



**UNIVERSIDAD DE EL SALVADOR**  
Facultad Multidisciplinaria De Occidente  
Departamento de Ingeniería y Arquitectura  
Ingeniería en Desarrollo de Software



Asignatura:

### **Introducción Software Libre**

Ciclo 2 / 2025

#### **Tema: Proyecto Final**

#### **Coordinador de Cátedra:**

Ing. Graciela Albaluz Escalante Grijalva

#### **Tutor GT02:**

Ing. Graciela Albaluz Escalante Grijalva

#### **Alumno/a:**

Alyson Pamela Alvarado Jovel

Alison Patricia Barrillas Castillo

Héctor Danilo Benítez Ortéz

Medardo Oseas Villalobos Meléndez

#### **Carnet:**

VM24009

AJ24002

BO16004

BC24013

## FASE 1: PREPARACIÓN DEL ENTORNO SERVIDOR

PREPARACIÓN DEL ENTORNO SERVIDOR: Configuración inicial del sistema Linux incluyendo la creación de usuarios (adminsys, tecnico, visitante), grupos (soporte, web) y estructura de directorios en / proyecto /, asignando permisos específicos para garantizar la seguridad y organización del sistema.

### Paso 1: Estado inicial (para capturas)

Entrada:

```
root@mariccy:/home/mariccy# echo "==== ESTADO INICIAL DEL SISTEMA ==="
hostname
getent passwd | grep -E "(adminsys|tecnico|visitante)" || echo "Usuarios no existen"
getent group | grep -E "(soporte|web)" || echo "Grupos no existen"
ls -la /proyecto 2>/dev/null || echo "Directorio /proyecto no existe"
```

Salida:

```
==== ESTADO INICIAL DEL SISTEMA ===
servidor-grupo01
adminsys:x:1001:1001:Administrador del Sistema:/home/adminsys:/bin/bash
tecnico:x:1002:1002:Técnico de Soporte:/home/tecnico:/bin/bash
visitante:x:1003:1003:Usuario Visitante:/home/visitante:/bin/bash
webdev:x:1006:
web_team:x:1011:
soporte:x:1014:adminsys,tecnico
web:x:1015:visitante
total 24
drwxr-xr-x  6 root root    4096 Nov 17 16:55 .
drwxr-xr-x 24 root root    4096 Nov 17 16:55 ..
drwxr-xr-x  2 root root    4096 Nov 17 16:56 capturas
drwxrwsr-x  2 root soporte 4096 Nov 17 16:55 datos
drwxr-xr-x  2 root root    4096 Nov 17 16:55 scripts
drwxrwsr-x  2 root web     4096 Nov 17 16:55 web
root@mariccy:/home/mariccy#
```

### Paso 2: Configurar hostname

Entrada:

```
root@mariccy:/home/mariccy# echo "1. CONFIGURANDO HOSTNAME"
sudo hostnamectl set-hostname servidor-grupo01
hostnamectl status
```

Salida:

```
root@mariccy:/home/mariccy# hostnamectl status
hostnamectl status
1. CONFIGURANDO HOSTNAME
    Static hostname: servidor-grupo01
        Icon name: computer-desktop
        Chassis: desktop 🖥
    Machine ID: 21534f550cb64d69a0b2a5d40dd30700
        Boot ID: ad0bf1645b584726a91dacfe4dc08871
    Operating System: Ubuntu 24.04.3 LTS
        Kernel: Linux 6.14.0-35-generic
        Architecture: x86-64
    Hardware Vendor: ASRock
    Hardware Model: A520M-HDV
    Firmware Version: P2.30
    Firmware Date: Thu 2022-10-20
    Firmware Age: 3y 4w 1d
root@mariccy:/home/mariccy#
```

Paso 3: Crear grupos

Entrada:

```
root@mariccy:/home/mariccy# echo "2. CREANDO GRUPOS"
sudo groupadd soporte
sudo groupadd web
getent group soporte web
root@mariccy:/home/mariccy#
```

