer una redirección en el firewall de las peticiones que nos lleguen a la IP pública redirigirlas a la IP del servidor web

Ahora tendremos que cambiar ciertos archivos de configuración donde se apunta a nuestro dominio, el primero de ellos el archivo de configuración de apache que lo encontramos en la siguiente localización:

/etc/apache2/sites-available/

cambiamos donde se indica “ServerName” y “ServerAlias” por el nombre del dominio, luego habilitamos el sitio we con “sudo a2ensite nombreDeDominio.conf” y reiniciamos Apache.

6.3 Cambios en base de datos

Después de que realicemos estos pasos iremos a nuestro PhpMyAdmin, para cambiar el enlace de acceso a nuestro pagina web por el nuevo nombre de dominio esto lo hemos cambiado durante el desarrollo en la figura 5.25.

6.4 Certificado SSL

Por último, es fundamental utilizar un certificado SSL, el cual podemos obtener de forma gratuita a través de Let's Encrypt o comprarlo mediante un proveedor de dominios o una empresa certificadora. Mi recomendación es utilizar un certificado de Let's Encrypt, que se renueva automáticamente cada 3 meses, lo que facilita su gestión y garantiza la seguridad continua del sitio web. Virtualmin puede gestionar esta renovación automática, lo que reduce el esfuerzo requerido. Sin embargo, comprar un certificado SSL también es una opción válida y puede ofrecer ventajas adicionales, como soporte más especializado y garantía aumentada, dependiendo del proveedor.

CAPÍTULO 7
Pruebas

Una vez todo en funcionamiento, tuvimos que realizar diferentes pruebas para comprobar que efectivamente esta todo en correcto funcionamiento. Esto es de máxima necesidad para que luego no nos encontremos con ningún tipo de error cuando queramos poner en producción nuestro servidor.

7.1 Pruebas de funcionalidad

7.1.1. Pruebas de Navegación

En esta prueba, nos vamos a asegura de que al acceder a nuestra página web, todos los enlaces efectivamente responden y nos llevan a donde queremos, para ello simplemente accedemos al sitio web como si fuésemos cualquier usuario y en el caso de la página que hemos migrado pues solo tenemos unos pocos links a los que acceder, accedemos a ellos y vemos que efectivamente no nos da ningún error.

7.1.2. Pruebas de Funcionalidad del Usuario

La página web que hemos migrado no tiene muchas funcionalidades para el usuario, ya que la pagina web migrada es una página donde hay publicados diferentes artículos y la gente accede a leer los artículos, por lo tanto solo hay un buscador, comprobamos el correcto funcionamiento del buscador y vemos que efectivamente va todo perfecto.

7.2 Pruebas de Compatibilidad

7.2.1. Compatibilidad entre navegadores

Para comprobar este apartado, hemos intentado acceder a la web desde diferentes navegadores para ver si en alguno de ellos daba algún error, ha funcionado correctamente en todos, hemos accedido desde Firefox, Google Chrome, Microsoft Edge y safari, la búsqueda ha sido efectiva en todos los casos.

7.2.2. Compatibilidad entre dispositivos

Para comprobar la compatibilidad entre dispositivos he accedido desde mi propio ordenador que es el principal objetivo de una página web de estas características y además de dos terminales móviles, uno con sistema operativo Android y otro con IOS.

7.3 Pruebas de seguridad

7.3.1. Pruebas de acceso al panel de administración

En este apartado nos hemos querido asegurar que las reglas de restricción de IPs funcionan correctamente, para ello hemos intentado acceder al panel de administración de virtualmin mediante el uso de otro ordenador el cual tiene asignada un IP diferente a la mía y como hemos configurado solo el acceso desde la IP de nuestro portátil desde el otro ordenador de sobremesa es incapaz de acceder por lo que podemos confirmar que esta medida de seguridad es completamente valida.

