



S.I.G.A.E.

Albisoft
Sistemas Operativos III

Rol	Apellido	Nombre	Cédula Identidad	Email	Tel./Cel.
Coordinador	Benavides	Moises	6.285.082-9	mbenavides@scuolaitaliana.edu.uy	096087002
Sub-coordinador	Easton	Mateo	5.645.982-3	measton@scuolaitaliana.edu.uy	098610119
Integrante 1	Carsillo	Gianni	5.567.221-6	gcarsillo@scuolaitaliana.edu.uy	092871428
Integrante 2	Charlo	Lucas	5.773.871-7	lcharlo@scuolaitaliana.edu.uy	092643467
Integrante 3	Migueluez	Mateo	5.640.554-3	mmigueluez@scuolaitaliana.edu.uy	096128643

Docente: Farias, Gustavo

Fecha de entrega: 23/9/2024

Segunda entrega

SIM

Índice

1. Introducción al documento.....	4
1.1 Objetivos del proyecto.....	4
1.2 Alcance del proyecto.....	4
1.3 Metodología.....	4
2. Estudio de roles de usuarios.....	5
2.1 Administrador de Sistemas.....	5
2.2 Administrador de Bases de Datos.....	5
2.3 Administrador de Seguridad.....	5
2.4 Administrador de Redes.....	5
2.5 Administrador de Respaldos y Recuperación.....	6
2.6 Soporte Tecnico.....	6
3. Diagrama de Navegabilidad e Implementación de Shell Script para Administración de Usuarios y Grupos.....	6
3.1 Estructura del diagrama.....	6
3.2 Mapa conceptual.....	7
3.3 Shell Script para Administración de Usuarios y Grupos.....	8
3.4 Descripción de funciones del script.....	11
4. Relevamiento y Justificación del Sistema Operativo a Utilizar Tanto en las Terminales de los Usuarios como en el Servidor.....	13
4.1 Comparación de Sistemas Operativos.....	13
4.2 Justificación de Ubuntu Server.....	15
4.3 Implementación y Mantenimiento.....	16
5. Manual de Instalación del Sistema Operativo en el Servidor Seleccionado.....	16
5.1 Propósito del Manual.....	16
5.2 Descripción General de Ubuntu Server 22.04 LTS.....	16
5.3 Audiencia Objetivo.....	17
5.4 Requisitos previos.....	17
5.5 Preparativos.....	18
5.6 Inicialización.....	18
5.7 Conclusión de la instalación.....	24
6. Implementación de Máquina Virtual o Contenedor Conteniendo el Script de Administración.....	25
6.1 Preparativos.....	25
6.2 Credenciales de Acceso.....	25
6.3 Implementación del Script.....	25
6.4 Uso de la Máquina Virtual.....	26
7. Nuevos desarrollos (Segunda entrega).....	26
7.1 Usuarios de Proyecto Creados de Acuerdo al Estudio de Roles.....	26
7.2 Menús para los usuarios del Sistema, 1ra. Versión navegable y aplicable.....	30
7.3 Código del menú de Scripts.....	31
7.4 Menú para el Operador del Centro de Cómputos (Administrador del Sistema),	

1ra. versión.....	36
Código del Script.....	36
8. Configuraciones de Red y Servicios.....	42
8.1 Configuraciones de red en las terminales y el servidor.....	42
8.2 Configuración del servicio SSH en el cliente y el servidor (v).....	45
8.3 Archivos crontab con rutinas de backup y sus correspondientes scripts para el administrador.....	48
8.4 Instalación y configuración del servicio web http.....	51
8.5 Instalación y configuración de la base de datos MySQL.....	53
9. Scripts de Gestión y Consultas a la Base de Datos.....	59
9.2 Menú principal y funcionalidades.....	59
9.3 Ejecución y ejemplos de código.....	60
10. Menú Integrado de Gestión de Scripts.....	68
10.1 Funcionalidad del Menú Principal.....	68
10.2 Código del Script Principal.....	69
10.3 Descripción de las Funciones del Script.....	71
Glosario.....	72
Aplicaciones Usadas.....	72
Conclusión Final.....	73
Fuentes Consultadas.....	73
Hoja testigo.....	75

1. Introducción al documento

El presente documento ha sido elaborado con el objetivo de representar la carpeta correspondiente denominada “Sistemas Operativos III”, asignatura obligatoria del tercer año del bachillerato tecnológico en informática. En él, se encarga la primera entrega, como también la segunda entrega. Así mismo, se realizan las correcciones dadas para la primera entrega, como también se adjuntan los puntos solicitados para la segunda entrega.

1.1 Objetivos del proyecto

El objetivo del proyecto es desarrollar una plataforma web para el taller mecánico AIO, que permita la gestión y administración eficiente de los servicios comerciales ofrecidos, tales como diagnóstico vehicular, alineación y balanceo, y la venta de neumáticos. Asimismo, la página web facilitará la gestión de las reservas y la disponibilidad de plazas de estacionamiento en el servicio de parking.

1.2 Alcance del proyecto

El proyecto contempla ofrecer una gama de servicios mecánicos, incluyendo diagnóstico vehicular, alineación, balanceo, venta de neumáticos y servicio de parking automotriz. La empresa cuenta con la infraestructura adecuada y personal certificado para la realización de estos trabajos.

Se establece un número limitado de vehículos que pueden ser atendidos y estacionados simultáneamente, lo que, dependiendo de la carga de trabajo, podría ocasionar tiempos de espera prolongados.

Asimismo, es fundamental cumplir con las normativas legales en términos de zonificación, seguridad y medio ambiente. El servicio de parking no estará disponible para vehículos de gran porte.

Es necesario disponer de una cuenta de correo electrónico con dominio @gmail para la recepción de notificaciones. Además, las reservas físicas pueden ser realizadas sin necesidad de registro, agregando directamente al usuario final.

1.3 Metodologia

La metodología seleccionada para la gestión de este proyecto es **Scrumban**, una estrategia ágil que integra las mejores prácticas de **Scrum** y **Kanban**. Scrumban

resulta particularmente adecuada para entornos de desarrollo de software dinámicos y complejos, donde la flexibilidad, la mejora continua y la eficiencia son aspectos fundamentales.

2. Estudio de roles de usuarios

La identificación de roles de usuarios en un sistema operativo define permisos y responsabilidades específicas para gestionar el sistema de manera eficiente. Los roles identificados incluyen el Administrador de Sistemas, Administrador de Bases de Datos, Administrador de Seguridad, Administrador de Redes, Operador de Respaldos y el Soporte Técnico. Cada rol tiene accesos y tareas específicas para garantizar la operación segura y efectiva del sistema. Descripción de los distintos roles identificados:

2.1 Administrador de Sistemas

El administrador de sistemas es el encargado de configurar y mantener el hardware y software del servidor, gestión de usuarios y permisos, monitorear el rendimiento y ajustes del servidor para la eficiencia, documentación configuración, procedimientos y problemas. Así mismo, se encarga de la gestión y administración de los servicios que se encuentran en el mismo.

2.2 Administrador de Bases de Datos

El Administrador de Bases de Datos es responsable de gestionar las bases de datos del sistema, asegurando su seguridad y disponibilidad. Diseña y configura estructuras de bases de datos, implementa políticas de seguridad y realiza copias de seguridad. Además, tiene privilegios para crear, modificar y eliminar bases de datos, así como establecer políticas de seguridad y control de acceso.

2.3 Administrador de Seguridad

El Administrador de Seguridad es el responsable de protegerlo contra amenazas mediante el control de accesos, monitoreo de actividades, actualización de software y aplicación de medidas de protección de datos. Su objetivo es asegurar la integridad, confidencialidad y disponibilidad del servidor y sus datos.

2.4 Administrador de Redes

El administrador de redes, se encarga del mantenimiento y configuración de distintos medios de red que se conectan al sistema, tales como routers, switches y firewalls. Así mismo monitorea la red y verifica el correcto funcionamiento de cada uno. Cuenta con permisos sobre el área de comunicación de redes y tecnologías.

2.5 Administrador de RespalDOS y Recuperación

El administrador de respaldos y recuperación, es el encargado de mantener los datos respaldados de manera segura y puedan ser recuperados en caso de fallos, pérdida de datos o desastres. Este rol se enfoca en diseñar, implementar y gestionar políticas y procedimientos de respaldos (backup) y recuperación de datos.

2.6 Soporte Técnico

El Soporte Técnico a Usuarios Finales es el encargado de asistir y resolver problemas técnicos que enfrentan los usuarios de una organización con sus equipos, software o sistemas. Su función principal es garantizar que los empleados puedan utilizar la tecnología de manera eficiente y sin interrupciones.

3. Diagrama de Navegabilidad e Implementación de Shell Script para Administración de Usuarios y Grupos

El diagrama de navegabilidad del sistema proporciona una representación visual de las principales funciones y flujos de navegación del sistema. Este diagrama ilustra cómo los usuarios interactúan con las diferentes secciones y opciones disponibles, facilitando la comprensión de procesos y la estructura del sistema. Su objetivo es ofrecer una visión clara y organizada de la navegación, permitiendo una mejor planificación e implementación de las funcionalidades del sistema.

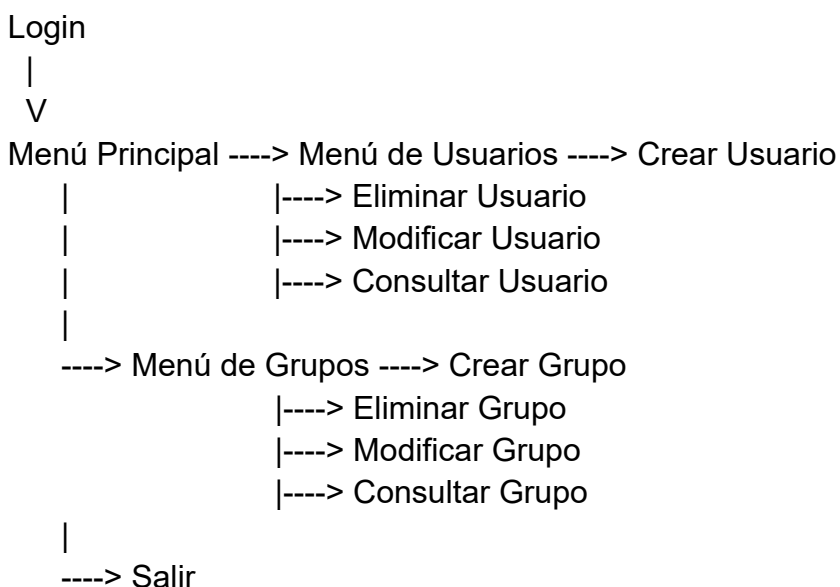
3.1 Estructura del diagrama

El diagrama se encuentra dividido de la siguiente manera:

1. **Inicio**
 - El programa comienza con la función **login**.
2. **Login**
 - Se solicita el nombre de usuario.
 - **Condición:** ¿El usuario existe?
 - **Sí:** El usuario inicia sesión correctamente y se dirige al **Menú Principal**.
 - **No:** Se muestra un mensaje de error y el programa se cierra.
3. **Menú Principal**
 - Opción 1: **Operaciones con Usuarios** → Dirige al **Menú de Usuarios**.
 - Opción 2: **Operaciones con Grupos** → Dirige al **Menú de Grupos**.
 - Opción 0: **Salir** → Termina el programa.

- **Condición:** ¿Opción válida? Si no, muestra "Opción no válida" y vuelve a mostrar el menú.
4. **Menú de Usuarios**
- Opción 1: **Crear Usuario** → Llama a la función `create_user`.
 - Opción 2: **Eliminar Usuario** → Llama a la función `delete_user`.
 - Opción 3: **Modificar Usuario** → Llama a la función `modify_user`.
 - Opción 4: **Consultar Usuario** → Llama a la función `query_user`.
 - Opción 0: **Volver al Menú Principal**.
 - **Condición:** ¿Opción válida? Si no, muestra "Opción no válida" y vuelve a mostrar el menú de usuarios.
5. **Menú de Grupos**
- Opción 1: **Crear Grupo** → Llama a la función `create_group`.
 - Opción 2: **Eliminar Grupo** → Llama a la función `delete_group`.
 - Opción 3: **Modificar Grupo** → Llama a la función `modify_group`.
 - Opción 4: **Consultar Grupo** → Llama a la función `query_group`.
 - Opción 0: **Volver al Menú Principal**.
 - **Condición:** ¿Opción válida? Si no, muestra "Opción no válida" y vuelve a mostrar el menú de grupos.
6. **Operaciones específicas:**
- Para cada operación de usuario/grupo (crear, eliminar, modificar, consultar):
 - **Condición:** Ejecuta los comandos utilizando `run_sudo`.
 - **Si el comando falla:** Muestra "Error" y espera 2 segundos.
 - **Si el comando tiene éxito:** Muestra un mensaje de éxito y espera a que el usuario presione Enter.
7. **Fin**