Salida:

```
root@mariccy:/home/mariccy# echo "2. CREANDO GRUPOS"
2. CREANDO GRUPOS
groupadd: group 'soporte' already exists
groupadd: group 'web' already exists
soporte:x:1014:adminsystecniko
web:x:1015:visitante
root@mariccy:/home/mariccy#
```

Paso 4: Crear usuarios

Entrada:

```
root@mariccy:/home/mariccy# echo "3. CREANDO USUARIOS"
sudo useradd -m -s /bin/bash -c "Administrador del Sistema" adminsyst
sudo useradd -m -s /bin/bash -c "Técnico de Soporte" tecniko
sudo useradd -m -s /bin/bash -c "Usuario Visitante" visitante

# Verificar creación
id adminsyst
id tecniko
id visitante
```

**Salida:**

```
root@mariccy:~# 3. CREANDO USUARIOS
useradd: user 'adminsys' already exists
useradd: user 'tecnico' already exists
useradd: user 'visitante' already exists
uid=1001(adminsys) gid=1001(adminsys) groups=1001(adminsys),27(sudo),1014(soporte)
uid=1002(tecnico) gid=1002(tecnico) groups=1002(tecnico),1014(soporte)
uid=1003(visitante) gid=1003(visitante) groups=1003(visitante),1015(web)
root@mariccy:/home/mariccy#
```

**Paso 5: Asignar contraseñas**

```
root@mariccy:/home/mariccy# echo "4. ASIGNANDO CONTRASEÑAS"
echo "adminsys:Admin1235d!" | sudo chpasswd
echo "tecnico:Techno34/a!" | sudo chpasswd
echo "visitante:Visit2024!" | sudo chpasswd
4. ASIGNANDO CONTRASEÑAS
root@mariccy:/home/mariccy#
```

**Paso 6: Configurar membresías de grupos**

**Entrada:**

```
root@mariccy:/home/mariccy# echo "5. CONFIGURANDO GRUPOS DE USUARIOS"
sudo usermod -aG sudo adminsys
sudo usermod -aG soporte adminsys
sudo usermod -aG soporte tecnico
sudo usermod -aG web visitante

# Verificar
echo "Grupos de adminsys:"
groups adminsys
echo "Grupos de tecnico:"
groups tecnico
echo "Grupos de visitante:"
groups visitante
```

**Salida:**

```
5. CONFIGURANDO GRUPOS DE USUARIOS
Grupos de adminsys:
adminsys : adminsys sudo soporte
Grupos de tecnico:
tecnico : tecnico soporte
Grupos de visitante:
visitante : visitante web
root@mariccy:/home/mariccy#
```

## Paso 7 : Crear estructura de directorios

Entrada:

```
root@mariccy:/home/mariccy# echo "6. CREANDO ESTRUCTURA DE DIRECTORIOS"
sudo mkdir -p / proyecto
sudo mkdir -p / proyecto/datos
sudo mkdir -p / proyecto/web
sudo mkdir -p / proyecto/scripts
sudo mkdir -p / proyecto/capturas

# Verificar
ls -la / proyecto
```

Salida:

```
ls -la / proyecto
6. CREANDO ESTRUCTURA DE DIRECTORIOS
total 24
drwxr-xr-x 6 root root 4096 Nov 17 16:55 .
drwxr-xr-x 24 root root 4096 Nov 17 16:55 ..
drwxr-xr-x 2 root root 4096 Nov 17 16:56 capturas
drwxrwsr-x 2 root soporte 4096 Nov 17 16:55 datos
drwxr-xr-x 2 root root 4096 Nov 17 16:55 scripts
drwxrwsr-x 2 root web 4096 Nov 17 16:55 web
root@mariccy:/home/mariccy#
```

