Sistemas Operativos III

TechBoss

| Rol | Apellido | Nombre | C.I. | E-mail | Tel/Cel |
| --- | --- | --- | --- | --- | --- |
| Coordinador | Veneri | Martin | 5.559.252-7 | martinchico27@gmail.com | 095278843 |
| Sub-coordinador | Bobadilla | Julieta | 5.435.903–9 | julietabobadilla123@gmail.com | 093366757 |
| Integrante 1 | Fernandez | Victoria | 5.565.388-8 | Victoriadollanartr777@gmail.com | 093354000 |
| Integrante 2 | Mendez | Andrew | 5.450.638-7 | Andrewnicolas145@gmail.com | 097088873 |

**Docente: Martínez, Santiago**





ÍNDICE

[**PRIMERA ENTREGA 2**](#_heading=h.1fob9te)

[Relevamiento y justificación del Sistema Operativo a utilizar tanto en las terminales 2](#_heading=h.3znysh7)

[Manual de instalación del Sistema Operativo en el servidor 4](#_heading=h.2et92p0)

[INSTALACIÓN DE DOCKER 9](#_heading=h.2s8eyo1)

[Instalación de un servidor LAMP en la máquina virtual 10](#_heading=h.17dp8vu)

[**SEGUNDA ENTREGA 10**](#_heading=h.26in1rg)

[Estudio de los diferentes roles de los usuarios del servidor 10](#_heading=h.lnxbz9)

[**Relevamiento y justificación del Sistema Operativo a utilizar tanto en las terminales de los usuarios como en el servidor 11**](#_heading=h.35nkun2)

[Instalación de un servidor LAMP en la máquina virtual 12](#_heading=h.qhna9tojzc2t)

[Usuarios necesarios en el sistema operativo creados 12](#_heading=h.uwauonofxadc)

[de acuerdo al estudio de roles 12](#_heading=h.a1nm0s6eanhr)

[**TERCERA ENTREGA 13**](#_heading=h.2bn6wsx)

[**Roles de usuarios y contraseñas 13**](#_heading=h.qsh70q)

[**Scripts creados, su comandos y funciones 14**](#_heading=h.3as4poj)

[SCRIPT DE DOCKER 14](#_heading=h.1pxezwc)

[SCRIPT DE FILTRADO DE IPs 15](#_heading=h.49x2ik5)

[SCRIPT DE FIREWALL 16](#_heading=h.2p2csry)

[SCRIPT DE GRUPOS 18](#_heading=h.147n2zr)

[SCRIPT DE MENÚ DE ADMINISTRACIÓN 19](#_heading=h.3o7alnk)

[SCRIPT DE GESTIÓN SERVICIOS 20](#_heading=h.ihv636)

[SCRIPT DE RED 21](#_heading=h.1hmsyys)

[SCRIPT DE RESPALDO 22](#_heading=h.2grqrue)

[SCRIPT DE SSH 23](#_heading=h.3fwokq0)

[SCRIPT DE USUARIOS 24](#_heading=h.4f1mdlm)

[**Contenedores Docker 25**](#_heading=h.2u6wntf)

[**Instalación de un servidor LAMP en la máquina virtual 25**](#_heading=h.19c6y18)

[Respaldo Remoto 27](#_heading=h.3tbugp1)

[Investigación y documentación del mecanismo Crontab 27](#_heading=h.nmf14n)

[Configuración de Firewall 28](#_heading=h.mjnvlz3t6r5p)

[Archivos crontab con rutinas de backup y sus correspondientes scripts para el administrador 29](#_heading=h.37m2jsg)

[**Bibliografía 29**](#_heading=h.1mrcu09)

# PRIMERA ENTREGA

## Relevamiento y justificación del Sistema Operativo a utilizar tanto en las terminales

¿POR QUÉ FEDORA SERVER 38?

Fedora Server 38 es la última versión de la distribución Linux Fedora orientada a

servidores. Utilizar Fedora Server 38 puede tener varias ventajas dependiendo del uso

que le vayas a dar. En nuestro caso las características que buscamos estaban ya

implementadas en esta nueva versión de Fedora, las cuales son las siguientes:

* Docker: soporte nativo para Docker, lo que facilita la creación y gestión de

contenedores.

* Herramientas y servicios actualizados: versiones actualizadas de las

principales herramientas y servicios utilizados en entornos de servidor, como el

kernel de Linux, el gestor de paquetes DNF, el servidor web Apache, el servidor

de bases de datos MySQL/MariaDB y muchas otras.

* Soporte para arquitecturas múltiples: ofrece soporte para diferentes

arquitecturas, lo que te permite ejecutarlo en una variedad de sistemas y

dispositivos, como servidores x86\_64, ARM y POWER.

¿QUÉ ES DOCKER?

Docker es una plataforma de software que permite crear, probar e implementar

aplicaciones rápidamente. Docker empaqueta el software en unidades estandarizadas

llamadas contenedores que tienen todo lo que el software necesita para ejecutarse,

incluidas bibliotecas, herramientas del sistema, código y tiempo de ejecución. Con

Docker, se puede implementar y escalar rápidamente aplicaciones en cualquier entorno

y saber que su código se ejecutará.

¿POR QUÉ DOCKER?

