



**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**  
**LICENCIATURA EN CIBERSEGURIDAD**  
**DESARROLLO WEB**

**Proyecto Semestral**

**Integrantes:**

**Eduardo Samaniego 8-964-2469**

**Ricardo Guardado 8-999-1102**

**Boris De León 8-997-94**

**Alan Quiroz 8-1009-2330**

**Luis Sánchez 8-976-1365**

**Azael Acosta 8-1001-717**

**Tema:**

**PHP+MariaDB+CSS+HTML**

**Grupo:**

**1S3122**

**Profesor:**

**José Chirú**

**SEMESTRE II, 2023**

**Fecha:**

**Viernes 15 de diciembre de 2023**

## Índice

Índice de figura .....	4
Introducción .....	9
Documentación de la red .....	10
Configuración del servidor Apache .....	15
Creación de la máquina virtual .....	15
Instalación del servidor para la maquina apache .....	18
Instalación del net-tools. ....	26
Instalación del apache y las librerías del php. ....	27
Servidor de bases de datos.....	29
Creación de máquina virtual .....	29
Instalación del servidor para MariaDB .....	30
Instalación del paquete Net-Tools.....	35
Configuración de ip estatica (192.168.0.23 Provisional) .....	35
Carga y configuración de la base de datos. ....	37
Configuración del mariadb y apache en ApacheServer .....	41
IP final servidor Apache .....	42
IP Final servidor MariaDb.....	42
Importando el proyecto a Apache .....	43
Creación de Nuevos Usuarios .....	45
Error del CSS .....	47
Resolución del error.....	48
Hardening .....	51
SonarCloud para escaneo de vulnerabilidades .....	58
Vulnerabilidades encontradas y parcheadas en el desarrollo del proyecto .....	63
Consejos de vulnerabilidades a nivel de servidores .....	63
Consejos de vulnerabilidades a nivel de redes.....	64
Vulnerabilidades a nivel de escaneo.....	67
Vulnerabilidades encontradas.....	68
Consejos para mitigar estas vulnerabilidades.....	72

Conclusión .....	76
Bibliografía.....	77

## Índice de figura

figura 01. Tp-link-Deco M4 Sistema Wi-Fi.....	10
Figura 02. app Tp-link-Deco. ....	10
figura 03. App configuración Tp-link-Deco.....	11
figura 04. App configuración Tp-link-Deco.....	11
figura 05. App configuración Tp-link-Deco.....	12
figura 06. App configuración Tp-link-Deco.....	12
figura 07. App configuración en más. ....	13
figura 08. App configuración en más. ....	13
figura 09. App configuración conexión a Internet. ....	13
figura 10. App configuraciones avanzadas.....	14
figura 11. Configuración IP LAN .....	14
figura 12. Configuración del servicio DHCP.....	15
figura 13. Diagrama de nuestra Intarnet.....	15
figura 14. Nombre y sistema operativo.....	16
figura 15. Hardware .....	16
figura 17. resumen .....	17
figura 18. Adaptador de red .....	18
figura 19. Adaptador de red 2.....	18
figura 20. Inicio instalador.....	19
figura 21. idioma .....	19
figura 22. actualizaciones.....	20
figura 23. teclado .....	20
figura 24. Adaptador de red instalación.....	21
figura 25. Disco duro .....	21
figura 26. Storage configuración .....	22
figura 27. perfiles .....	22
figura 28. openssh .....	23
figura 29. Servicios opcionales.....	24
figura 30. Instalación.....	24
figura 31. Instalación del sistema.....	25
figura 32. Inicio por powershell .....	25

figura 33. Updates .....	26
figura 34. Upgrades .....	26
figura 35. Instalación de net-tools .....	26
figura 36. Configuración del archivo .....	27
figura 37. Tarjeta provisional .....	27
figura 38. Aplicando la configuración.....	27
figura 39. Instalación del apache y parches .....	27
figura 40. Instalación del módulo.....	28
figura 41. Apache2 .....	28
figura 42. mysql.....	29
figura 43. RAM.....	29
figura 44. Resumen del sistema .....	30
figura 45. idioma .....	30
figura 46. teclado .....	31
figura 47. Resumen de las conexiones .....	31
figura 48. almacenamiento .....	32
figura 49. Perfil del sistema.....	32
figura 50. Instalando el openssh .....	33
figura 51. Instalación del sistema.....	33
figura 52. Ingreso al sistema .....	34
figura 53. update .....	34
figura 54. upgrade .....	35
figura 55. Instalacion de net-tools .....	35
figura 56. Ingreso por ssh.....	36
figura 57. Entramos al config .....	36
figura 58. IP .....	36
figura 59. Aplicamos la configuración .....	37
figura 60. Viendo la interfaz de ip .....	37
figura 61. Ingresando la base .....	37
figura 62. Instalacion del mariadb server.....	38
figura 63. Instalacion segura .....	38
figura 64. Cambiando la password.....	39
figura 65. Quitando los usuarios anónimos .....	39

figura 66. Quitamos los test .....	39
figura 67. Terminando configuración .....	39
figura 68. Entrando al mariadb .....	40
figura 69. Creando la base de datos.....	40
figura 70. Creando el usuario de base.....	40
figura 71. Abriendo el puerto.....	40
figura 72. Configurando el mariadb .....	40
figura 73. Comentando el bind-address.....	41
figura 74. Importando la base de datos .....	41
figura 75. Instalando el cliente.....	41
figura 76. Cambiando el usuario de base de datos.....	41
figura 77. Cambiando la ip del apache .....	42
figura 78. Cambiando la ip del mariadb .....	42
figura 79. Conectando de manera remota.....	42
figura 80. Seteando las credenciales dentro de la app .....	43
figura 81. Instalando unrar.....	43
figura 82. Poniendo los archivos .....	44
figura 83. Archivos en su carpeta.....	44
figura 84. Comprobando la app .....	44
figura 85. Panel .....	45
figura 86. Error de permisos.....	45
figura 87. Permisos.....	45
figura 88. Usuario del profesor .....	46
figura 89. Usuario de boris .....	46
figura 90. Usuario de Eduardo .....	47
figura 91. Usuarios .....	47
figura 92. Error de css.....	48
figura 94. Error de referencias .....	48
figura 95. Nuevos archivos .....	49
figura 96. importando los nuevos archivos.....	49
figura 97. Descomprimiendo los archivos.....	50
figura 98. Entrando a la carpeta.....	50
figura 99. Moviendo las carpetas.....	50

figura 100. Creando un nuevo css .....	50
figura 101. Ingresando las instrucciones.....	51
figura 102. layout .....	51
figura 103. Cambiando las referencias del header .....	51
figura 104. Reiniciando apache2 .....	51
figura 105. Borramos el index .....	51
figura 106. Borramos info.php .....	52
figura 107. Vemos el baner .....	52
figura 108. Nuevas políticas .....	52
figura 109. Volvemos a ver el baner .....	53
figura 110. Instalamos fail2ban .....	53
figura 111. Vemos la configuración.....	53
figura 112. Vemos los intentos.....	54
figura 113. Instalamos fail2ban en mariadb.....	54
figura 114. nmap .....	54
figura 115. Instalando el unattended-upgrades .....	55
figura 116. Decidiendo .....	55
figura 117. Viendo la configuración .....	55
figura 118. Verificamos las actualizaciones.....	56
figura 119. Instalamos el paquete en el mariadb .....	56
figura 120. Quitamos el index .....	57
figura 121. Abrimos los puertos.....	57
figura 122. Vemos los puertos .....	57
figura 123. Vemos los puertos de mariadb .....	58
figura 124. Entrando a Sonar Cloud .....	58
figura 125. Creando el repositorio en GitHub.....	59
figura 126. Arrastrando los archivos .....	59
figura 127. repositorio.....	60
figura 128. Configurando el SonarCloud .....	60
figura 129. Dándole acceso al repositorio.....	61
figura 130. Realizando el análisis .....	61
figura 131. Resumen de escaneo .....	62
figura 132. Viendo los puntos de compromiso .....	62

figura 133. Cantidad escaneada en Sonarcloud.....	63
figura 134. Cantidad escaneada en Codacy .....	63
figura 135. Configuración de contraseñas .....	64
Figura 136. Supervisión de subidas y descargas .....	65
Figura 137. Restricción del equipo por medio de su Mac.....	65
Figura 138. Configuración del modo parental y mas .....	66
Figura 139. Configuración de bloqueo al acceso a internet.....	66
Figura 140. Historial de trafico de un equipo cliente.....	67



## Introducción

En el vasto panorama de la seguridad informática, la ciberseguridad en el ámbito del desarrollo web trasciende la aparente simplicidad de HTML y CSS en el front-end. Se extiende al complejo universo del back-end, un terreno tan extenso y diverso como el propio cosmos. En un mundo donde los ataques cibernéticos son moneda corriente y la certeza de la seguridad absoluta en la red es una quimera, este proyecto se adentra en la creación de una base sólida. Cada semana, exploramos diversas implementaciones de tecnologías clave como PHP, MySQL (MariaDB) y Apache, desplegadas en un servidor, inmersos en un entorno controlado y orientado a objetos.

El propósito fundamental de este proyecto es destacar la importancia de realizar análisis de seguridad exhaustivos. Esto incluye el escaneo de vulnerabilidades y el análisis estático de código, con el fin de identificar y corregir de manera proactiva vulnerabilidades críticas, entre otros modelos de amenazas. Este proceso se lleva a cabo en una red local, proporcionando un entorno ejemplar que refleja lo que enfrentaremos en el ámbito laboral como futuros agentes de ciberseguridad.

Enfatizamos que este proyecto no solo se trata de aprendizaje teórico, sino de una inmersión práctica en las complejidades y desafíos del desarrollo web seguro. Estamos forjando las habilidades necesarias para no solo entender la ciberseguridad, sino también para aplicarla de manera efectiva en un mundo digital cada vez más interconectado y amenazante.

## Documentación de la red

### TP-Link Smart Connect

TP-Link Smart Connect permite que el en Router asigne automáticamente los dispositivos conectados a la banda Wi-Fi que proporciona la velocidad más rápida. Al equilibrar la carga y asignar dispositivos a la banda más adecuada, Smart Connect puede reducir retrasos e interrupciones.

En nuestro caso contamos con un modelo **AC1200**



**figura 01. Tp-link-Deco M4 Sistema Wi-Fi.**

La banda de 5 GHz alcanza hasta los 867 Mbps para que puedas jugar en línea y transmitir en HD simultáneamente. Ambas bandas funcionan al mismo tiempo para satisfacer las necesidades de múltiples tareas en diferentes dispositivos.

Dicho esto, procedemos a instalar la app que soporte y sea disponible con el modelo a configurar en este caso muestra **intarnet**.

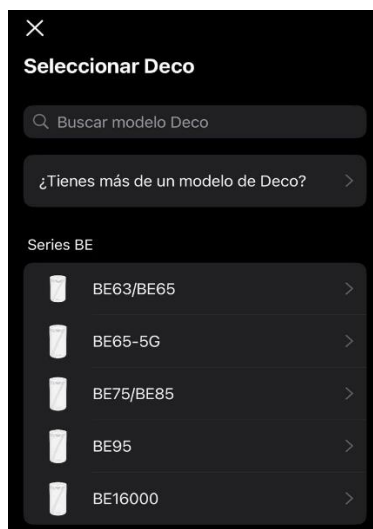
Dentro, la app **Tp-link-Deco** procedemos a instalar y seguir los siguientes pasos.



**Figura 02. app Tp-link-Deco.**

## Paso 1

Recordamos el modelo **AC1200** para la preselección del router.



*figura 03. App configuración Tp-link-Deco.*

## Paso 2

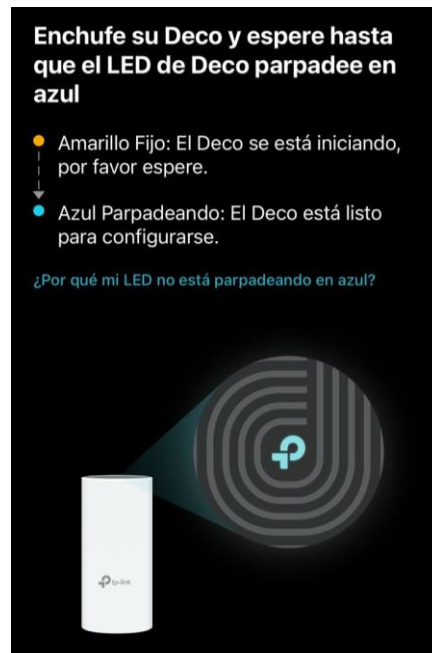
Seleccionamos en que área estará el router ubicado ya que, si no elegimos dormitorio u oficina, la aplicación no procesadora a configurar la red.



*figura 04. App configuración Tp-link-Deco.*

## Paso 3

Procedemos a seguir las indicaciones de la aplicación como se muestra en la próxima figura.



**figura 05. App configuración Tp-link-Deco.**

#### **Paso 4**

Colocamos las credenciales de la internet.

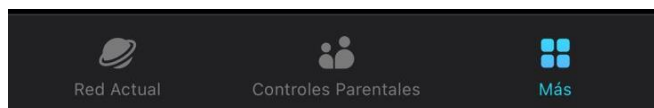


**figura 06. App configuración Tp-link-Deco.**

#### **Paso 5**

Ahora transformamos la ip del router a estática.

Para ello primero vamos al icono inferior derecha.



**figura 07. App configuración en más.**



**figura 08. App configuración en más.**

Luego nos vamos a ajustes de wifi y cambiamos la ip a estática la **conexión a internet**.



**figura 09. App configuración conexión a Internet.**

## Paso 6

Procedemos a configurar **IP LAN** y el servicio de **DHCP**.



*figura 10. App configuraciones avanzadas*



*figura 11. Configuración IP LAN*

<

Servidor DHCP

Guardar

3 Dirección IP asignada

IP Inicial

192.168.0.2

IP final

192.168.0.22

Puerta de Enlace Predeterminada

192.168.0.1

DNS Primario (Opcional)

DNS Secundario (Opcional)

**figura 12. Configuración del servicio DHCP**

Al final nuestra conexión Intranet estaría diagramada de la siguiente forma:

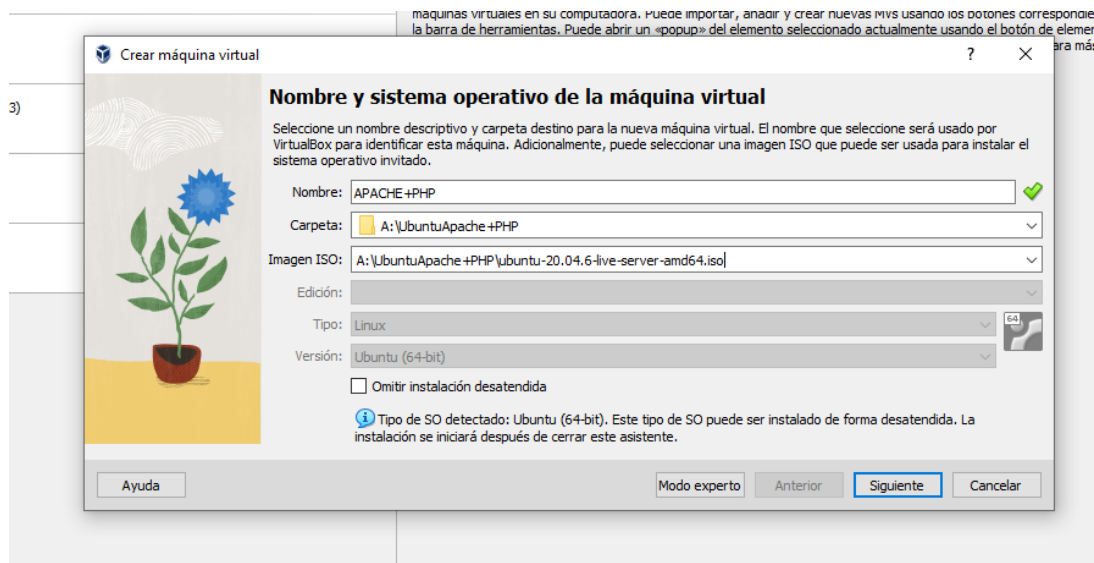


**figura 13. Diagrama de nuestra Intranet.**

## Configuración del servidor Apache

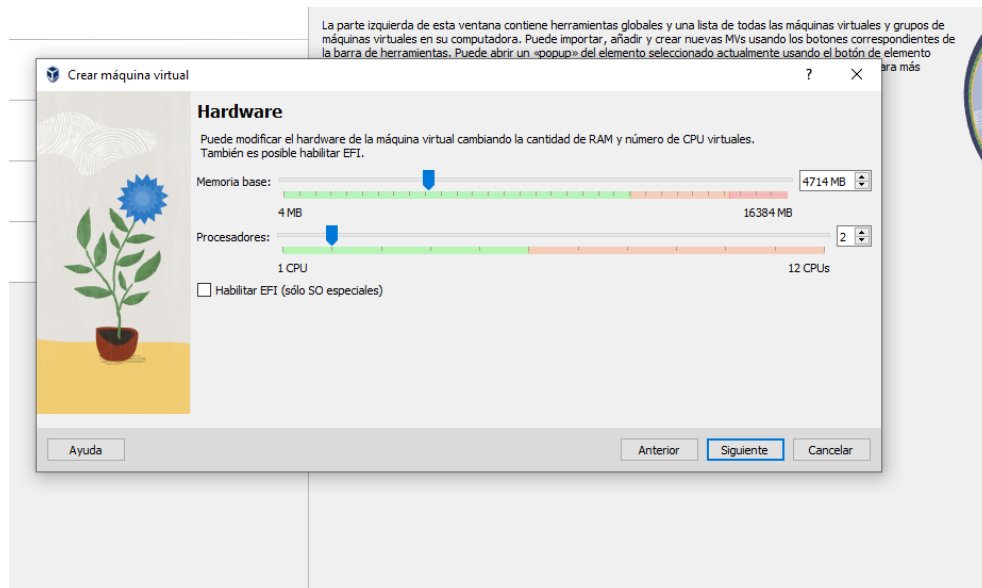
### Creación de la máquina virtual

Primero crearemos una máquina virtual, siguiendo los requerimientos del docente, ponemos el nombre y la carpeta de dirección de la máquina virtual.



**figura 14. Nombre y sistema operativo**

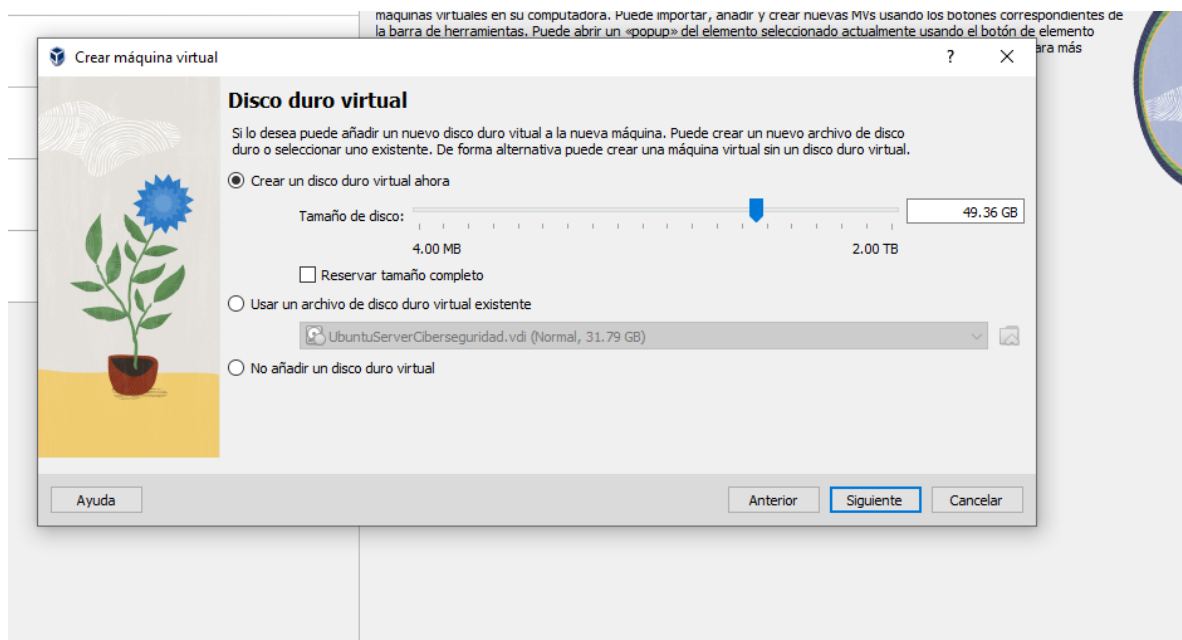
Posteriormente vamos a asignar la cantidad de memoria RAM y procesadores.