3.2 Mapa conceptual



3.3 Shell Script para Administración de Usuarios y Grupos

ESTE SCRIPT SE DEBE LLAMAR ./gest_scripts.sh

```
#!/bin/bash
```

```
# Función para ejecutar comandos con sudo sin contraseña
```

```
run_sudo() {  
    sudo -n "$@"  
    if [ $? -ne 0 ]; then  
        echo "Error: No se pudo ejecutar el comando con privilegios. Verifique la  
configuración de sudoers."  
        sleep 2  
    fi  
}
```

```
login() {  
    clear  
    read -p "Usuario: " usuario  
    if id "$usuario" &>/dev/null; then  
        echo "Inicio de sesión exitoso"  
        sleep 2  
        main_menu  
    else  
        echo "Usuario no existe"  
        sleep 2  
        exit 1  
    fi  
}
```

```
main_menu() {  
    while true; do  
        clear  
        echo "Menu Principal"  
        echo "1. Operaciones con Usuarios"  
        echo "2. Operaciones con Grupos"  
        echo "0. Salir"  
        read -p "Seleccione una opcion: " opcion  
        case $opcion in  
            1) user_menu ;;  
            2) group_menu ;;  
            0) exit 0 ;;  
            *) echo "Opcion no valida"; sleep 1 ;;  
        esac  
    done  
}
```



```

        esac
    done
}

user_menu() {
    while true; do
        clear
        echo "Operaciones con Usuarios"
        echo "1. Crear Usuario"
        echo "2. Eliminar Usuario"
        echo "3. Modificar Usuario"
        echo "4. Consultar Usuario"
        echo "0. Volver al Menu Principal"
        read -p "Seleccione una Opcion: " opcion
        case $opcion in
            1) create_user ;;
            2) delete_user ;;
            3) modify_user ;;
            4) query_user ;;
            0) return ;;
            *) echo "Opcion no valida"; sleep 1 ;;
        esac
    done
}

group_menu() {
    while true; do
        clear
        echo "Operaciones con Grupos"
        echo "1. Crear Grupo"
        echo "2. Eliminar Grupo"
        echo "3. Modificar Grupo"
        echo "4. Consultar Grupo"
        echo "0. Volver al Menu Principal"
        read -p "Seleccione una Opcion: " opcion
        case $opcion in
            1) create_group ;;
            2) delete_group ;;
            3) modify_group ;;
            4) query_group ;;
            0) return ;;
            *) echo "Opcion no valida"; sleep 1 ;;
        esac
    done
}

```

```

}

create_user() {
    read -p "Nombre del usuario a crear: " username
    run_sudo adduser --disabled-password --gecos "" "$username"
    echo "Usuario agregado con exito"
    read -p "Presione Enter para continuar"
}

delete_user() {
    read -p "Nombre del usuario a eliminar: " username
    run_sudo deluser "$username"
    echo "Usuario eliminado con exito"
    read -p "Presione Enter para continuar"
}

modify_user() {
    read -p "Nombre del usuario a modificar: " username
    run_sudo passwd "$username"
    read -p "Presione Enter para continuar"
}

query_user() {
    read -p "Nombre del usuario a consultar: " username
    id "$username"
    read -p "Presione Enter para continuar"
}

create_group() {
    read -p "Nombre del grupo a crear: " groupname
    run_sudo addgroup "$groupname"
    echo "Grupo agregado con exito"
    read -p "Presione Enter para continuar"
}

delete_group() {
    read -p "Nombre del grupo a eliminar: " groupname
    run_sudo delgroup "$groupname"
    echo "Grupo eliminado con exito"
    read -p "Presione Enter para continuar"
}

modify_group() {
    read -p "Nombre del grupo a modificar: " groupname

```

```

read -p "Nombre del usuario a agregar al grupo: " username
run_sudo adduser "$username" "$groupname"
read -p "Presione Enter para continuar"
}

```

```

query_group() {
    read -p "Nombre del grupo a consultar: " groupname
    getent group "$groupname"
    read -p "Presione Enter para continuar"
}

```

login

3.4 Descripción de funciones del script

1. run_sudo

- **Descripción:** Esta función ejecuta comandos con privilegios de superusuario (**sudo**) sin solicitar contraseña.
- **Funcionamiento:**
 - Ejecuta el comando proporcionado con **sudo -n**, que no pide la contraseña.
 - Si el comando falla (código de salida diferente de 0), muestra un mensaje de error indicando que no se pudo ejecutar el comando con privilegios de superusuario.
- **Propósito:** Facilitar la ejecución de comandos administrativos sin requerir interacción adicional del usuario para ingresar una contraseña.

2. login

- **Descripción:** Controla el inicio de sesión del usuario en el sistema.
- **Funcionamiento:**
 - Solicita un nombre de usuario.
 - Verifica si el usuario existe mediante el comando **id**.
 - Si el usuario existe, muestra un mensaje de inicio de sesión exitoso y redirige al **Menú Principal**.
 - Si el usuario no existe, muestra un mensaje de error y finaliza el script.
- **Propósito:** Validar la identidad del usuario antes de permitir el acceso a las funciones administrativas.

3. main_menu

- **Descripción:** Proporciona el **Menú Principal** donde se ofrecen opciones para operar con usuarios, grupos o salir del programa.
- **Funcionamiento:**
 - Muestra las opciones de:
 - 1: Operaciones con Usuarios.
 - 2: Operaciones con Grupos.
 - 0: Salir.
 - Según la opción seleccionada, redirige al menú correspondiente o sale del programa.
- **Propósito:** Ofrecer un punto de inicio para acceder a las operaciones de usuarios y grupos o salir del sistema.

4. user_menu

- **Descripción:** Proporciona el **Menú de Operaciones con Usuarios**.
- **Funcionamiento:**
 - Muestra las opciones de:
 - 1: Crear Usuario.
 - 2: Eliminar Usuario.
 - 3: Modificar Usuario.
 - 4: Consultar Usuario.
 - 0: Volver al Menú Principal.
 - Ejecuta la operación correspondiente según la opción seleccionada.
- **Propósito:** Facilitar las operaciones de creación, eliminación, modificación y consulta de usuarios.

5. group_menu

- **Descripción:** Proporciona el **Menú de Operaciones con Grupos**.
- **Funcionamiento:**
 - Muestra las opciones de:
 - 1: Crear Grupo.
 - 2: Eliminar Grupo.
 - 3: Modificar Grupo.
 - 4: Consultar Grupo.
 - 0: Volver al Menú Principal.
 - Ejecuta la operación correspondiente según la opción seleccionada.
- **Propósito:** Facilitar las operaciones de creación, eliminación, modificación y consulta de grupos.

6. Operaciones específicas

Estas funciones ejecutan comandos del sistema para gestionar usuarios y grupos.

- **create_user:**

- Solicita un nombre de usuario y utiliza `run_sudo` para crear el usuario con `adduser`, sin contraseña, y con información básica vacía.
- **delete_user:**
 - Solicita un nombre de usuario y utiliza `run_sudo` para eliminar al usuario con `deluser`.
- **modify_user:**
 - Solicita un nombre de usuario y permite cambiar la contraseña usando `passwd`.
- **query_user:**
 - Solicita un nombre de usuario y muestra información detallada del usuario usando `id`.
- **create_group:**
 - Solicita un nombre de grupo y utiliza `run_sudo` para crear el grupo con `addgroup`.
- **delete_group:**
 - Solicita un nombre de grupo y utiliza `run_sudo` para eliminar el grupo con `delgroup`.
- **modify_group:**
 - Solicita un nombre de grupo y un usuario para añadirlo al grupo con `adduser`.
- **query_group:**
 - Solicita un nombre de grupo y muestra información del grupo con `getent group`.

4. Relevamiento y Justificación del Sistema Operativo a Utilizar Tanto en las Terminales de los Usuarios como en el Servidor.

A continuación, se realiza una comparación de sistemas operativos similares para justificar la elección del sistema operativo que mejor se adapta tanto a las terminales de los usuarios como al servidor.

4.1 Comparación de Sistemas Operativos

En esta sección se presentan las características de Ubuntu Server, AlmaLinux y Rocky Linux, analizando sus ventajas, desventajas y la justificación de la selección del sistema adecuado.

4.1.1 Ubuntu Server

Ventajas:

- **Amplio soporte y comunidad:** Desarrollado por Canonical, Ubuntu Server cuenta con una gran comunidad de usuarios y desarrolladores que proporcionan documentación extensa y soporte.
- **Soporte LTS (Long Term Support):** Canonical ofrece versiones con soporte extendido (LTS), asegurando actualizaciones de seguridad y correcciones de errores durante cinco años o más.
- **Compatibilidad:** Es compatible con una amplia variedad de hardware y software, siendo especialmente popular en entornos de servidores web, bases de datos y servicios en la nube.
- **Facilidad de uso:** Ubuntu Server es amigable para los administradores de sistemas, con herramientas fáciles de configurar.

Desventajas:

- **Menor compatibilidad con entornos Red Hat:** Puede presentar dificultades en ciertos entornos empresariales que siguen directamente el ecosistema Red Hat.

4.1.2 AlmaLinux

Ventajas:

- **Continuidad del ecosistema RHEL:** AlmaLinux es un fork de CentOS, creado después de que Red Hat cambió la dirección de CentOS. Ofrece la misma estabilidad que Red Hat Enterprise Linux (RHEL) sin los costos asociados.
- **Comunidad activa:** Aunque es un proyecto relativamente nuevo, la comunidad de desarrolladores y usuarios ha crecido rápidamente, proporcionando soporte y actualizaciones regulares.
- **100% compatible con RHEL:** Esto lo convierte en una excelente opción para aquellos que buscan la estabilidad y confiabilidad de RHEL, sin las restricciones de licencias.

Desventajas:

- **Relativa juventud:** Al ser un proyecto nuevo, AlmaLinux aún no tiene la misma trayectoria de estabilidad y confianza que otros sistemas operativos más consolidados.
- **Menor enfoque en la facilidad de uso:** Aunque estable y robusto, puede no ser tan fácil de manejar para administradores menos experimentados.

4.1.3 Rocky Linux

Ventajas:

- **Estabilidad:** Al igual que AlmaLinux, Rocky Linux también es un clon binario de RHEL. Fue creado como respuesta a la decisión de Red Hat de finalizar CentOS tal como se conocía.
- **Misión a largo plazo:** El fundador de Rocky Linux, Gregory Kurtzer, fue también el creador de CentOS, y ha establecido el proyecto con el objetivo de ser un reemplazo a largo plazo para CentOS.
- **Comunidad dedicada:** Rocky Linux ha ganado rápidamente adeptos dentro de la comunidad empresarial, especialmente aquellos que dependen de la estabilidad y compatibilidad de RHEL.

Desventajas:

- **Juventud del proyecto:** A pesar del rápido crecimiento, Rocky Linux es aún joven en comparación con otros sistemas operativos, lo que podría plantear dudas sobre su trayectoria a largo plazo.
- **Recursos limitados:** Al ser un proyecto de código abierto mantenido por la comunidad, los recursos disponibles podrían no ser tan amplios como los de otros sistemas operativos respaldados por empresas comerciales.

4.2 Justificación de Ubuntu Server

Ubuntu Server ha sido seleccionado para ser utilizado en este proyecto debido a su equilibrio entre facilidad de uso, soporte a largo plazo, compatibilidad y características avanzadas.

Estabilidad y Confiabilidad

Desarrollado por la empresa **Canonical**, Ubuntu Server ha demostrado ser una plataforma robusta y confiable para servidores en diversos entornos. Canonical ofrece soporte profesional y una amplia gama de actualizaciones, lo que asegura su estabilidad a largo plazo.

Facilidad de Uso y Administración

Ubuntu Server es conocido por ser relativamente sencillo de configurar y administrar, en comparación con otros sistemas basados en Linux. Las herramientas de administración y su vasta comunidad lo hacen adecuado para administradores con experiencia variada, desde principiantes hasta expertos.

Compatibilidad

Ubuntu Server es compatible con una amplia variedad de hardware y software, lo que lo hace ideal para entornos mixtos. Su capacidad de integración con tecnologías como Docker, Kubernetes y OpenStack lo convierte en una opción ideal para servidores web, bases de datos, y aplicaciones en la nube.

4.3 Implementación y Mantenimiento

La implementación de **Ubuntu Server 22.04 LTS** será realizada en un entorno virtualizado, ya sea mediante máquinas virtuales o contenedores, permitiendo una gestión controlada de los recursos y garantizando un entorno replicable para pruebas y producción.

4.3.1 Facilidad de Instalación

La instalación de Ubuntu Server es rápida y directa, con una interfaz de instalación intuitiva que permite configurar rápidamente los componentes esenciales del servidor.

4.3.2 Soporte a Largo Plazo

Canonical ofrece versiones **LTS** con soporte extendido por hasta cinco años, lo que garantiza que los servidores recibirán actualizaciones de seguridad y mantenimiento por un periodo prolongado, reduciendo los riesgos de vulnerabilidades a largo plazo.

Conclusión

Después de analizar **Ubuntu Server**, **AlmaLinux**, y **Rocky Linux**, se concluye que **Ubuntu Server** es la opción más adecuada para este proyecto debido a su estabilidad, facilidad de administración, soporte a largo plazo y compatibilidad con diversas tecnologías. Además, la extensa comunidad de usuarios y el soporte de Canonical aseguran que Ubuntu Server seguirá siendo una opción fiable y flexible en el futuro.

5. Manual de Instalación del Sistema Operativo en el Servidor Seleccionado

5.1 Propósito del Manual

El siguiente manual de instalación se proporciona con la finalidad de brindar una guía rápida y eficiente de instalación del sistema operativo Ubuntu Server 22.04.

5.2 Descripción General de Ubuntu Server 22.04 LTS

Ubuntu Server 22.04 LTS es una versión diseñada para entornos de servidor, ofreciendo estabilidad y soporte por cinco años. Incluye un kernel actualizado de Linux, herramientas de seguridad avanzadas como AppArmor y opciones de cifrado de disco. Es compatible con tecnologías de cloud y virtualización, como OpenStack y Kubernetes, y soporta hardware moderno. También ofrece

herramientas de gestión y automatización, como Landscape y Juju, para facilitar la administración de servidores.

5.3 Audiencia Objetivo

La audiencia objetivo para un manual de instalación de Ubuntu Server 22.04 es el administrador de sistemas y el profesional de TI con experiencia básica en servidores. También le resulta útil al estudiante de informática que quiere aprender sobre la gestión de servidores.

5.4 Requisitos previos

Requerimientos Mínimos

Hardware

1. **Procesador:** CPU de 1 GHz (x86_64 o ARMv8)
2. **Memoria RAM:** 512 MB
3. **Almacenamiento:** 2.5 GB de espacio libre en disco para una instalación mínima
4. **Conexión a Internet:** Para descargar actualizaciones y paquetes adicionales (opcional pero recomendado)
5. **Dispositivo de arranque:** Unidad USB o CD/DVD para el medio de instalación

Software

Los requerimientos mínimos de software para instalar Ubuntu Server 22.04 son:

1. **Arquitectura de CPU:** Procesador x86_64 (64 bits) o ARMv8.
2. **Sistema Operativo:** Compatible con arquitectura de 64 bits.
3. **Almacenamiento:** Espacio suficiente en disco para la instalación del sistema operativo (al menos 2.5 GB).
4. **Dispositivo de Arranque:** Unidad USB, CD/DVD o soporte para arranque por red.