Docker ofrece una forma eficiente y portátil de desarrollar, desplegar y ejecutar

aplicaciones, con ventajas como la portabilidad, el aislamiento, la eficiencia en el uso de

recursos, la reproducibilidad y la facilidad de despliegue. Estas ventajas hacen que

Docker sea una opción popular en el mundo del desarrollo de software.

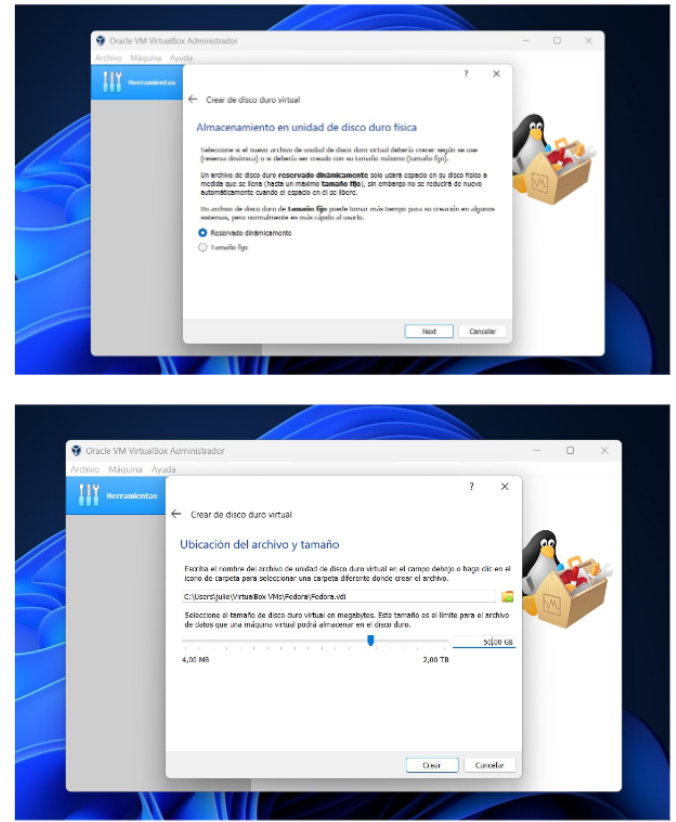
¿CÓMO CREAR DOCKER?

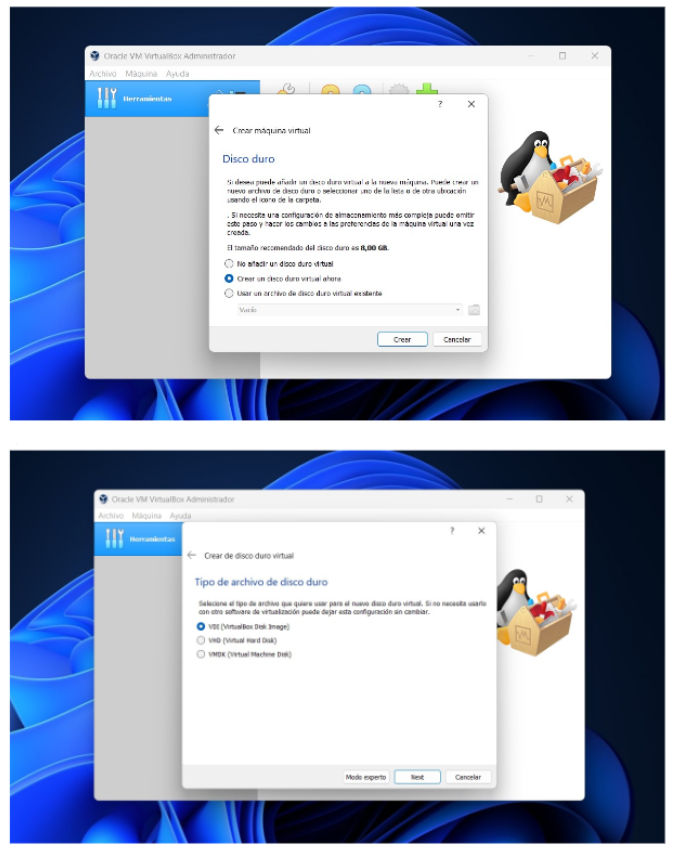
A continuación, mostraré algunos comandos de Docker para crear y gestionar contenedores en

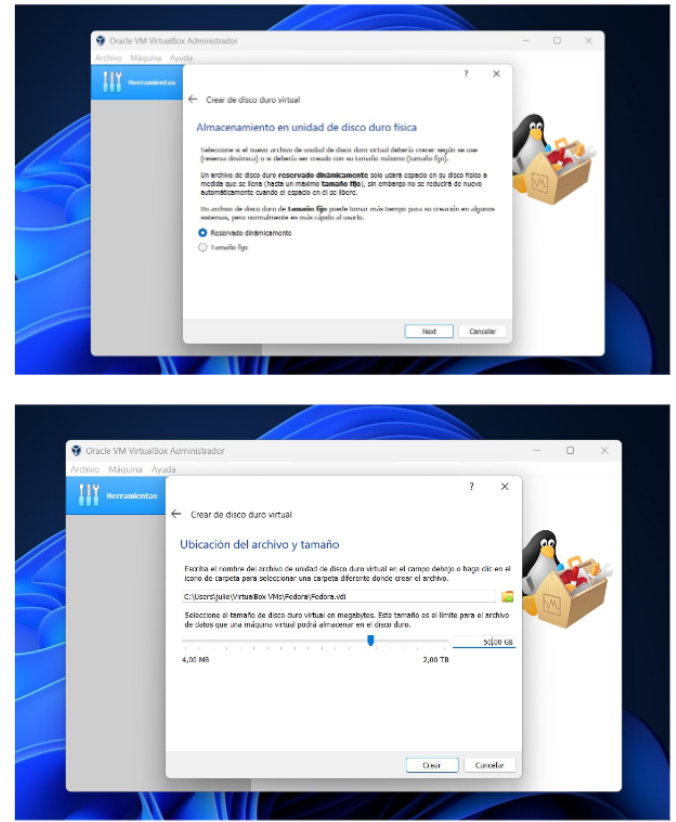
Linux:

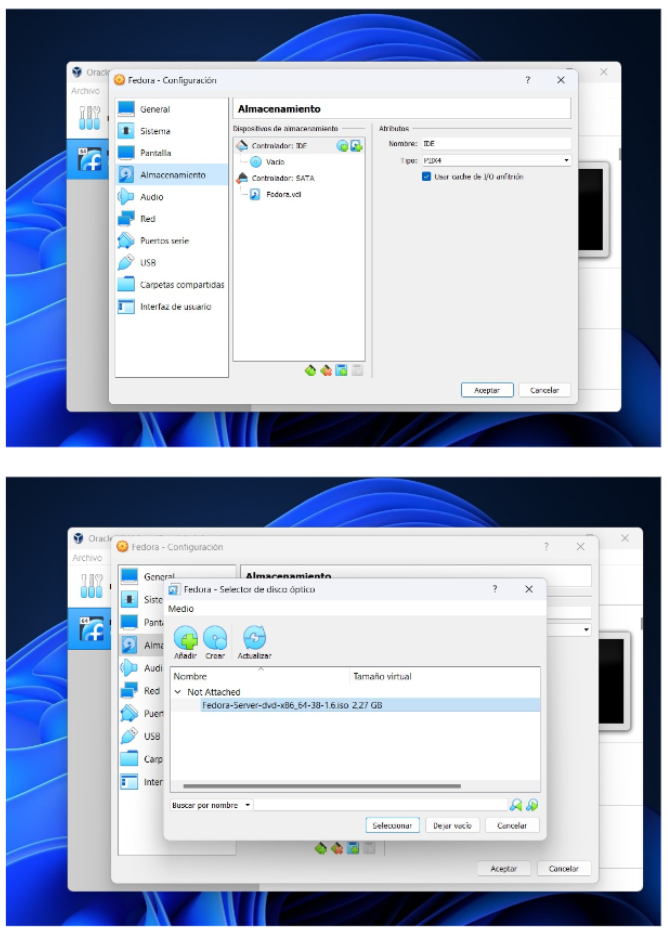
* systemctl status/stop/start/enable docker
* docker run
* docker pull
* docker ps
* docker system prune

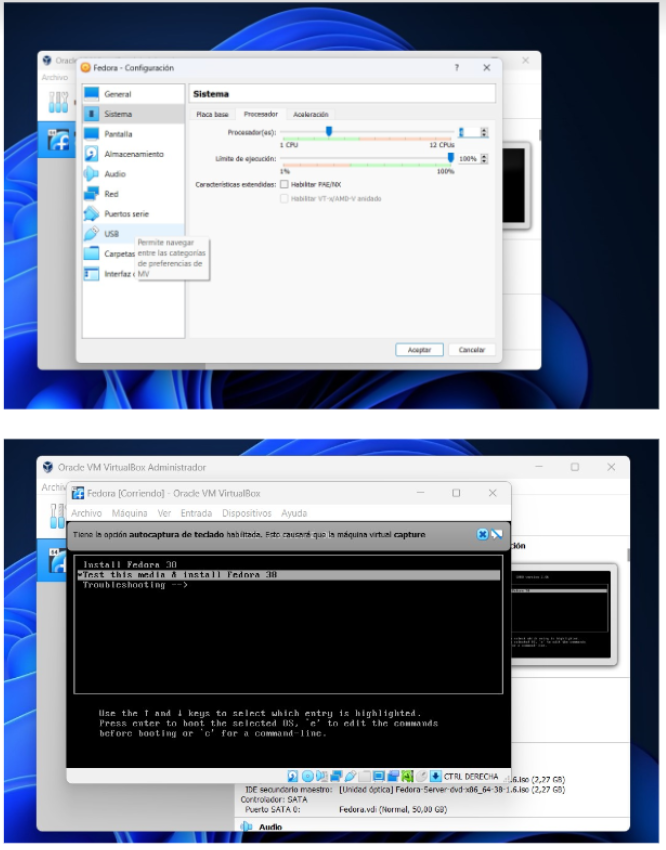
## Manual de instalación del Sistema Operativo en el servidor











## 

## 

## 

## 

## INSTALACIÓN DE DOCKER

1. Se recomienda remover versiones anteriores antes de instalar un nuevo Docker:

$ sudo dnf remove docker docker-client docker-client-latest docker-common docker-

latest docker-latest-logrotate docker-logrotate docker-selinux docker-engine-selinux

docker-engine

2. Instalamos plugins para el administrador de paquetes dnf y lo configuramos:

$ sudo dnf -y install dnf-plugins-core

$ sudo dnf config-manager --add-repo

<https://download.docker.com/linux/fedora/docker-ce.repo>

3. Instalamos paquetes (este comando instala Docker, pero no lo inicia):

$ sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-

compose-plugin

4. Verificamos el estado del servicio:

$ sudo systemctl status docker

Si está activo aparecerá en verde "active(running)".