**figura 15. Hardware**

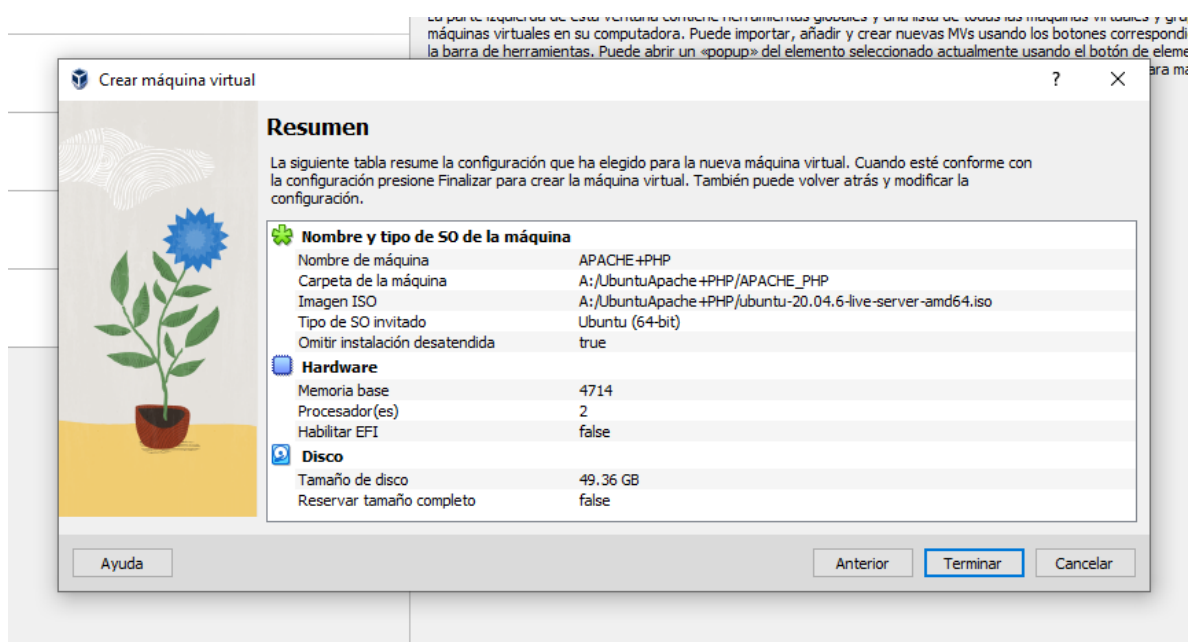
Seleccionamos la cantidad de espacio del tamaño del disco en este caso tenemos 49 GB de espacio, más que suficiente para una configuración sencilla.





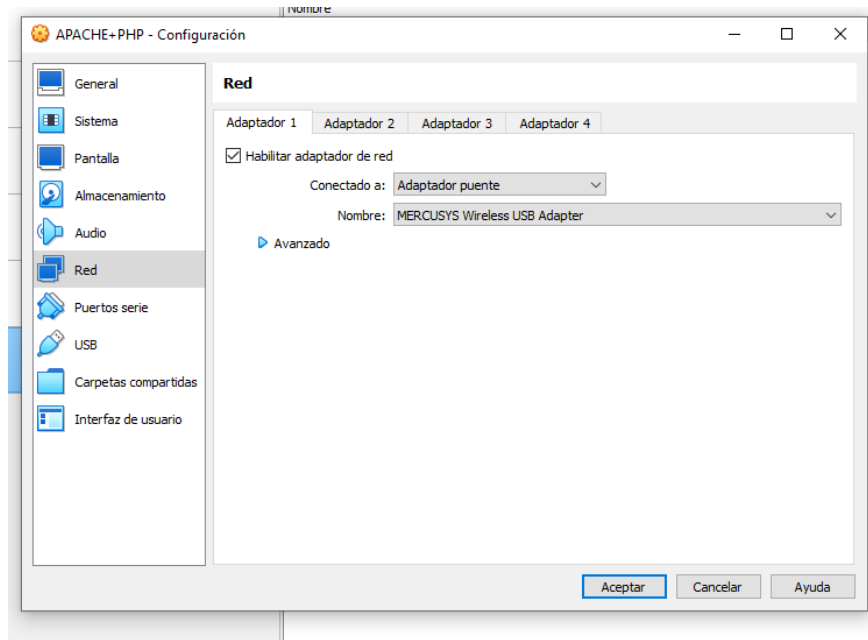
**figura 16. Disco duro virtual**

Posteriormente nos mostrará un resumen de nuestra máquina virtual.



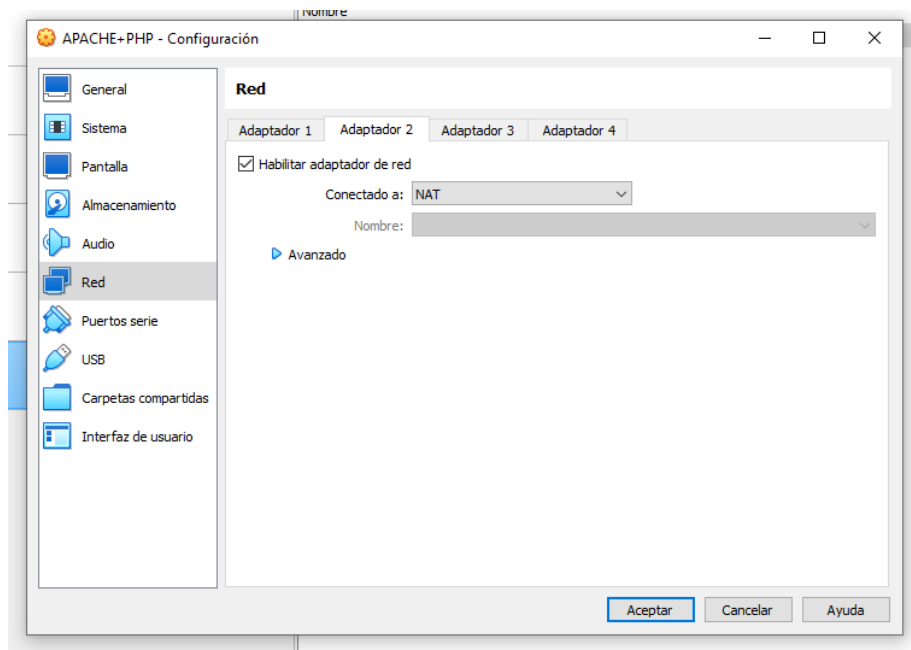
**figura 17. resumen**

En la sección de adaptadores de las tarjetas vamos a agregar una tarjeta de red en modo adaptador puente o bridge para que replique la red que tenemos en nuestro router.



**figura 18. Adaptador de red**

También agregaremos una tarjeta en NAT para que tener internet en esta computadora.



**figura 19. Adaptador de red 2**

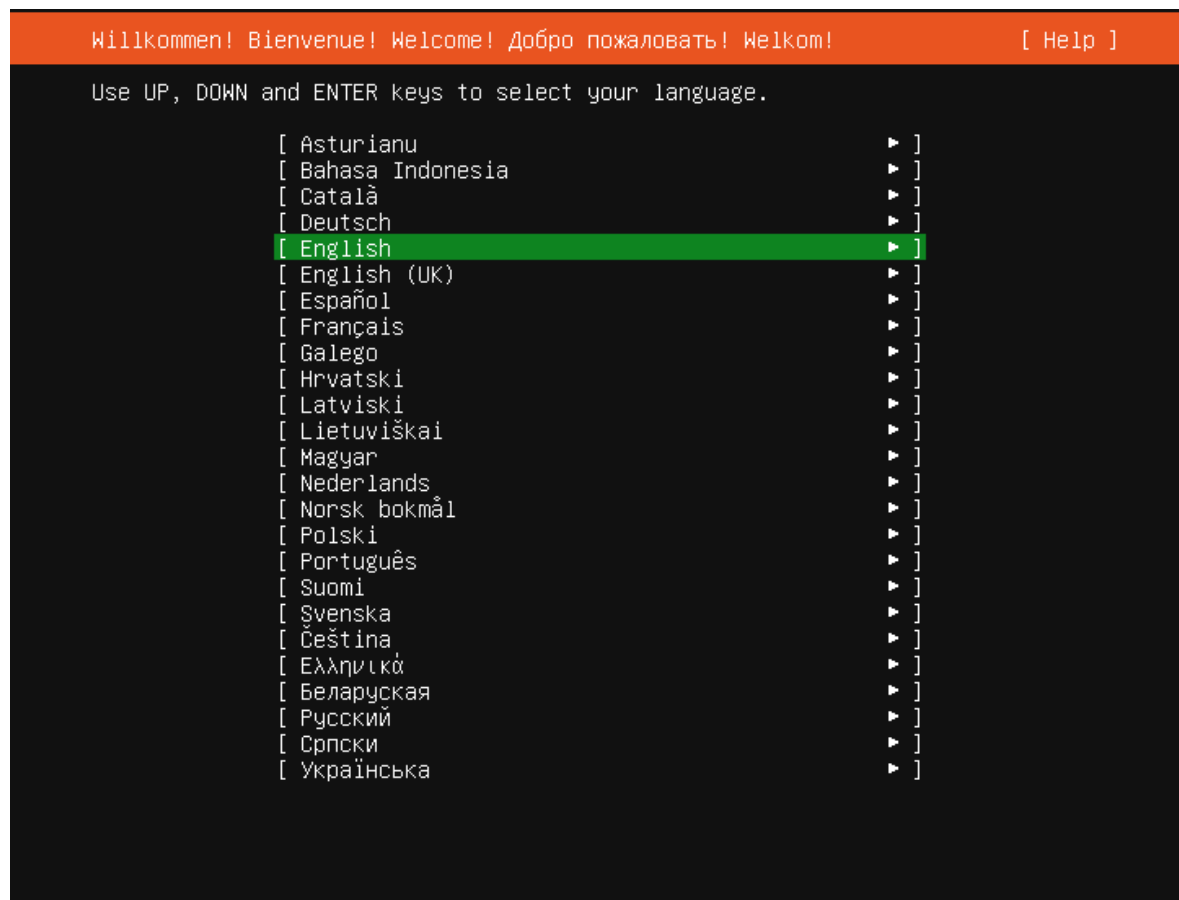
### **Instalación del servidor para la maquina apache**

Al iniciar la maquina tendremos que esperar a que los servicios inicien.

```
[ 0.108252] RETbleed: WARNING: Spectre v2 mitigation leaves CPU vulnerable to
RETbleed attacks, data leaks possible!
[ 1.367571] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send host log messa
ge.
[ 1.368206] [drm:vmw_host_log [vmwgfx]] *ERROR* Failed to send host messa
ge.
.
Checking integrity, this may take some time (or try: fsck.mode=skip)
.....
Check finished: no errors found.
```

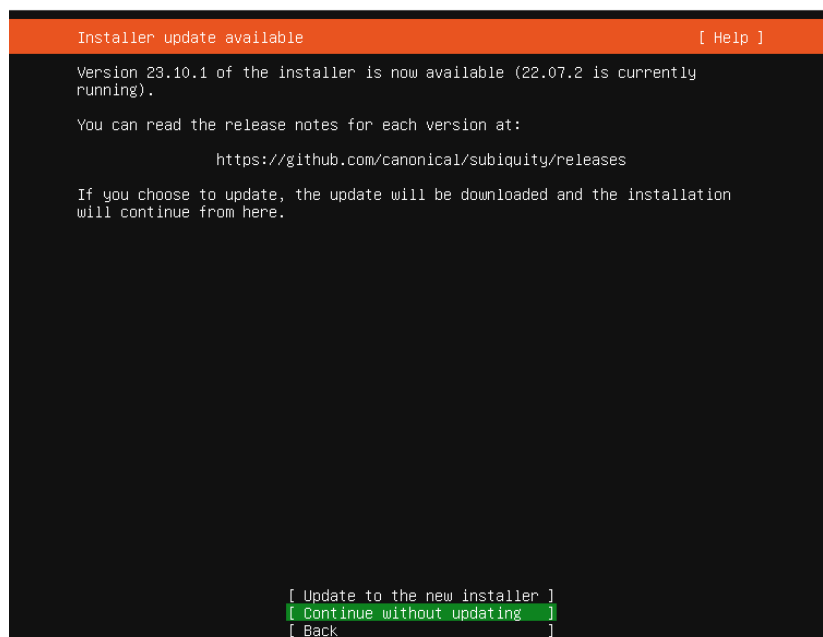
**figura 20. Inicio instalador**

Elegimos nuestro idioma.



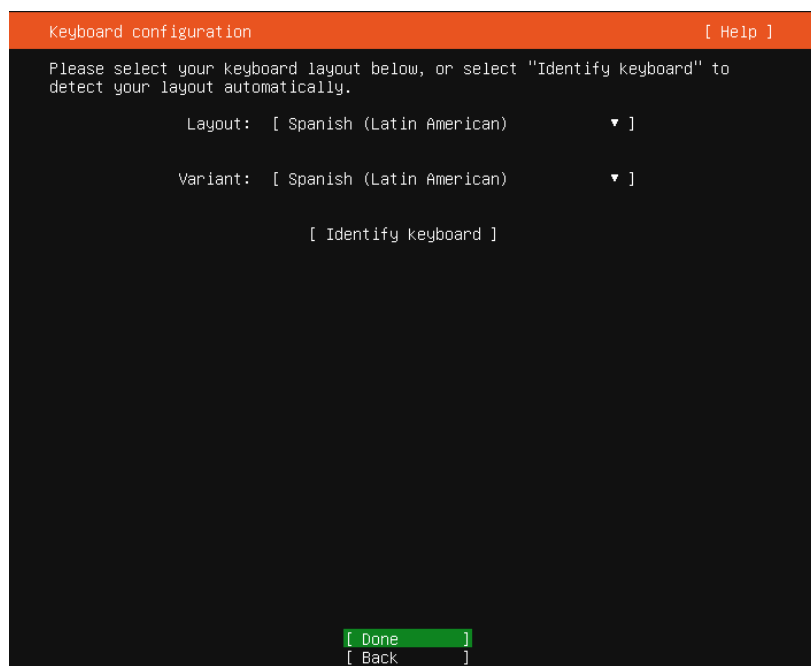
**figura 21. idioma**

En este punto nos va a preguntar si queremos iniciar con el instalador o con actualizaciones para esto luego actualizaremos todo así que continuamos solamente.



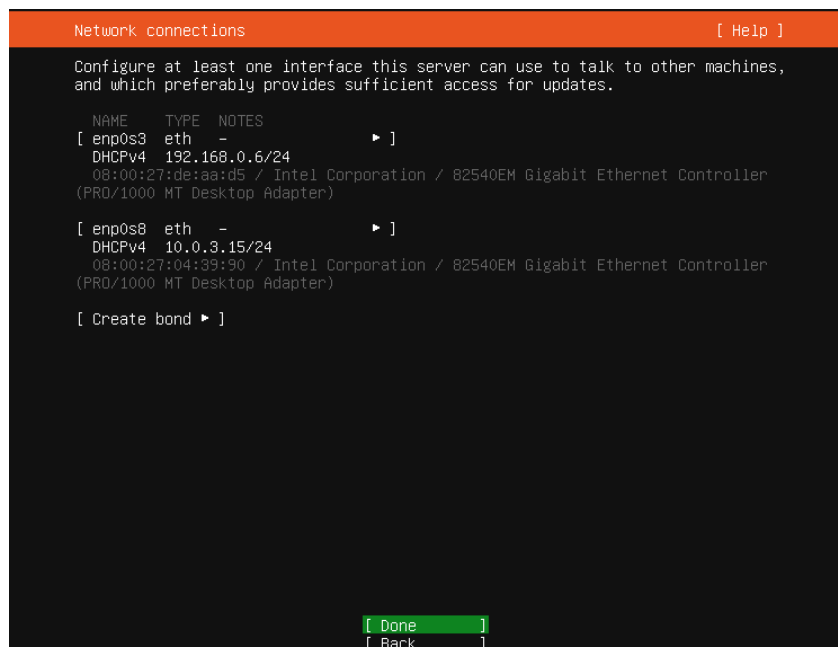
**figura 22. actualizaciones**

Elegimos nuestro teclado que en este caso es el latinoamericano.



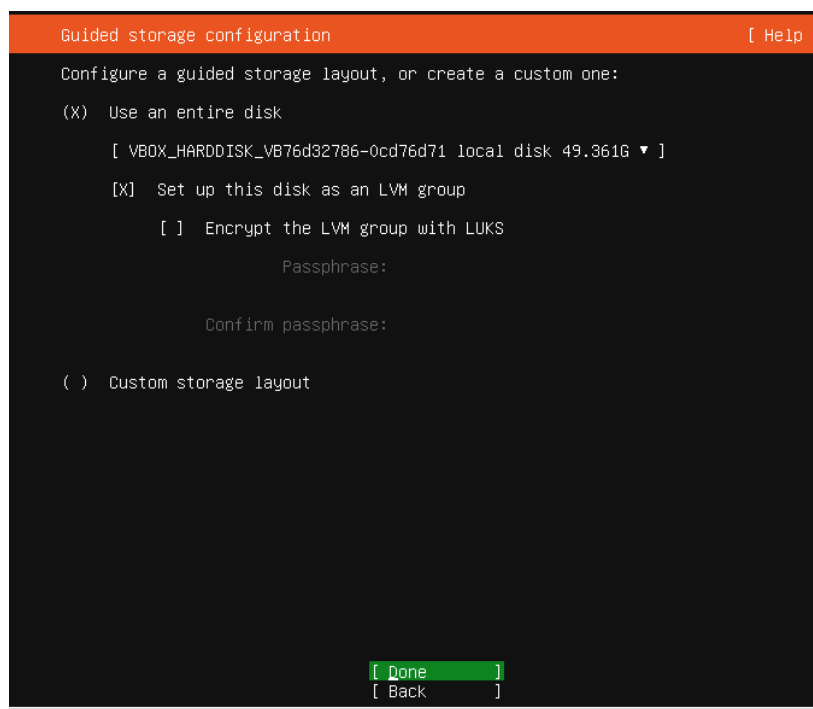
**figura 23. teclado**

Nos va a mostrar un ligero resumen de nuestras tarjetas de red con sus respectivas ip, posteriormente se van a cambiar de manera estática.



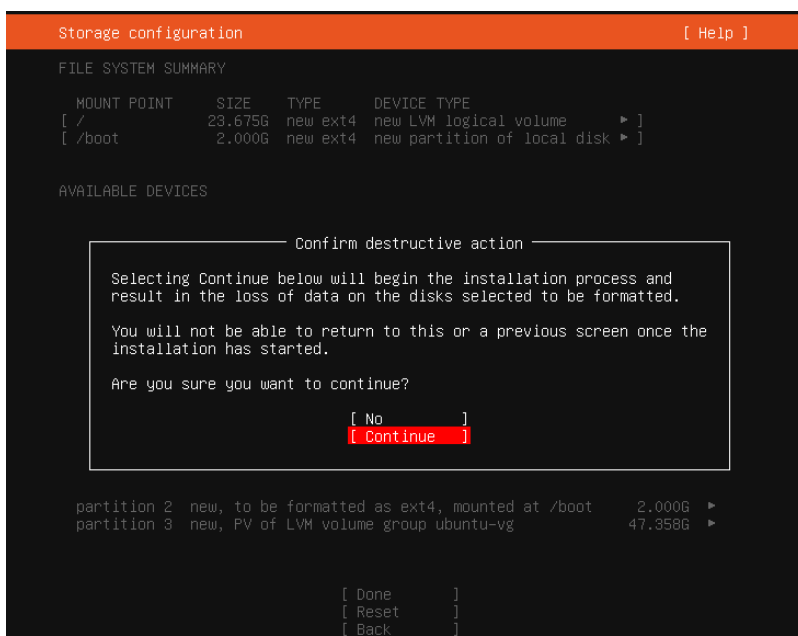
**figura 24. Adaptador de red instalación**

En este punto nos va a preguntar si queremos usar todo el espacio, para este punto si queremos hacerlo, por ende, apretamos en done.



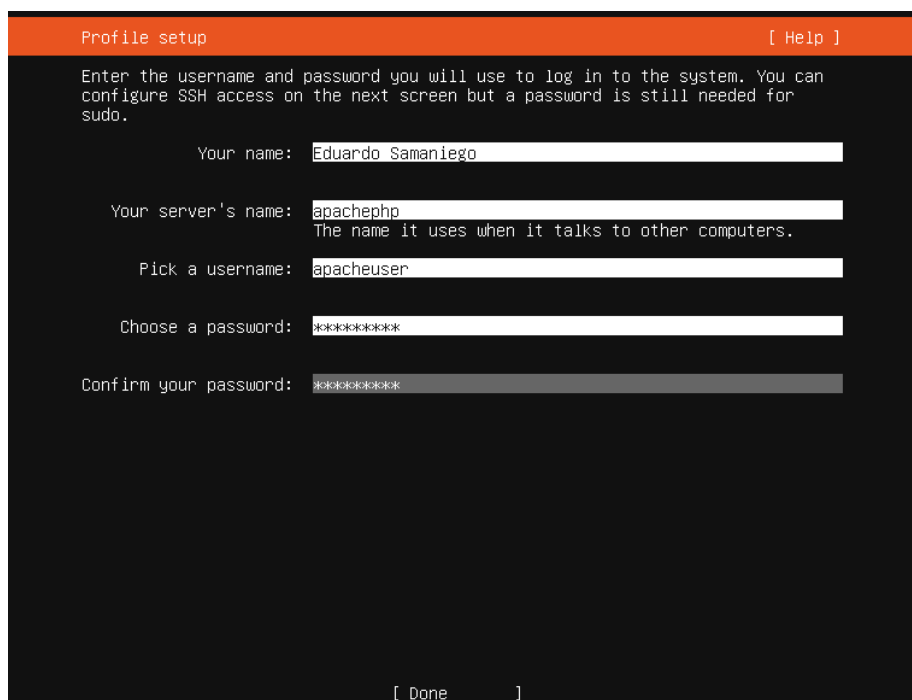
**figura 25. Disco duro**

Continuamos con la instalación sin problemas.