Requerimientos recomendados

Hardware

1. **Procesador:** CPU de múltiples núcleos (x86_64 o ARMv8) a 3 GHz o superior, con soporte para virtualización (VT-x o AMD-V).
2. **Memoria RAM:** 8 GB o más para manejar cargas de trabajo intensivas y ejecutar múltiples aplicaciones y servicios simultáneamente.
3. **Almacenamiento:**
 - 40 GB o más de espacio libre en disco para una instalación completa con todos los paquetes necesarios.

- Almacenamiento SSD para un rendimiento óptimo y tiempos de acceso rápidos.
- Discos adicionales para almacenamiento de datos y backups.
- 4. **Conexión a Internet:** Conexión de alta velocidad y baja latencia, idealmente con redundancia para asegurar alta disponibilidad.
- 5. **Dispositivo de Arranque:** Unidad USB 3.0 o superior, o arranque por red para instalaciones rápidas y flexibles.

Software

1. **Arquitectura de CPU:** Procesador x86_64 (64 bits) o ARMv8 con soporte para virtualización (VT-x o AMD-V).
2. **Sistema Operativo:** Compatible con arquitectura de 64 bits, con soporte para UEFI.
3. **Almacenamiento:**
 - Al menos 40 GB de espacio libre en disco para el sistema operativo y aplicaciones adicionales.
4. **Dispositivo de Arranque:** Unidad USB 3.0, CD/DVD, o soporte para arranque por red (PXE boot).

5.5 Preparativos

Descarga de la Imagen ISO

- Guía detallada para la instalación de la imagen ISO desde el sitio oficial.

Creación de un USB de Arranque

- Lista de software recomendado.
- Pasos detallados para crear la unidad de arranque.

Configuración de la Máquina Virtual

- Especificación recomendados de CPU, RAM y almacenamiento para la máquina virtual.

5.6 Inicialización

Se procederá a brindar una guía de instalación detallada para personal amateur en sistemas de información y computación.

Crear un Medio Booteable de Ubuntu Server

Get Ubuntu Server

Option 2: Manual server installation

USB or DVD image based physical install

- ✓ OS security guaranteed until April 2025
- ✓ Extended security maintenance until April 2030
- ✓ Commercial support for enterprise customers

[Download Ubuntu Server 22.04 LTS](#)

[Alternative downloads >](#)

[Alternative architectures >](#)

[Read the Ubuntu Server 22.04 LTS release notes](#)



Option 1 - Instant Ubuntu VMs

Option 2 - Manual server installation

Option 3 - Automated server provisioning

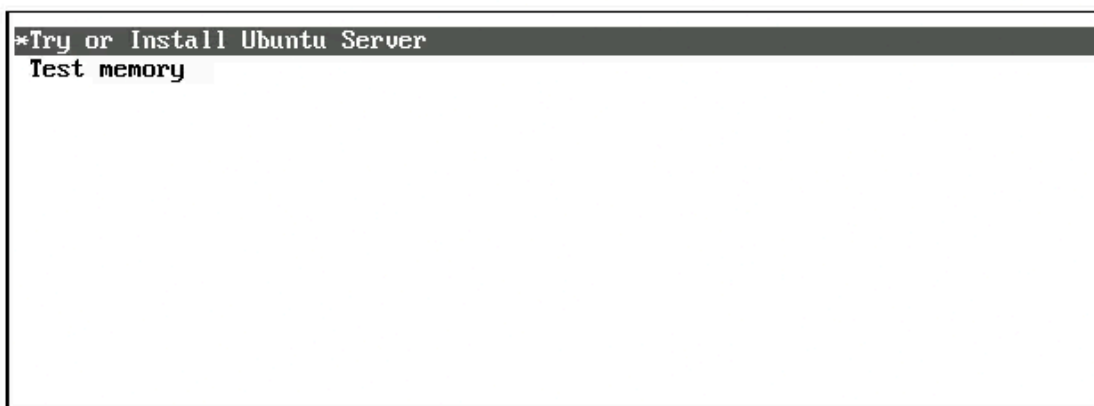
Seleccione la "Opción 2 - Manual server installation" para descargar el medio de instalación y poder instalar manualmente el servidor de Ubuntu. Fuente:

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

5.6.2 Iniciar la instalación

Una vez que haya creado su medio de instalación, podrá instalar y configurar su servidor de Ubuntu. Para ello, vincule su servidor con el USB booteable e inicie el dispositivo. Si está utilizando un CD o DVD, introdúzcalo y reinicie el servidor. Después de que se haya leído el archivo de imagen, aparecerá una ventana con el bootloader GNU GRUB. Ahora, solo confirme la entrada "Try or install Ubuntu" para empezar a instalar el servidor de Ubuntu.

GNU GRUB version 2.06



Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 13s.

Ventana del bootloader GNU GRUB

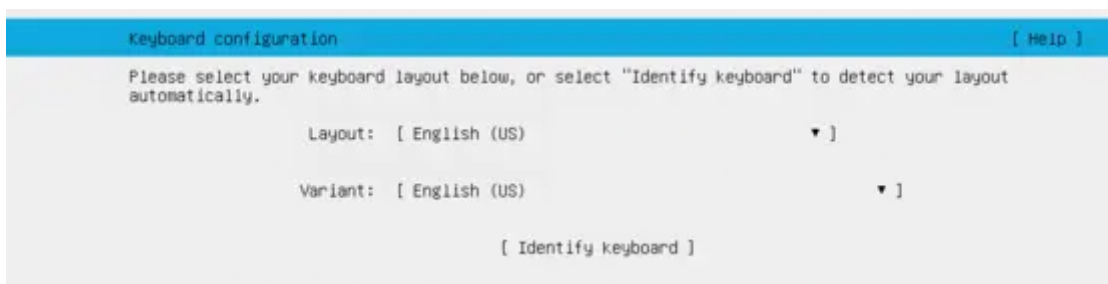
5.6.3 Seleccionar el idioma del sistema y la distribución del teclado

Luego de una breve preconfiguración, se iniciará el **asistente de instalación** del software Ubuntu Server. A modo de diferencia del propio sistema operativo, este posee una interfaz gráfica de usuario. En primer lugar, elige el lenguaje del sistema, que también se utilizara como idioma de instalación.



Instalación de Ubuntu Server: Selección de idioma

Una vez elegido el idioma general del sistema, le corresponderá elegir la distribución del teclado. Busque **manualmente el idioma del teclado** entre las opciones disponibles en la lista o utilice la opción **"Reconocer teclado"** para compartir su idioma con el asistente.

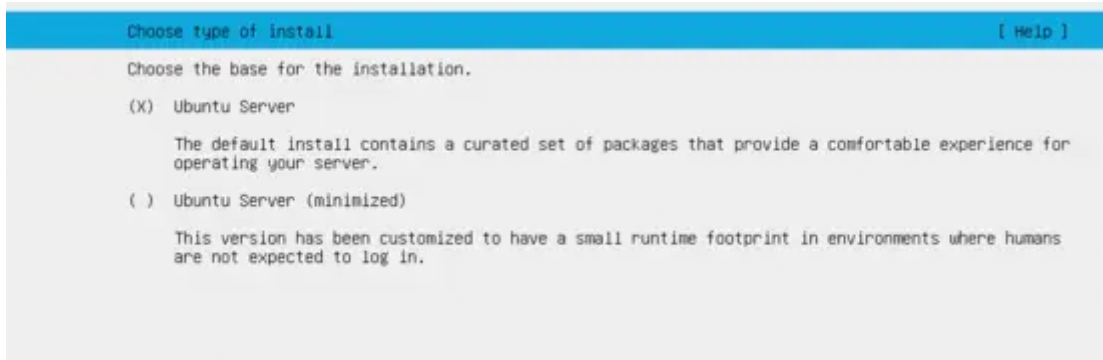


Configurar Ubuntu Server: teclado

5.6.4 Seleccionar la variante del servidor Ubuntu

En el siguiente paso sienta la base de la instalación: si elige la instalación estándar "Ubuntu server", obtendrá un paquete de software básico con el que tendrá acceso a todas las herramientas para trabajar cómodamente con el servidor de Ubuntu. Si decide optar por la opción "Ubuntu Server (minimized)", perderá algunas herramientas de usuario durante la instalación. Esta variante consume menos recursos, pero solo conviene cuando se tiene previsto utilizar poco el servidor.

Para elegir una de las dos opciones, marque con una **cruz la opción correspondiente**



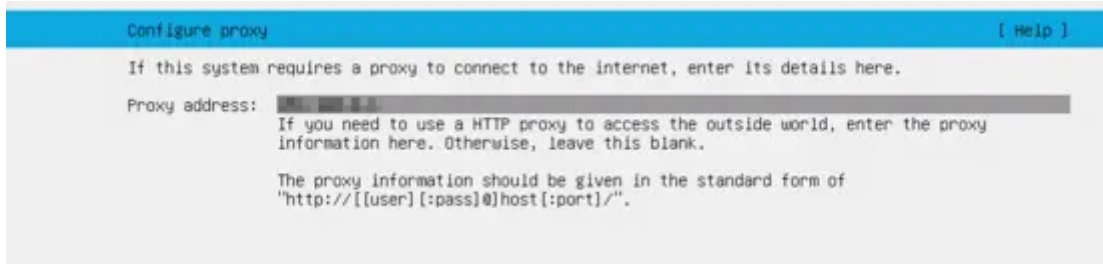
Instalar Ubuntu Server: elegir el tipo de instalación

5.6.5 Configuración de red y proxy

Ahora es el turno de configurar la conexión de red y el servidor proxy.

Si su servidor Ubuntu ya está conectado a una red con un servicio de DHCP activo, el adaptador de red adecuado debería estar pre configurado automáticamente. Si no puede o no desea establecer una conexión en este momento, avance seleccionando "Continuar sin red".

Si su servidor Ubuntu debe utilizar un servidor proxy para conectarse a Internet, introduzca la dirección individual del proxy HTTP en el siguiente diálogo.



Instalación Ubuntu Server: Configuración Proxy

5.6.6 Configurar el espejo del archivo de Ubuntu

Los numerosos paquetes de software que forman parte de la distribución Ubuntu se actualizan regularmente y se almacenan en el archivo público "archive.ubuntu.com/ubuntu". Para no depender exclusivamente de este archivo, existen espejos de archivo de Ubuntu para repartir la carga y por seguridad. Introduzca la dirección de un espejo si desea desviarse del archivo estándar.



Configurar el espejo del archivo Ubuntu

5.6.7 Administrar el espacio de almacenamiento

Se recomienda la opción que viene preseleccionada, es decir, un disco duro completo para el servidor Ubuntu. En la siguiente ventana, se puede ver un resumen del sistema de archivos. Existe la posibilidad de que se encuentre un gran número de particiones, aunque no se haya seleccionado la opción de partición, y se debe principalmente ya que Ubuntu reserva automáticamente pequeñas particiones del disco duro para los archivos importantes del sistema y el bootloader.



Configuración de almacenamiento del servidor Ubuntu

En cuanto confirme la configuración del espacio, recibirá un aviso indicando que los datos actuales del disco afectado serán eliminados. Para continuar con la instalación, debe aceptar que se borren.

5.6.8 Configurar el perfil

Cree una nueva cuenta de usuario para iniciar sesión en su servidor Ubuntu. Introduzca su nombre, nombre de usuario y contraseña. Además, deberá asignar un nombre a su servidor, que se utilizará al comunicarse con otros ordenadores.

Ubuntu server: configurar el perfil

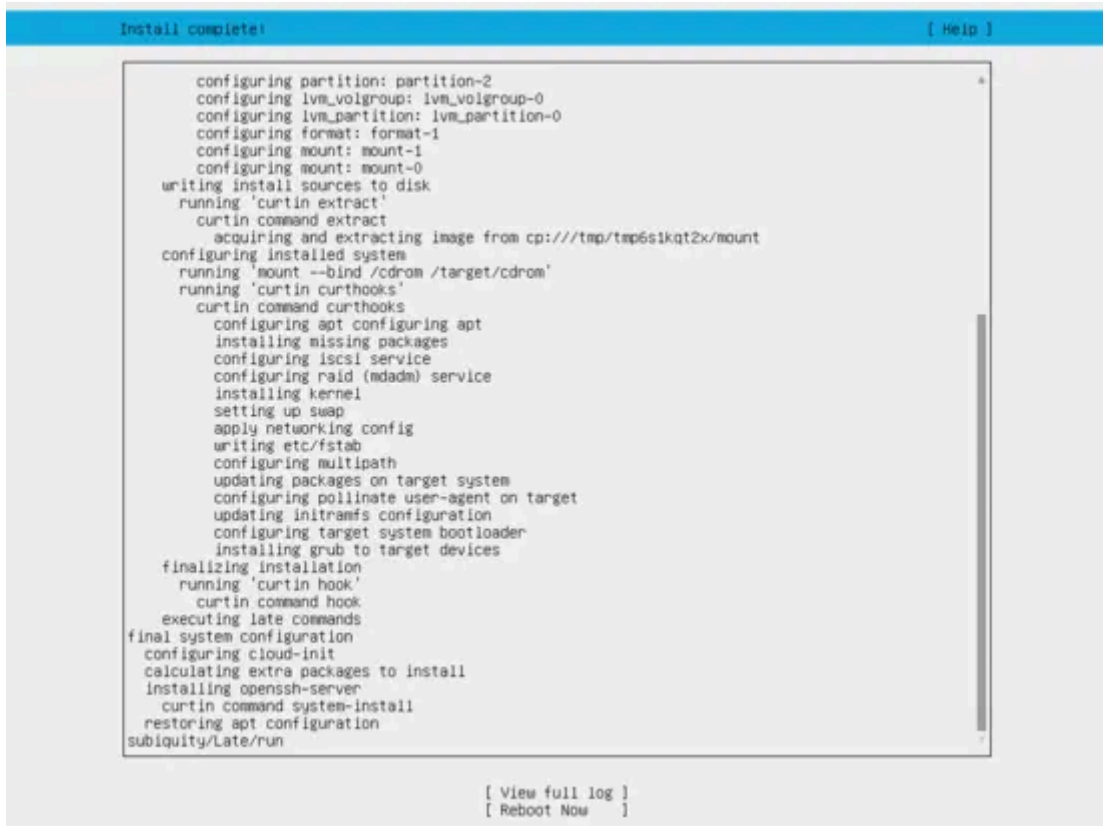
5.6.9 Configuración de SSH

SSH es una tecnología importante para acceder a sus servidores de manera segura desde la distancia. Por lo tanto, durante la instalación de su servidor Ubuntu, podrá instalar directamente el paquete del programa OpenSSH colocando una cruz en el paréntesis. Si ya tiene claves SSH válidas, puede importarlas.