Con el comando "docker images" se listan las imágenes que podemos utilizar para

nuestros contenedores. En la lista debemos visualizar la imagen "hello-world” la cual

tras ejecutarla con el comando "docker run hello-world" nos hará saber que todo está

marchando bien con nuestra instalación de Docker.

## Instalación de un servidor LAMP en la máquina virtual

Para la instalación del servidor LAMP tendremos que hacer lo siguiente en la terminal:

1. Ir a Docker Hub y buscar las imágenes que queremos instalar en Docker, o sea

una imagen de Apache, otra de MySql y otra de PHP.

2. Luego en la terminal ejecutamos los comandos correspondientes para instalar la

imagen. Para PHP-Apache usaremos el comando:

$ sudo docker pull php:8.7.1-apache-bullseye

Instala PHP con la etiqueta correspondiente (8-7.1-apache-bullseye).

Para MySql usaremos el mismo comando:

$ sudo docker pull mysql

Instala la última versión por defecto porque no agregamos etiqueta.

# 

# SEGUNDA ENTREGA

## Estudio de los diferentes roles de los usuarios del servidor

**ADMINISTRADORA**

Será capaz de administrar sin restricción alguna el sistema.

**SUPERVISORES**

Estarán pendientes al bienestar del sistema como por ejemplo el

almacenamiento, sesiones y su información, conexiones y más sin embargo

deben de recurrir al administrador debido a que no pueden realizar ningún cambio.

**MANTENIMIENTO DEL SISTEMA**

Estará constantemente haciendo actualizaciones para que el sistema no quede

desactualizado, mejorando el rendimiento del sistema y agregando nuevas

funcionalidades de ser necesario.

# Relevamiento y justificación del Sistema Operativo a utilizar tanto en las terminales de los usuarios como en el servidor

**¿POR QUÉ FEDORA SERVER 38?**

Fedora Server 38 es la última versión de la distribución Linux Fedora orientada a

servidores. Utilizar Fedora Server 38 puede tener varias ventajas dependiendo del uso

que le vayas a dar. En nuestro caso las características que buscamos estaban ya

implementadas en esta nueva versión de Fedora, las cuales son las siguientes:

* Docker: soporte nativo para Docker, lo que facilita la creación y gestión de

contenedores.

* Herramientas y servicios actualizados: versiones actualizadas de las

principales herramientas y servicios utilizados en entornos de servidor, como el

kernel de Linux, el gestor de paquetes DNF, el servidor web Apache, el servidor

de bases de datos MySQL/MariaDB y muchas otras.

* Soporte para arquitecturas múltiples: ofrece soporte para diferentes

arquitecturas, lo que te permite ejecutarlo en una variedad de sistemas y

dispositivos, como servidores x86\_64, ARM y POWER.

**¿QUÉ ES DOCKER?**

Docker es una plataforma de software que permite crear, probar e implementar

aplicaciones rápidamente. Docker empaqueta el software en unidades estandarizadas

llamadas contenedores que tienen todo lo que el software necesita para ejecutarse,

incluidas bibliotecas, herramientas del sistema, código y tiempo de ejecución. Con

Docker, se puede implementar y escalar rápidamente aplicaciones en cualquier entorno

y saber que su código se ejecutará.

**¿POR QUÉ DOCKER?**

Docker ofrece una forma eficiente y portátil de desarrollar, desplegar y ejecutar

aplicaciones, con ventajas como la portabilidad, el aislamiento, la eficiencia en el uso de

recursos, la reproducibilidad y la facilidad de despliegue. Estas ventajas hacen que

Docker sea una opción popular en el mundo del desarrollo de software.

**¿CÓMO CREAR DOCKER?**

A continuación, mostraré comandos de Docker para crear y gestionar contenedores en

Linux:

• systemctl status/stop/start/enable docker

• docker run

• docker pull

• docker ps

• docker system prune

## Instalación de un servidor LAMP en la máquina virtual

Para la instalación del servidor LAMP tendremos que hacer lo siguiente en la terminal:

1. Ir a Docker Hub y buscar las imágenes que queremos instalar en Docker, o sea

una imagen de Apache, otra de MySql y otra de PHP.

1. Luego en la terminal ejecutamos los comandos correspondientes para instalar la

imagen.

* Para Apache usaremos el comando:

$ sudo docker pull httpd

Este comando “docker pull” instalara la imagen de Apache, la última versión por

defecto ya que no le pusimos una etiqueta.

* Para PHP usaremos el mismo comando:

$ sudo docker pull php:8.7.1-apache-bullseye

Instala PHP con la etiqueta correspondiente (8-7.1-apache-bullseye).

* Para MySql usaremos el mismo comando:

$ sudo docker pull mysql

Instala la última versión por defecto porque no agregamos etiqueta.