## Paso 8: Configurar permisos

Entrada:

```
drwxrwsr-x 2 root web 4096 Nov 17 16:55 web
root@mariccy:/home/mariccy# echo "7. CONFIGURANDO PERMISOS"

# Directorio datos - grupo soporte
sudo chgrp soporte / proyecto/datos
sudo chmod 775 / proyecto/datos
sudo chmod g+s / proyecto/datos

# Directorio web - grupo web
sudo chgrp web / proyecto/web
sudo chmod 775 / proyecto/web
sudo chmod g+s / proyecto/web

# Directorios scripts y capturas - root
sudo chown root:root / proyecto/scripts
sudo chown root:root / proyecto/capturas
sudo chmod 755 / proyecto/scripts
sudo chmod 755 / proyecto/capturas

# Verificar permisos
ls -ld / proyecto/ / proyecto/*/
```

## Salida:

```
cd .. cd /proyecto/ /proyecto/ /  
7. CONFIGURANDO PERMISOS  
drwxr-xr-x 6 root root 4096 Nov 17 16:55 /proyecto/  
drwxr-xr-x 2 root root 4096 Nov 17 16:56 /proyecto/capturas/  
drwxrwsr-x 2 root soporte 4096 Nov 17 16:55 /proyecto/datos/  
drwxr-xr-x 2 root root 4096 Nov 17 16:55 /proyecto/scripts/  
drwxrwsr-x 2 root web 4096 Nov 17 16:55 /proyecto/web/  
root@mariccy:/home/mariccy#
```

## Paso 9: Verificación final

### Entrada:

```
root@mariccy:/home/mariccy# echo "8. VERIFICACIÓN FINAL"  
echo "==== HOSTNAME ==="  
hostname  
  
echo "==== USUARIOS Y GRUPOS ==="  
echo "adminsys: $(groups adminsys)"  
echo "tecnico: $(groups tecnico)"  
echo "visitante: $(groups visitante)"  
  
echo "==== ESTRUCTURA DE DIRECTORIOS ==="  
ls -la /proyecto  
  
echo "==== PERMISOS SGID ==="  
find /proyecto -type d -perm -g+s 2>/dev/null  
  
echo "==== GRUPOS DEL SISTEMA ==="  
getent group soporte web
```

## Salida:

```
getent group soporte web  
8. VERIFICACIÓN FINAL  
==== HOSTNAME ===  
servidor-grupo01  
==== USUARIOS Y GRUPOS ===  
adminsys: adminsys : adminsys sudo soporte  
tecnico: tecnico : tecnico soporte  
visitante: visitante : visitante web  
==== ESTRUCTURA DE DIRECTORIOS ===  
total 24  
drwxr-xr-x 6 root root 4096 Nov 17 16:55 .  
drwxr-xr-x 24 root root 4096 Nov 17 16:55 ..  
drwxr-xr-x 2 root root 4096 Nov 17 16:56 capturas  
drwxrwsr-x 2 root soporte 4096 Nov 17 16:55 datos  
drwxr-xr-x 2 root root 4096 Nov 17 16:55 scripts  
drwxrwsr-x 2 root web 4096 Nov 17 16:55 web  
==== PERMISOS SGID ===  
/proyecto/web  
/proyecto/datos  
==== GRUPOS DEL SISTEMA ===  
soporte:x:1014:adminsys,tecnico  
web:x:1015:visitante  
root@mariccy:/home/mariccy#
```

## PARTE 2 - AUTOMATIZACIÓN Y MONITOREO

AUTOMATIZACIÓN Y MONITOREO: Desarrollo de scripts bash para el monitoreo automatizado del sistema y configuración de tareas programadas con cron que ejecutan reportes periódicos sobre el estado del servidor, usuarios conectados, uso de recursos y contenedores activos.