Ubuntu Server: configurar SSH

5.6.10 Completar la instalación de Ubuntu server

Tras configurar SSH, el asistente iniciará automáticamente la instalación propiamente dicha. Si la instalación se ha realizado correctamente, aparecerá el mensaje "Instalación completa".



```
Install complete: [ Help ]

configuring partition: partition-2
configuring lvm_voigroup: lvm_voigroup-0
configuring lvm_partition: lvm_partition-0
configuring format: format-1
configuring mount: mount-1
configuring mount: mount-0
writing install sources to disk
running 'curtin extract'
curtin command extract
  acquiring and extracting image from cp:///tmp/tmp6s1kqt2x/mount
configuring installed system
running 'mount --bind /cdrom /target/cdrom'
running 'curtin curthooks'
curtin command curthooks
  configuring apt
  configuring apt
  installing missing packages
  configuring iscsi service
  configuring raid (mdadm) service
  installing kernel
  setting up swap
  apply networking config
  writing etc/fstab
  configuring multipath
  updating packages on target system
  configuring pollinate user-agent on target
  updating initramfs configuration
  configuring target system bootloader
  installing grub to target devices
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
  configuring cloud-init
  calculating extra packages to install
  installing openssh-server
  curtin command system-install
  restoring apt configuration
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

Ubuntu Server: notificación de instalación completa

Extraiga el medio booteable y seleccione la opción "Reiniciar ahora" para completar la instalación.

5.7 Conclusión de la instalación

La instalación de Ubuntu Server 22.04 es un proceso directo que incluye la creación de un medio booteable, configuración de red, y opciones de seguridad como SSH. Siguiendo los pasos indicados, podrá tener su servidor configurado y operativo de manera eficiente.

En este manual, se ha optado por invertir los colores de las capturas de pantalla, cambiando el esquema original (negro con naranja) para minimizar el gasto de tinta durante la impresión. Esta medida busca evitar el uso excesivo de tinta negra, que suele ser costosa y se gasta rápidamente en áreas extensas. Al invertir los colores, se logra una impresión más eficiente y económica, sin comprometer la claridad y legibilidad de las imágenes.

6. Implementación de Máquina Virtual o Contenedor Conteniendo el Script de Administración

6.1 Preparativos

En esta sección se describen los preparativos realizados antes de la implementación del script en la máquina virtual. La máquina virtual ya contiene el entorno necesario para ejecutar el script, y se realizaron los siguientes pasos previos:

- **Verificación del estado y configuración de la máquina virtual:** Asegurarse de que la máquina virtual esté correctamente configurada y funcionando.
- **Transferencia del script `gest_scripts.sh` a la máquina virtual:** Utilizar un método de transferencia adecuado para llevar el script a la máquina virtual.

6.2 Credenciales de Acceso

Las credenciales de acceso al servidor son las siguientes:

MÁQUINA VIRTUAL:

USUARIO: aio

CONTRASEÑA: sim

MYSQL:

USUARIO: 'root'

CONTRASEÑA: "

6.3 Implementación del Script

6.3.1 Asignación de Permisos

Para ejecutar el script de manera eficiente, se le deberán de otorgar permisos de ejecución. Para hacerlo, se deberá ingresar la siguiente línea:

```
chmod +x "gest_scripts.sh"
```

6.3.2 Ejecución del script

Pasos para ejecutar el script en la máquina virtual:

```
./gest_scripts.sh
```

6.4 Uso de la Máquina Virtual

La máquina virtual ha sido descargada y almacenada en un pendrive. Para utilizar la máquina virtual:

1. **Inserte el pendrive:** Conecte el pendrive a la computadora donde se utilizará la máquina virtual.
2. **Inicie la Máquina Virtual:** Use su software de virtualización preferido (como VirtualBox o VMware) para abrir e iniciar la máquina virtual desde el pendrive.

Esto concluye la sección de implementación de la máquina virtual conteniendo el script de administración.

7. Nuevos desarrollos (Segunda entrega)

7.1 Usuarios de Proyecto Creados de Acuerdo al Estudio de Roles

Dentro del marco del proyecto informático S.I.G.A.E., se ha determinado una estructura de roles, los cuales presentan distintas características y responsabilidades según su área de trabajo. Cada uno de estos roles se aplicará a diferentes usuarios del sistema, con el fin de asegurar una administración eficiente y organizada del entorno. Este documento tiene como objetivo estandarizar el proceso de creación de usuarios y asignación de permisos, permitiendo su replicabilidad y manteniendo la consistencia en la gestión del sistema.

Identificación de Roles:

En el marco del proyecto informático S.I.G.A.E., se ha definido una estructura de roles para garantizar una gestión eficiente y segura del sistema. Los roles se organizan de la siguiente manera:

- **Administrador de sistemas:** Responsable de la configuración, mantenimiento y supervisión de la infraestructura del sistema.
- **Administrador de bases de datos:** Encargado del diseño, mantenimiento y gestión de las bases de datos, asegurando su disponibilidad y seguridad.
- **Administrador de seguridad:** Gestiona los aspectos de seguridad del sistema, incluyendo el control de accesos, monitoreo de actividades sospechosas y aplicación de políticas de protección.
- **Administrador de redes:** Se ocupa de la configuración y monitoreo de los componentes de red, asegurando una conectividad estable y segura.

- **Administrador de respaldos y recuperación:** Responsable de la creación de copias de seguridad y la restauración de datos en caso de fallos o desastres.
- **Soporte técnico:** Proporciona asistencia técnica a los usuarios del sistema, resolviendo problemas y asegurando el correcto uso de los recursos tecnológicos.

Estructura Jerárquica del Sistema:

```

/ - Administrador de sistemas
/ - - - Administrador de bases de datos
    / - - - Administrador de seguridad
    / - - - Administrador de redes
    / - - - Administrador de respaldos y recuperación
/ - Soporte técnico

```

Esta estructura permite una clara distinción de responsabilidades, con el administrador de sistemas a la cabeza de las operaciones globales, y los demás roles especializados bajo su supervisión.

Ejecución de Actividades:

Creación de Grupos:

Para gestionar los permisos de manera eficiente, se han creado grupos específicos para cada rol. Los siguientes comandos permiten la creación de los grupos necesarios:

```

sudo groupadd sysadmin
sudo groupadd dbadmin
sudo groupadd secadmin
sudo groupadd netadmin
sudo groupadd backupadmin
sudo groupadd support

```

Creación de Usuarios y Asignación de Roles:

Cada usuario será asignado a uno de los grupos definidos anteriormente, garantizando así que cada uno cuente con los permisos y privilegios necesarios para su rol. A continuación, se muestran los comandos utilizados para la creación de usuarios y la asignación de roles:

```

# Creación de usuario para el Administrador de Sistemas
sudo useradd -m -s /bin/bash -G sysadmin adminsys

```

```
sudo passwd adminsys
```

```
# Creación de usuario para el Administrador de Bases de Datos
```

```
sudo useradd -m -s /bin/bash -G dbadmin admindb
```

```
sudo passwd admindb
```

```
# Creación de usuario para el Administrador de Seguridad
```

```
sudo useradd -m -s /bin/bash -G secadmin adminsec
```

```
sudo passwd adminsec
```

```
# Creación de usuario para el Administrador de Redes
```

```
sudo useradd -m -s /bin/bash -G netadmin adminnet
```

```
sudo passwd adminnet
```

```
# Creación de usuario para el Administrador de RespalDOS y Recuperación
```

```
sudo useradd -m -s /bin/bash -G backupadmin adminbackup
```

```
sudo passwd adminbackup
```

```
# Creación de usuario para el Soporte Técnico
```

```
sudo useradd -m -s /bin/bash -G support techsupport
```

```
sudo passwd techsupport
```

(*) El comando `passwd` establece la contraseña para el usuario.

Configuración de Permisos para Cada Rol:

Es necesario configurar los permisos adecuados para cada rol a través del archivo `sudoers`, el cual se edita utilizando el siguiente comando:

```
bash
```

Copiar código

```
sudo visudo
```

A continuación, se agregan las siguientes líneas al archivo `sudoers` para otorgar los permisos correspondientes a cada grupo:

Permisos para el Administrador de Sistemas:

```
%sysadmin ALL=(ALL) ALL
```

Permisos para el Administrador de Bases de Datos:

```
%dbadmin ALL=(ALL) NOPASSWD: /usr/bin/mysql,  
/usr/bin/mysqldump, /usr/bin/mysqladmin
```

Permisos para el Administrador de Seguridad:

```
%secadmin ALL=(ALL) NOPASSWD: /usr/bin/iptables,  
/usr/bin/ufw, /usr/bin/fail2ban-client
```

Permisos para el Administrador de Redes:

```
%netadmin ALL=(ALL) NOPASSWD: /usr/sbin/ifconfig,  
/usr/sbin/ip, /usr/sbin/route, /usr/bin/nmcli
```

Permisos para el Administrador de Respaldos y Recuperación:

```
%backupadmin ALL=(ALL) NOPASSWD: /usr/bin/rsync,  
/usr/bin/tar, /usr/sbin/restore, /usr/sbin/dump
```

Permisos para el Soporte Técnico:

```
%support ALL=(ALL) NOPASSWD: /usr/bin/top, /usr/bin/htop,  
/usr/bin/journalctl, /usr/bin/systemctl status
```

Auditoría del sistema

Usuario	Rol	Grupo	Funcion
adminsyst	Administrador de sistemas	sysadmin	Gestión de la infraestructura tecnológica
admindb	Administrador de bases de datos	dbadmin	Diseño y control de la base de datos
adminsec	Administrador de seguridad	secadmin	Seguridad de procesos y recursos
adminnet	Administrador de	netadmin	Funcionamiento del software y

	redes		hardware de redes
adminbackup	Administrador de respaldos y recuperación	backupadmin	Protección y restauración de datos
techsupport	Soporte técnico	support	Satisfacción del usuario y resolución de problemas

Verificación

Se proporciona una guía detallada sobre la creación de los usuarios de proyecto de acuerdo a los diferentes roles del sistema. Así mismo, se crean grupos los cuales verifican que el usuario tenga los permisos necesarios para su rol. Se facilita el correcto funcionamiento del sistema y una manera organizada de la utilización de los recursos tecnológicos.

7.2 Menús para los usuarios del Sistema, 1ra. Versión navegable y aplicable.

Se procede con la creación del primer menú para los usuarios del sistema, siendo esta la 1ra. versión navegable y aplicable. Para ello, se presenta una representación gráfica primaria sobre la escala jerárquica y la relación de los usuarios con respecto a sus distintas áreas de trabajo.

Menú Principal

Administración

Administrador de sistemas

- Gestión de usuarios
- Configuración del sistema
- Registro de actividad

Administrador de bases de datos

- Gestión de bases de datos
- Consultas y reportes
- Respaldo y restauración

Administrador de seguridad

- Control de acceso
- Registro de eventos
- Configuración de políticas

Administrador de redes

- Configuración de red
- Monitoreo de dispositivos
- Troubleshooting

Administrador de backups y recuperación

Programación de respaldos

Restauración de datos

Registro de actividades

Soporte técnico

Gestión de tickets

Documentación técnica

Contacto

7.3 Código del menú de Scripts

Luego de referenciar el mapa conceptual sobre la estructura del menú principal, se procede con la implementación del shell script, el cual contiene cada función de su respectiva área de trabajo.