7.3.2. Pruebas de Autenticación y autorización

Nos aseguramos de que la autenticación y la autorización funcionan correctamente en nuestro servidor, para ello de los usuarios que tenemos creados hemos dado un rol a cada uno, entre ellos el rol de administrador y otro de editor, para esos roles que son los que se van a tener que autenticar para acceder al panel de administración de WordPress les he activado el doble factor de verificación y efectivamente este funciona con la app de Google y de esta forma también comprobamos que efectivamente, la url para acceder al login ha cambiado para poder evitar ataques por fuerza bruta.

Además comprobamos que efectivamente un usuario que accede a la página web no puede acceder a nada que tenga que ver con la administración del sitio web.

7.3.3. Pruebas de BackUp y Recuperación

Para hacer una prueba de que las copias de seguridad están haciéndose correctamente y todo funciona correctamente, lo que debemos de hacer es ir dentro del panel de administración de WordPress a “Herramientas”->“Copias de seguridad” y ahí dentro nos dirigimos a ejecutar ahora mismo una copia de seguridad.

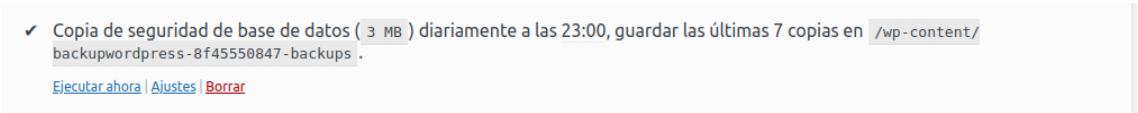

Figura 7.1: Copia de seguridad de Prueba

Una vez hecha la copia de seguridad nos dirigimos al directorio que nos indica donde tenemos almacena la copia de seguridad de nuestra base de datos, en nuestro caso en wp-content/.

Ahora nos disponemos a cambiar algo dentro de una de las entradas que tenemos ya creadas o creamos una nueva ,para que cuando restauremos la copia de seguridad veamos si ha desaparecido lo que hemos creado posterior a la copia de seguridad para confirmar que efectivamente la copia de seguridad ha tenido efecto.

Ahora nos dirigimos al directorio que se nos indica y veremos que encontramos diferentes backups con la extensión .sql, de entre todos elegimos el que esta creado con la fecha y hora que nosotros hemos creado manualmente.

Una vez localizado el archivo .sql que queremos restaurar nos dirigimos al panel de PhpMyAdmin, una vez dentro elegimos la base de datos que queremos restaurar en nuestro caso la base de datos del WordPress, seleccionamos la opción de “Importar” e importamos el archivo sql que hemos localizado anteriormente, una vez hecho, nos dirigimos

a nuestro WordPress y vemos que efectivamente eso que habíamos creado nuevo, luego de la creación de la copia de seguridad de nuestra base de datos ha desaparecido.

CAPÍTULO 8
Conclusiones

A lo largo de este proyecto de final de grado, hemos creado y configurado un servidor web seguro y robusto, garantizando su estabilidad y funcionalidad a largo plazo. Utilizamos la última versión de Ubuntu LTS 22.04, lo que nos proporciona una base sólida y actualizable para el servidor.

Optamos por Virtualmin como panel de control, facilitando la gestión y monitorización del servidor. Migramos exitosamente un sitio web de WordPress, superando problemas de compatibilidad con PHP gracias a herramientas de debug y recursos en línea.

Una vez operativo, securizamos el servidor y el sitio web usando FirewallD, herramientas de Virtualmin y plugins de WordPress. Implementamos actualizaciones automáticas y copias de seguridad para asegurar el mantenimiento continuo.

Este trabajo ha sido un reto que me ha hecho aprender mucho en estos campos además de ver la gran cantidad de formas que hay de hacer una misma cosa, viendo que no hay una solución única, que hay miles de soluciones y todas validas, por eso hay que tener la mente abierta a trabajar de formas diferentes a las habituales, también nos ha quedado demostrado que la planificación y ejecución meticulosa a la hora de la creación de un servidor es clave.