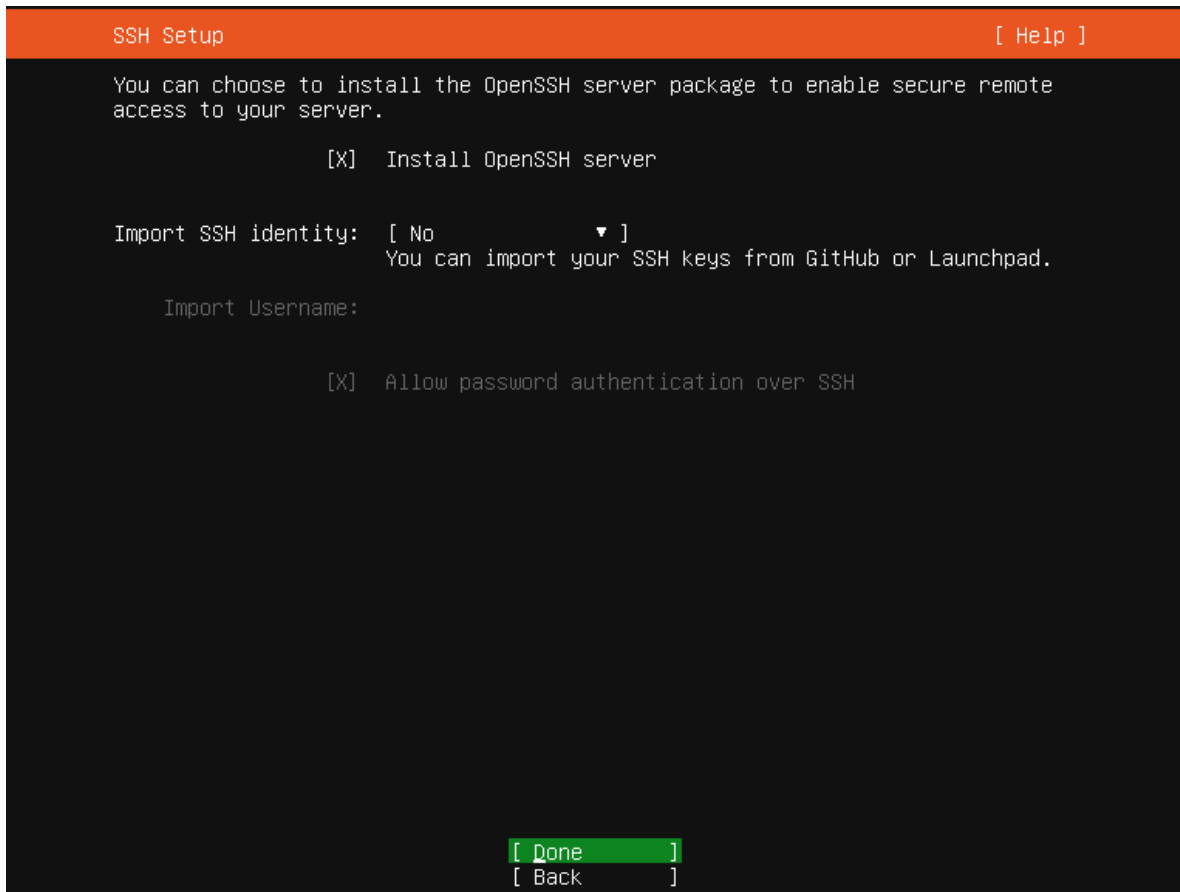
**figura 26. Storage configuración**

Agregamos temas de usuarios, nombres y contraseñas.



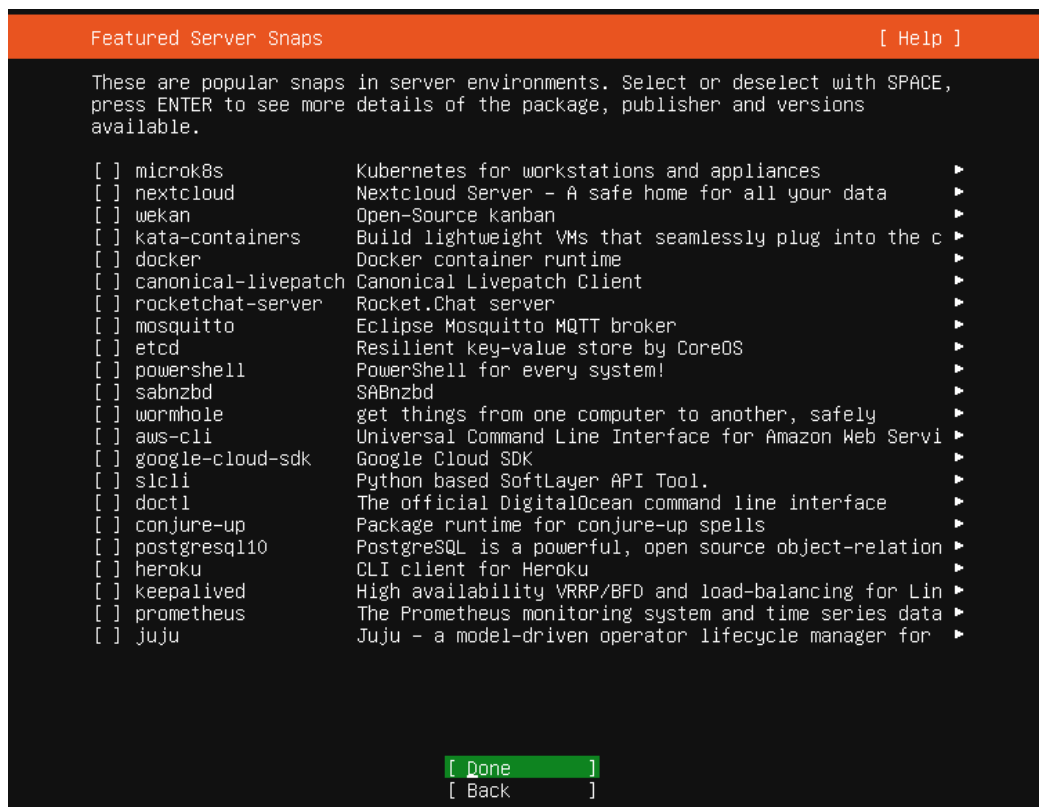
**figura 27. perfiles**

Instalamos el OpenSSH para poder luego conectarnos por ssh a través de PowerShell.



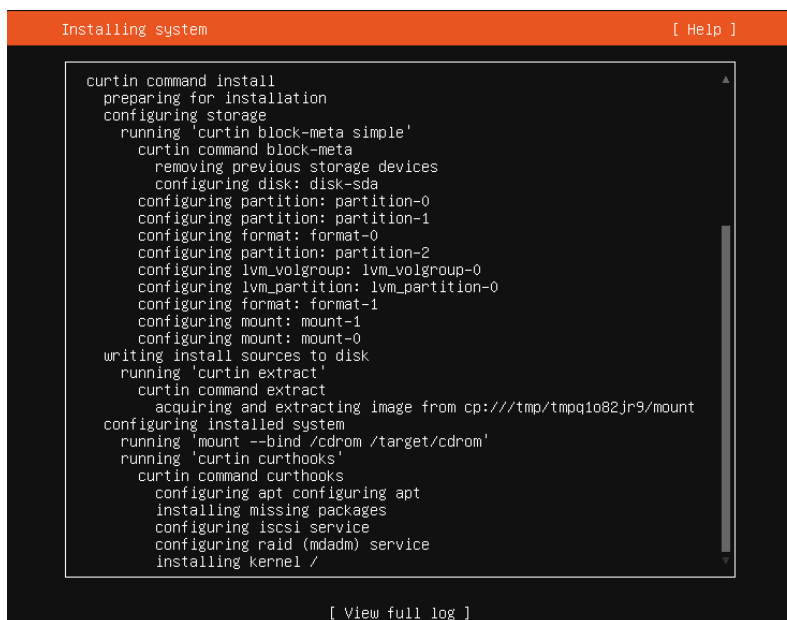
**figura 28. openssh**

Luego nos va a mostrar un apartado para ingresar servicios que queramos instalar y que son comunes, nosotros no vamos a agregar nada, si luego queremos instalarlo lo haremos manualmente.



**figura 29. Servicios opcionales**

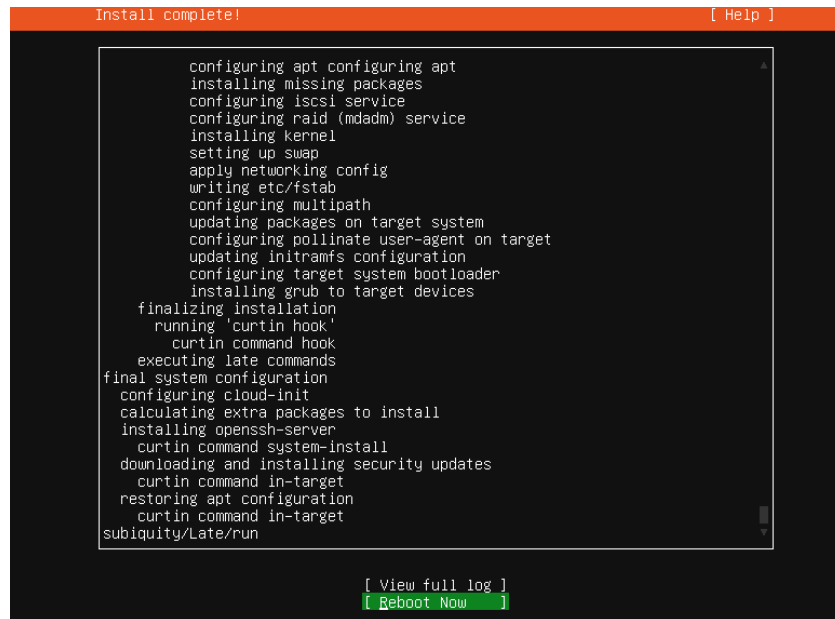
Iniciará el proceso de instalación del servidor, esto puede tomar varios minutos.



**figura 30. Instalación**

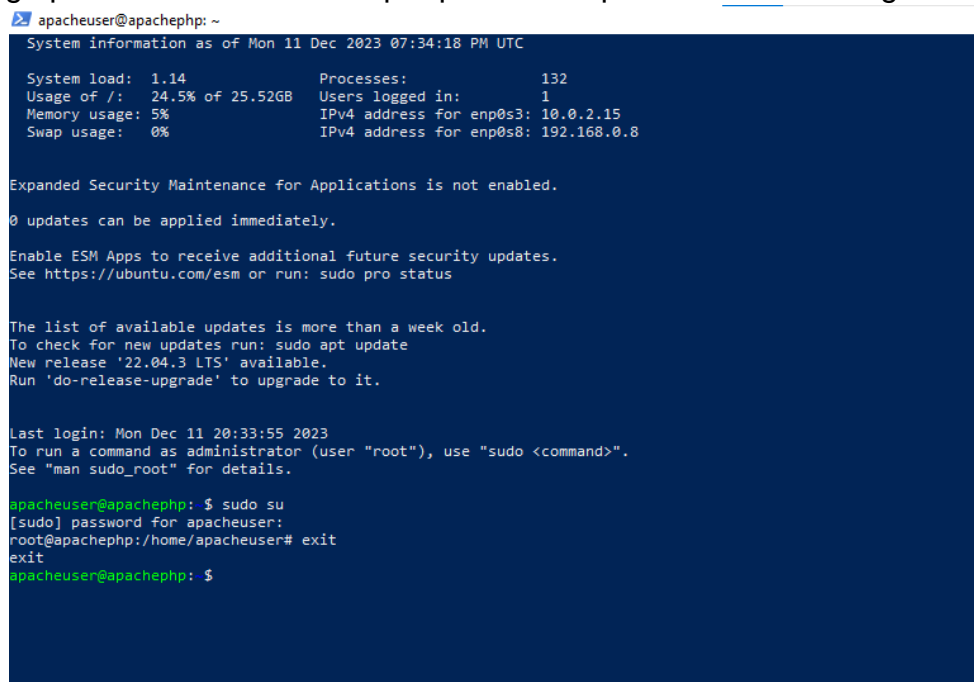
Una vez terminado podemos reiniciar el sistema y tendremos instalada la máquina virtual con el Ubuntu server.





**figura 31. Instalación del sistema**

Luego podremos conectarnos por powershell para hacer las configuraciones.



**figura 32. Inicio por powershell**

Cuando hacemos esto podremos hacer un update de los repositorios.

```

apacheuser@apachephp: ~
apacheuser@apachephp: $ sudo apt-get update
Hit:1 http://pa.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://pa.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://pa.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://pa.archive.ubuntu.com/ubuntu focal-security InRelease
Get:5 http://pa.archive.ubuntu.com/ubuntu focal/main Translation-en [506 kB]
Get:6 http://pa.archive.ubuntu.com/ubuntu focal/restricted Translation-en [6,212 B]
Get:7 http://pa.archive.ubuntu.com/ubuntu focal/universe Translation-en [5,124 kB]
Get:8 http://pa.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:9 http://pa.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [487 kB]
Get:10 http://pa.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [358 kB]
Get:11 http://pa.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [273 kB]
Get:12 http://pa.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7,484 B]
Fetched 6,867 kB in 23s (295 kB/s)

```

**figura 33. Updates**

Posteriormente hacemos un upgrade de los mismos.

```

apacheuser@apachephp: $ sudo apt-get upgrade
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  ubuntu-advantage-tools
The following packages will be upgraded:
  apparmor apport apt apt-utils bolt bsdutils cloud-init distro-info distro-info-data fdisk fwupd fwupd-signed iptables kpartx libapparmor1 libapt-pkg6.0 libblkid1 libfdisk1 libfwupd2
  libfwupdplugin5 libgpgme11 libip4tc2 libipset2 libmount1 libnetplan0 libnss-systemd libpam-systemd libsmartcols1 libsystemd0 libudev1 libunwind8 libuuid1 libxtables12 mount multipath-tools
  netplan.io python3-apport python3-debian python3-distro-info python3-problem-report python3-software-properties rsync software-properties-common sosreport systemd systemd-sysv
  systemd-timesyncd tcldata udev urf update-notifier-common util-linux uuid-runtime
53 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
Need to get 16.8 MB of archives.
After this operation, 550 kB disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 bsdutils amd64 1:2.34-0.1ubuntu9.4 [63.0 kB]
Get:2 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 libblkid1 amd64 2.34-0.1ubuntu9.4 [137 kB]
Get:3 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 libuuid1 amd64 2.34-0.1ubuntu9.4 [28.0 kB]
Get:4 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 libfdisk1 amd64 2.34-0.1ubuntu9.4 [174 kB]
Get:5 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 libmount1 amd64 2.34-0.1ubuntu9.4 [150 kB]
Get:6 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 libsmartcols1 amd64 2.34-0.1ubuntu9.4 [100 kB]
Get:7 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 fdisk amd64 2.34-0.1ubuntu9.4 [119 kB]
Get:8 http://pa.archive.ubuntu.com/ubuntu focal-updates/main amd64 util-linux amd64 2.34-0.1ubuntu9.4 [1,021 kB]

```

**figura 34. Upgrades**

## Instalación del net-tools.

Instalamos el net-tools para poder utilizar el ifconfig en el servidor con el comando `sudo apt install net-tools`.

Ki

```

apacheuser@apachephp: $ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 196 kB of archives.
Selecting previously unselected package net-tools.ace will be used.
(Reading database ... 72340 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
apacheuser@apachephp: $

```

**figura 35. Instalación de net-tools**

Configuración de la IP estática del servidor (192.168.0.22 Provisional)

En Ubuntu server para configurar la red tendremos que entrar al archivo de configuración de red como se detalla en la imagen.

```
Last login: Mon Dec 11 19:34:19 2023 from 192.168.0.2
apacheuser@apachephp:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

**figura 36. Configuración del archivo**

Posteriormente cambiamos todo el texto como se detalla a continuación.

```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.0.22/24
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
      routes:
        - to: default
          via: 192.168.0.1
  version: 2
```

**figura 37. Tarjeta provisional**

Aplicamos los cambios con netplan apply.

```
[sudo] password for apacheuser:
apacheuser@apachephp:~$ sudo netplan apply
apacheuser@apachephp:~$
```

**figura 38. Aplicando la configuración**

### Instalación del apache y las librerías del php.

Instalamos los servicios como se detalla a continuación con `sudo apt install apache2 php libapache2-mod-php`.

```
[sudo] password for apacheuser:
apacheuser@apachephp:~$ sudo netplan apply
apacheuser@apachephp:~$ sudo apt install apache2 php libapache2-mod-php
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 php-common php7.4 php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
  ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser php-pear openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php libapache2-mod-php7.4 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 php php-common php7.4 php7.4-cli php7.4-common php7.4-json
  php7.4-opcache php7.4-readline ssl-cert
0 upgraded, 20 newly installed, 0 to remove and 1 not upgraded.
Need to get 5,877 kB of archives.
After this operation, 26.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

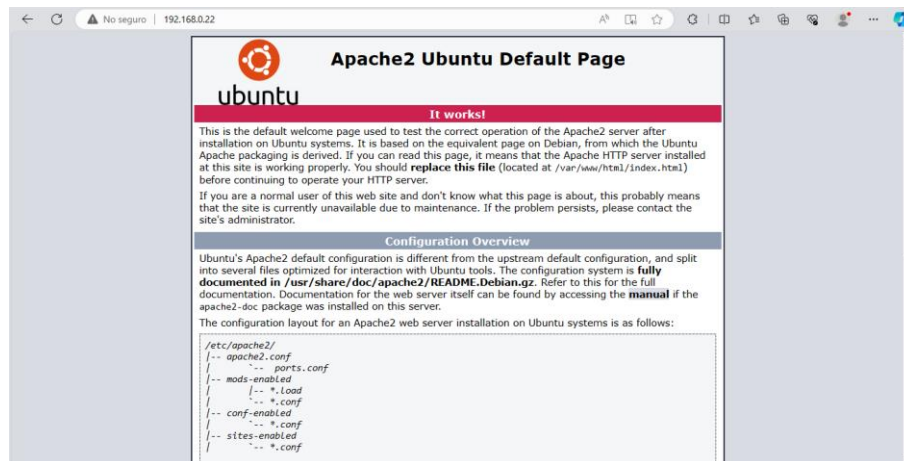
**figura 39. Instalación del apache y parches**