```
#!/bin/bash
```

```
# Función para limpiar la pantalla
```

```
clear_screen() {  
    clear  
}
```

```
# Función para mostrar el menú principal
```

```
show_main_menu() {  
    clear_screen  
    echo "==== Menú Principal ====  
    echo "1. Administración"  
    echo "2. Soporte técnico"  
    echo "0. Salir"  
    echo "=====  
    read -p "Seleccione una opción del Menú Principal: " option  
    echo  
    return $option  
}
```

```
# Función para mostrar el menú de Administración
```

```
show_admin_menu() {  
    clear_screen  
    echo "==== Administración ====  
    echo "1. Administrador de sistemas"  
    echo "2. Administrador de bases de datos"  
    echo "3. Administrador de seguridad"  
    echo "4. Administrador de redes"  
    echo "5. Administrador de backups y recuperación"
```

```

    echo "0. Volver al menú principal"
    echo "===== "
    read -p "Seleccione una opción de Administración: " admin_option
    echo
    return $admin_option
}

# Función para mostrar el menú de Soporte técnico
show_support_menu() {
    clear_screen
    echo "===== Soporte técnico ====="
    echo "1. Gestión de tickets"
    echo "2. Documentación técnica"
    echo "3. Contacto"
    echo "0. Volver al menú principal"
    echo "===== "
    read -p "Seleccione una opción de Soporte técnico: " support_option
    echo
    return $support_option
}

# Funciones del Administrador de sistemas
manage_users() {
    while true; do
        clear_screen
        echo "===== Gestión de usuarios ====="
        echo "1. Agregar usuario"
        echo "2. Modificar usuario"
        echo "3. Eliminar usuario"
        echo "4. Listar usuarios"
        echo "0. Volver al menú anterior"
        echo "===== "
        read -p "Seleccione una opción de Gestión de usuarios: " user_option
        echo

        case $user_option in
            1)
                read -p "Ingrese el nombre del nuevo usuario: " new_user
                sudo useradd -m "$new_user"
                sudo passwd "$new_user"
                echo "Usuario $new_user agregado."
                ;;
            2)
                read -p "Ingrese el nombre del usuario a modificar: " mod_user

```



```

        read -p "Ingrese el nuevo grupo para el usuario: " new_group
        sudo usermod -aG "$new_group" "$mod_user"
        echo "Usuario $mod_user modificado."
        ;;
    3)
        read -p "Ingrese el nombre del usuario a eliminar: " del_user
        sudo userdel -r "$del_user"
        echo "Usuario $del_user eliminado."
        ;;
    4)
        echo "Lista de usuarios:"
        cut -d: -f1 /etc/passwd
        ;;
    0)
        break
        ;;
    *)
        echo "Opción inválida"
        ;;
esac
read -n 1 -s -r -p "Presione cualquier tecla para continuar"
done
}

system_config() {
    while true; do
        clear_screen
        echo "==== Configuración del sistema ====="
        echo "1. Actualizar software del sistema"
        echo "2. Configurar zona horaria"
        echo "3. Verificar espacio en disco"
        echo "0. Volver al menú anterior"
        echo "===== "
        read -p "Seleccione una opción de Configuración del sistema: " config_option
        echo

        case $config_option in
            1)
                sudo apt update && sudo apt upgrade -y
                echo "Sistema actualizado."
                ;;
            2)
                sudo dpkg-reconfigure tzdata
                echo "Zona horaria actualizada."

```

```

        ;;
    3)
        df -h
        ;;
    0)
        break
        ;;
    *)
        echo "Opción inválida"
        ;;
esac
read -n 1 -s -r -p "Presione cualquier tecla para continuar"
done
}

```

Funciones del Administrador de bases de datos

```

manage_databases() {
    while true; do
        clear_screen
        echo "===== Gestión de bases de datos ====="
        echo "1. Crear base de datos"
        echo "2. Eliminar base de datos"
        echo "3. Listar bases de datos"
        echo "4. Ejecutar consulta SQL"
        echo "0. Volver al menú anterior"
        echo "===== "
        read -p "Seleccione una opción de Gestión de bases de datos: " db_option
        echo

        case $db_option in
            1)
                read -p "Ingrese el nombre de la nueva base de datos: " new_db
                mysql -u root -e "CREATE DATABASE $new_db;"
                echo "Base de datos $new_db creada."
                ;;
            2)
                read -p "Ingrese el nombre de la base de datos a eliminar: " del_db
                mysql -u root -e "DROP DATABASE $del_db;"
                echo "Base de datos $del_db eliminada."
                ;;
            3)
                echo "Lista de bases de datos:"
                mysql -u root -e "SHOW DATABASES;"
                ;;

```

```

4)
    read -p "Ingrese la consulta SQL: " sql_query
    read -p "Ingrese el nombre de la base de datos: " db_name
    mysql -u root -D "$db_name" -e "$sql_query"
    ;;
0)
    break
    ;;
*)
    echo "Opción inválida"
    ;;
esac
read -n 1 -s -r -p "Presione cualquier tecla para continuar"
done
}

```

Bucle principal del menú

```

while true; do
    show_main_menu
    option=$?

    case $option in
        1)
            while true; do
                show_admin_menu
                admin_option=$?
                case $admin_option in
                    1) manage_users ;;
                    2) manage_databases ;;
                    0) break ;;
                    *) echo "Opción inválida" ;;
                esac
            done
            ;;
        2)
            while true; do
                show_support_menu
                support_option=$?
                case $support_option in
                    1) manage_tickets ;;
                    2) show_documentation ;;
                    3) show_contact_info ;;
                    0) break ;;
                    *) echo "Opción inválida" ;;
                esac
            done
            ;;
    esac
done

```

```

        esac
    done
    ;;
0)
    exit 0
    ;;
*)
    echo "Opción inválida"
    ;;
esac
done

```

Conclusión

Este script shell implementa un sistema de menús interactivo que organiza las funciones de administración y soporte técnico en submenús accesibles. Proporciona una estructura navegable para tareas comunes como gestión de usuarios, bases de datos y tickets, sirviendo como una primera versión funcional para la interacción de diferentes roles en el sistema.

7.4 Menú para el Operador del Centro de Cómputos (Administrador del Sistema), 1ra. versión.

Dentro del script proporcionado anteriormente, se implementa el menú para el Operador del Centro de Cómputos, representado por el Administrador del Sistema. Esta primera versión navegable incluye las funciones clave necesarias para la gestión del sistema y la administración eficiente de recursos.

Funcionalidades Principales

1. **Gestión de Usuarios (manage_users):** Permite agregar, modificar, eliminar y listar usuarios en el sistema. También incluye funciones para cambiar contraseñas y permisos.
2. **Configuración del Sistema (system_config):** Facilita la actualización del software del sistema, configuración de la zona horaria, y verificación del uso del espacio en disco y memoria.
3. **Registro de Actividad (activity_log):** Permite ver registros del sistema, registros de autenticación y errores, así como realizar búsquedas específicas en los logs del sistema.

Código del Script

```
#!/bin/bash

clear_screen() {
    clear
}

show_admin_menu() {
    clear_screen
    echo "==== Menú del Administrador del Sistema ====="
    echo "1. Gestión de usuarios"
    echo "2. Configuración del sistema"
    echo "3. Registro de actividad"
    echo "0. Salir"
    echo "===== "
    read -p "Seleccione una opción: " option
    echo
    return $option
}

manage_users() {
    while true; do
        clear_screen
        echo "==== Gestión de usuarios ====="
        echo "1. Agregar usuario"
        echo "2. Modificar usuario"
        echo "3. Eliminar usuario"
        echo "4. Listar usuarios"
        echo "5. Cambiar contraseña de usuario"
        echo "6. Cambiar permisos de usuario"
        echo "0. Volver al menú anterior"
        echo "===== "
        read -p "Seleccione una opción: " user_option
        echo

        case $user_option in
            1)

```

```

        read -p "Ingrese el nombre del nuevo
usuario: " new_user
        sudo useradd -m "$new_user"
        sudo passwd "$new_user"
        echo "Usuario $new_user agregado."
        ;;
2)
        read -p "Ingrese el nombre del usuario a
modificar: " mod_user
        read -p "Ingrese el nuevo grupo para el
usuario: " new_group
        sudo usermod -aG "$new_group" "$mod_user"
        echo "Usuario $mod_user modificado."
        ;;
3)
        read -p "Ingrese el nombre del usuario a
eliminar: " del_user
        sudo userdel -r "$del_user"
        echo "Usuario $del_user eliminado."
        ;;
4)
        echo "Lista de usuarios:"
        cut -d: -f1 /etc/passwd
        ;;
5)
        read -p "Ingrese el nombre del usuario para
cambiar contraseña: " user
        sudo passwd "$user"
        ;;
6)
        read -p "Ingrese el nombre del usuario para
cambiar permisos: " user
        read -p "Ingrese el nuevo grupo para el
usuario: " group
        sudo usermod -aG "$group" "$user"
        ;;
0)
        break

```

```

        ;;
    *)
        echo "Opción inválida"
        ;;
    esac
    read -n 1 -s -r -p "Presione cualquier tecla para
continuar"
done
}

```

```

system_config() {
    while true; do
        clear_screen
        echo "==== Configuración del sistema ====="
        echo "1. Actualizar software del sistema"
        echo "2. Configurar zona horaria"
        echo "3. Verificar espacio en disco"
        echo "4. Verificar uso de memoria"
        echo "5. Configurar red"
        echo "0. Volver al menú anterior"
        echo "===== "
        read -p "Seleccione una opción: " config_option
        echo

        case $config_option in
            1)
                sudo apt update && sudo apt upgrade -y
                echo "Sistema actualizado."
                ;;
            2)
                sudo dpkg-reconfigure tzdata
                echo "Zona horaria actualizada."
                ;;
            3)
                df -h
                ;;
            4)
                free -h

```

```

        ;;
    5)
        sudo nano /etc/network/interfaces
        echo "Configuración de red actualizada."
        ;;
    0)
        break
        ;;
    *)
        echo "Opción inválida"
        ;;
esac
read -n 1 -s -r -p "Presione cualquier tecla para
continuar"
done
}

```

```

activity_log() {
    while true; do
        clear_screen
        echo "==== Registro de actividad ====="
        echo "1. Ver registros del sistema"
        echo "2. Ver registros de autenticación"
        echo "3. Ver registros de errores"
        echo "4. Buscar en registros"
        echo "0. Volver al menú anterior"
        echo "===== "
        read -p "Seleccione una opción: " log_option
        echo

        case $log_option in
            1)
                sudo less /var/log/syslog
                ;;
            2)
                sudo less /var/log/auth.log
                ;;
            3)

```



```

        sudo less /var/log/syslog | grep 'error'
        ;;
4)
    read -p "Ingrese término de búsqueda: "
search_term

    sudo grep "$search_term" /var/log/syslog
    ;;
0)
    break
    ;;
*)
    echo "Opción inválida"
    ;;
esac
    read -n 1 -s -r -p "Presione cualquier tecla para
continuar"
done
}

admin_menu() {
    while true; do
        show_admin_menu
        admin_option=$?
        case $admin_option in
            1) manage_users ;;
            2) system_config ;;
            3) activity_log ;;
            0) exit 0 ;;
            *) echo "Opción inválida" ;;
        esac
    done
}
admin_menu

```

Conclusión

El menú para el Administrador del Sistema permite gestionar diversas funciones clave del sistema a través de un menú interactivo. Las funciones principales incluyen:

- **Gestión de Usuarios:** Agregar, modificar, eliminar y listar usuarios, así como cambiar contraseñas y permisos.
- **Configuración del Sistema:** Actualizar software, configurar la zona horaria y verificar el uso de recursos como disco y memoria.
- **Registro de Actividad:** Ver y buscar en los registros del sistema y de autenticación.

Este script organiza de manera eficiente las tareas administrativas, facilitando la gestión y mantenimiento del sistema para el Operador del Centro de Cómputos.

8. Configuraciones de Red y Servicios

8.1 Configuraciones de red en las terminales y el servidor

Introducción

Este documento detalla la configuración de red implementada en nuestro entorno de servidor y terminales como parte del proyecto de software en curso. El objetivo principal es establecer una conectividad estable y segura entre el servidor y las terminales, facilitando la comunicación eficiente dentro de nuestra infraestructura. Esta configuración es crucial para el funcionamiento óptimo de nuestros sistemas y servicios.

Requisitos previos:

- Acceso al servidor con permisos de administrador
- Acceso a las terminales (clientes) para configuraciones de red
- Instalación de **openssh-server** para la gestión de conexiones remotas

Configuración del servidor

Detalles del servidor

- Sistema operativo: Ubuntu server 24.04 64bits
- Función: Servidor principal para el proyecto de software, gestionando las operaciones del sistema y proporcionando servicios centralizados a las terminales conectadas.

Configuración de red

- Interfaz de red: enp0s3
- Dirección IP: 192.168.0.100
- Máscara de subred: 255.255.255.0
- Puerta de enlace: 192.168.0.1
- Servidores DNS: 8.8.8.8, 8.8.4.4

Archivo de configuración Netplan

A continuación, se detalla el archivo de configuración Netplan el cual define una interfaz Ethernet estática en un servidor Ubuntu. La interfaz enp0s3 se le asigna a la dirección IP estática 192.168.0.100 con una máscara de subred /24, se configura una ruta por defecto a través de la puerta de enlace 192.168.0.1, y se establece DNS de Google (8.8.8.8, 8.8.4.4).

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.0.100/24
      routes:
        - to: default
          via: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Procedimientos de configuración

1. Edición del archivo Netplan:

Para comenzar a editar el archivo netplan, se debe escribir el siguiente comando:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

2. Aplicación de la configuración

Para aplicar los cambios

```
sudo netplan apply
```

3. Verificación de la configuración

ip a

Configuración de las terminales

Las terminales están configuradas para conectarse al servidor a través de una conexión de red puente. Esto permite que las terminales y el servidor estén en la misma red local, facilitando la comunicación directa.

Pruebas de conectividad

- Ping al servidor desde una terminal: `ping 192.168.0.100`
- Ping desde el servidor a la puerta de enlace: `ping 192.168.0.1`
- Ping a un servidor DNS externo: `ping 8.8.8.8`
- Prueba de resolución DNS: `ping google.com`

Todas las pruebas realizadas mostraron 0% de pérdida de paquetes, indicando una conectividad estable y fiable.

Seguridad

- Estado del firewall: `sudo ufw status`
- Puertos abiertos: 22 (SSH)

Mantenimiento y resolución de problemas

Comando para reiniciar el servicio de red:

```
sudo systemctl restart systemd-networkd
```

Backup de configuración

Procedimiento para respaldar la configuración de red:

```
sudo cp /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.yaml.bak
```

Conclusion

La configuración de red implementada proporciona una base sólida para la comunicación entre el servidor Ubuntu 24.04 y las terminales del proyecto. Se ha establecido una conectividad estable y se han realizado pruebas exitosas de ping y resolución DNS. Esta configuración permite un entorno de red robusto para el desarrollo y la operación del proyecto de software en curso. Se recomienda realizar revisiones periódicas de la configuración para asegurar un rendimiento óptimo continuo.

8.2 Configuración del servicio SSH en el cliente y el servidor (v).

Introducción

El presente documento expone de manera clara y detallada los pasos necesarios para configurar el servicio SSH tanto en el servidor como en el cliente. SSH (Secure Shell) es una herramienta clave para establecer conexiones seguras en redes no confiables, proporcionando un entorno de administración remota con protocolos de cifrado sólidos que protegen la información sensible durante la transmisión.

La correcta configuración de SSH garantiza que los accesos a servidores remotos sean seguros y controlados, minimizando riesgos de acceso no autorizado y fortaleciendo la seguridad de la infraestructura tecnológica.

Configuración del servidor SSH

Instalación del servidor SSH

El primer paso para habilitar el acceso remoto, es instalar el servidor SSH en el servidor. Para hacerlo, se escribe el siguiente comando:

```
sudo apt-get update  
sudo apt-get install openssh-server
```

Estos comandos se encargan de actualizar los repositorios de paquetes, y de instalar el paquete necesario para habilitar el servicio SSH.

Verificación del estado del servicio SSH

Luego de realizada la instalación, se debe verificar que el servicio se encuentre activo y funcionando. Para esto, se escribe el siguiente comando:

```
sudo systemctl status ssh
```

Este comando busca informar el estado de SSH. En caso de que el servicio no esté activo, se puede realizar el siguiente comando.