Lo que queda claro, es que la contante evolución en el campo de las tecnologías web hace esencial el mantenerse actualizado con las nuevas tecnologías emergentes para poder hacer uso de ellas, ya que son herramientas fantásticas.

Además he podido poner en práctica muchas de las cosas aprendidas en la universidad y mucho más, ya que he tenido que investigar sobre ciertos campos en los cuales el conocimiento previo no era suficiente, las herramientas que he utilizado, que ya conocía previamente gracias a mis estudios cursados, han sido sobre todo el manejo con sistemas operativos linux, en el cual me he sentido muy cómodo y seguro a la hora de utilizar todo tipo de comandos, además del uso de virtualbox como herramienta para crear mi máquina virtual, que ha sido el entorno de trabajo en muchas de las practicas de la universidad, como el uso de Virtualmin, que ha sido el único panel de control del que ya tenía un conocimiento previo, por una de las asignaturas de la rama de Tecnologías de la información, lo que me ha hecho darme cuenta que la universidad te da las herramientas necesarias para que luego tu puedas ser autosuficiente en tu trabajo y autoabastecerte de conocimientos y métodos para poder hace uso de ellos.

Bibliografía

[1] Blog tool, publishing platform, and cms 8211; wordpress.org. Disponible en https://wordpress.org/.

[2] Linux.org. Disponible en https://www.linux.org/.

[3] Welcome! - the apache http server project. Disponible en https://httpd.apache.org/.

[4] Mysql. Disponible en https://www.mysql.com/.

[5] Php: Hypertext preprocessor. Disponible en https://www.php.net/.

[6] Francisco Rubén Serrano López. *Plan de migración de un servidor web basado en Windows Server 2008 e IIS*. PhD thesis, Universitat Politècnica de València, 2011.

[7] Javier Hernández Sanz. *Desarrollo completo de un sitio web con buenas prácticas*. PhD thesis, Universitat Politècnica de València, 2018.

[8] Bitnami: Packaged applications for any platform - cloud, container, virtual machine. Disponible en https://bitnami.com/.

[9] Hosting platform of choice. Disponible en https://www.cpanel.net/.

[10] Plesk Documentation Team. Documentation and help portal for plesk obsidian. Disponible en https://docs.plesk.com/.

[11] Virtualmin — open source web hosting control panel. Disponible en https://www.virtualmin.com/.

[12] Vesta control panel. Disponible en https://vestacp.com/.

[13] Ispconfig hosting control panel. Disponible en https://www.ispconfig.org/.

[14] Oracle vm virtualbox. Disponible en https://www.virtualbox.org/.

[15] DigitalOcean. Tutorials | digitalocean. Disponible en https://www.digitalocean.com/community/tutorials/.

[16] WordPress Codex. Hardening wordpress – advanced administration handbook | developer.wordpress.org. Disponible en https://wordpress.org/support/article/hardening-wordpress/.

[17] Sucuri. Wordpress security: How to secure amp; protect wordpress (2024 guide). Disponible en https://sucuri.net/guides/wordpress-security/.

[18] WPBeginner. The ultimate wordpress security guide - step by step (2024). Disponible en https://www.wpbeginner.com/wordpress-security/.

APÉNDICE A

ODS (OBJETIVOS DE DESARROLLO SOSTENIBLE

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				x
ODS 2. **Hambre cero.**				x
ODS 3. **Salud y bienestar.**		x		
ODS 4. **Educación de calidad.**				x
ODS 5. **Igualdad de género.**				x
ODS 6. **Agua limpia y saneamiento.**				x
ODS 7. **Energía asequible y no contaminante.**				x
ODS 8. **Trabajo decente y crecimiento económico.**		x		
ODS 9. **Industria, innovación e infraestructuras.**				
ODS 10. **Reducción de las desigualdades.**		x		
ODS 11. **Ciudades y comunidades sostenibles.**				x
ODS 12. **Producción y consumo responsables.**				x
ODS 13. **Acción por el clima.**				x
ODS 14. **Vida submarina.**				x
ODS 15. **Vida de ecosistemas terrestres.**				x
ODS 16. **Paz, justicia e instituciones sólidas.**				x
ODS 17. **Alianzas para lograr objetivos.**				x