## Usuarios necesarios en el sistema operativo creados

## de acuerdo al estudio de roles

1. Técnico del sistema

2. Administrador

3. Usuario

# TERCERA ENTREGA

## Roles de usuarios y contraseñas

| **USUARIOS** | **PASSWORD** |
| --- | --- |
| Tecnico | tecnicousuario |
| Administrador | administrador |
| Usuario | usuario |

TÉCNICO

El técnico será el encargado de gestionar y mantener el control. Tendrá varias tareas como por ejemplo:

* Instalación y Configuración: Instalar, configurar y mantener el sistema operativo en los servidores
* Resolución de Problemas: Diagnosticar y solucionar problemas relacionados con el sistema operativo, como errores, fallas en el rendimiento del sistema
* Administración de Usuarios: Gestionar cuentas de usuario, permisos y accesos para garantizar la seguridad y la eficiencia en el uso del sistema
* Respaldos y Recuperación: Configurar y gestionar procesos de respaldo para proteger los datos y establecer procedimientos de recuperación en caso de pérdida de información
* Automatización de Tareas: Utilizar scripts y herramientas de automatización para realizar tareas repetitivas y mejorar la eficiencia del sistema

ADMINISTRADOR

Estarán pendientes al bienestar del sistema como por ejemplo el almacenamiento, sesiones y su información, conexiones y más sin embargo deben de recurrir al técnico debido a que no pueden realizar ningún cambio.

USUARIO

El usuario es un individuo el cual será capaz de interactuar con el dispositivo, siendo las siguientes las acciones que podrá realizar:

* Inicio de sesión: Cuando un usuario accede al sistema, generalmente debe iniciar sesión proporcionando un nombre de usuario y una contraseña. Esto autentica al usuario y le otorga acceso al dispositivo
* Interfaz Gráfica: El usuario puede interactuar con el dispositivo a través de la interfaz gráfica de usuario (GUI). En una GUI, los usuarios utilizan mouse y teclado para interactuar con ventanas, iconos y menús
* Ejecución de Aplicaciones y Programas: Los usuarios pueden ejecutar aplicaciones y programas
* Cierre de Sesión y Apagado: Al finalizar su sesión, los usuarios cierran sesión y apagan el dispositivo según la situación

## Scripts creados, su comandos y funciones

docker.sh

filtradoip.sh

firewall.sh

grupos.sh

menu\_de\_gestion.sh

menu\_servicios.sh

red.sh

respaldo.sh

ssh.sh

usuarios.sh

### SCRIPT DE DOCKER

#/bin/bash

while true;

do

clear

echo “Menú de Gestión de Contenedores de Docker”

echo “1. Listar contenedores en ejecución”

echo “2. Crear un nuevo contenedor”

echo “3. Iniciar un contenedor”

echo “4. Detener un contenedor”

echo “5. Eliminar un contenedor”

echo “6. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. docker ps

;;

1. read -p “Ingrese el nombre del nuevo contenedor: “ nombre

read -p “Ingrese la imagen a utilizar: “ imagen

docker run -d – name $nombre $imagen

;;

1. read -p”Ingrese el nombre del contenedor que desea iniciar: “ nombre\_ini

docker start $nombre\_ini

;;

1. read -p”Ingrese el nombre del contenedor que desea detener: “ nombre\_det

docker stop $nombre\_det

;;

1. read -p “Ingrese el nombre del contenedor que desea eliminar: “ nombre\_elim

docker rm $nombre\_elim

;;

1. echo “Saliendo del menú”

exit 0

;;

\*) echo “Opción incorrecta. Seleccionar una opción válida.”

;;

esac

read -p “Presionar ENTER para continuar.”

done

***Este script proporciona una interfaz simple de un menú que permite gestionar los contenedores de Docker, ya sea crearlos, iniciarlos, eliminarlos. También tiene una opción de salir al menú principal y si ingresa una opción no válida se le pide ingresar una que sea válida.***

### SCRIPT DE FILTRADO DE IPs

#/bin/bash

while true;

do

clear

echo “Menú de Filtrado de IPs”

echo “1. Habilitar IP”

echo “2. Deshabilitar IP”

echo “3. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. read -p “Ingrese la dirección IP que desea habilitar: “ ip\_si

sudo firewall-cmd –permanent –add-source=$ip\_si

sudo firewall-cmd –reload

;;

1. read -p “Ingrese la dirección IP que desea habilitar: “ ip\_no

sudo firewall-cmd –permanent –add-rich-rule= rule family=’ipv4’ source address= $ip\_no drop

sudo firewall-cmd –reload

;;

1. read -p “Saliendo del menú”

exit 0

;;

\*) echo “Opción incorrecta. Seleccionar una opción válida.”

;;

esac

read -p “Presionar ENTER para continuar.”

done

***Este script proporciona una interfaz básica de un menú para gestionar IPs. Este menú te da dos opciones simples; habilitar y deshabilitar IPs. También cuenta con una opción para salir al menú y en caso de que se ingrese una opción inválida te pedirá que ingreses una opción válida.***

### SCRIPT DE FIREWALL

#/bin/bash

while true;

do

clear

echo “Menú de Gestión de Firewall”

echo “1. Mostrar reglas de Firewall”

echo “2. Agregar nueva regla al Firewall”

echo “3. Eliminar regla de Firewall”

echo “4. Habilitar el Firewall”

echo “5. Detener el Firewall”

echo “6. Estado del Firewall”

echo “7. Filtrado de IPs ”

echo “8. Salir”

read -p “Seleccione una opción: ” op

case $op in

1. echo “Reglas actuales del Firewall”

sudo iptables -L -n -v

;;

1. echo “Agregar regla al Firewall”

read -p “Ingrese nueva regla: “ nueva\_regla

sudo iptables $nueva\_regla

echo “Regla agregada exitosamente al Firewall!”