```
sudo systemctl start ssh
```

Configuración del archivo sshd_config

El archivo de configuración del servidor SSH, se encuentra localizado en la ruta `/etc/ssh/sshd_config`. Este archivo contiene diversas opciones de personalización de seguridad y funcionamiento, tales como el puerto en que escucha el servidor, la habilitación o deshabilitación de la autenticación por clave pública, y la limitación de accesos por usuarios.

para editar el archivo, se debe de ejecutar:

```
sudo nano /etc/ssh/sshd_config
```

Luego, se recomienda los siguientes ajustes en el archivo

Puerto de escucha (es el puerto predeterminado)
`Port 22`

Permitir solo autenticación co
`PasswordAuthentication no`
`ChallengeResponseAuthentication no`

Permitir solo ciertos usuarios (opcional)
`AllowUsers usuario1 usuario2`

Deshabilitar el acceso del usuario root (opcional)
`PermitRootLogin no`

Reglas de cifrado recomendadas
`Ciphers aes256-ctr,aes192-ctr,aes128-ctr`

Reinicio del servicio SSH

Luego de realizar los cambios en el archivo de configuración, se necesita reiniciar el servicio para aplicar los cambios. Para ello, se escribe el siguiente comando:

```
sudo systemctl restart ssh
```

Este comando aplicará la configuración establecida, y reiniciará el servicio.

Verificación del funcionamiento del servicio

Para asegurarse de que el servicio esté en correcto funcionamiento, se deberá de seguir los siguientes pasos:

1. Comprobar el estado del servicio:

```
sudo systemctl status ssh
```

Se verifica que el estado del servicio esté en “active (running)”.

2. Verificar que el puerto de SSH esté escuchando:

```
sudo netstat -tuln | grep 22
```

Este comando verifica que el puerto 22 está escuchando, lo cual indica si el servicio está aceptando conexiones.

Conclusión

La configuración adecuada del servicio SSH tanto en el servidor como en el cliente es un paso fundamental para garantizar conexiones remotas seguras y eficaces. El proceso de instalación, verificación y ajuste de las configuraciones básicas del servicio, como el puerto de escucha y las reglas de autenticación, permite un control riguroso sobre los accesos al servidor, mitigando riesgos y fortaleciendo la seguridad de la infraestructura tecnológica.

SSH se posiciona como una herramienta clave en la administración de sistemas, especialmente en entornos donde la seguridad de la información es prioritaria. La correcta implementación de este protocolo contribuye al robustecimiento de la infraestructura de red y al aseguramiento de la comunicación remota.

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

Introducción

Esta documentación describe la configuración y funcionamiento del sistema de backups automáticos para la base de datos en el servidor Ubuntu. El objetivo es garantizar la seguridad y disponibilidad de los datos mediante la creación y almacenamiento de copias de seguridad periódicas.

Configuración del entorno

Requisitos

- **Sistema Operativo:** Ubuntu Server 24.04
- **Base de Datos:** MySQL
- **Usuario de MySQL:** root
- **Directorio de Backup:** `/var/lib/backup`
- **Script de Backup:** `backup.sh`

Configuración del directorio del backup

```
sudo mkdir -p /var/lib/backup
sudo chown $USER:$USER /var/lib/backup
sudo chmod 755 /var/lib/backup
```

Creación del script de backup

Ubicación del script: `/var/lib/backup/backup.sh`

Contenido del script:

```
#!/bin/bash

BACKUP_DIR="/var/lib/backup"
DATE=$(date +%Y%m%d%H%M)
DB_NAME="aio_db"
DB_USER="root"
DB_PASSWORD=""
BACKUP_FILE="$BACKUP_DIR/db_backup_$DATE.sql"

mysqldump -u $DB_USER -p$DB_PASSWORD $DB_NAME > $BACKUP_FILE
```



```
if [ $? -eq 0 ]; then
    echo "Backup realizado con éxito: $BACKUP_FILE"
else
    echo "Error en el backup"
fi
```

Permisos para el backup:

```
sudo chmod +x /var/lib/backup/backup.sh
```

Configuracion de cron

Creacion de la tarea cron

Primero, se debe editar Crontab

```
crontab -e
```

Se designará un backup diario a las 2 AM

```
0 2 * * * /var/lib/backup/backup.sh
```

Verificacion del servicio de cron

Verificacion del estado del servicio

```
sudo systemctl status cron
```

Verificacion y pruebas

Prueba de ejecucion manual del script

```
/var/lib/backup/backup.sh
```

Con este comando se verifica que el archivo de backup se cree correctamente en la ubicación.

Revision de logs de cron

```
sudo less /var/log/syslog
```

Con este comando se permite asegurar que el script se ejecute segun lo programado.

Verificación de archivos de backup

```
ls -l /var/lib/backup/
```

Confirma la existencia y la creación de los archivos de backup.

Mantenimiento y recuperación

- **Pruebas de Restauración:** Realizar pruebas periódicas de restauración de backups para validar la integridad de los archivos de respaldo.

Notificaciones

Agregar Notificaciones por Correo Electrónico:

El script de backup incluye una funcionalidad para enviar notificaciones por correo electrónico, que permite alertar al usuario sobre el resultado del backup. Estas notificaciones son útiles para informar tanto los éxitos como los errores del proceso de respaldo, asegurando que el administrador esté al tanto del estado de los backups sin necesidad de revisar manualmente.

```
if [ $? -eq 0 ]; then
    echo "Backup realizado con éxito: $BACKUP_FILE" | mail -s "Backup Exitoso"
    gcarsillo@scuolaitaliana.edu.uy
else
    echo "Error en el backup" | mail -s "Error en Backup"
    gcarsillo@scuolaitaliana.edu.uy
fi
```

Conclusion

La implementación de un sistema de backups automáticos es crucial para asegurar la disponibilidad y la integridad de los datos en el servidor. La configuración detallada en esta documentación proporciona una solución robusta que incluye la creación de backups periódicos, verificación del servicio de cron y notificaciones por correo electrónico. Mantener el sistema de backups funcionando correctamente es esencial para prevenir la pérdida de datos y garantizar una recuperación efectiva en caso de fallos. Es recomendable revisar y ajustar las configuraciones según las necesidades específicas y realizar pruebas periódicas para asegurar la fiabilidad del sistema de backups.

8.4 Instalación y configuración del servicio web http

Introducción:

En el ámbito de la administración de sistemas, la instalación y configuración de un servidor web es una tarea fundamental para ofrecer servicios web a través de la red. Apache, de gran conocimiento mundialmente, es uno de los servidores web más utilizados en el mundo, conocido por su robustez y flexibilidad. A continuación, se procede a brindar una guía detallada del proceso de instalación y configuración del servicio web http.

Actualización del sistema

Antes de comenzar la instalación, se debe asegurar que el sistema se encuentre actualizado.

```
sudo apt update # Actualiza la lista de paquetes disponibles  
sudo apt upgrade -y # Actualiza todos los paquetes instalados a la última versión
```

Instalación de Apache

A continuación se procede con la instalación del servidor apache.

```
sudo apt install apache2 -y # Instala el servidor apache a su última versión disponible
```

Verificar la instalación

Luego de la instalación, apache debería de comenzar a ejecutarse automáticamente, para verificar su estado, puede hacerse con:

```
sudo systemctl status apache2 # Verifica el estado de apache2
```

Verificar la instalación desde un navegador

Puede verificarse la instalación desde un navegador web, introduciendo la dirección ip del servidor en la URL o a localhost si estás trabajando en el mismo servidor.

```
http://192.168.1.100 # Dirección hipotética del servidor
```

Si todo es correcto, se debería de ver la página de bienvenida de apache.

Configuración de Apache

Configuración de Archivos Virtuales (Virtual Hosts)

1. Crear un nuevo archivo de configuración para el sitio

Crea un archivo de configuración para la página web en /etc/netplan/sites-available/.

```
sudo nano /etc/apache2/sites-available/aio.conf # Crea un archivo de configuración para la página web aio.conf
```

2. Agregar la configuración básica

Debe de ponerse algo como:

```
<VirtualHost *:80>
    ServerAdmin webmaster@aio.com
    ServerName aio.com
    ServerAlias www.aio.com
    DocumentRoot /var/www/aio
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

3. Crear el directorio para la página web

Se crea el directorio donde se aloja el sitio web.

```
sudo mkdir -p /var/www/aio
```

4. Hacer que apache use el nuevo archivo de configuración

Para el siguiente paso, se procede a habilitar el nuevo archivo de configuración y deshabilitar el archivo por defecto.

```
sudo a2ensite aio.conf
sudo a2dissite 000-default.conf
```

5. Reiniciar apache

Se aplican los cambios reiniciando apache.

```
sudo systemctl restart apache2
```

Verificar la configuración

Para verificar la configuración, se coloca un archivo index.html en el directorio /var/www/aio para verificar que el servidor esté sirviendo el contenido correcto.

```
sudo nano /var/www/aio/index.html
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Aio</title>
</head>
<body>
  <h1>¡Bienvenido al sitio web de Aio!</h1>
</body>
</html>
```

Luego de esto, guarde y cierre el archivo. Navegue a <http://aio.com> (o la dirección ip del servidor) en el navegador.

Hacer el servicio HTTP arrancar al inicio

Es recomendable configurar Apache para que inicie automáticamente cuando el servidor se reinicie:

```
sudo systemctl enable apache2 # Configurar Apache para arrancar automáticamente
```

Conclusión

Con esto, se ha completado la instalación y configuración básica del servidor web Apache. Ahora el servidor está listo para servir contenido web y puede personalizarse más según las necesidades que se presenten.

8.5 Instalación y configuración de la base de datos MySQL.

Introducción

MySQL es uno de los sistemas de gestión de bases de datos relacionales más populares en el mundo, utilizado ampliamente en aplicaciones web y empresariales por su alto rendimiento, escalabilidad y fiabilidad. En este documento, se detalla el proceso de instalación y configuración de MySQL en un

servidor Ubuntu, así mismo, los pasos para la creación de la base de datos y su configuración respectivamente para audio.

Actualizar el sistema

Al momento de realizar una instalación, se recomienda actualizar el sistema para evitar problemas de compatibilidad.

```
sudo apt update  
sudo apt upgrade -y
```

Instalación de MySQL

Instalación del servidor MySQL.

```
sudo apt install mysql-server -y
```

Esto instalará MySQL y lo iniciará automáticamente.

Verificar el estado del servicio MySQL

Luego de la instalación, mysql debería de iniciarse automáticamente, para comprobarlo, puede hacerse de la siguiente manera:

```
sudo systemctl status mysql
```

Ejecutar el script de seguridad de MySQL

MySQL ofrece un script llamado `mysql_secure_installation` que ayuda a asegurar la instalación eliminando configuraciones por defecto inseguras. Para ejecutarlo, usa:

```
sudo mysql_secure_installation
```

Opciones del script de seguridad:

1. Configurar una contraseña para el usuario root.
2. Eliminar el acceso remoto del usuario root.
3. Eliminar usuarios anónimos.
4. Eliminar la base de datos de prueba.
5. Recargar las tablas de privilegios.

Verificar el correcto funcionamiento de MySQL

Para verificar que MySQL está funcionando correctamente y que el nuevo usuario pueda conectarse a la base de datos, se hace mediante el siguiente comando:

```
mysql -u usuario -p
```

Se pedirá la contraseña asignada al usuario.