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

A.1 Objetivo de Desarrollo Sostenible 3: Salud y bienestar

El ODS 3 se centra en garantizar una vida sana y asegurar el bienestar para todos en todas las edades. Aunque puede que no sea evidente, la configuración de un servidor para alojar sitios web puede tener un impacto importante en la salud y el bienestar, especialmente en el ámbito de la salud mental.

En primer lugar, un servidor bien configurado y seguro garantiza que los sitios web, incluidos los que brindan servicios de salud mental y bienestar, estén siempre disponibles y sean de fácil acceso. Esto es crucial para las plataformas que ofrecen asesoramiento psicológico, consultas médicas en línea y recursos de autoayuda. La disponibilidad constante de estos servicios puede hacer una gran diferencia en la vida de las personas que buscan apoyo.

Además, la facilidad de administración del servidor, que hemos obtenido mediante la elección de un panel de control adecuado, reduce en gran cantidad el estrés y la carga de trabajo del personal encargado de su mantenimiento. Un entorno de trabajo menos estresante y más manejable contribuye directamente al bienestar mental de los trabajadores de tecnología de la información (TI).

Por otro lado, la migración de sitios web a un servidor más eficiente puede mejorar la velocidad y la estabilidad de las páginas, proporcionando una mejor experiencia de usuario.

A.2 Objetivo de Desarrollo Sostenible 8: Trabajo decente y crecimiento económico

El Objetivo de Desarrollo Sostenible (ODS) 8 se centra en potenciar el crecimiento económico inclusivo y sostenible, el empleo pleno y productivo para todos. La creación y configuración de un servidor para alojar sitios web, como el que se ha realizado en el proyecto, tiene un impacto significativo en este objetivo.

El desarrollo de infraestructura tecnológicas, como servidores para alojar sitios web, facilita la creación de nuevas oportunidades de empleo y ayuda al crecimiento de la economía digital. Los sitios web migrados y alojados en el nuevo servidor pueden incluir tiendas online, plataformas de servicios y portales de contenido que generan empleo y promueven el comercio electrónico. Esto es especialmente relevante en un mundo cada vez más digitalizado, donde la presencia onlone es crucial para la competitividad de las empresas.

Además, la configuración de un servidor eficiente y seguro permite a las pequeñas y medianas empresas (PYMES) acceder a este tipo de tecnnologias sin la necesidad de inversiones iniciales grandes. Esto contribuye a un entorno más equitativo donde las PYMES pueden competir con grandes corporaciones, hasta cierto punto, fomentando un crecimiento económico más inclusivo y sostenible.

A.3 Objetivo de Desarrollo Sostenible 10: Reducción de las desigualdades

El ODS 10 busca reducir la desigualdad. La implementación de un servidor accesible y eficiente contribuye a este objetivo al proporcionar herramientas tecnológicas que pueden ser utilizadas por individuos y organizaciones de diferentes contextos económicos y sociales.

Al facilitar la migración y el alojamiento de sitios web, el proyecto permite a las pequeñas empresas y a los emprendedores acceder a una infraestructura de calidad, sin los altos costos que conlleva el establecimiento de servidores. Esto reduce las barreras de entrada al mercado digital, permitiendo a más personas participar en la economía onlie.

Además, la disponibilidad de recursos y servicios online puede ayudar a reducir las desigualdades geográficas. Las personas en áreas rurales o desfavorecidas pueden acceder a la misma información y oportunidades que aquellas en áreas urbanas, contribuyendo a una distribución más equitativa de los recursos y oportunidades.