Posteriormente dentro del /var/www/html podemos crear el archivo info.php para ver si se instaló correctamente el módulo de php e instalamos el módulo de php-mysql.

```
apacheuser@apachephp:/var/www/html$ sudo nano info.php
apacheuser@apachephp:/var/www/html$ sudo apt install php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  php7.4-mysql
The following NEW packages will be installed:
  php-mysql php7.4-mysql
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 123 kB of archives.
After this operation, 487 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

**figura 40. Instalación del módulo**

Ingresamos a la página desde la ip y podemos ver que tenemos el apache activo.



**figura 41. Apache2**

Si entramos al info.php podemos ver que tenemos acceso al mysql que es necesario para utilizar php.

libXML streams			enabled
----------------	--	--	---------

mysqli			enabled
--------	--	--	---------

mysqli Support		enabled
Client API library version	mysqli 7.4.3-4ubuntu2.19	
Active Persistent Links	0	
Inactive Persistent Links	0	
Active Links	0	

Directive	Local Value	Master Value
mysqli.allow_local_infile	Off	Off
mysqli.allow_persistent	On	On
mysqli.default_host	no value	no value
mysqli.default_port	3306	3306
mysqli.default_pw	no value	no value
mysqli.default_socket	no value	no value
mysqli.default_user	no value	no value
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off
mysqli.rollback_on_cached_plink	Off	Off

mysqli		enabled
--------	--	---------

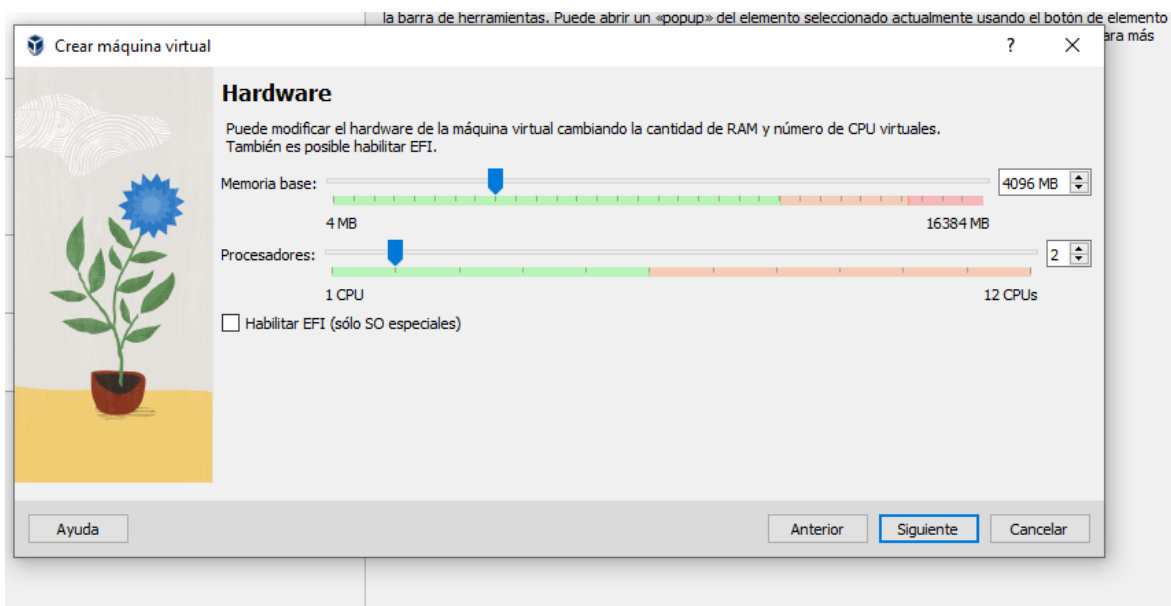
Version	mysqli 7.4.3-4ubuntu2.19	
Compression	supported	
core SSL	supported	
extended SSL	supported	

**figura 42. mysqli**

## Servidor de bases de datos

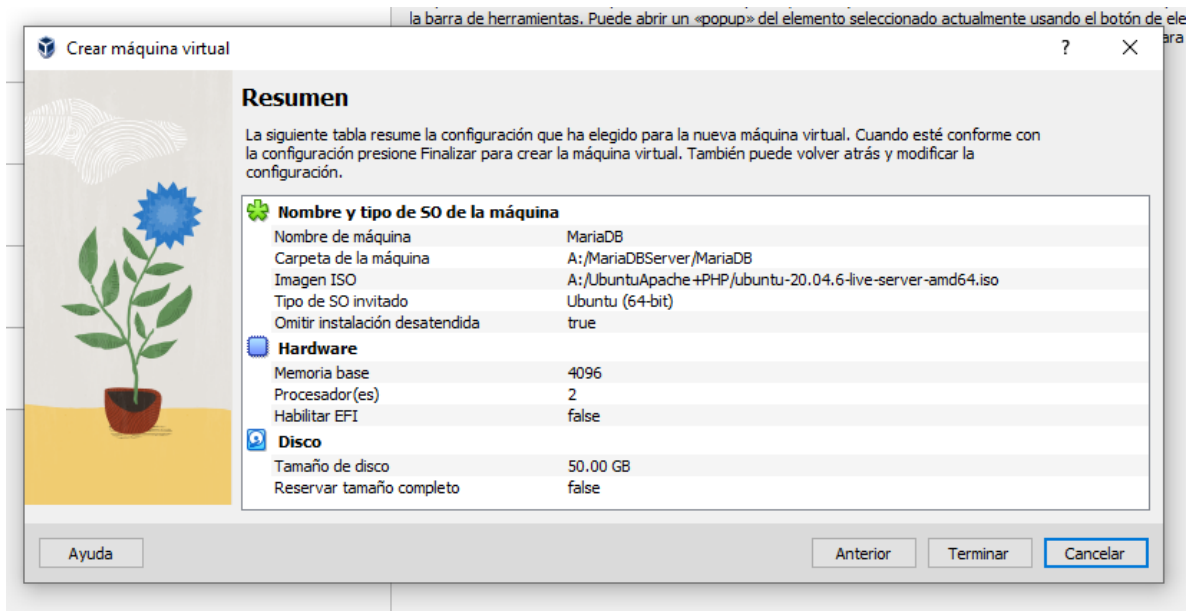
### Creación de máquina virtual

Vamos a crear ahora el servidor mariadb así que veremos el proceso de instalación para esto pondremos 4gb de RAM en este caso.



**figura 43. RAM**

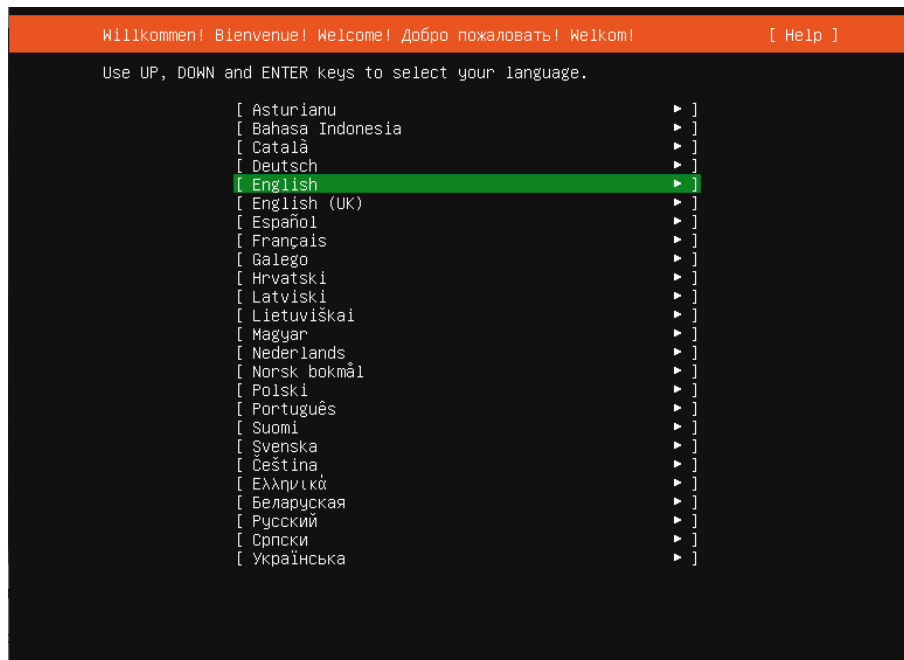
Y también veremos un resumen de cómo está configurado.



**figura 44. Resumen del sistema**

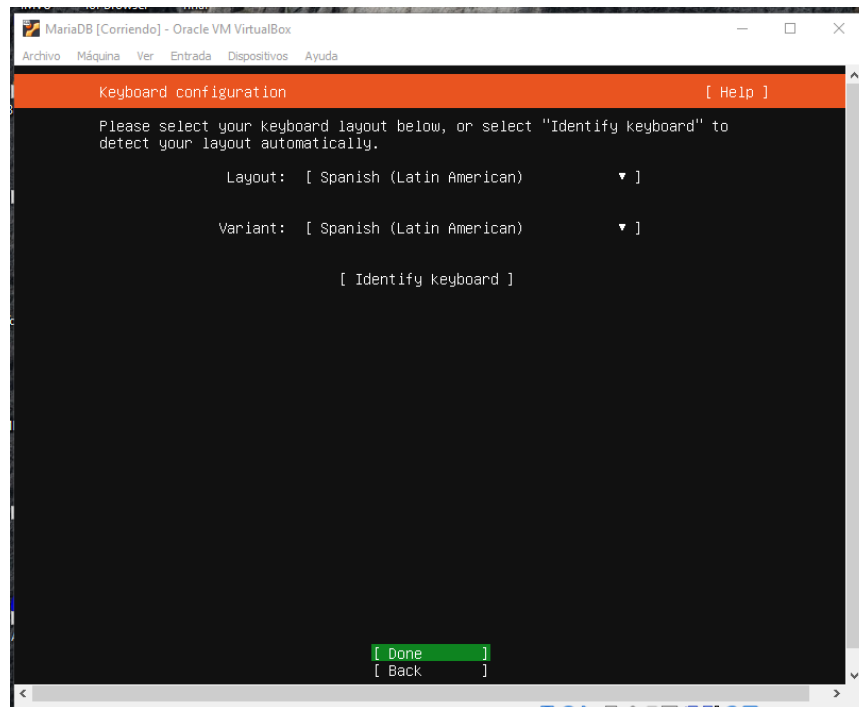
## Instalación del servidor para MariaDB

Ingresamos el idioma en este caso es inglés.



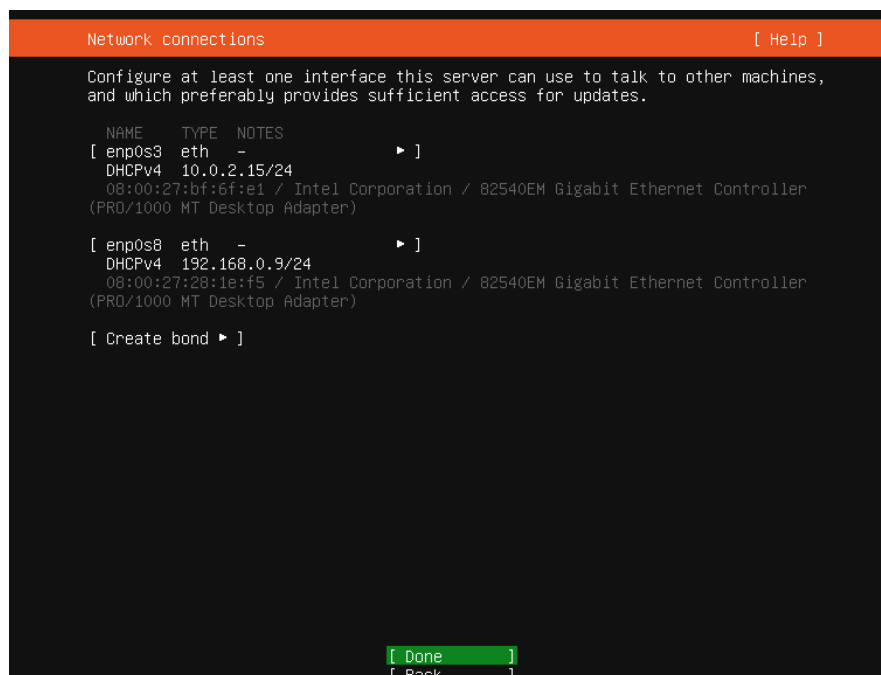
**figura 45. idioma**

Elegimos el teclado que es el latino en este caso es el latinoamericano.



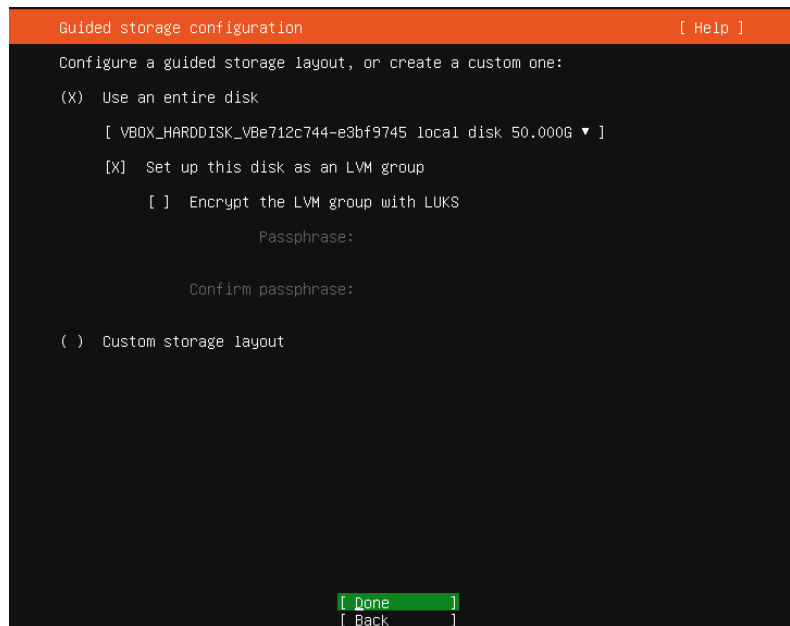
**figura 46. teclado**

Vemos un resumen de los diferentes adaptadores de red.



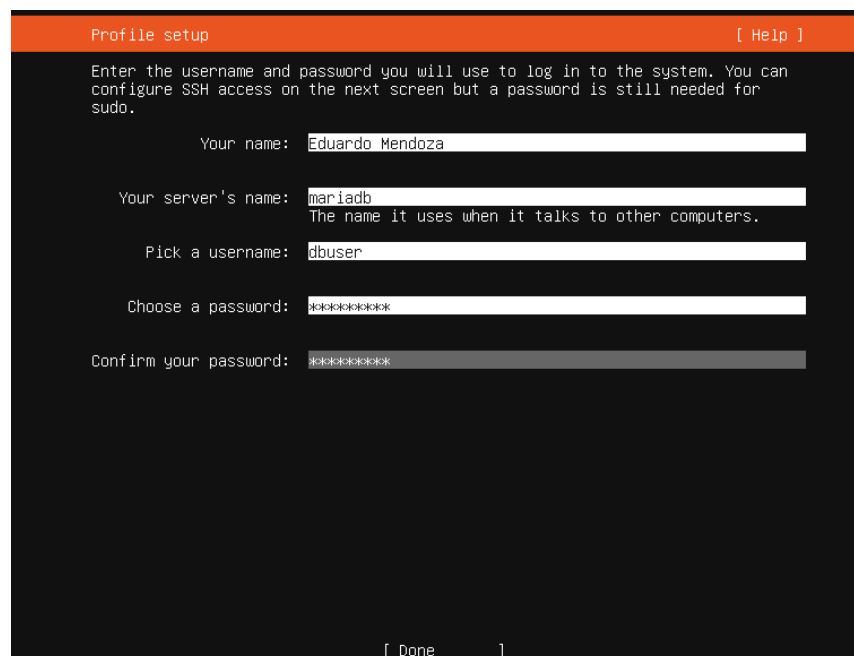
**figura 47. Resumen de las conexiones**

Nos pregunta si vamos a usar todo el disco en este caso si lo vamos a usar todo.



**figura 48. almacenamiento**

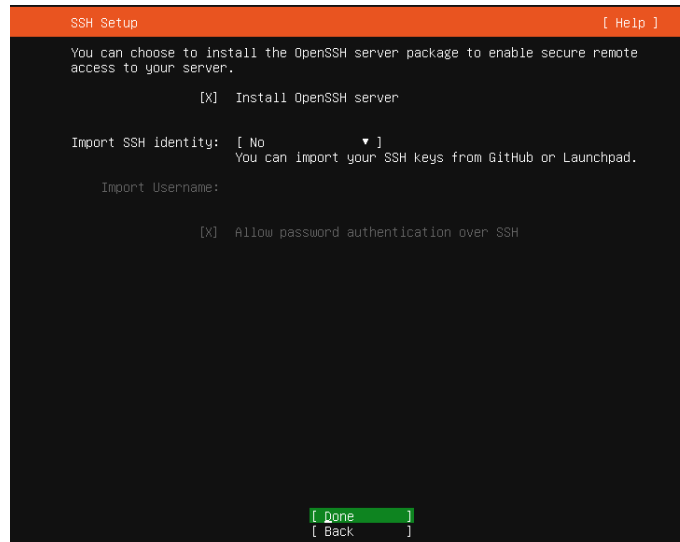
Luego nos preguntará todo lo conveniente a el servidor como usuarios, nombres, etc.



**figura 49. Perfil del sistema**

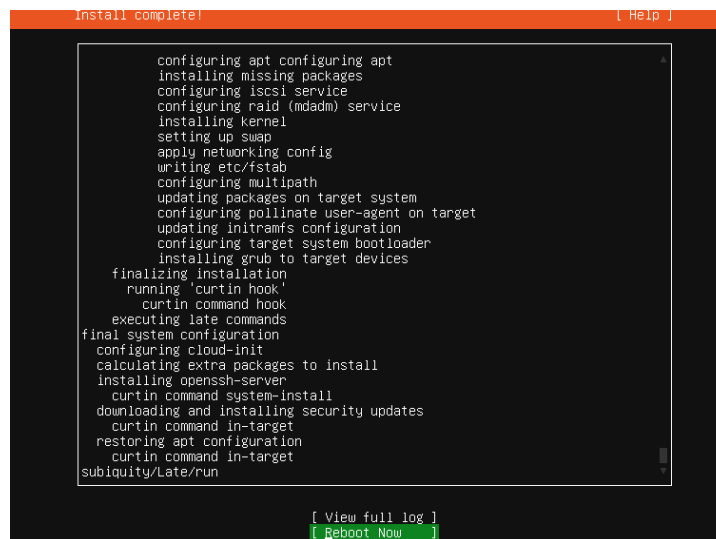
Instalamos el openssh para posteriormente conectarnos por ssh.





**figura 50. Instalando el openssh**

Empezará la instalación del servidor.



**figura 51. Instalación del sistema**

Una vez instalado podemos iniciar sesión en el servidor.

```

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-169-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Mon 11 Dec 2023 08:19:36 PM UTC

System load:  0.44           Processes:           112
Usage of /:   26.7% of 23.45GB Users logged in:      0
Memory usage: 5%            IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%            IPv4 address for enp0s8: 192.168.0.9

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dbuser@mariadb:~$

```

**figura 52. Ingreso al sistema**

Hacemos una update de los servidores.

```

dbuser@mariadb:~$ sudo apt update
[sudo] password for dbuser:
Hit:1 http://pa.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://pa.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://pa.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://pa.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... 4%

```

**figura 53. update**

También hacemos un upgrade.

```

54 packages can be upgraded. Run 'apt list --upgradable' to see them.
dbuser@mariadb:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  ubuntu-pro-client-l10n
The following packages will be upgraded:
  apparmor apport apt apt-utils bolt bsdtails cloud-init distro-info distro-info-data fdisk fwupd
  fwupd-signed iptables kpartx libapparmor1 libapt-pkg6.0 libblkid1 libfdisk1 libfwupd2
  libfwupdplugin5 libgpgme11 libip4tc2 libip6tc2 libmount1 libnetplan0 libnss-systemd
  libpam-systemd libsmartcols1 libsystemd0 libudev1 libunwind8 libuuid1 libxtables12 mount
  multipath-tools netplan.io python3-apport python3-debian python3-distro-info
  python3-problem-report python3-software-properties rsync software-properties-common sosreport
  systemd systemd-sysv systemd-timesyncd tzdata ubuntu-advantage-tools udev ufw
  update-notifier-common util-linux uuid-runtime
54 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 17.0 MB of archives.
After this operation, 212 kB disk space will be freed.
Do you want to continue? [Y/n]

```

**figura 54. upgrade**

### Instalación del paquete Net-Tools.

Y instalamos el paquete de net-tools.

```

dbuser@mariadb:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://pa.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git201806
1ubuntu1 [196 kB]
Fetched 196 kB in 1s (167 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 72402 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
dbuser@mariadb:~$ _

```

**figura 55. Instalacion de net-tools**

### Configuración de ip estatica (192.168.0.23 Provisional)

Posteriormente nos conectamos por ssh al servidor.

```

PS C:\Users\YuRuMeng> ssh dbuser@192.168.0.9
The authenticity of host '192.168.0.9 (192.168.0.9)' can't be established.
ECDSA key fingerprint is SHA256:6ELV9moayVuLc2Qcudehm8PbsscAleX+X3R1FaptsmY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.9' (ECDSA) to the list of known hosts.
dbuser@192.168.0.9's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-169-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon 11 Dec 2023 08:29:17 PM UTC

System load:  0.0          Processes:      128
Usage of /:   26.8% of 23.45GB Users logged in: 1
Memory usage: 7%          IPv4 address for enp0s3: 10.0.2.15
Swap usage:  0%           IPv4 address for enp0s8: 192.168.0.9

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon Dec 11 20:19:37 2023

```

**figura 56. Ingreso por ssh**

Modificamos el archivo 00-installer-config. yaml.

```

Last login: Mon Dec 11 20:19:37 2023
dbuser@mariadb:~$ sudo nano /etc/netplan/00-installer-config.yaml

```

**figura 57. Entramos al config**

Lo detallamos como a continuación y cambiamos sus direcciones IP.

```

GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.0.23/24
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
      routes:
        - to: default
          via: 192.168.0.1
  version: 2

```

**figura 58. IP**

Cargamos los nuevos servicios con sudo netplan apply.

```

dbuser@mariadb:~$ sudo nano /etc/netplan/00-installer-config.yaml
[sudo] password for dbuser:
dbuser@mariadb:~$ dbuser@mariadb:~$ sudo netplan apply

```

**figura 59. Aplicamos la configuración**

Y podemos ver la tarjeta de red.

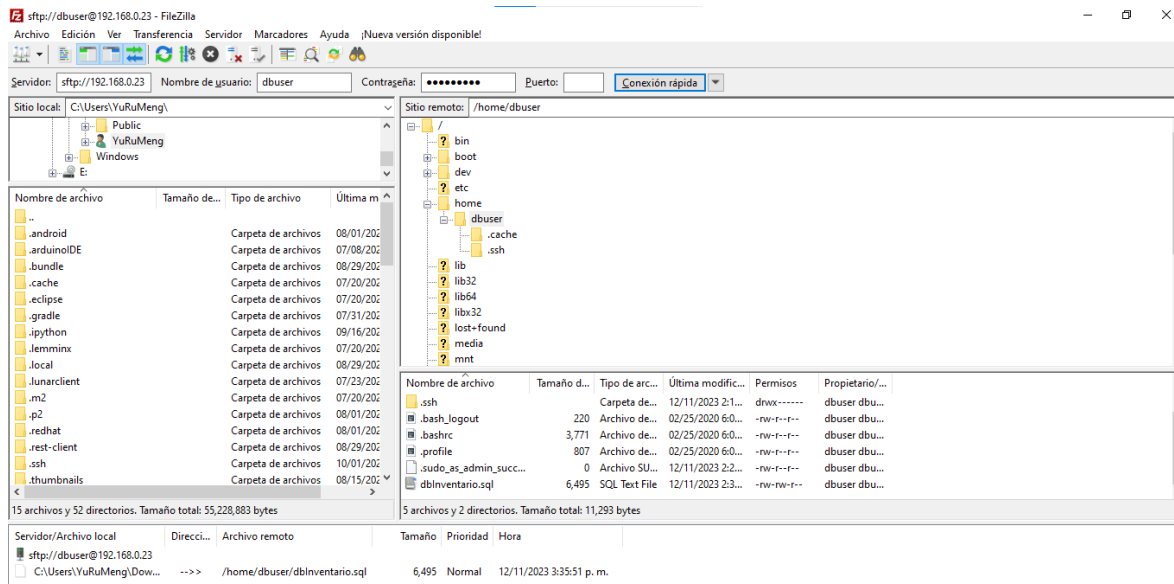
```
dbuser@mariadb:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:febf:6fe1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:bf:6f:e1 txqueuelen 1000 (Ethernet)
    RX packets 8943 bytes 13264094 (13.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3387 bytes 230008 (230.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.23 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe28:1ef5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:28:1e:f5 txqueuelen 1000 (Ethernet)
    RX packets 4127 bytes 5206127 (5.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1700 bytes 153688 (153.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**figura 60. Viendo la interfaz de ip**

## Carga y configuración de la base de datos.

Abrimos el servidor a través de FileZilla y entramos a la carpeta del usuario dbuser, posteriormente cargamos el archivo de base de datos.