Creación de la base de datos de AIO

Para crear la base de datos de audio, se proporciona el siguiente código, actualizado a su última versión.

```
-- CREACION DE TABLAS
```

```
-- Cliente
```

```
CREATE TABLE cliente (  
    id INT NOT NULL,  
    ci INT NOT NULL UNIQUE,  
    email VARCHAR(63) NOT NULL UNIQUE,  
    nombre VARCHAR(23) NOT NULL,  
    apellido VARCHAR(23) NOT NULL,  
    telefono INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
-- Vehiculo
```

```
CREATE TABLE vehiculo (  
    matricula VARCHAR(8) NOT NULL,  
    marca VARCHAR(32) NULL,  
    modelo VARCHAR(32) NULL,  
    tipo ENUM('auto', 'moto', 'camioneta', 'camion', 'utilitario') NOT NULL,  
    color VARCHAR(6) NULL,  
    PRIMARY KEY (matricula)  
);
```

```
-- tiene
```

```
CREATE TABLE tiene (  
    id_cliente INT NOT NULL,  
    matricula VARCHAR(8) NOT NULL,  
    PRIMARY KEY (id_cliente, matricula),  
    FOREIGN KEY (id_cliente) REFERENCES cliente(id),  
    FOREIGN KEY (matricula) REFERENCES vehiculo(matricula)  
);
```

```

-- Producto
CREATE TABLE producto (
    id INT NOT NULL,
    upc VARCHAR(13) NOT NULL UNIQUE,
    precio DECIMAL(10,2) NOT NULL,
    marca VARCHAR(23) NOT NULL,
    fecha_creacion DATETIME NOT NULL,
    stock INT NOT NULL,
    PRIMARY KEY (id)
);

-- Neumatico
CREATE TABLE neumatico (
    id_producto INT NOT NULL,
    tamano VARCHAR(16) NOT NULL,
    modelo VARCHAR(23) NOT NULL,
    tipo CHAR(2) NOT NULL,
    PRIMARY KEY (id_producto),
    FOREIGN KEY (id_producto) REFERENCES producto(id)
);

-- Otro_producto
CREATE TABLE otro_producto (
    id_producto INT NOT NULL,
    nombre VARCHAR(63),
    PRIMARY KEY (id_producto),
    FOREIGN KEY (id_producto) REFERENCES producto(id)
);

-- Transaccion
CREATE TABLE transaccion (
    id_producto INT NOT NULL,
    cantidad INT NOT NULL,
    tipo ENUM('ingreso', 'egreso'),
    fecha DATETIME NOT NULL,
    PRIMARY KEY (id_producto),
    FOREIGN KEY (id_producto) REFERENCES producto(id)
);

-- Servicio
CREATE TABLE servicio (
    id INT NOT NULL,
    matricula VARCHAR(8) NOT NULL,

```



```

    precio DECIMAL(10,2) NOT NULL,
    fecha_inicio DATETIME NOT NULL,
    fecha_final DATETIME NOT NULL,
    estado ENUM('pendiente', 'realizado', 'cancelado'),
    PRIMARY KEY (id),
    FOREIGN KEY (matricula) REFERENCES vehiculo(matricula)
);

-- Taller
CREATE TABLE taller (
    id_servicio INT NOT NULL,
    tipo VARCHAR(3) NOT NULL,
    descripcion TEXT NOT NULL,
    tiempo_estimado INT NOT NULL,
    diagnostico TEXT NULL,
    PRIMARY KEY (id_servicio),
    FOREIGN KEY (id_servicio) REFERENCES servicio(id)
);

-- Parking
CREATE TABLE parking (
    id_servicio INT NOT NULL,
    largo_plazo BOOLEAN NOT NULL,
    tipo_plaza ENUM('auto', 'moto'),
    PRIMARY KEY (id_servicio),
    FOREIGN KEY (id_servicio) REFERENCES servicio(id)
);

-- Numero de plaza
CREATE TABLE numero_plaza (
    numero_plaza INT NOT NULL,
    id_servicio INT NOT NULL,
    PRIMARY KEY (numero_plaza, id_servicio),
    FOREIGN KEY (id_servicio) REFERENCES parking(id_servicio)
);

-- Ejecutivo
CREATE TABLE ejecutivo (
    id_empleado INT NOT NULL,
    PRIMARY KEY (id_empleado)
);

-- realiza
CREATE TABLE realiza (

```

```

    id_empleado INT NOT NULL,
    id_servicio INT NOT NULL,
    PRIMARY KEY (id_empleado, id_servicio),
    FOREIGN KEY (id_empleado) REFERENCES ejecutivo(id_empleado),
    FOREIGN KEY (id_servicio) REFERENCES taller(id_servicio)
);

-- Orden
CREATE TABLE orden (
    id INT NOT NULL,
    id_cliente INT NOT NULL,
    total DECIMAL(10,2) NOT NULL,
    fecha_orden DATETIME NOT NULL,
    estado_pago ENUM('no pago', 'pago', 'cancelado') NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_cliente) REFERENCES cliente(id)
);

-- Detalle de orden de servicio
CREATE TABLE detalle_orden_servicio (
    id_servicio INT NOT NULL,
    id_orden INT NOT NULL,
    PRIMARY KEY (id_servicio),
    FOREIGN KEY (id_servicio) REFERENCES servicio(id),
    FOREIGN KEY (id_orden) REFERENCES orden(id)
);

-- Detalle de orden de producto
CREATE TABLE detalle_orden_producto (
    id_producto INT NOT NULL,
    id_orden INT NOT NULL,
    cantidad INT NOT NULL,
    PRIMARY KEY (id_producto, id_orden),
    FOREIGN KEY (id_producto) REFERENCES producto(id),
    FOREIGN KEY (id_orden) REFERENCES orden(id)
);

```

Iniciar MySQL Automaticamente al arrancar

Para aplicar la configuracion de inicio automatico de MySQL cada vez que se inicie el servidor, se hace con el siguiente comando:

```
sudo systemctl enable mysql
```

Conclusion

En este documento se cubrieron los pasos necesarios para la instalación y configuración básica de MySQL en Ubuntu, además de la creación de una base de datos con sus respectivas tablas. Estos pasos forman la base para gestionar la información del sistema AIO de forma segura y eficiente.

9. Scripts de Gestión y Consultas a la Base de Datos

Introducción

Este script es una interfaz de línea de comandos para interactuar con una base de datos de gestión de servicios automotrices. Permite a los usuarios con diferentes roles acceder a consultas específicas sobre clientes, vehículos, productos, servicios y transacciones.

Requisitos

- **Sistema Operativo:** Ubuntu Server 24.04
- **Base de Datos:** MySQL
- **Base de Datos:** `aio_db`
- **Usuario de MySQL:** `root`
- **Contraseña de MySQL:** (vacía)

9.2 Menú principal y funcionalidades

El script ofrece un menú basado en roles para acceder a consultas SQL específicas:

1. **Administrador de Sistemas**
 - Listar clientes, servicios, órdenes y productos.
 - Consultar detalles específicos de clientes, servicios, órdenes y productos.
2. **Gerente**
 - Consultar informes de ventas, reservas y disponibilidad de servicios.
3. **Jefe de Servicios de Diagnóstico**
 - Consultar citas programadas, datos de citas específicas, historial de diagnóstico y número de diagnósticos realizados.
4. **Jefe de Servicios de Alineación y Balanceo**
 - Consultar número de alineaciones, balanceos, servicios completos y estado del vehículo.

5. Ejecutivo de Servicios

- Consultar servicios realizados, programados y la agenda de servicios.

6. Cajero

- Consultar transacciones, ingresos diarios y mensuales.

7. Operador de Respaldos

- Consultar estado e historial de respaldos (aquí se puede agregar más funcionalidad según sea necesario).

8. Cliente

- Consultar reservas realizadas, historial de servicios y costo total de servicios.

Estructura del script

El script está estructurado en varias funciones, cada una correspondiente a un menú específico basado en el rol del usuario. La función principal (`while true`) permite al usuario seleccionar un rol y acceder al menú adecuado.

Funciones:

run_query: Ejecuta una consulta SQL proporcionada.

admin_menu: Menú para el Administrador de Sistemas.

manager_menu: Menú para el Gerente.

diagnostic_menu: Menú para el Jefe de Servicios de Diagnóstico.

alignment_menu: Menú para el Jefe de Servicios de Alineación y Balanceo.

executive_menu: Menú para el Ejecutivo de Servicios.

cashier_menu: Menú para el Cajero.

backup_operator_menu: Menú para el Operador de Respaldos.

client_menu: Menú para el Cliente.

9.3 Ejecución y ejemplos de código

1. Ejecutar el script

```
./gestion_de_informes.sh
```

2. Selección de rol

Introduzca el número correspondiente según el rol al que desee ingresar.

3. Selección de consulta

Una vez dentro del menú del rol seleccionado, introduzca el número correspondiente a la consulta deseada.

4. Salir

Presione la tecla 0 para salir de cualquier menú.

Ejemplo de código

```
#!/bin/bash
```

```
DB_NAME="aio_db"
```

```
DB_USER="root"
```

```
DB_PASS=""
```

```
function run_query {  
    local query="$1"  
    mysql -u"$DB_USER" -p"$DB_PASS" "$DB_NAME" -e "$query"  
}
```

```
function admin_menu {  
    while true; do  
        clear  
        echo "Administrador de sistemas"  
        hecho "1. Listar todos los clientes"  
        echo "2. Consultar detalles de un cliente específico"  
        echo "..."  
        read -p "Selecciona una opción: " option  
  
        case $option in  
            1) run_query "SELECT * FROM cliente LIMIT 7";;  
            2) read -p "Introduce el ID del cliente: " id; run_query "SELECT * FROM  
cliente WHERE id=$id LIMIT 7";;  
            ...  
            0) break;;  
            *) echo "Opción no válida";;  
        esac  
        read -p "Presiona Enter para continuar..."  
    done  
}
```

```
# Menú principal
```

```
while true; do
```

```

clear
echo "Sistema de Gestión"
echo "1. Administrador de sistemas"
echo "2. Gerente"
echo "..."
read -p "Selecciona un rol: " role

case $role in
    1) admin_menu;;
    2) manager_menu;;
    ...
    0) exit;;
    *) echo "Opción no válida";;
esac
done

```

5. Código final

```

#!/bin/bash

DB_NAME="aio_db"
DB_USER="root"
DB_PASS=""

function run_query {
    local query="$1"
    mysql -u"$DB_USER" -p"$DB_PASS" "$DB_NAME" -e "$query"
}

function admin_menu {
    while true; do
        clear
        echo "Administrador de sistemas"
        echo "1. Listar todos los clientes"
        echo "2. Consultar detalles de un cliente específico"
        echo "3. Consultar todos los servicios"
        echo "4. Consultar detalles de un servicio específico"
        echo "5. Consultar todas las órdenes"
        echo "6. Consultar detalles de una orden específica"
        echo "7. Consultar todos los productos"
        echo "8. Consultar detalles de un producto específico"
        echo "0. Salir"
    done
}

```

```

read -p "Selecciona una opción: " option

case $option in
    1) run_query "SELECT * FROM cliente LIMIT 7";;
    2) read -p "Introduce el ID del cliente: " id; run_query "SELECT * FROM
cliente WHERE id=$id LIMIT 7";;
    3) run_query "SELECT * FROM servicio LIMIT 7";;
    4) read -p "Introduce el ID del servicio: " id; run_query "SELECT * FROM
servicio WHERE id=$id LIMIT 7";;
    5) run_query "SELECT * FROM orden LIMIT 7";;
    6) read -p "Introduce el ID de la orden: " id; run_query "SELECT * FROM
orden WHERE id=$id LIMIT 7";;
    7) run_query "SELECT * FROM producto LIMIT 7";;
    8) read -p "Introduce el ID del producto: " id; run_query "SELECT * FROM
producto WHERE id=$id LIMIT 7";;
    0) break;;
    *) echo "Opción no válida";;
esac
read -p "Presiona Enter para continuar..."
done
}

```

```

function manager_menu {
while true; do
clear
echo "Gerente"
echo "1. Consultar informes de ventas por período"
echo "2. Consultar ventas por tipo de servicio"
echo "3. Consultar ventas por cliente"
echo "4. Consultar reservas por cliente"
echo "5. Consultar reservas por período"
echo "6. Consultar disponibilidad de los servicios"
echo "7. Consultar el rendimiento de cada servicio"
echo "0. Salir"
read -p "Selecciona una opción: " option

case $option in
    1) echo "No hay informes predefinidos para ventas por período";;
    2) echo "No hay datos suficientes para ventas por tipo de servicio";;
    3) run_query "SELECT id_cliente, COUNT(*) AS ventas FROM orden
GROUP BY id_cliente LIMIT 7";;
    4) run_query "SELECT id_cliente, COUNT(*) AS reservas FROM orden
GROUP BY id_cliente LIMIT 7";;

```

```

        5) run_query "SELECT fecha_orden, COUNT(*) AS reservas FROM orden
GROUP BY fecha_orden LIMIT 7";;
        6) run_query "SELECT matricula, COUNT(*) AS disponibilidad FROM
servicio GROUP BY matricula LIMIT 7";;
        7) run_query "SELECT id, COUNT(*) AS rendimiento FROM servicio
GROUP BY id LIMIT 7";;
        0) break;;
        *) echo "Opción no válida";;
    esac
    read -p "Presiona Enter para continuar..."
done
}

```

```

function diagnostic_menu {
    while true; do
        clear
        echo "Jefe de Servicios de Diagnóstico"
        echo "1. Consultar todas las citas de diagnóstico programadas"
        echo "2. Consultar datos de una cita específica"
        echo "3. Consultar el historial de diagnóstico de un cliente"
        echo "4. Consultar el número de diagnósticos realizados"
        echo "0. Salir"
        read -p "Selecciona una opción: " option

        case $option in
            1) run_query "SELECT * FROM servicio LIMIT 7";;
            2) read -p "Introduce el ID del servicio: " id; run_query "SELECT * FROM
servicio WHERE id=$id LIMIT 7";;
            3) read -p "Introduce el ID del cliente: " id; run_query "SELECT * FROM
servicio WHERE matricula IN (SELECT matricula FROM tiene WHERE
id_cliente=$id) LIMIT 7";;
            4) run_query "SELECT COUNT(*) AS numero_diagnosticos FROM
servicio LIMIT 7";;
            0) break;;
            *) echo "Opción no válida";;
        esac
        read -p "Presiona Enter para continuar..."
    done
}

```

```

function alignment_menu {
    while true; do
        clear
        echo "Jefe de Servicios de Alineación y Balanceo"

```



```

echo "1. Consultar el número de alineaciones realizadas"
echo "2. Consultar el número de balanceos realizados"
echo "3. Consultar el número de servicios completos realizados"
echo "4. Consultar el estado del vehículo antes y después de los servicios"
echo "0. Salir"
read -p "Selecciona una opción: " option

case $option in
    1) run_query "SELECT tipo, COUNT(*) AS cantidad FROM taller WHERE
tipo='AL' GROUP BY tipo LIMIT 7";;
    2) run_query "SELECT tipo, COUNT(*) AS cantidad FROM taller WHERE
tipo='BL' GROUP BY tipo LIMIT 7";;
    3) run_query "SELECT COUNT(*) AS servicios_completos FROM taller
WHERE tipo='AL' OR tipo='BL' LIMIT 7";;
    4) read -p "Introduce el ID del servicio: " id; run_query "SELECT matricula
FROM servicio WHERE id=$id LIMIT 7";;
    0) break;;
    *) echo "Opción no válida";;
esac
read -p "Presiona Enter para continuar..."
done
}

function executive_menu {
while true; do
clear
echo "Ejecutivo de Servicios"
echo "1. Consultar servicios realizados"
echo "2. Consultar servicios programados"
echo "3. Consultar la agenda de servicios"
echo "0. Salir"
read -p "Selecciona una opción: " option

case $option in
    1) run_query "SELECT * FROM servicio WHERE estado='realizado' LIMIT
7";;
    2) run_query "SELECT * FROM servicio WHERE estado='pendiente'
LIMIT 7";;
    3) run_query "SELECT * FROM servicio LIMIT 7";;
    0) break;;
    *) echo "Opción no válida";;
esac
read -p "Presiona Enter para continuar..."
done
}