### Paso 1. Crear script de monitoreo

```
root@mariccy:/home/mariccy# sudo cat > / proyecto/scripts/reporte_sistema.sh << 'EOF'
#!/bin/bash
echo =====
echo " REPORTE DEL SISTEMA - $(date)"
echo =====
echo "Fecha y hora actual: $(date)"
echo "Nombre del host: $(hostname)"
echo "Usuarios conectados: $(who | wc -l)"
echo -----
echo "Espacio libre en disco:"
df -h | awk 'NR==2{print " " $4 " libres de " $2 " (" $5 " usado)"}'
echo "Memoria RAM disponible:"
free -h | awk 'NR==2{print " " $7 " disponibles de " $2}'
if command -v docker &> /dev/null; then
    echo "Contenedores Docker activos: $(docker ps -q | wc -l)"
else
    echo "Contenedores Docker activos: Docker no instalado"
fi
echo =====
echo ""
EOF
root@mariccy:/home/mariccy#
```

### Paso 2 :Dar permisos de ejecución

```
Lvi
root@mariccy:/home/mariccy# sudo chmod +x / proyecto/scripts/reporte_sistema.sh
root@mariccy:/home/mariccy#
```

### Paso 3 : Verificar creación del script

```
root@mariccy:/home/mariccy# ls -l / proyecto/scripts/reporte_sistema.sh
-rwxr-xr-x 1 root root 737 Nov 17 18:41 / proyecto/scripts/reporte_sistema.sh
root@mariccy:/home/mariccy#
```

## Paso 4: Probar el script de monitoreo

```
root@mariccy:/home/mariccy# ./proyecto/scripts/reporte_sistema.sh
=====
===== REPORTE DEL SISTEMA - Mon Nov 17 06:44:00 PM CST 2025 =====
=====
Fecha y hora actual: Mon Nov 17 06:44:00 PM CST 2025
Nombre del host: servidor-grupo01
Usuarios conectados: 3
-----
Espacio libre en disco:
  581G libres de 915G (34% usado)
Memoria RAM disponible:
  5.6Gi disponibles de 14Gi
Contenedores Docker activos: Docker no instalado
=====
```

## Paso 5: Verificar que funciona

```
root@mariccy:/home/mariccy# echo "Código de salida: $?"
Código de salida: 0
root@mariccy:/home/mariccy# 
```

## Paso 6: Crear directorio de logs

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo mkdir -p /var/log/proyecto
root@mariccy:/home/mariccy# 
```

## 7. Crear archivo de log

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo touch /var/log/proyecto/reporte_sistema.log
root@mariccy:/home/mariccy# 
```

## 8. Dar permisos al log

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo chmod 644 /var/log/proyecto/reporte_sistema.log
root@mariccy:/home/mariccy# 
```

## 9. Verificar directorio de logs

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# ls -la /var/log/proyecto/
total 12
drwxr-xr-x  2 root root  4096 Nov 17 18:48 .
drwxrwxr-x 22 root syslog 4096 Nov 17 16:59 ..
-rw-r--r--  1 root root   3581 Nov 17 17:08 parte1.log
-rw-r--r--  1 root root     0 Nov 17 18:48 reporte_sistema.log
root@mariccy:/home/mariccy# 
```

## 10. Verificar cron jobs actuales

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# crontab -l 2>/dev/null || echo "No hay cron jobs configurados"
No hay cron jobs configurados
root@mariccy:/home/mariccy#
```

## 11. Configurar cron job

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# (crontab -l 2>/dev/null; echo "*/30 * * * * / proyecto/scripts/report
e_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1") | crontab -
root@mariccy:/home/mariccy#
```

## 12. Verificar cron job configurado

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# crontab -l | grep reporte_sistema
*/30 * * * * / proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1
root@mariccy:/home/mariccy#
```

## 13. Verificar estado del servicio cron

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo systemctl status cron --no-pager | head -5
● cron.service - Regular background program processing daemon
  Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: enabled)
  Active: active (running) since Mon 2025-11-17 10:10:25 CST; 8h ago
    Docs: man:cron(8)
   Main PID: 1426 (cron)
      Tasks: 1 (since Mon 2025-11-17 10:10:25 CST)
     CGroup: /system.slice/cron.service
             └─1426 /usr/sbin/cron -n
root@mariccy:/home/mariccy#
```