**figura 61. Ingresando la base**

Cuando hagamos esto. Tendremos una copia de la base de datos dentro del servidor, luego vamos a instalar el mariadbserver con `sudo apt install mariadb-server`.

```

dbinventario.sql
dbuser@mariadb:~$ sudo apt install mariadb-server
[sudo] password for dbuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 libbcgi-fast-perl libbcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libmysqlclient21 libsnappy1v5 libterm-readkey-perl libtimedate-perl
  liburi-perl mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server-10.3 mariadb-server-core-10.3
  mysql-common socat
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl
  libwww-perl mailx mariadb-test tinyca
The following NEW packages will be installed:
  galera-3 libbcgi-fast-perl libbcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libmysqlclient21 libsnappy1v5 libterm-readkey-perl libtimedate-perl
  liburi-perl mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server mariadb-server-10.3
  mariadb-server-core-10.3 mysql-common socat
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 21.5 MB of archives.
After this operation, 175 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

**figura 62. Instalacion del mariadb server**

E iniciaremos la instalación segura del mariadb, para esto veremos con Mysql\_secure\_Installation podremos ver lo necesario para hacer este proceso.

```

dbuser@mariadb:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

```

**figura 63. Instalacion segura**

Una vez iniciamos nos va a hacer una serie de preguntas, la primera es que, si queremos cambiar la contraseña del root, lo hacemos

```

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
Remove anonymous users? [Y/n]

```

**figura 64. Cambiando la password**

Quitamos los usuarios anónimos, no queremos ningún usuario anónimo en nuestra base de datos.

```

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? [Y/n] n

```

**figura 65. Quitando los usuarios anónimos**

Luego removemos los test que vienen por defecto.

```

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
Reload privilege tables now? [Y/n]

```

**figura 66. Quitamos los test**

Y finalmente tendremos la base de datos securizada.

```

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
dbuser@mariadb:~$

```

**figura 67. Terminando configuración**

Si nos intentamos conectar veremos que con el root podemos entrar.

```

Thanks for using MariaDB!
dbuser@mariadb:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.3.38-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>

```

**figura 68. Entrando al mariadb**

Creamos la base de datos dbInventario.

```

MariaDB [(none)]> CREATE DATABASE dbInventario;
Query OK, 1 row affected (0.000 sec)

```

**figura 69. Creando la base de datos**

Creamos posteriormente el usuario de la base de datos con todos los privilegios.

```

MariaDB [(none)]> CREATE USER 'apacheuser'@'192.168.0.22' IDENTIFIED BY 'granyayol';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'apacheuser'@'192.168.0.22' WITH GRANT OPTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

```

**figura 70. Creando el usuario de base**

Abrimos el puerto 3306 que es el puerto de base de datos y activamos el firewall.

```

bye
dbuser@mariadb:~$ sudo ufw allow 3306
Rules updated
Rules updated (v6)
dbuser@mariadb:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
dbuser@mariadb:~$

```

**figura 71. Abriendo el puerto**

Vamos a la configuración del mariaDB para permitir los usuarios externos.

```

dbuser@mariadb:~$
dbuser@mariadb:~$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

```

**figura 72. Configurando el mariadb**



Comentamos las líneas mostradas a continuación.

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address            = 127.0.0.1
#
```

**figura 73. Comentando el bind-address**

Hacemos la copia de la base de datos a la que ya tenemos.

```
dbuser@mariadb:~$ sudo mysql -u root -p dbInventario < dbInventario.sql
Enter password:
dbuser@mariadb:~$
```

**figura 74. Importando la base de datos**

Configuración del mariadb y apache en ApacheServer

Desde el servidor apache tendremos que instalar el mysql-client

```
apacheuser@apachephp:/var/www/html$ sudo apt-get install mysql-client
[sudo] password for apacheuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  mysql-client-8.0 mysql-client-core-8.0 mysql-common
The following NEW packages will be installed:
  mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common
0 upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 5,118 kB of archives.
After this operation, 74.7 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

**figura 75. Instalando el cliente**

Para este punto ya habíamos cambiado las direcciones ip a las nuevas que eran 192.168.0.16 para el mariadb y 192.168.0.15 Para el apache véase la configuración de anteriores capturas para seteo de las direcciones estáticas.

```

MariaDB [(none)]> CREATE USER 'apache2'@'192.168.0.15' IDENTIFIED BY 'granyayo1';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'apache2'@'192.168.0.15' WITH GRANT OPTION;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MariaDB server version for the right syntax to use near 'GRANT ALL PRIVILEGES ON *.* TO 'apache2'@'1
92.168.0.15' WITH GRANT OPTION' at line 1
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'apache2'@'192.168.0.15' WITH GRANT OPTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]>
```

**figura 76. Cambiando el usuario de base de datos**

## IP final servidor Apache

```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.0.15/24
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
      routes:
        - to: default
          via: 192.168.0.1
  version: 2
```

*figura 77. Cambiando la ip del apache*

## IP Final servidor MariaDb

```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.0.16/24
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
      routes:
        - to: default
          via: 192.168.0.1
  version: 2
```

*figura 78. Cambiando la ip del mariadb*

Posteriormente intentamos ingresar al mariadb desde el servidor apache y podremos ingresar

```
apacheuser@apachephp:~$ mysql -u apache2 -p -h 192.168.0.16
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 200
Server version: 5.5.5-10.3.38-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

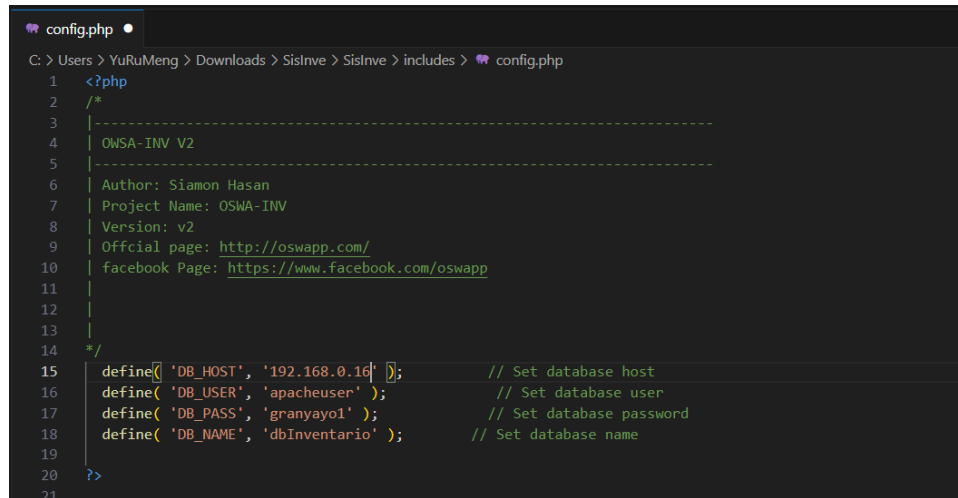
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> _
```

*figura 79. Conectando de manera remota*

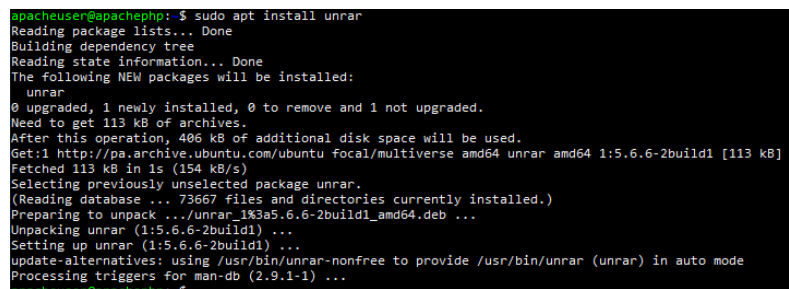
Agregamos en el documento las direcciones ip y el usuario, la pass y el nombre de la base de datos.



```
config.php
C: > Users > YuRuMeng > Downloads > Sislne > Sislne > includes > config.php
1  <?php
2  /*
3  -----
4  | OWSA-INV V2
5  | -----
6  | Author: Siamon Hasan
7  | Project Name: OWSA-INV
8  | Version: v2
9  | Official page: http://oswapp.com/
10 | Facebook Page: https://www.facebook.com/oswapp
11 |
12 |
13 |
14 */
15 define('DB_HOST', '192.168.0.16'); // Set database host
16 define('DB_USER', 'apacheuser'); // Set database user
17 define('DB_PASS', 'granyayo1'); // Set database password
18 define('DB_NAME', 'dbInventario'); // Set database name
19
20 ?>
21
```

**figura 80. Seteando las credenciales dentro de la app**

En el servidor instalamos el unrar para poder descomprimir algunos documentos dentro del apache server.

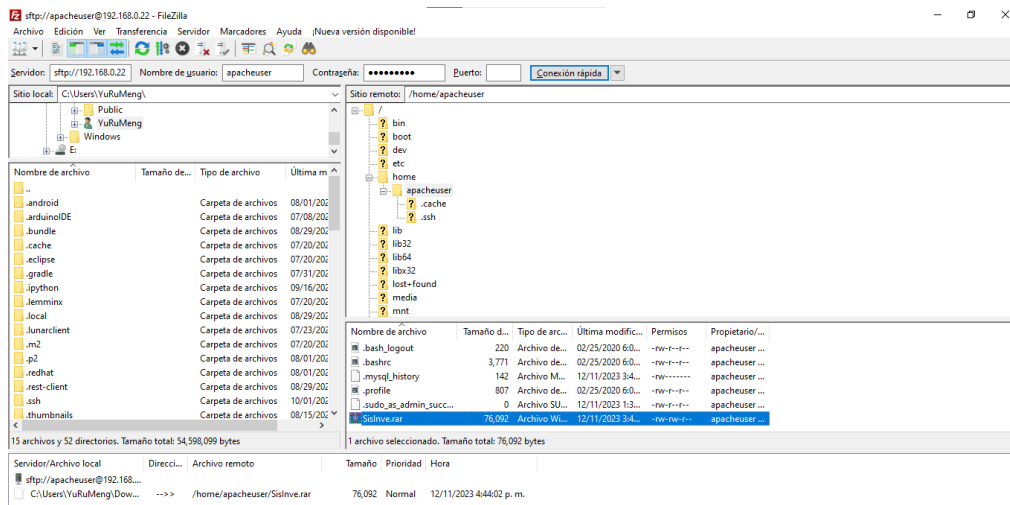


```
apacheuser@apachephp:~$ sudo apt install unrar
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  unrar
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 113 kB of archives.
After this operation, 406 kB of additional disk space will be used.
Get:1 http://pa.archive.ubuntu.com/ubuntu/focal/multiverse/amd64/unrar amd64 1:5.6.6-2build1 [113 kB]
Fetched 113 kB in 1s (154 kB/s)
Selecting previously unselected package unrar.
(Reading database ... 73667 files and directories currently installed.)
Preparing to unpack .../unrar_1%3a5.6.6-2build1_amd64.deb ...
Unpacking unrar (1:5.6.6-2build1) ...
Setting up unrar (1:5.6.6-2build1) ...
update-alternatives: using /usr/bin/unrar-nonfree to provide /usr/bin/unrar (unrar) in auto mode
Processing triggers for man-db (2.9.1-1) ...
```

**figura 81. Instalando unrar**

## Importando el proyecto a Apache

Debemos arrastrar el archivo rar al servidor apache



**figura 82. Poniendo los archivos**

Ponemos ingresamos todos los archivos de la aplicación web en /var/www/html

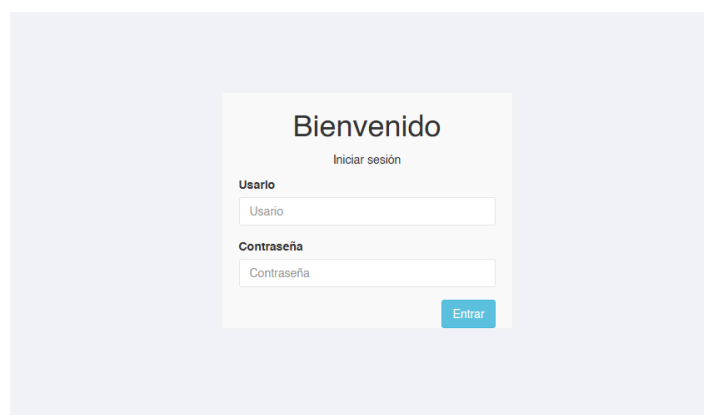
```

apacheuser@apachephp:/var/www/html$ ls
add_group.php      delete_categorie.php  edit_user.php      media.php
add_product.php    delete_group.php     group.php           monthly_sales.php
add_sale.php        delete_media.php      home.php            oswa_inv.sql
add_user.php        delete_product.php    includes            product.php
admin.php           delete_sale.php       index.php           profile.php
ajax.php            delete_user.php       info.php            sale_report_process.php
auth.php            edit_account.php      layouts             sales.php
auth_v2.php         edit_categorie.php    libs               sales_report.php
categorie.php       edit_group.php        LICENSE            SisInve.rar
change_password.php edit_product.php      login_v2.php        Uploads
daily_sales.php     edit_sale.php         logout.php          users.php

```

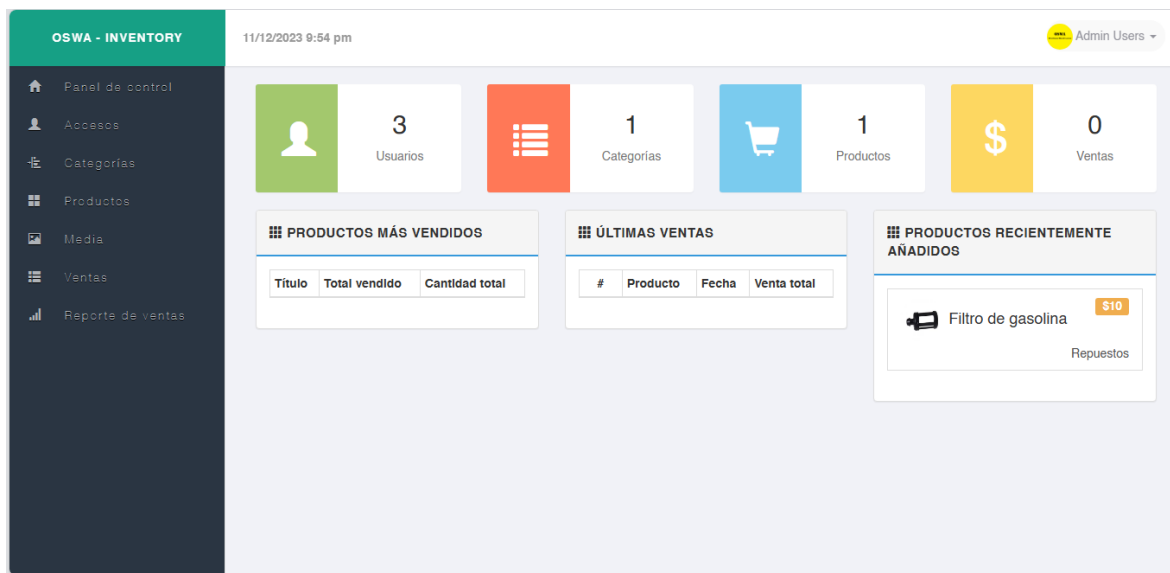
**figura 83. Archivos en su carpeta**

Si entramos a la página vemos que ya carga.



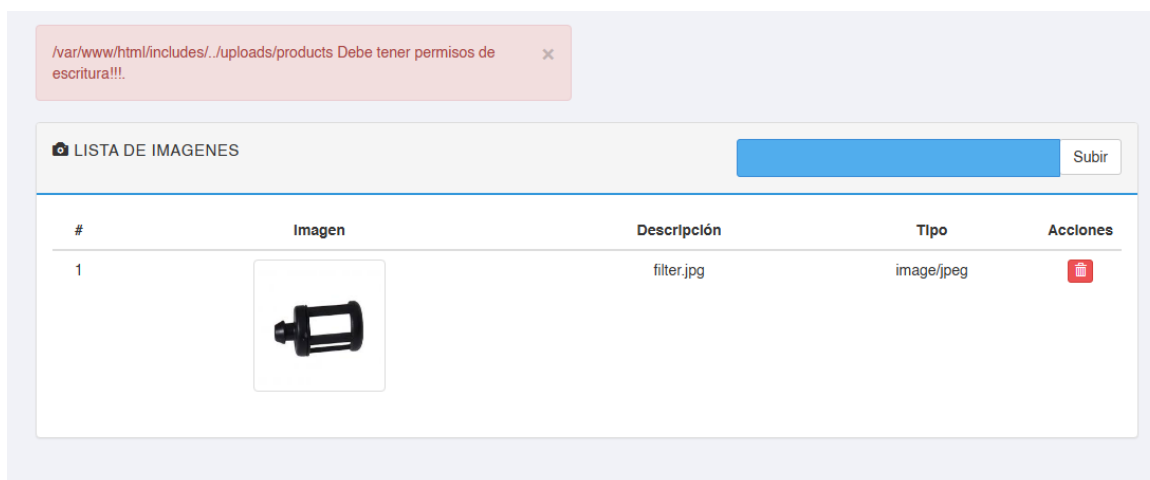
**figura 84. Comprobando la app**

Entramos al panel y podremos ver que efectivamente se puede ingresar.



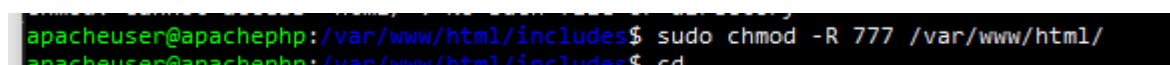
**figura 85. Panel**

Si intentamos subir nuevos productos vemos que no tenemos permisos, para lo mismo por ende tenemos que darle esos permisos.



**figura 86. Error de permisos**

Con `sudo chmod -R 777 /var/www/html/` se arregla facilmente

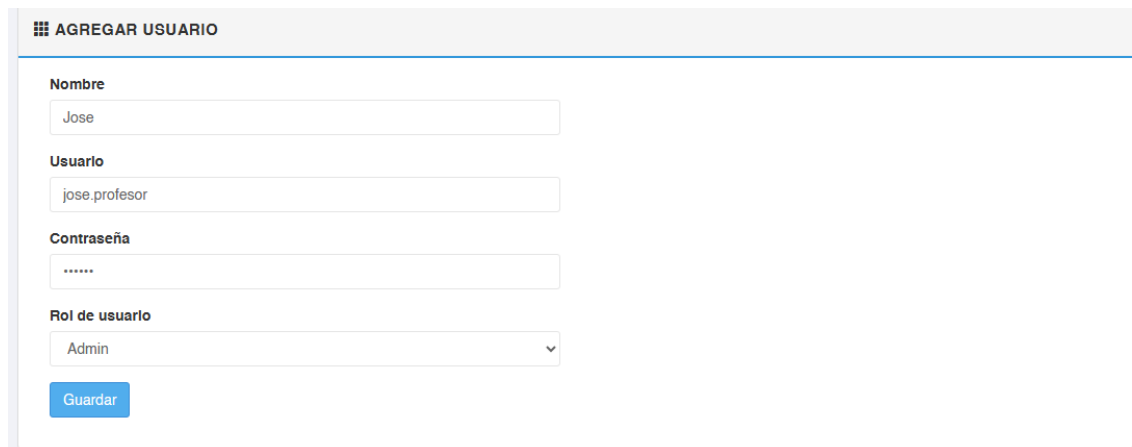


**figura 87. Permisos**

### Creación de Nuevos Usuarios

La app trae 3 usuarios por defecto Special con contraseña Special , Admin con contraseña Admin y User con contraseñar User, esos fueron eliminados por nosotros para evitar fallos de seguridad.

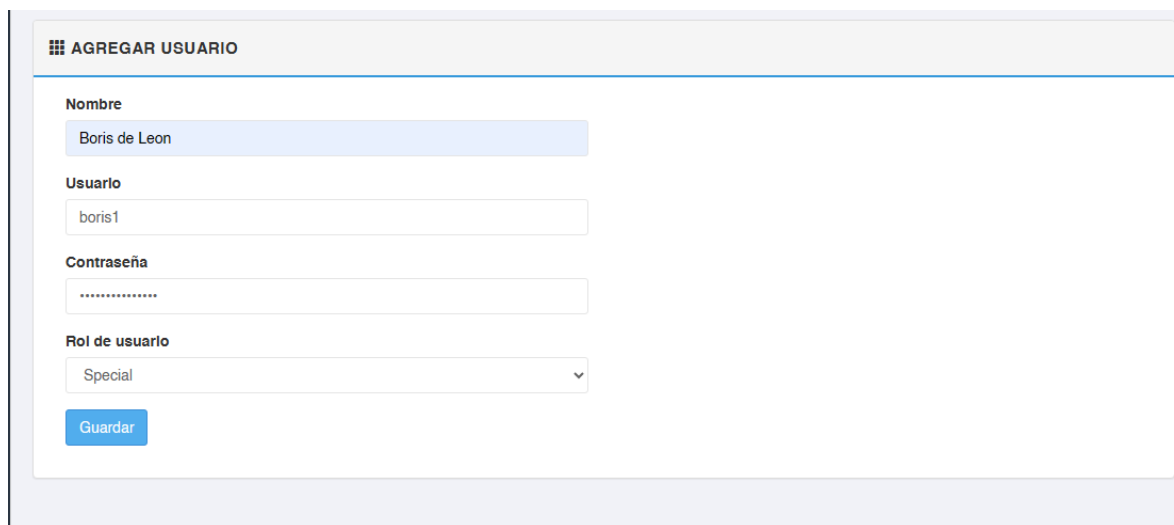
Agregaremos 3 nuevos usuarios, el primero será el usuario del profesor.



The screenshot shows a web form titled "AGREGAR USUARIO" with a grid icon. It contains four input fields: "Nombre" with the value "Jose", "Usuario" with the value "jose.profesor", "Contraseña" with masked characters "\*\*\*\*\*", and "Rol de usuario" with a dropdown menu showing "Admin". A blue "Guardar" button is at the bottom.

***figura 88. Usuario del profesor***

Usuario: Jose.profesor Contraseña: jose1#



The screenshot shows the same "AGREGAR USUARIO" form. The "Nombre" field now contains "Boris de Leon", the "Usuario" field contains "boris1", and the "Contraseña" field contains masked characters "\*\*\*\*\*". The "Rol de usuario" dropdown menu now shows "Special". The blue "Guardar" button remains at the bottom.

***figura 89. Usuario de boris***

Usuario: Boris1    Contraseña: boris123456789#

**Nombre**

Eduardo Samaniego

**Usuario**

edu1720 de usuario

**Contraseña**

edu123#



**Rol de usuario**

User










Guardar

**figura 90. Usuario de Eduardo**

**Usuario:** Edu1720 **Contraseña:** Edu123#

Finalmente tendremos los nuevos 3 usuarios en la plataforma.

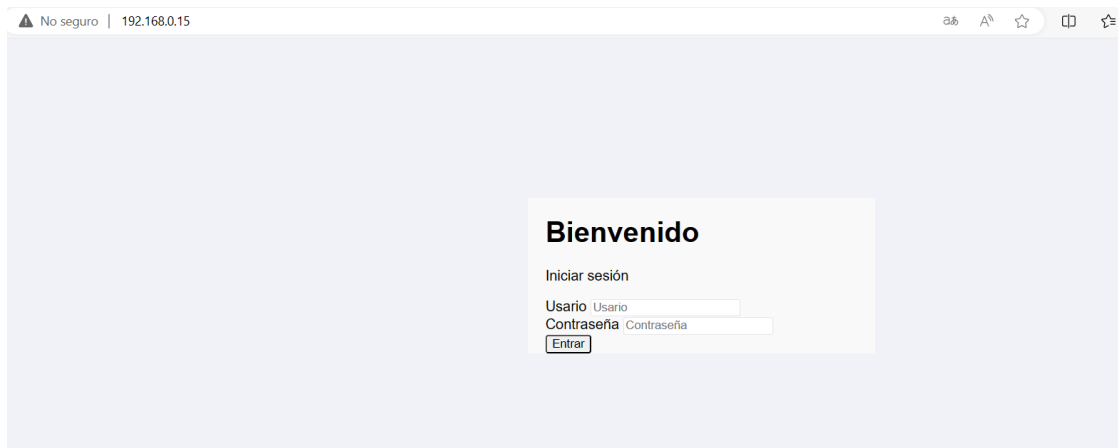
11/12/2023 10:43 pm  José Javier Chirú

USUARIOS							AGREGAR USUARIO
#	Nombre	Usuario	Rol de usuario	Estado	Último login	Acciones	
1	Boris De Leon	Boris1	Special	Activo	11/12/2023 10:42:49 pm	 	
2	Eduardo Samaniego	Edu1720	User	Activo	11/12/2023 10:35:58 pm	 	
3	José Javier Chirú	Jose.profesor	Admin	Activo	11/12/2023 10:43:52 pm	 	

**figura 91. Usuarios**

### Error del CSS

Al levantar los servidores dentro de una red LAN en este caso a la "Internet"; nos proporciona un error en el css, al usuario cliente debido a las referencias que están predeterminadas en la aplicación del servidor de manera online.



**figura 92. Error de css**

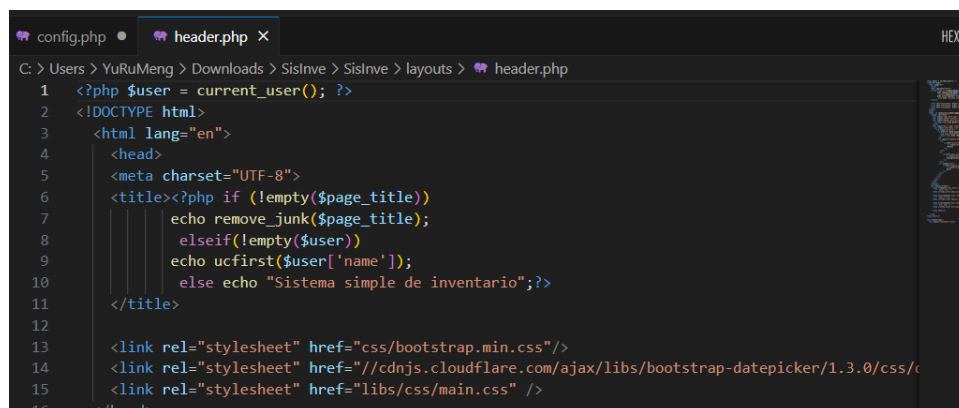
Si intentamos entrar al usuario vemos que no podemos entrar.



**figura 93. Dentro css**

### Resolución del error

Para resolver este error tuvimos que recurrir al profesor, como podemos ver en la siguiente imagen el error es al referenciar al estar referenciados en archivos de internet, como en una intranet no tenemos conexión fuera del propio servidor los css no se pueden bajar por ende da error al cargar el css.



**figura 94. Error de referencias**

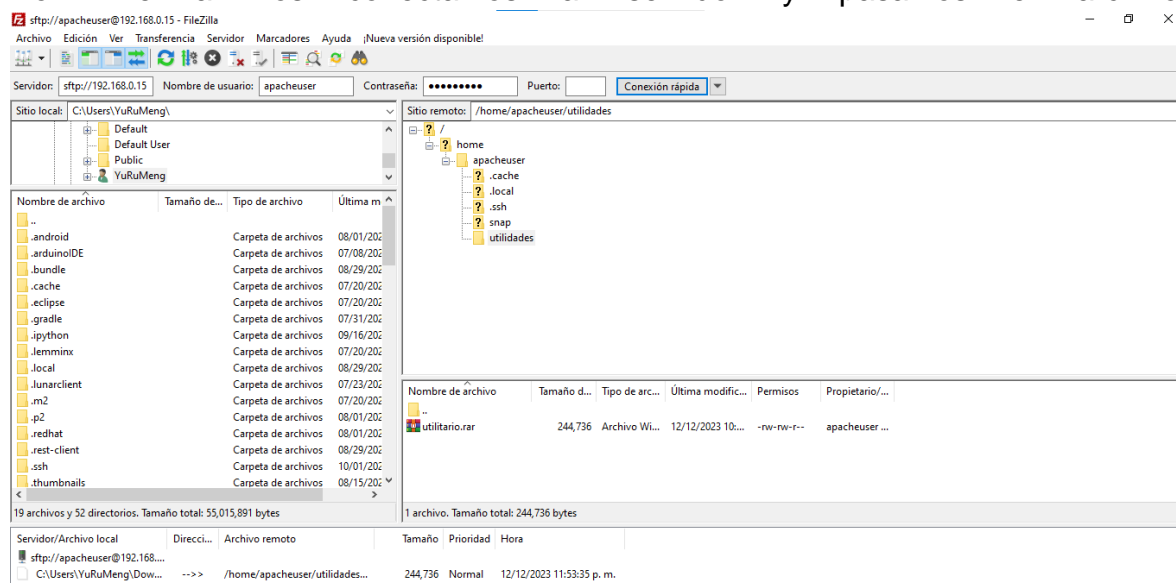


El profesor nos ayudó a crear 3 nuevos archivos que tendrían el css.

css	757,917	113,358	Carpeta de archivos	12/13/2023 12:...
fonts	215,721	109,303	Carpeta de archivos	12/13/2023 12:...
js	103,981	22,739	Carpeta de archivos	12/13/2023 12:...

**figura 95. Nuevos archivos**

Por FileZilla nos conectamos al servidor y pasamos el archivo.



**figura 96. importando los nuevos archivos**

Posteriormente dentro del server descomprimos el archivo.