;;

1. echo “Eliminar regla del Firewall”

read -p “Ingrese la regla que desea eliminar: “ eliminar\_regla

sudo iptables -D $eliminar\_regla

echo “Regla eliminada exitosamente del Firewall!”

;;

1. sudo systemctl start firewalld

;;

1. sudo systemctl stop firewalld

;;

1. sudo systemctl status firewalld

;;

1. echo “Ejecutando script de Filtrado de IPs”

./filtradoip.sh

;;

1. echo “Saliendo del menú de Firewall”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Este script nos muestra la interfaz de un menú de gestión de Firewall. En este menú se muestran varias opciones: mostrar las reglas del Firewall, agregar o eliminar alguna regla de Firewall, habilitar o deshabilitar el Firewall y mostrar el estado de este. Además tiene una opción que te lleva a otro script de Filtrado de IPs y otra que te permite salir al menú. Como los demás scripts, al ingresar una opción inválida te pedirá que ingreses una oposición válida.***

### SCRIPT DE GRUPOS

#/bin/bash

while true;

do

clear

echo “Menú de Gestión de Grupos”

echo “1. Crear grupo”

echo “2. Eliminar grupo”

echo “3. Listar grupos”

echo “4. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. read -p “Ingrese el nombre del grupo: ” gruponuev

sudo groupadd $gruponuev

echo “El grupo “ $gruponuev “ se ha creado!”

;;

1. read -p “Ingrese el nombre del grupo que desea eliminar: “ delgrupo

sudo groupdel $delgrupo

echo “El grupo “ $delgrupo “ se eliminó correctamente!”

;;

1. echo “Listado de grupos: “

cut -d: -f1 /etc/group

;;

1. recho “Saliendo del Menú de Gestión de Grupos”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Este script nos muestra un menú interactivo el cual te permite gestionar grupos. Puedes crearlos, eliminarlos o listarlos. Además de una opción de salir del menú y, además, si ingresas una opción inválida te pedirá que lo intentes de nuevo. Es, ciertamente, un menú básico de gestión pero funcional.***

### SCRIPT DE MENÚ DE ADMINISTRACIÓN

#/bin/bash

while true;

do

clear

echo “Menú de Administración”

echo “1.Gestionar Usuarios”

echo “2. Gestionar grupos”

echo “3. Gestionar Red”

echo “4. Gestionar Firewall”

echo “5. Gestionar Servicios”

echo “6. Realizar respaldo local”

echo “7. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. echo “Ejecutando script de gestión de usuarios…”

./usuarios.sh

;;

1. echo “Ejecutando script de gestión de grupos…”

./grupos.sh

;;

1. echo “Ejecutando script de gestión de red…”

./red.sh

;;

1. echo “Ejecutando script de gestión de firewall…”

./firewall.sh

;;

1. echo “Ejecutando script de gestión de servicios…”

./menu\_servicios.sh

;;

1. echo “Ejecutando script de respaldo local…”

./respaldo.sh

;;

1. echo “Saliendo del menú de Administración”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Este es otro script con un menú básico(el cual es el menú inicial), pero este contiene otros scripts que se ejecutarán según la opción elegida por el usuario.***

### 

### SCRIPT DE GESTIÓN SERVICIOS

#/bin/bash

while true;

do

clear

echo “Menú de Gestión de Servicios”

echo “1. Gestionar Docker”

echo “2. Gestionar SSH”

echo “3. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. echo “Ejecutando menú de Docker…”

./docker.sh

;;

1. echo “Ejecutando menú de SSH…”

./ssh.sh

;;

1. echo “Saliendo del menú de Gestión de Servicios”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Este script contiene un menú el cual, según la opción elegida por el usuario, ejecuta un script ya sea del servicio de Docker o el servicio de SSH.***

### 

### SCRIPT DE RED

#/bin/bash

while true;

do

clear

echo “Menú de Configuración de Red”

echo “1. Mostrar información de red”

echo “2. Configurar dirección IP”

echo “3. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. echo “Información de Red”

ifconfig

;;

1. read -p “Ingrese el nombre de la interfaz de red: “ nom\_interfaz

read -p “Ingrese la nueva dirección IP: “ new\_ip

sudo ifconfig $nom\_interfaz $new\_ip

echo “La dirección IP “ $nom\_interfaz “ ha sido configurada exitosamente!”