```

```

}

function cashier_menu {
    while true; do
        clear
        echo "Cajero"
        echo "1. Consultar todas las transacciones"
        echo "2. Consultar ingresos por día"
        echo "3. Consultar ingresos por mes"
        echo "0. Salir"
        read -p "Selecciona una opción: " option

        case $option in
            1) run_query "SELECT * FROM transaccion LIMIT 7";;
            2) run_query "SELECT DATE(fecha) AS fecha, SUM(cantidad) AS
ingresos FROM transaccion GROUP BY DATE(fecha) LIMIT 7";;
            3) run_query "SELECT DATE_FORMAT(fecha, '%Y-%m') AS mes,
SUM(cantidad) AS ingresos FROM transaccion GROUP BY
DATE_FORMAT(fecha, '%Y-%m') LIMIT 7";;
            0) break;;
            *) echo "Opción no válida";;
        esac
        read -p "Presiona Enter para continuar..."
    done
}

function backup_operator_menu {
    while true; do
        clear
        echo "Operador de Respaldos"
        echo "1. Consultar estado de los respaldos"
        echo "2. Consultar historial de respaldos"
        echo "0. Salir"
        read -p "Selecciona una opción: " option

        case $option in
            1) echo "No hay datos predefinidos para el estado de los respaldos";;
            2) echo "No hay datos predefinidos para el historial de respaldos";;
            0) break;;
            *) echo "Opción no válida";;
        esac
        read -p "Presiona Enter para continuar..."
    done
}

```

```

function client_menu {
    while true; do
        clear
        echo "Cliente"
        echo "1. Consultar reservas realizadas"
        echo "2. Consultar historial de servicios"
        echo "3. Consultar el costo total de los servicios"
        echo "0. Salir"
        read -p "Selecciona una opción: " option

        case $option in
            1) read -p "Introduce tu ID de cliente: " id; run_query "SELECT * FROM
orden WHERE id_cliente=$id LIMIT 7";;
            2) read -p "Introduce tu ID de cliente: " id; run_query "SELECT * FROM
servicio WHERE matricula IN (SELECT matricula FROM tiene WHERE
id_cliente=$id) LIMIT 7";;
            3) read -p "Introduce tu ID de cliente: " id; run_query "SELECT SUM(total)
AS costo_total FROM orden WHERE id_cliente=$id LIMIT 7";;
            0) break;;
            *) echo "Opción no válida";;
        esac
        read -p "Presiona Enter para continuar..."
    done
}

while true; do
    clear
    echo "Sistema de Gestión"
    echo "1. Administrador de sistemas"
    echo "2. Gerente"
    echo "3. Jefe de Servicios de Diagnóstico"
    echo "4. Jefe de Servicios de Alineación y Balanceo"
    echo "5. Ejecutivo de Servicios"
    echo "6. Cajero"
    echo "7. Operador de RespalDOS"
    echo "8. Cliente"
    echo "0. Salir"
    read -p "Selecciona un rol: " role

    case $role in
        1) admin_menu;;
        2) manager_menu;;
        3) diagnostic_menu;;
    esac
done

```

```
4) alignment_menu;;
5) executive_menu;;
6) cashier_menu;;
7) backup_operator_menu;;
8) client_menu;;
0) exit;;
*) echo "Opción no válida";;
esac
done
```

El resultado de cada consulta, se encuentra limitado a una respuesta de 7 valores. Esto se hizo con la finalidad de evitar largas filas de datos y optimizar el funcionamiento para una mayor compresión y legibilidad.

Conclusión

Este script proporciona una solución flexible y eficaz para la gestión de servicios automotrices, permitiendo una interacción simplificada con la base de datos a través de un menú basado en roles. La estructura modular del script asegura que sea fácil de mantener y actualizar, permitiendo a los administradores adaptar el sistema a nuevas necesidades o consultas a medida que evolucionen los requisitos del negocio. Con una interfaz intuitiva y consultas bien definidas, el script facilita la toma de decisiones y la gestión eficiente de la información.

10. Menú Integrado de Gestión de Scripts

Introducción

Este script proporciona un menú principal que facilita la gestión y el acceso a todos los scripts desarrollados en las secciones anteriores. El objetivo es centralizar y simplificar la interacción con los diferentes componentes del sistema, permitiendo a los usuarios acceder de manera rápida y sencilla a cada una de las funcionalidades configuradas, como la gestión de la base de datos, la edición de crontabs, la creación de usuarios y grupos, las modificaciones de la red, y los backups automáticos.

10.1 Funcionalidad del Menú Principal

El menú principal conecta y facilita el acceso a los siguientes scripts:

- **Script de Gestión de la Base de Datos:** Permite realizar consultas y administrar los registros en la base de datos `aio_db`.

- **Script de Gestión de Usuarios y Grupos:** Facilita la creación, modificación, y eliminación de usuarios y grupos en el sistema.
- **Script de Backups Automáticos:** Realiza copias de seguridad de la base de datos y otros archivos importantes.
- **Script de Configuración de Red:** Administra la configuración de red y servicios en el servidor.
- **Gestión de Crontabs:** Administra las rutinas de backup y permite la configuración y revisión de tareas programadas.

10.2 Código del Script Principal

```
#!/bin/bash

BD_SCRIPT="/home/aio/segunda_entrega/ix./gestion_de_informes
.sh"
USUARIOS_SCRIPT="/home/aio/segunda_entrega/ii./menu_usuarios
.sh"
RED_CONFIG="/etc/netplan/50-cloud-init.yaml"
BACKUP_SCRIPT="/var/lib/backup/backup.sh"

limpiar_pantalla() {
    clear
}

mostrar_menu() {
    echo "==== Menú Principal ====="
    echo "1. Acceder al script de base de datos"
    echo "2. Acceder al script de crontab"
    echo "3. Acceder al script de creación de usuarios y
grupos"
    echo "4. Modificar la configuración de red"
    echo "5. Acceder al script de backup"
    echo "0. Salir"
    echo "===== "
    echo -n "Elige una opción: "
}

ejecutar_opcion() {
    case $1 in
        1)
```

```

        limpiar_pantalla
        bash "$BD_SCRIPT"
        sleep 2
        ;;
2)
    limpiar_pantalla
    echo "Abriendo crontab para editar..."
    sleep 2
    crontab -e
    ;;
3)
    limpiar_pantalla
    bash "$USUARIOS_SCRIPT"
    sleep 2
    ;;
4)
    limpiar_pantalla
    echo "Abriendo la configuración de red para
editar..."
    sleep 2
    sudo nano "$RED_CONFIG"
    ;;
5)
    limpiar_pantalla
    bash "$BACKUP_SCRIPT"
    sleep 2
    ;;
0)
    limpiar_pantalla
    echo "Saliendo..."
    exit 0
    ;;
*)
    limpiar_pantalla
    echo "Opción no válida, intenta de nuevo."
    ;;
esac
}

```

```
while true; do
    mostrar_menu
    read -r opcion
    ejecutar_opcion "$opcion"
done
```

10.3 Descripción de las Funciones del Script

- **limpiar_pantalla**: Limpia la pantalla antes de ejecutar cada acción, manteniendo la interfaz del menú organizada y fácil de leer.
- **mostrar_menu**: Muestra el menú principal con las opciones disponibles, permitiendo al usuario seleccionar la acción deseada.
- **ejecutar_opcion**: Ejecuta la opción seleccionada por el usuario mediante un **case**, llamando a los scripts correspondientes o comandos específicos.
 - **bash "\$BD_SCRIPT"**: Ejecuta el script de gestión de la base de datos, proporcionando acceso a funciones de consulta y administración de registros.
 - **crontab -e**: Abre el editor de tareas programadas de **crontab**, permitiendo la configuración y revisión de rutinas de backup.
 - **bash "\$USUARIOS_SCRIPT"**: Ejecuta el script de gestión de usuarios y grupos, facilitando la creación, modificación y eliminación de usuarios y grupos en el sistema.
 - **sudo nano "\$RED_CONFIG"**: Abre el archivo de configuración de red con permisos de superusuario para editar los parámetros de red.
 - **bash "\$BACKUP_SCRIPT"**: Ejecuta el script de backups automáticos, configurando y gestionando tareas de respaldo de datos.

Conclusión

Este menú principal unifica el acceso a todos los scripts desarrollados para el proyecto, permitiendo una gestión centralizada y simplificada de todas las funciones del sistema. Es una herramienta fundamental para los operadores y administradores, ya que mejora la eficiencia y facilita el mantenimiento del entorno de trabajo. A medida que se desarrollen nuevas funcionalidades o se actualicen los scripts existentes, este menú puede ser fácilmente modificado para incorporar los nuevos cambios, asegurando que siempre se mantenga como una solución flexible y accesible.

Glosario

1. **SSH (Secure Shell):** Protocolo de red utilizado para operar servicios de red de forma segura a través de una conexión cifrada.
2. **Crontab:** Archivo de configuración utilizado en sistemas Unix y Linux para programar tareas que se ejecutan automáticamente en horarios específicos.
3. **Script:** Secuencia de comandos que se ejecutan de manera automática para realizar tareas específicas en un sistema operativo.
4. **Backup:** Copia de seguridad de datos que permite recuperar la información en caso de pérdida o falla del sistema.
5. **Virtual Host:** Configuración que permite que un servidor web aloje múltiples dominios en una misma IP.
6. **LTS (Long Term Support):** Versiones de software que reciben soporte y actualizaciones durante un periodo extendido.
7. **MySQL:** Sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado para gestionar datos en aplicaciones web.
8. **Apache:** Servidor web de código abierto que gestiona solicitudes HTTP y sirve contenido a través de Internet.
9. **Netplan:** Herramienta de configuración de red para sistemas basados en Linux, especialmente Ubuntu.
10. **Scrumban:** Metodología ágil que combina las prácticas de Scrum y Kanban para la gestión de proyectos.

Aplicaciones Usadas

1. **Ubuntu Server 22.04 LTS (Máquina Virtual):** Sistema operativo utilizado como base para el servidor del proyecto, ejecutado en una máquina virtual para facilitar la gestión y la implementación de servicios críticos como Apache, MySQL y SSH. Proporciona estabilidad y soporte a largo plazo, ideal para entornos empresariales.
2. **VirtualBox 7.0:** Software de virtualización utilizado para crear y gestionar la máquina virtual en la que se ejecuta el servidor Ubuntu. Permite la simulación de un entorno de servidor real, facilitando la configuración, pruebas y despliegue de las aplicaciones sin necesidad de hardware adicional.
3. **Gestor de Scripts de Administración (GSA 1.2):** Aplicación de línea de comandos desarrollada internamente para la gestión de usuarios, grupos y configuración del sistema en el servidor. Permite la automatización de tareas administrativas y proporciona una interfaz simplificada para la ejecución de scripts de mantenimiento y configuración.
4. **Backup Manager Pro 2.5:** Herramienta automatizada de gestión de backups, utilizada para programar y ejecutar copias de seguridad de la

- base de datos y archivos críticos del servidor. Incluye funcionalidades de notificación por correo electrónico y gestión de tareas cron.
5. **Netplan Configurator 1.0:** Aplicación utilizada para la configuración y gestión de las redes del servidor. Facilita la edición de archivos de configuración Netplan, asegurando una conectividad estable y segura entre las terminales y el servidor.
 6. **MySQL Workbench 8.0:** Herramienta gráfica para la administración y diseño de bases de datos MySQL. Utilizada para visualizar, modificar y optimizar la estructura de la base de datos implementada en el servidor.

Conclusión Final:

Este documento ha detallado el proceso de instalación, configuración y gestión de un servidor Linux orientado a administrar un taller mecánico y su plataforma web asociada. Se utilizaron tecnologías probadas como Ubuntu Server, MySQL y Apache para crear un sistema robusto, seguro y fácilmente administrable. La implementación de un menú integrado facilita el acceso a diversas funciones críticas, permitiendo una gestión eficiente de usuarios, bases de datos, configuraciones de red y tareas programadas de backup.

El uso de scripts automatizados y una estructura modular simplifica la administración y mejora la eficiencia del personal técnico. La metodología ágil Scrumban aplicada garantiza una respuesta dinámica a los cambios y necesidades del entorno operativo, promoviendo la mejora continua y la adaptación ágil a los desafíos tecnológicos.

Fuentes Consultadas

1. **García, A. (2023).** *Implementación de Servidores Linux para Servicios Web*. Editorial Técnica.
2. **López, M. (2022).** *Seguridad y Configuración de SSH en Redes Empresariales*. Ediciones de Sistemas.
3. **Martínez, J. & Pérez, L. (2024).** *Gestión de Bases de Datos con MySQL: Guía Completa*. Editorial Informática Avanzada.
4. **Canonical (2024).** *Documentación Oficial de Ubuntu Server 22.04 LTS*. Disponible en: www.ubuntu.com/docs/server.
5. **Apache Software Foundation (2023).** *Guía de Instalación y Configuración de Apache HTTP Server*. Disponible en: www.apache.org/docs.
6. **MySQL Documentation (2024).** *MySQL 8.0 Reference Manual*. Oracle Corporation. Disponible en: www.mysql.com/docs.
7. **Netplan.io (2023).** *Configuración de Redes con Netplan en Ubuntu*. Disponible en: www.netplan.io.
8. **AlmaLinux Foundation (2024).** *Comparación de Distribuciones Linux: AlmaLinux vs Ubuntu*. Disponible en: www.almaLinux.org/docs.

9. **Rocky Linux Documentation (2024).** *Rocky Linux para Servidores Empresariales.* Disponible en: www.rockylinux.org/docs.
10. **OpenSSH Developers (2023).** *Configuración Segura de SSH para Administradores de Sistemas.* Disponible en: www.openssh.com/docs.
11. **Scrumban Methodology (2024).** *Guía de Prácticas Ágiles: Scrumban en Desarrollo de Software.* Editorial Ágil.

Hoja testigo