```

apacheuser@apachephp:~/utilidades$ unrar x utilitario.rar
UNRAR 5.61 beta 1 freeware      Copyright (c) 1993-2018 Alexander Roshal
Extracting from utilitario.rar
Creating      utilitario                OK
Creating      utilitario/css            OK
Extracting    utilitario/css/bootstrap-theme.css      OK
Extracting    utilitario/css/bootstrap-theme.css.map  OK
Extracting    utilitario/css/bootstrap-theme.min.css  OK
Extracting    utilitario/css/bootstrap.css            OK
Extracting    utilitario/css/bootstrap.css.map        OK
Extracting    utilitario/css/bootstrap.min.css        OK
Creating      utilitario/fonts          OK
Extracting    utilitario/fonts/glyphicons-halflings-regular.eot  OK
Extracting    utilitario/fonts/glyphicons-halflings-regular.svg  OK
Extracting    utilitario/fonts/glyphicons-halflings-regular.ttf  OK
Extracting    utilitario/fonts/glyphicons-halflings-regular.woff  OK
Extracting    utilitario/fonts/glyphicons-halflings-regular.woff2 OK
Creating      utilitario/js             OK
Extracting    utilitario/js/bootstrap.js             OK
Extracting    utilitario/js/bootstrap.min.js         OK
Extracting    utilitario/js/npm.js               OK
All OK

```

**figura 97. Descomprimiendo los archivos**

Vemos que dentro de la carpeta tendremos los 3 archivos.

```

utilitario utilitario.rar
apacheuser@apachephp:~/utilidades$ cd utilitario/
apacheuser@apachephp:~/utilidades/utilitario$ ls
css  fonts  js
apacheuser@apachephp:~/utilidades/utilitario$

```

**figura 98. Entrando a la carpeta**

Los movemos a /var/www/html/

```

apacheuser@apachephp:~/utilidades$ cd utilitario/
apacheuser@apachephp:~/utilidades/utilitario$ ls
css  fonts  js
apacheuser@apachephp:~/utilidades/utilitario$ sudo mv * /var/www/html/
apacheuser@apachephp:~/utilidades/utilitario$ cd
apacheuser@apachephp:~$ cd /var/www/html/
apacheuser@apachephp:/var/www/html$ ls
add_group.php  categorie.php  delete_sale.php  fonts  LICENSE  sale_report_process.php
add_product.php  change_password.php  delete_user.php  group.php  login_v2.php  sales.php
add_sale.php  css  edit_account.php  home.php  logout.php  sales_report.php
add_user.php  daily_sales.php  edit_categorie.php  include  media.php  SisInve.rar
admin.php  delete_categorie.php  edit_group.php  index.php  monthly_sales.php  uploads
ajax.php  delete_group.php  edit_product.php  js  oswa_inv.sql  users.php
auth.php  delete_media.php  edit_sale.php  layout  product.php
auth_v2.php  delete_product.php  edit_user.php  libs  profile.php

```

**figura 99. Moviendo las carpetas**

Dentro de la carpeta css tendremos que crear un dateTime.min.css

```

apacheuser@apachephp:/var/www/html/css$ sudo nano dateTime.min.css
[sudo] password for apacheuser:

```

**figura 100. Creando un nuevo css**

Ingresamos el css a este archivo.

```
GNU nano 4.8                                dateTime.min.css
*!
* DatePicker for Bootstrap
*
* Copyright 2012 Stefan Petre
* Improvements by Andrew Rowls
* Licensed under the Apache license v2.0
* http://www.apache.org/licenses/LICENSE-2.0
*
*/.datepicker[padding:4px;border-radius:4px,direction:ltr].datepicker-inline[width:220px].datepicker.datepicker-rtl{d
```

**figura 101. Ingresando las instrucciones**

Dentro de layout entremos al header.

```
pacheuser@apachephp:/var/www/html$ cd layouts/
pacheuser@apachephp:/var/www/html/layouts$ ls
admin_menu.php footer.php header.php special_menu.php user_menu.php
pacheuser@apachephp:/var/www/html/layouts$ nano header.php
```

**figura 102. layout**

Finalmente cambiamos las referencias del css.

```
GNU nano 4.8                                header.php
<?php $user = current_user(); ?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title><?php if (!empty($page_title))
    echo remove_junk($page_title);
    elseif(!empty($user))
    echo ucfirst($user['name']);
    else echo "Sistema simple de inventario";?>
</title>
<link rel="stylesheet" href="css/bootstrap.min.css"/>
<link rel="stylesheet" href="css/dateTime.min.css" />
<link rel="stylesheet" href="libs/css/main.css" />
</head>
<body>
```

**figura 103. Cambiando las referencias del header**

Luego reiniciamos el apache2.

```
apacheuser@apachephp:/var/www/html/layouts$ apacheuser@apachephp:/var/www/html/layouts$ sudo systemctl restart apache2
apacheuser@apachephp:/var/www/html/layouts$
```

**figura 104. Reiniciando apache2**

## Hardening

Este va a ser un hardening sencillo en ambos servidores, primero, en el servidor apache utilizando `rm -r` eliminamos los archivos `index.html` y `info.php` que fueron creados al inicio.

```
apacheuser@apachephp:/var/www/html$ rm -r index.html
apacheuser@apachephp:/var/www/html$ ls
```

**figura 105. Borrarnos el index**

```
apacheuser@apachephp:/var/www/html$ rm -r info.php
```

**figura 106. Borramos info.php**

Deshabilitar el ServerToken y el ServerSignature

Deshabilita la opción ServerSignature y ServerTokens para ocultar información sobre la versión de Apache. Si usamos curl para ver el baner vemos que nos sale la versión de apache.

```
dbuser@mariadb:~$ curl -I http://192.168.0.22
HTTP/1.1 200 OK
Date: Mon, 11 Dec 2023 22:07:07 GMT
Server: Apache/2.4.41 (Ubuntu)
Set-Cookie: PHPSESSID=95b3grv9d05u32qgqa3r19r9dd; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
```

**figura 107. Vemos el baner**

Dentro del repositorio de apache tenemos que activar la regla.

```
GNU nano 4.8 /etc/apache2/apache2.conf
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives. See also the AllowOverride
# directive.
#
AccessFileName .htaccess
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<FilesMatch "^\.ht">
    Require all denied
</FilesMatch>
#
# The following directives define some format nicknames for use with
# a CustomLog directive.
#
# These deviate from the Common Log Format definitions in that they use %O
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
ServerTokens Prod
ServerSignature Off
```

**figura 108. Nuevas políticas**

Si hacemos la prueba nuevamente ya no nos sale el tipo de servidor ni su versión.

```
dbuser@mariadb:~$ curl -I http://192.168.0.22
HTTP/1.1 200 OK
Date: Mon, 11 Dec 2023 22:08:49 GMT
Server: Apache
Set-Cookie: PHPSESSID=q9eb8jlu47jnfhvs2279f5ea33; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
```

**figura 109. Volvemos a ver el baner**

Instalaremos el Fail2ban para ataques de crackeos de contraseña y ataques de fuerza bruta.

```
apacheuser@apachephp:~$ sudo apt install fail2ban
[sudo] password for apacheuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 1 not upgraded.
Need to get 444 kB of archives.
After this operation, 2,400 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

**figura 110. Instalamos fail2ban**

Posteriormente si vamos a la configuración del fail2ban vemos que son 10 m para el ban y el número de intentos fallidos son 5.

```
apacheuser@apachephp:~$ sudo nano /etc/fail2ban/jail.conf
apacheuser@apachephp:~$
```

**figura 111. Vemos la configuración**

```
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime = 10m

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 10m

# "maxretry" is the number of failures before a host get banned.
maxretry = 5

# "maxmatches" is the number of matches stored in ticket (resolvable via tag <matches> in actions).
maxmatches = %(maxretry)s

# "backend" specifies the backend used to get filter modification
```

**figura 112. Vemos los intentos**

Hacemos lo mismo con el servidor de mariadb.

```
dbuser@mariadb:~$ sudo systemctl restart mariadb
dbuser@mariadb:~$ sudo apt install fail2ban
[sudo] password for dbuser:
granySorry, try again.
[sudo] password for dbuser:
sudo: 1 incorrect password attempt
dbuser@mariadb:~$ sudo apt install fail2ban
[sudo] password for dbuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 444 kB of archives.
After this operation, 2,400 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

**figura 113. Instalamos fail2ban en mariadb**

Ahora intentaremos hacerles escaneo de puertos y realmente no nos deja ver nada.

```
SYN Stealth Scan Timing: About 42.50% done; ETC: 20:27 (0:00:14 remaining)
Nmap scan report for 192.168.0.15
Host is up (0.00086s latency).
All 1000 scanned ports on 192.168.0.15 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 34.88 seconds
└─(kali@kali)-[~]
```

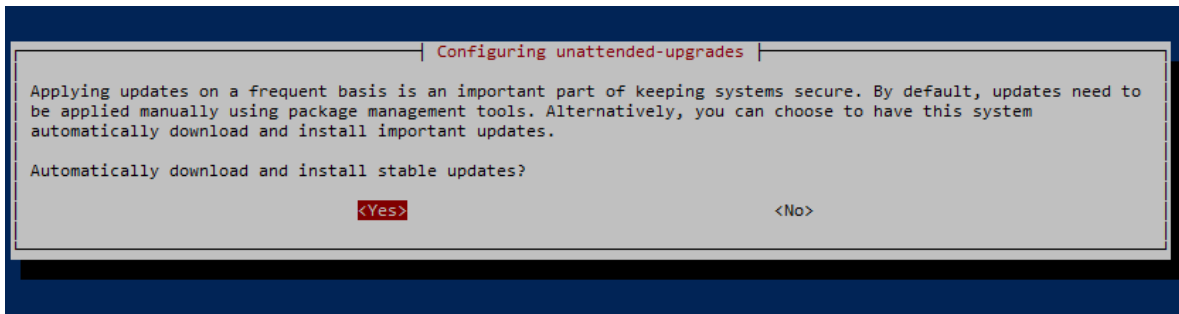
**figura 114. nmap**

Instalaremos la función de unattended-upgrades para que el sistema se actualice automáticamente.

```
l: Unable to locate package unattended-upgrades
apacheuser@apachephp:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades
apacheuser@apachephp:~$
```

### **figura 115. Instalando el unattended-upgrades**

Después de ingresar el comando ponemos Yes.



### **figura 116. Decidiendo**

Y Entramos a la configuración del desatendido.

```
apacheuser@apachephp:~$ cd /etc/ap
apache2/  apparmor/  apparmor.d/  apport/      apt/
apacheuser@apachephp:~$ cd /etc/ap
apache2/  apparmor/  apparmor.d/  apport/      apt/
apacheuser@apachephp:~$ cd /etc/ap
apache2/  apparmor/  apparmor.d/  apport/      apt/
apacheuser@apachephp:~$ cd /etc/apt/apt.conf.d/
apacheuser@apachephp:/etc/apt/apt.conf.d$ ls
01autoremove      15update-stamp      20auto-upgrades    50command-not-found  99update-notifier
01-vendor-ubuntu  20apt-esm-hook.conf  20packagekit       50unattended-upgrades
10periodic        20archive           20snapd.conf       70debconf
apacheuser@apachephp:/etc/apt/apt.conf.d$ nano 50unattended-upgrades
apacheuser@apachephp:/etc/apt/apt.conf.d$ sudo nano 50unattended-upgrades
apacheuser@apachephp:/etc/apt/apt.conf.d$ apacheuser@apachephp:/etc/apt/apt.conf.d$
```

### **figura 117. Viendo la configuración**

Aquí podemos modificar todos, nos cercioramos de que las distros de seguridad estén activas.

```

GNU nano 4.8                               50unattended-upgrades
// Automatically upgrade packages from these (origin:archive) pairs
//
// Note that in Ubuntu security updates may pull in new dependencies
// from non-security sources (e.g. chromium). By allowing the release
// pocket these get automatically pulled in.
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-upgrades
    // should also install from here by default.
    "${distro_id}ESMA:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    //
    "${distro_id}:${distro_codename}-updates";
    //
    "${distro_id}:${distro_codename}-proposed";
    //
    "${distro_id}:${distro_codename}-backports";
};

// Python regular expressions, matching packages to exclude from upgrading
Unattended-Upgrade::Package-Blacklist {
    // The following matches all packages starting with linux-
    // "linux-";

    // Use $ to explicitly define the end of a package name. Without
    // the $, "libc6" would match all of them.
    // "libc6$";
    // "libc6-dev$";
    // "libc6-i686$";

    // Special characters need escaping
    // "libstdc\+\+6$";

    // The following matches packages like xen-system-amd64, xen-utils-4.1,
    // xenstore-utils and libxenstore3.0
    // "(lib)?xen(store)?";

```

**figura 118. Verificamos las actualizaciones**

Posteriormente hacemos lo mismo con el servidor de bases de datos.

```

dbuser@mariadb:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades_

```

**figura 119. Instalamos el paquete en el mariadb**

Ahora vamos a quitar el index of que nos aparece a la hora de poder navegar entre directorios para eso nos vamos a la configuración de apache y quitamos de la parte de /var/www/ quitamos los indexes.