;;

1. echo “Saliendo del menú de configuración de red”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Script para gestionar la Red el cual nos puede mostrar información de la red o configurar IPs. Este es uno de los scripts un poco más complejos que los anteriores porque más allá de que sea otro menú de gestión, este conlleva muchos más comandos. No deja de ser un menú básico pero con comandos más difíciles.***

### 

### SCRIPT DE RESPALDO

#/bin/bash

dir\_respaldo = “/home/tecnico/respaldo”

while true;

do

clear

echo “Menú de Gestión de Respaldo Local”

echo “1. Crear copia de seguridad”

echo “2. Listar archivos de respaldo”

echo “3. Salir”

read -p “Ingrese una opción: ” op

case $op in

# Le pide al usuario que ingrese el nombre del archivo o directorio que quiere respaldar, y se almacena en la variable **arch\_dir**

1. *read -p “Ingrese el nombre del archivo o directorio a respaldar: “ arch\_dir*

# Con el comando **date** se obtiene la fecha y hora en un formato y el formato es **“%Y%m%d%H%M%S”**.Todo eso se almacena en la variable **fecha**

*fecha = $(date + “%Y%m%d%H%M%S”)*

# Se crea la variable **destino** que va a contener la ruta y el nombre del archivo/directorio que queremos respaldar. La variable **respaldo** representa la ubicación en donde se almacenan los respaldos, y con la variable **fecha** se crea el nombre del archivo junto con la fecha y hora actual y la extensión **.tar.gz**

*destino = $dir\_respaldo/respaldo\_$fecha.tar.gz*

# Aquí se crea un archivo de respaldo comprimido con la **extensión tar** y los parámetros que se muestran, siendo **f** el que especifica el nombre del archivo de respaldo y la variable **arch\_dir** nos dice la ubicación del archivo a respaldar

*tar -czvf $destino $arch\_dir*

*echo “Copia de seguridad creada en: ” $destino*

;;

1. a echo “Archivos de respaldo en “ $dir\_respaldo ”:”

-ls -lh $dir\_respaldo

;;

1. echo “Saliendo del menú de respaldo local…”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Este es uno de los scripts más importantes el cual tiene un menú para respaldar ya sea archivos o directorios. Uno de los más complejos hasta ahora.***

### 

### SCRIPT DE SSH

#/bin/bash

while true;

do

clear

echo “Menú de Gestión de SSH”

echo “1. Iniciar el servicio SSH”

echo “2. Detener el servicio SSH”

echo “3. Reiniciar el servicio SSH”

echo “4. Ver el estado del servicio del SSH”

echo “5. Editar la configuración de SSH”

echo “6. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. sudo systemctl start sshd

;;

1. sudo systemctl stop sshd

;;

1. sudo systemctl restart sshd

;;

1. sudo systemctl status sshd

;;

1. sudo nano /etc/ssh/sshd\_config

sudo systemctl reload sshd

;;

1. echo “Saliendo del menú de respaldo local…”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Un script de un menú de gestión del servicio SSH básico, con opciones como iniciar, detener, reiniciar o ver el estado del servicio. Simple pero funcional.***

### 

### SCRIPT DE USUARIOS

#/bin/bash

while true;

do

clear

echo “Menú de Gestión deUsuarios”

echo “1. Agregar usuario ”

echo “2. Eliminar usuario”

echo “3. Listar usuarios”

echo “4. Salir”

read -p “Ingrese una opción: ” op

case $op in

1. read -p “Ingrese un nuevo usuario: “ newuser

sudo adduser $newuser

echo “Usuario “ $newuser “ agregado!”

;;

1. a read -p “Ingrese el nombre del usuario que desea eliminar: “ deluxe

sudo deluser $delusu

echo “Usuario ” $delusu “ se ha eliminado!”

;;

1. echo “Listado de usuarios”

cut -d: f1 /etc/passwd

;;

1. echo “Saliendo del menú de respaldo local…”

exit 0

;;

\*) echo “Opción incorrecta. Ingrese una opción válida.”

esac

read -p “Presione ENTER para continuar.”

done

***Este script es un menú de gestión de usuarios. Este menú ofrece las opciones de crear, eliminar o listar usuarios. Contiene también una opción de salir del menú y en caso de ingresar una opción inválida te dice que intentes de nuevo.***

## Contenedores Docker

Para los contenedores de Docker instalamos las imágenes de php:7.0-apache y de mysql:8.0.

## Instalación de un servidor LAMP en la máquina virtual

Para la instalación del servidor LAMP tendremos que hacer lo siguiente en la terminal:

1. Ir a Docker Hub y buscar las imágenes que queremos instalar en Docker, o sea una imagen de Apache, otra de MySql y otra de PHP.
2. Luego en la terminal ejecutamos los comandos correspondientes para instalar la imagen.

Para PHP usaremos el mismo comando:

$ sudo docker pull php:7.0-apache

Instala PHP con la etiqueta correspondiente (versión 7.0 con Apache).