```
apacheuser@apachephp: /etc/apache2
GNU nano 4.8
includeOptional mods-enabled/*.conf
# Include list of ports to listen on
Include ports.conf

# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

**figura 120. Quitamos el index**

Posteriormente activamos el firewall y agregamos el puerto 80.

```
apacheuser@apachephp:/etc/apache2$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
apacheuser@apachephp:/etc/apache2$ sudo ufw allow 80
Rule added
Rule added (v6)
apacheuser@apachephp:/etc/apache2$
```

**figura 121. Abrimos los puertos**

Podemos ver que puertos tenemos abiertos.

```
Active Internet Connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                     LISTEN
tcp6       0      0 :::22                   :::*                     LISTEN
udp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
udp        0      0 10.0.2.15:68            0.0.0.0:*
```

**figura 122. Vemos los puertos**

También vemos el firewall en el servidor de bases de datos.

```
dbuser@mariadb:~$ sudo ufw status
[sudo] password for dbuser:
Status: active

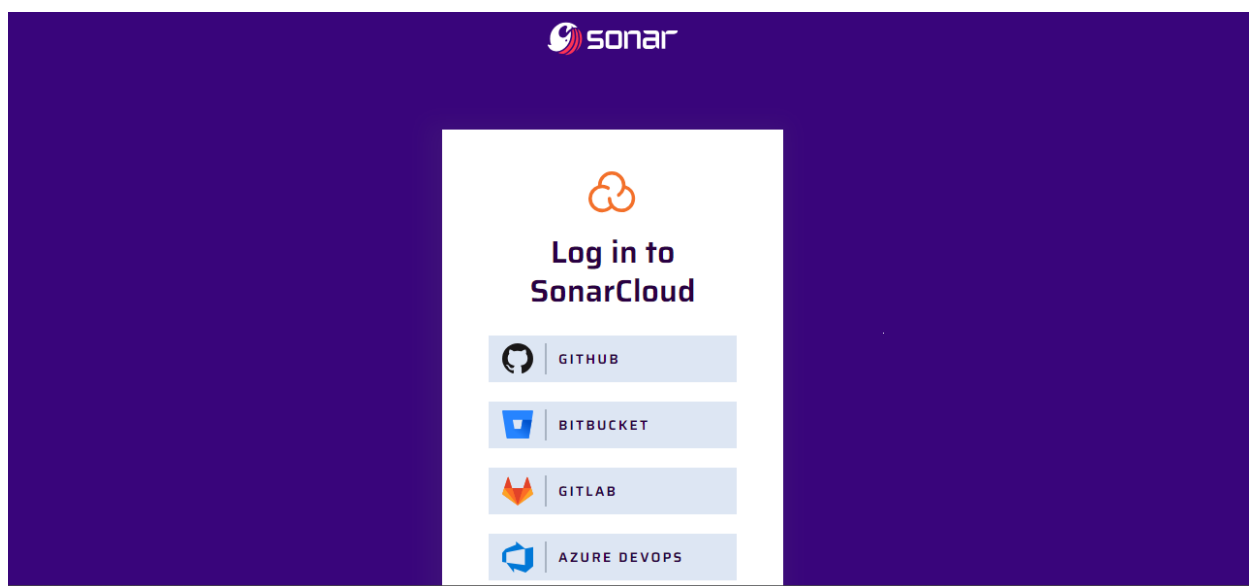
To Action From
--
3306 ALLOW Anywhere
3306 (v6) ALLOW Anywhere (v6)

dbuser@mariadb:~$
```

*figura 123. Vemos los puertos de mariadb*

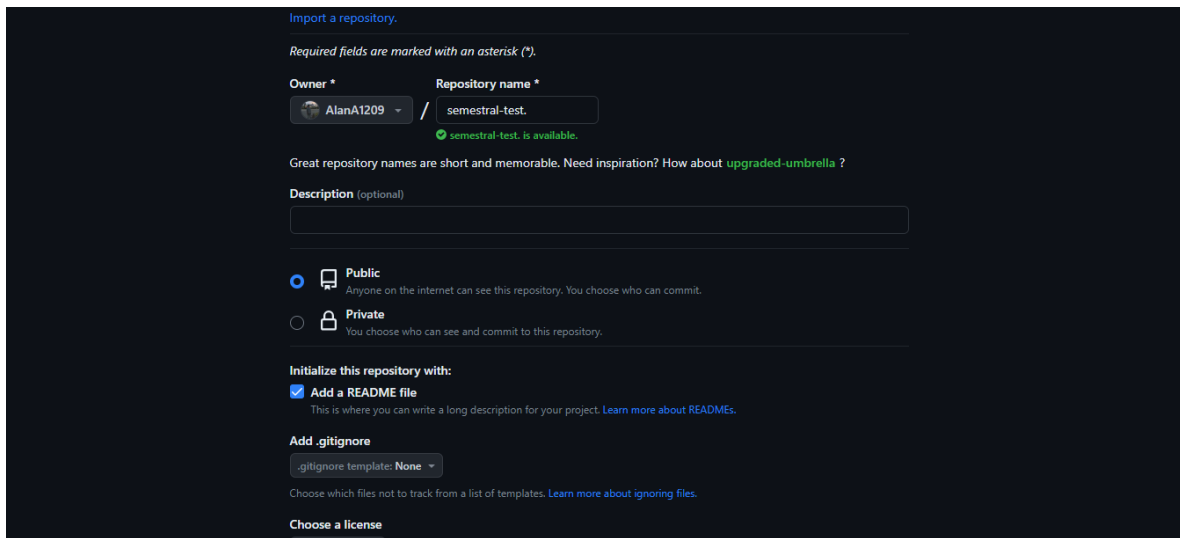
## SonarCloud para escaneo de vulnerabilidades

Iniciamos buscando en Google Sonar Cloud



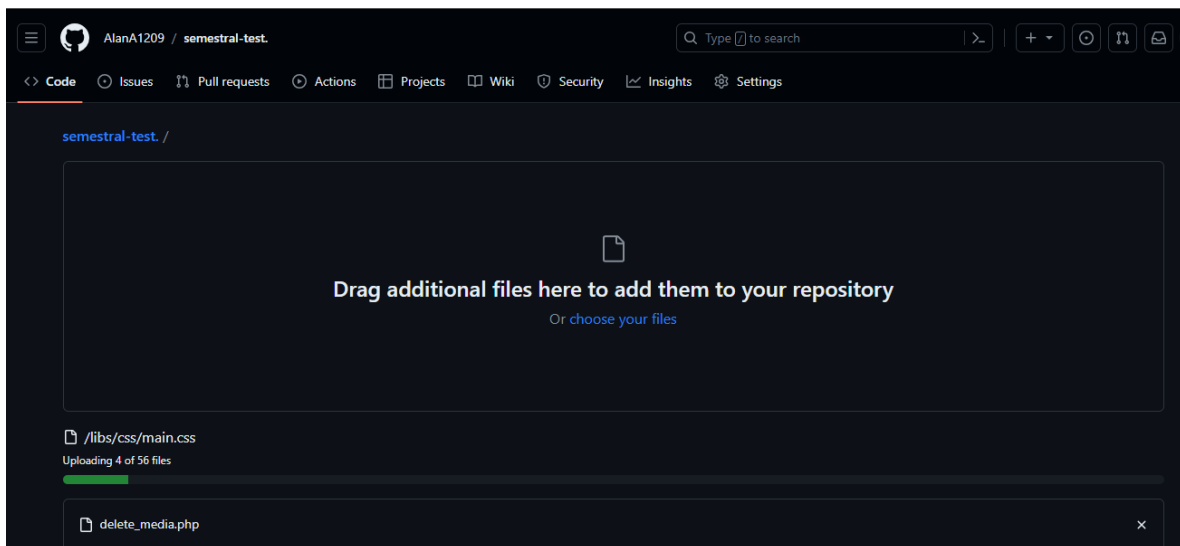
*figura 124. Entrando a Sonar Cloud*

Importamos un nuevo repositorio de GitHub.



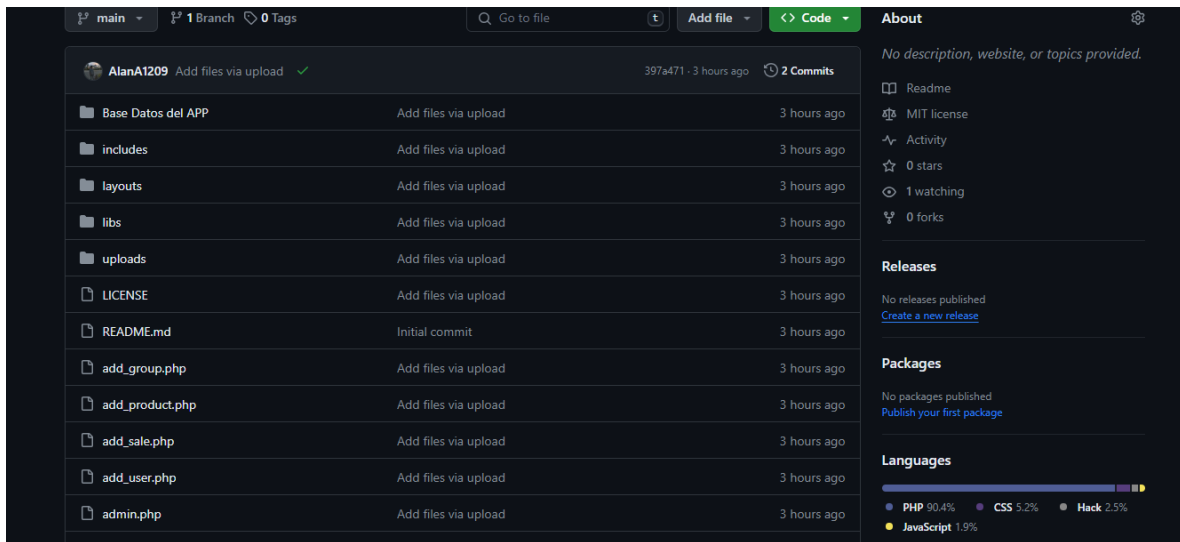
**figura 125. Creando el repositorio en GitHub**

Arrastramos los archivos al nuevo repositorio.



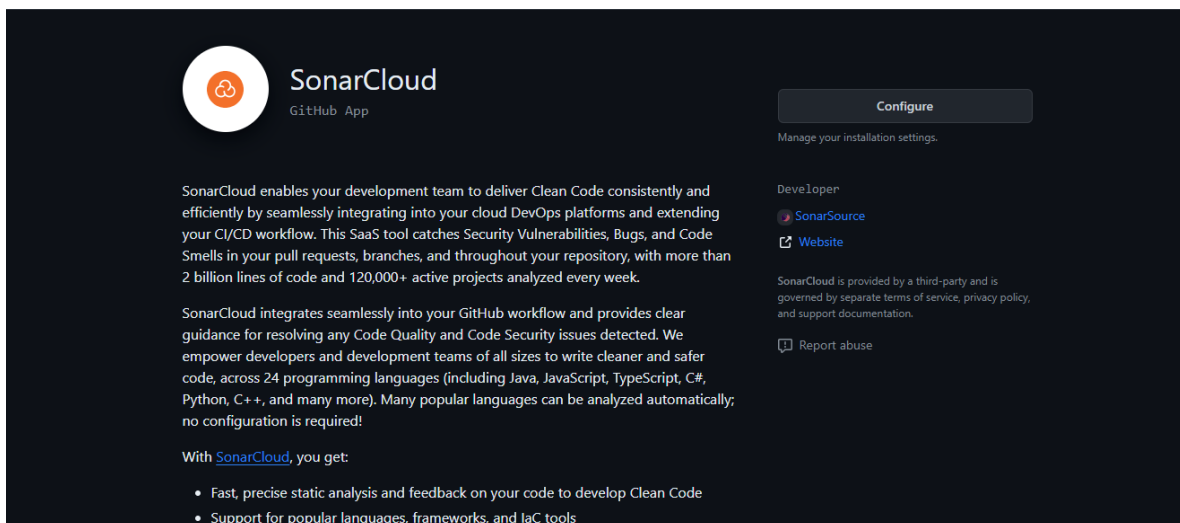
**figura 126. Arrastrando los archivos**

Observamos el Repositorio ya creado.



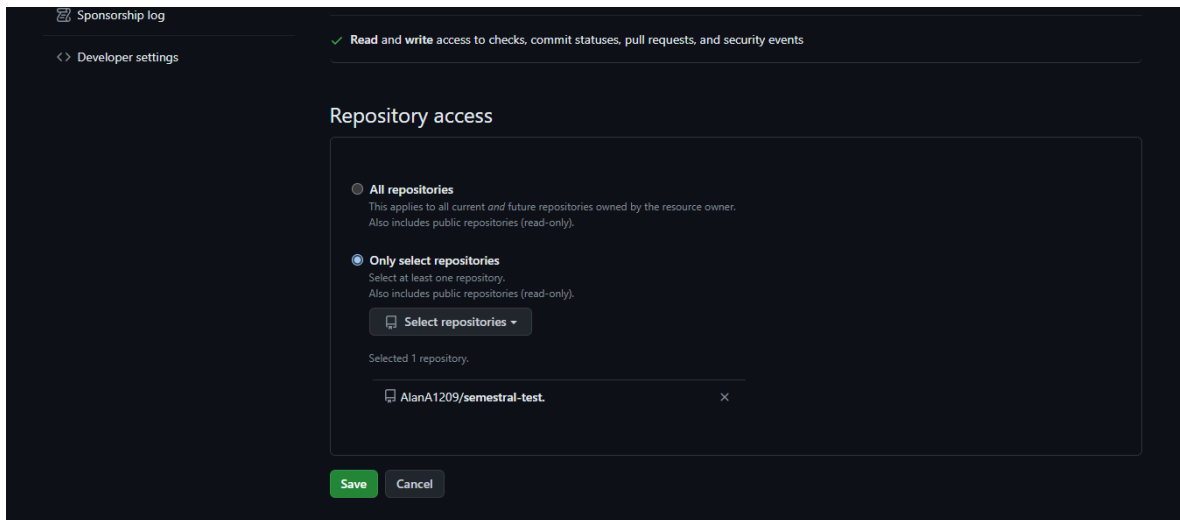
**figura 127. repositorio**

Procedemos a configurar SonarCloud.



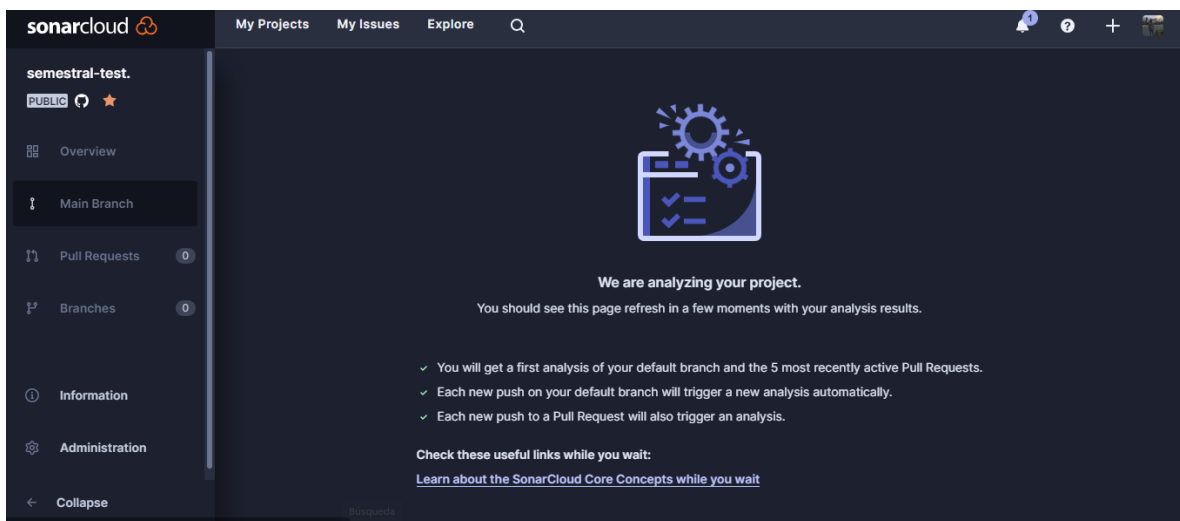
**figura 128. Configurando el SonarCloud**

Seleccionamos el repositorio creado.



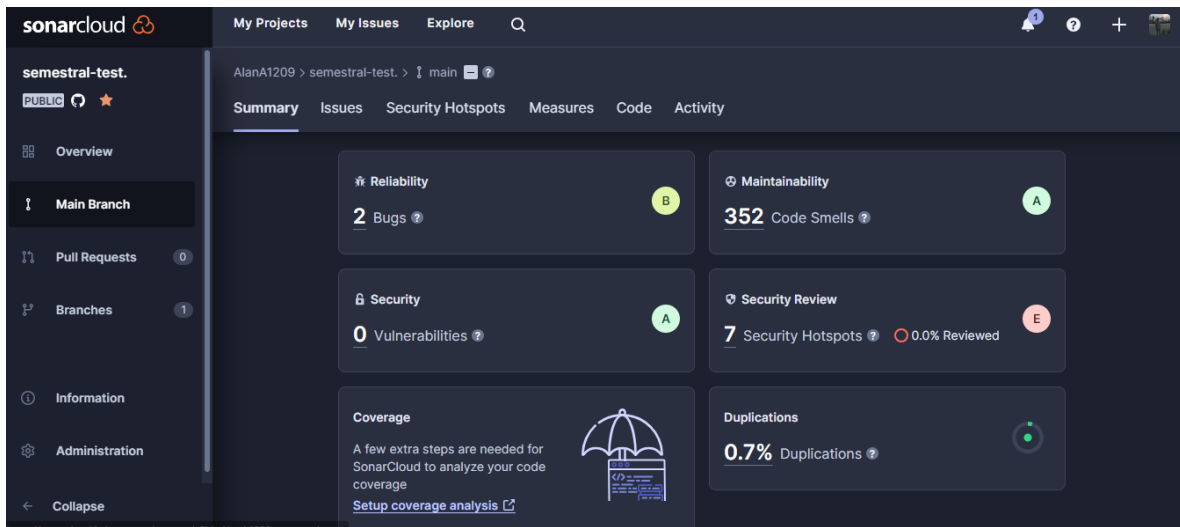
**figura 129. Dándole acceso al repositorio**

Esperamos a que se realice el análisis.



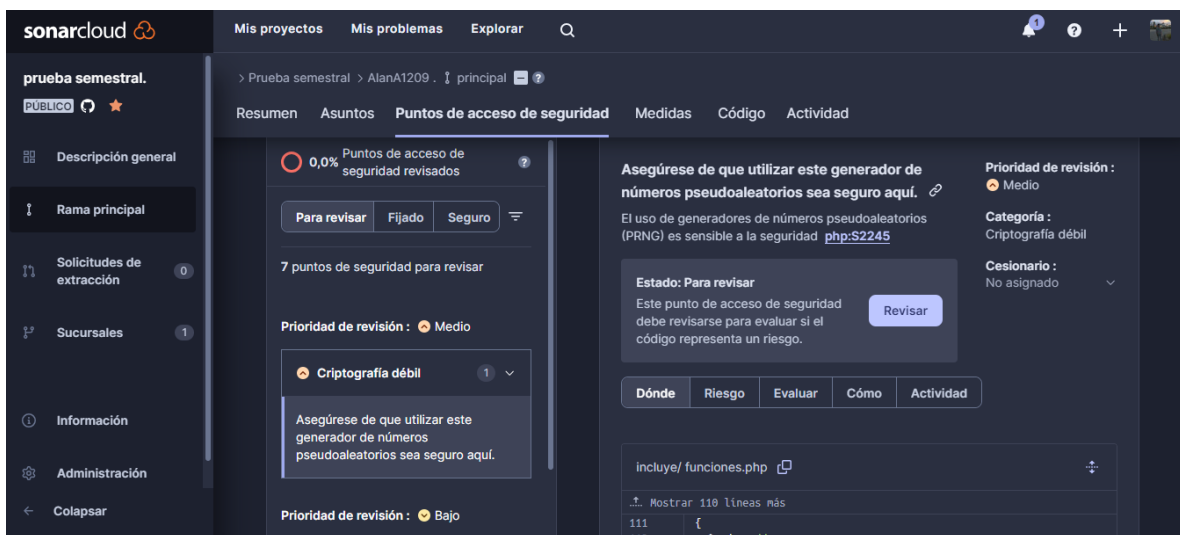
**figura 130. Realizando el análisis**

Observamos el resumen del escaneo.



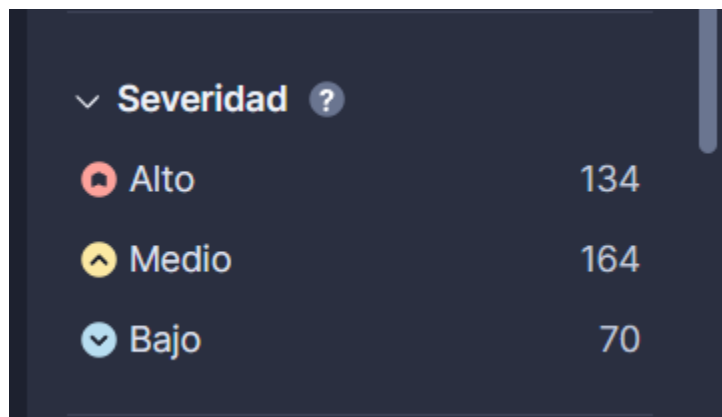
**figura 131. Resumen de escaneo**

Podemos ver los puntos de accesos de seguridad que pueden ser comprometidos.



**figura 132. Viendo los puntos de compromiso**

Para revisar el resto del análisis utilizamos el modelo stride para el análisis de vulnerabilidades, en los archivos adjuntos tenemos los dos archivos junto a las diapositivas. En donde se encontrarán todo el contenido del análisis. Hay que destacar que se encontramos en el análisis más de 368 problemas con la web SonarCloud.



**figura 133. Cantidad escaneada en Sonarcloud**

Mientras tanto se encontraron más de 1,544 entre errores y vulnerabilidades en el código con la plataforma Codacy.

#### Issues breakdown



**figura 134. Cantidad escaneada en Codacy**

## Vulnerabilidades encontradas y parcheadas en el desarrollo del proyecto

### Consejos de vulnerabilidades a nivel de servidores

Además del desarrollo de un escaneo de vulnerabilidades se descubrieron ciertas vulnerabilidades claras en la construcción del aplicativo, se tomó en cuenta primordialmente el desarrollo para mitigar un posible ataque que pueda sufrir en el futuro nuestra aplicación web, empezando por la implementación del Fail2Ban para

prevenir el ataque de contraseñas y vulnerabilidad que puedan ser explotados a través de diferentes aplicaciones. Por otro lado, también instalamos servicios para las actualizaciones automatizadas, hay que tomar en cuenta el hecho de que todos los parches de seguridad no son necesarios por ende la aplicación nos va a preguntar antes de actualizar cualquier cosa, por otra parte, debe tomarse en cuenta que una mala configuración de aplicativos puede llegar a una vulnerabilidad más grande, pero esto claro es mera suposición, se debe tomar en cuenta los diferentes apartados de seguridad no solamente en el área del servidor también en el área del administrador y sus dispositivos, una vez ingresados a la plataforma sus dispositivos pueden traer errores de seguridad, por ende se recomienda un escaneo de los diferentes dispositivos conectados a la red, esto mitigaría un poco los diferentes ataques provenientes a los diferentes dispositivos conectados, las vulnerabilidades provenientes de las contraseñas recomendamos el uso seguro de contraseñas con todos los requerimientos de seguridad, algo más que recomendamos es el seccionamiento de los directorios del servidor, utilizando Docker u otros servicios de contenedores para particionar los servicios y utilizar la aplicación de manera segura.

### **Consejos de vulnerabilidades a nivel de redes**

Dentro de las buenas prácticas de seguridad en contraseñas es contar con una contraseña robusta como se puede apreciar en la siguiente imagen:



**figura 135. Configuración de contraseñas**

Una tengamos una contraseña robusta, pasamos al apartado de seguridad en la capa del modelo OSI, específicamente a la capa 7 aplicativo.

Donde vamos a restringir dominios o sitios web que sean propenso posibles ataques o riesgos a ser víctimas de malwares o ransomwares. Por parte del cliente este caso si nuestra Internet cuentan con un proveedor de internet es necesario contar con control parental donde podemos observar el tráfico de lo que navega y a su vez administrar y detectar actividades sospechosas como veremos a continuación:





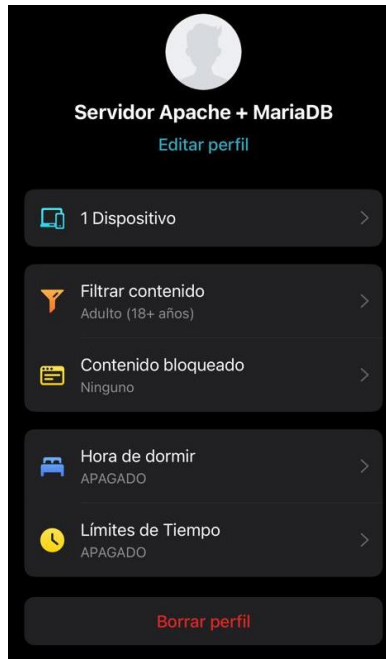
**Figura 136. Supervisión de subidas y descargas**

Así como también podemos restringir clientes a nuestra red como medida de seguridad y se puede ver en la siguiente imagen:



**Figura 137. Restricción del equipo por medio de su Mac.**

Ya para finalizar el control parental en las siguientes imágenes podemos ver en las diferentes funciones para tener un entorno controlado de la capa 7 en caso del cliente podemos bloquear también el acceso a la red como para el cliente en la segunda imagen:



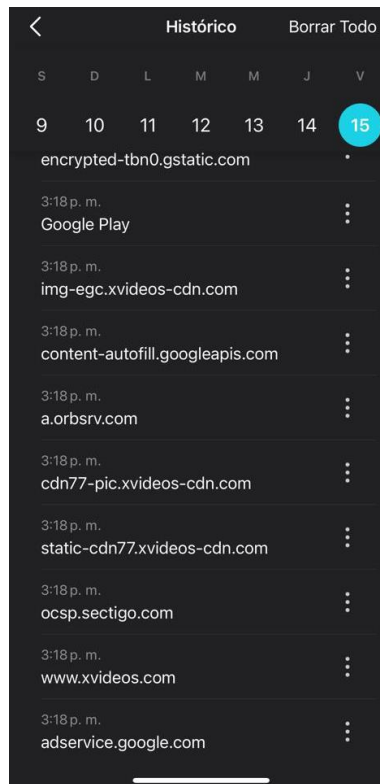
**Figura 138. Configuración del modo parental y mas**

Como se vio en la imagen anterior, apreciamos como también contamos con restricciones en cuanto límite de tiempo; esto a su vez es recomendable tenerlo habilitado para tener auditoria limpia.



**Figura 139. Configuración de bloqueo al acceso a internet**

Ya este apartado simplemente es más administrativo y dependerá en el entorno en el que estemos ya sea una oficina o un conjunto de departamentos.



***Figura 140. Historial de tráfico de un equipo cliente***

Ya para finalizar podemos apreciar cómo se puede ver el tráfico dentro de la aplicación los sitios visitados por el cliente... en este caso recordemos que si no tenemos los filtro y restricciones implantadas nos ocurrirá esta vulnerabilidad. Lo que se aconseja seguir las métricas y buenas prácticas para tener un entorno sano, versátil y contralado.

### **Vulnerabilidades a nivel de escaneo**

En base a los analizadores de códigos como sonarcloud y codacy, podemos ver que a nivel de vulnerabilidades en base al escaneo.

El analizador de código sonarcloud nos informan que tenemos en total 352 errores de código y a su vez estos necesitan mantenimiento, 2 errores a nivel de fiabilidad ubicado en la base de datos y en la sesión del programa. Sin embargo, en la evaluación de la seguridad nos muestra que tiene 7 puntos en el código que cuentan con una calificación de E, dándonos a entender que estos son puntos críticos y deben ser puntos prioritarios a revisar, las en los que se ubican esos puntos críticos son:

- **add\_user.php**
- **cambiar\_password.php**
- **edit\_user.php**

- **Incluye/funciones.php**
- **incluye/sql.p.p**

Luego, en utilizando el analizador de código codacy, nos registró que tenemos 1635 Issues en nuestra aplicación, de estos errores o vulnerabilidades tenemos:

- 1k de tipo Code Style.
- 502 de tipo security .
- Proponso a errores 38 códigos en nuestra aplicación.

### **Vulnerabilidades encontradas**

Dentro de las vulnerabilidades y errores encontradas en base a los analizadores de código estático como lo son sonarcloud y codacy, tenemos.

- 1) **Posible generador de números pseudoaleatorios inseguro:** El uso de generadores de números pseudoaleatorios (PRNG) es sensible a la seguridad. Por ejemplo, ha dado lugar en el pasado a las siguientes vulnerabilidades:
  - [CVE-2013-6386](#)
  - [CVE-2006-3419](#)
  - [CVE-2008-4102](#)

Cuando el software genera valores predecibles en un contexto que requiere imprevisibilidad, es posible que un atacante adivine el siguiente valor que y utilice esta suposición para hacerse pasar por otro usuario o acceder a información confidencial.

- 2) **Algoritmo hash débil (usuario):** Los algoritmos hash débiles son más propensos a ataques de fuerza bruta, donde un atacante intenta probar diferentes combinaciones de contraseñas hasta encontrar la correcta. Un buen algoritmo hash debería ser resistente a este tipo de ataques.
- 3) **Algoritmo hash débil (contraseña):** Cuando hablamos que un algoritmo hash es débil para contraseñas, significa que no es bueno para protegerlas. Puede ser rápido de calcular, lo que facilita a los atacantes probar muchas contraseñas rápidamente. También podría tener problemas, como dos contraseñas diferentes que generan el mismo código, lo cual es malo para la seguridad.
  - Ejemplos o Casos:

LinkedIn (2012): En 2012, LinkedIn sufrió una violación de seguridad en la que se filtraron millones de contraseñas de usuarios. Se descubrió que las contraseñas estaban almacenadas utilizando el algoritmo hash SHA-1 sin salting. Esto hizo que fuera más fácil para los atacantes realizar ataques de fuerza bruta y descifrar las contraseñas.

Yahoo (2013-2014): En varios incidentes entre 2013 y 2014, Yahoo sufrió violaciones de seguridad masivas. En algunos casos, las contraseñas se almacenaron utilizando algoritmos hash débiles y desactualizados, como MD5, lo que facilitó a los atacantes descifrar las contraseñas de los usuarios.

Adobe (2013): En 2013, Adobe experimentó una violación de seguridad en la que se filtraron datos de millones de usuarios. Las contraseñas se almacenaban utilizando el algoritmo hash SHA-1 sin salting, lo que permitió a los atacantes comprometer un gran número de cuentas.

- 4) **El código contiene expresiones de salida (Mala Práctica):** Si el código incluye expresiones de salida que presentan información sensible directamente al usuario, como contraseñas, tokens de sesión u otra información confidencial, existe el riesgo de que esta información se divulgue de manera insegura.

- Ejemplos o casos:

Ataque XSS en MySpace (2005): Descripción: En 2005, MySpace sufrió un ataque XSS masivo en el que los atacantes aprovecharon la falta de filtrado adecuado en la entrada de datos y la salida en la plataforma. Publicaron códigos JavaScript maliciosos en perfiles de usuario, que se ejecutaron cuando otros usuarios visitaron esas páginas. Este incidente afectó a millones de usuarios y llevó a la ejecución de código malicioso en los navegadores de las víctimas.

Divulgación de Información en GitHub (2016): Descripción: GitHub experimentó un problema en 2016 cuando algunos desarrolladores incluyeron en sus repositorios tokens de acceso y credenciales API en el código fuente. Estos secretos se presentaron en la salida de diffs, lo que permitió a atacantes potenciales encontrar y utilizar esas credenciales. Los atacantes pudieron aprovechar estas credenciales para acceder a repositorios privados y realizar acciones no autorizadas.

5) **Defina una constante (Mala Práctica):** El problema con tener cadenas duplicadas es que puede hacer que el código sea más difícil de mantener y refactorizar. La refactorización es el proceso de reestructurar el código sin cambiar su comportamiento externo. Aquí hay algunas vulnerabilidades que puede generar esta mala práctica:

- Inyección de Código: Si las cadenas duplicadas se utilizan en consultas a bases de datos o en construcciones de comandos de sistema, podría haber riesgos de inyección de código si una de las instancias no se maneja adecuadamente. Por ejemplo, si una cadena duplicada se utiliza en una consulta SQL y una instancia no se escapa correctamente, podría abrirse la puerta a ataques de inyección SQL.
- Ataques de Ingeniería Social: Mensajes inconsistentes o confusos en la interfaz de usuario pueden ser utilizados por atacantes en intentos de ingeniería social para confundir a los usuarios y llevarlos a realizar acciones no deseadas.

6) **El método query() contiene una expresión de salida. (Mala Práctica):** Si el método query() contiene una expresión de salida directa sin tener en cuenta la seguridad, puede considerarse una mala práctica por varias razones, especialmente si se trata de la salida de información sensible o si se presenta la información de una manera que podría ser aprovechada por un atacante. Aquí hay algunas razones por las cuales esto podría ser problemático:

- Ejemplos o casos:

Inyección de Código (SQL Injection): Si la salida directa en el método query() no se filtra o escapa adecuadamente, podría dar lugar a ataques de inyección de código SQL. Un atacante podría insertar instrucciones SQL maliciosas en los datos de entrada, comprometiendo la seguridad de la base de datos.

Divulgación de Información Confidencial: Si el método query() devuelve información sensible sin filtrar, como contraseñas o datos privados, podría exponer información que no debería ser accesible de manera directa.

Problemas de Seguridad: Dependiendo del contexto y de lo que haga el método query(), la salida directa podría presentar información técnica que podría ser utilizada por atacantes para encontrar debilidades en el sistema.

- 7) **El método db\_connect() contiene una expresión de salida. (Mala Práctica):** Si el método db\_connect() devuelve información técnica detallada sobre la conexión a la base de datos, como nombres de usuario, contraseñas o detalles de la configuración, podría exponer detalles internos que podrían ser aprovechados por un atacante.

- Caso hipotético

Supongamos que hay una aplicación web que maneja información financiera y utiliza una base de datos para almacenar datos de los usuarios. El desarrollador de la aplicación crea un método db\_connect() que devuelve información técnica detallada sobre la conexión a la base de datos:

Consecuencias:

Divulgación de Credenciales: La salida del método incluye el nombre de usuario y la contraseña de la base de datos. Si esta información se expone, un atacante podría utilizarla para acceder directamente a la base de datos.

Exposición de Detalles de Configuración: Se revelan detalles específicos de la configuración de la base de datos, como el host y el puerto. Esto proporciona a una atacante información valiosa sobre la infraestructura de la aplicación.

Riesgo de Ataques Específicos: Un atacante informado podría utilizar esta información para realizar ataques dirigidos, como intentos de fuerza bruta con las credenciales conocidas o intentos de explotar vulnerabilidades específicas en la versión de la base de datos.

### **Consejos para mitigar estas vulnerabilidades**

Considerando las amenazas previamente identificadas en el informe, se recomienda seguir las siguientes medidas:

**1) Para la amenaza de posible generador de números pseudoaleatorios inseguro se recomienda lo siguiente:**

- Pruebas y verificaciones de aleatoriedad: Realizando estas pruebas para verificar la calidad de los números generados, las pruebas de aleatoriedad pueden ayudar a identificar patrones o sesgos en los números generados.
- Implementación de ciclos de renovación de claves o tokens: consideramos que la rotación periódica de claves o tokens generados por el PRNG para limitar el impacto de un eventual descifrado o compromiso.
- Usar PRNG criptográficamente seguro: generadores de números pseudoaleatorios específicamente diseñados para aplicaciones de seguridad. Algunos lenguajes de programación ofrecen bibliotecas o funciones para PRNG seguros, como `random_bytes()` en PHP.

**2) Para la amenaza de uso de Algoritmo hash débil (usuario) recomendamos lo siguiente:**

- Actualización y migración: se debe planificar la actualización y migración a algoritmos más seguros lo antes posible. Actualiza las contraseñas almacenadas o la información de identificación utilizando los nuevos algoritmos.
- Usar iteraciones de hash (key stretching): Recomendamos realizar múltiples iteraciones del algoritmo de hash para aumentar la complejidad computacional del cálculo de contraseñas hash. Esto hace que los ataques de fuerza bruta sean más costosos en tiempo y recursos.
- Limitar la longitud y complejidad del nombre de usuario: se recomienda establecer límites razonables para la longitud del nombre de usuario y fomenta el uso de nombres de usuario que no revelen información sensible o que sean fáciles de adivinar.



3) Para la amenaza de uso de Algoritmo hash débil (contraseña) recomendamos lo siguiente:

- Actualización y migración: se debe planificar la actualización y migración a algoritmos más seguros lo antes posible. Actualiza las contraseñas almacenadas o la información de identificación utilizando los nuevos algoritmos.
- Usar algoritmos de hash robustos: Recomendamos usar algoritmos de hash robustos y adaptados para almacenar contraseñas, como bcrypt, Argon2, PBKDF2 o scrypt. Estos algoritmos están diseñados específicamente para proteger contraseñas.
- Recomendamos agregar salt a las contraseñas: Incluir una sal de aleatoriedad única para cada contraseña antes de aplicar el algoritmo de hash, ayudara a hacer más costoso el ataque de fuerza bruta y previene la creación de tablas de arco iris.

4) La amenaza de IP insegura recomendamos:

- Configurar firewall y filtrado de tráfico: Implementar reglas de firewall para filtrar el tráfico no deseado o malicioso hacia esa IP. Esto puede incluir restricciones de puertos, direcciones IP específicas o patrones de tráfico conocidos por ser riesgosos.
- Controlar el acceso y autenticación: Debe utilizar métodos sólidos de control de acceso. Esto podría incluir autenticación de dos factores o el uso de claves de acceso seguras.
- Encriptación de datos: Recomendamos asegurarse de que la comunicación esté encriptada mediante protocolos seguros como HTTPS.

5) El código de salida contiene expresiones esto se considera una mala práctica y recomendamos lo siguiente:

- Sanitizar los datos: Antes de mostrar datos provenientes de fuentes no confiables (como entrada de usuario o datos de la base de datos) en la salida, asegúrate de que estos datos estén sanitizados y limpios. Esto incluye eliminar caracteres especiales o escaparlos adecuadamente para evitar la ejecución de código malicioso.
- Utilizar funciones de escape apropiadas: Emplear las funciones específicas de escape proporcionadas por el lenguaje que están utilizando.
- Usar plantillas o motores de vistas: Considerar el uso de sistemas de plantillas o motores de vistas que automáticamente escapen o saniticen los datos antes de mostrarlos en el navegador. Por ejemplo, Twig para PHP.

6) La amenaza de no definir una constante recomendamos lo siguiente:

- Aplicar el principio de menor privilegio: recomendamos proporcionar solo la información necesaria en la salida. No expongas más información de la necesaria.
- Reevaluar constantemente su utilidad: A medida que evoluciona el proyecto, revisa regularmente las constantes existentes para asegurarte de que aún son necesarias. Elimina aquellas que ya no se utilizan o que han quedado obsoletas.
- Seguridad de gestión de claves: Considere usar la rotación regular de claves, o el almacenamiento seguro de estas claves (por ejemplo, usando servicios de administración de secretos).

7) Para la amenaza del método query (), recomendamos lo siguiente:

- Sanitizar los datos: Antes de mostrar datos provenientes de fuentes no confiables (como entrada de usuario o datos de la base de datos) en la salida, asegúrate de que estos datos estén sanitizados y limpios. Esto incluye eliminar caracteres especiales o escaparlos adecuadamente para evitar la ejecución de código malicioso.
- Validación y filtrado de datos: realizar una validación estricta de los datos de entrada para asegurarse de que se ajusten a ciertos criterios antes de ser

procesados o mostrados. Esto reduce la probabilidad de mostrar datos maliciosos.

- Auditorías de seguridad regulares: Realizar auditorías de seguridad periódicas para evaluar y mejorar continuamente la robustez de la generación de números aleatorios en tus sistemas.

## Conclusión

En el transcurso de este proyecto, hemos navegado por las complejidades de la ciberseguridad en el desarrollo web, descubriendo que va mucho más allá de la superficie aparente de HTML y CSS en el front-end. Nos hemos sumergido en el vasto universo del back-end, explorando la implementación y gestión de servicios críticos como MySQL (MariaDB), levantando servidores con sólidos fundamentos adquiridos a partir de materias como Gestión de Sistemas Operativos.

La interconexión de conocimientos provenientes de áreas como Redes nos permitió no solo levantar servidores, sino también configurar redes LAN de manera efectiva. La activación de servicios como MySQL (MariaDB) ha sido una tarea lograda gracias a la integración de conocimientos prácticos provenientes de diversos campos. Hemos aprendido a aplicar servicios como FTP, SSH y firewall para gestionar archivos y paquetes de manera segura en un entorno controlado, permitiéndonos realizar cambios de manera eficiente y segura.

La integración de conceptos de análisis de vulnerabilidades y verificación de códigos estáticos ha sido esencial. Hemos adquirido habilidades críticas para identificar y corregir posibles amenazas, así como para detectar malas prácticas en el código. Este enfoque proactivo en la seguridad no solo refuerza la robustez de nuestras aplicaciones, sino que también nos prepara para enfrentar desafíos del mundo real como profesionales de la ciberseguridad.

## Bibliografía

*Get Ubuntu Server / Download / Ubuntu.* (n.d.). Ubuntu.

<https://ubuntu.com/download/server/choose>

Documentation Group. (n.d.). *Welcome! - The Apache HTTP Server project.*

<https://httpd.apache.org/>

Documentation Group. (n.d.). *Welcome! - The Apache HTTP Server project.*

<https://httpd.apache.org/>

MariaDB.org. (2019, November 13). *MariaDB Foundation - MariaDB.org.*

<https://mariadb.org/>

Drake, M., & Boucheron, B. (2020, June 11). *Cómo instalar MariaDB en Ubuntu 20.04.*

DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-ubuntu-20-04-es>

Drake, M., & Boucheron, B. (2020, June 11). *Cómo instalar MariaDB en Ubuntu 20.04.*

DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-ubuntu-20-04-es>

*Security Tips - Apache HTTP Server Version 2.4.* (n.d.).

[https://httpd.apache.org/docs/2.4/misc/security\\_tips.html](https://httpd.apache.org/docs/2.4/misc/security_tips.html)

*Security Tips - Apache HTTP Server Version 2.4.* (n.d.).

[https://httpd.apache.org/docs/2.4/misc/security\\_tips.html](https://httpd.apache.org/docs/2.4/misc/security_tips.html)

Heidi, E. (2020, April 27). *Cómo instalar la pila Linux, Apache, MariaDB y PHP (LAMP) en CentOS 8.* DigitalOcean.

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mariadb-php-lamp-stack-on-centos-8-es>

B, G., & B, G. (2023, September 26). *¿Qué es Apache? Descripción completa.* Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-apache/>

Manuel Cabrera Caballero. (2023, February 19). *Como configurar ip fija en Ubuntu Server* [Video]. YouTube. <https://www.youtube.com/watch?v=4r2fdTCDzlQ>

Content Studio. (2022, June 18). *¿Qué es MariaDB?*

<https://www.purestorage.com/es/knowledge/what-is-mariadb.html>

Christian Torrico. (2019, June 30). *Hardening APACHE* [Video]. YouTube.

<https://www.youtube.com/watch?v=PxoIBH1pZMQ>

Secure Your Digital Life. (2020, April 6). *Harden a MariaDB database* [Video]. YouTube.

<https://www.youtube.com/watch?v=zkrUUs1jTDU>

TP-Link. (n.d.). *Sistema de Gestión de Red TP-LINK.* <https://www.tp-link.com/es/business-networking/accessory/tpnms/>

Comunicaciones Reunidas, SL. (n.d.). *Documentación TP-Link*.

<https://www.crsi.es/es/content/58-documentacion-tp-link>

*SonarCloud Documentation*. (n.d.). Sonar Docs. <https://docs.sonarsource.com/sonarcloud/>

support@codacy.com (Codacy Support). (n.d.). *Codacy docs*. <https://docs.codacy.com/>