Para MySQL usaremos el mismo comando:

$ sudo docker pull mysql:8.0

Instala la versión 8.0 de MySQL

Luego de instalar las imágenes vamos a correr las imágenes. Primero vamos a abrir el MySQL con el siguiente comando:

$ docker run -p 3306:3306 –name base\_de\_datos -v /home/tecnico/Escritorio/Datos/database:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=tecnico -d mysql:8.0

* docker run:
* 3306:3306: se refiere al puerto de MySQL. El primer 3306 representa al puerto donde se va a abrir en nuestra PC y el otro 3306 representa el puerto donde está el contenedor
* –name: le asigna un nombre a la imagen
* -v: crea un volumen donde se van a almacenar los datos de la base de datos en una carpeta. Para esto tenemos que crear una carpeta donde se van a almacenar a la cual le pusimos “Datos” y dentro de esa carpeta creamos una subcarpeta llamada “database”. Tenemos que especificar la ruta donde se van a almacenar los datos y la ruta del contenedor de MySQL
* -e MYSQL\_ROOT\_PASSWORD= : utilizamos este parámetro para asignar una contraseña, y la contraseña asignada es “tecnico”
* -d: parámetro utilizado para que corra en segundo plano

Ahora vamos a correr la imagen de PHP. Para eso utilizaremos el siguiente comando:

$ docker run -p 9090:80 -v /home/tecnico/Escritorio/Datos:/var/www/html –name servidor\_web -d –link base\_de\_datos php:7.0-apache

* docker run: iniciar un contenedor
* -p 9090:80: mapea los puertos entre el host y el contenedor
* -v: crea volúmenes
* –name: se le asigna un nombre al contenedor
* -d: ejecuta el contenedor en segundo plano
* –link: establece una conexión de red entre el contenedor “servidor\_web” y el contenedor “base\_de\_datos” que permite que se comuniquen entre sí

Luego vamos a crear un archivo llamado Dockerfile en donde vamos a personalizar varios parámetros:

FROM php:7.0-apache

RUN apt-get update && apt-get upgrade -y

RUN docker-php-ext-install mysqli

EXPOSE 80

Ahora levantaremos más de un contenedor con docker-compose.yml. Dentro de este archivo vamos a escribir cosas como:

-Versión de docker

-Construcción de contenedores mediante Dockerfile

-Volúmenes

-Redirección de puertos

Con el comando docker-compose up -d vamos a iniciar los contenedores, el -d hace que se ejecute en segundo plano.

## Respaldo Remoto

$ rsync -r /ORIGEN /DESTINO

## 

## Investigación y documentación del mecanismo Crontab

¿Qué es ‘Crontab’?

Crontab es una herramienta para automatizar tareas en sistemas Unix y Linux. Permite a los usuarios programar scripts, comandos o programas que deben ejecutarse de manera programada sin intervención manual. Es importante entender la sintaxis y la estructura del crontab para aprovechar al máximo esta funcionalidad y automatizar tareas de manera eficiente.

Estructura Básica de Crontab

El crontab es un archivo que contiene instrucciones cron. Cada usuario puede tener su propio archivo crontab.

Estructura:

* Minuto: se representa con un número(0-59)
* Hora: se representa con un número(0-23)
* Dia del mes: se representa con un número(1-31)
* Día de la semana: se representa con un número(0-6, siendo 0 el domingo) o abreviado en inglés(las tres primeras letras, ej. mon, sun, fri, etc)
* Comando: el comando a ejecutar

Uso básico:

Para editar el archivo crontab(también se puede utilizar los editores nano y vim):

$ crontab -e

Para listar el contenido de Crontab:

$ crontab -l

Para eliminar el archivo Crontab(del usuario):

$ crontab -r

## 

## Configuración de Firewall

En Fedora, el sistema de firewall utilizado por defecto es firewalld. Firewalld proporciona una interfaz dinámica y fácil de usar para administrar reglas de firewall. A continuación te mostraremos una guía de como usar firewalld y firewall cmd.

Instalación y estado de firewall:

* Firewall normalmente viene ya instalado, pero en caso de que no, puedes utilizar el siguiente comando:

$ sudo dnf install firewalld

* Hay que asegurarse de que el servicio esté iniciado y habilitado:

$ sudo systemctl start firewalld

$ sudo systemctl enable firewalld

Ademas debemos verificar el estado del servicio:

$ sudo systemctl status firewalld

Algunos usos:

* Ver zonas disponibles:

$ sudo firewall-cmd –get-zones

Firewall trabaja con “zonas” que se refiere a diferentes niveles de confianza a las interfaces de red. Algunas de estas zonas son “public”, “home”, etc.

* Ver configuración actual:

$ sudo firewall-cmd –list-all

* Agregar un puerto a una zona:

$ sudo firewall-cmd –zone=public –add-port=8080/tcp –permanent

* Eliminar un puerto de una zona:

$ sudo firewall-cmd –zone=public –remove-port=8080/tcp –permanent

* Asignarle a una interfaz una zona:

$ sudo firewall-cmd –zone=public –change-interface=eth0

## Archivos crontab con rutinas de backup y sus correspondientes scripts para el administrador

# Bibliografía

# 

* Firewall CentOS 7: Configurar, habilitar, deshabilitar y crear reglas <https://www.solvetic.com/tutoriales/article/3467-firewall-centos-7-configurar-habilitar-deshabilitar-crear-reglas/>
* LOGS- registro de eventos en Linux- comando JOURNALCTL <https://www.youtube.com/watch?v=WjWEqJwe29Q&t=160s>
* DOCKER COMPOSE- Servidor Apache PHP mysql Linux - LAMP <https://www.youtube.com/watch?v=cQ8CdSrv-1k&t=162s>
* RSYNC- RESPALDO REMOTO en Linux y sincronización de directorios. <https://www.youtube.com/watch?v=t18-C6f9y5Q>
* Utiliza Cron y Crontab para programar tareas en tu servidor <https://www.redeszone.net/tutoriales/servidores/cron-crontab-linux-programar-tareas/>