## 14. Verificar si cron está activo

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo systemctl is-active cron
active
root@mariccy:/home/mariccy#
```

## 15. Verificar si cron está habilitado

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo systemctl is-enabled cron
enabled
root@mariccy:/home/mariccy#
```

## 16. Probar ejecución automática

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# / proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_si  
stema.log 2>&1  
root@mariccy:/home/mariccy#
```

## 17. Verificar contenido del log

```
servicio.log 2>&1  
root@mariccy:/home/mariccy# tail -10 /var/log/proyecto/reporte_sistema.log  
Nombre del host: servidor-grupo01  
Usuarios conectados: 3  
-----  
Espacio libre en disco:  
 581G libres de 915G (34% usado)  
Memoria RAM disponible:  
 5.6Gi disponibles de 14Gi  
Contenedores Docker activos: Docker no instalado  
=====  
root@mariccy:/home/mariccy#
```

## 18. Verificación completa - Script

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# ls -la /proyecto/scripts/reporte_sistema.sh  
-rwxr-xr-x 1 root root 737 Nov 17 18:41 /proyecto/scripts/reporte_sistema.sh  
root@mariccy:/home/mariccy#
```

## 19. Verificación completa - Permisos

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# ls -l /proyecto/scripts/reporte_sistema.sh  
-rwxr-xr-x 1 root root 737 Nov 17 18:41 /proyecto/scripts/reporte_sistema.sh  
root@mariccy:/home/mariccy#
```

## 20. Verificación completa - Cron job

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# crontab -l | grep reporte_sistema  
*/30 * * * * /proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1  
root@mariccy:/home/mariccy#
```

## 21. Verificación completa - Archivo de log

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# ls -la /var/log/proyecto/reporte_sistema.log
-rw-r--r-- 1 root root 496 Nov 17 18:55 /var/log/proyecto/reporte_sistema.log
root@mariccy:/home/mariccy#
```

## 22. Verificación completa - Contenido del log

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# tail -5 /var/log/proyecto/reporte_sistema.log
Memoria RAM disponible:
 5,7Gi disponibles de 14Gi
Contenedores Docker activos: Docker no instalado
=====
root@mariccy:/home/mariccy#
```

## 23. Verificación completa - Servicio cron

```
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy#
root@mariccy:/home/mariccy# sudo systemctl is-active cron && echo "SI" || echo "NO"
active
SI
root@mariccy:/home/mariccy#
```

## PARTE 3: CONTROL DE VERSIONES

Implementación de un repositorio Git local y remoto en GitHub para gestionar el versionado del proyecto, permitiendo el trabajo colaborativo y el mantenimiento de un historial completo de todos los cambios realizados en la configuración y scripts.

## PARTE 4: DOCKER

Instalación y configuración del motor de contenedores Docker, integración de usuarios al grupo docker para permisos sin sudo, y verificación del funcionamiento mediante la ejecución del contenedor de prueba hello-world.

## Paso 1: Actualizar sistema e Instalar Docker

```
root@mariccy:/home/mariccy# sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:3 https://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:5 http://deb.anydesk.com all InRelease
Hit:6 https://deb.nodesource.com/node_20.x nodistro InRelease
Get:7 https://packages.microsoft.com/ubuntu/24.04/prod noble InRelease [3,600 B]
Get:8 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,211 B]
Hit:9 https://ppa.launchpadcontent.net/flatpak/stable/ubuntu noble InRelease
Get:10 https://packages.microsoft.com/ubuntu/24.04/prod noble/main armhf Packages [11.2 kB]
Get:11 https://packages.microsoft.com/ubuntu/24.04/prod noble/main amd64 Packages [67.9 kB]
Get:12 https://packages.microsoft.com/ubuntu/24.04/prod noble/main arm64 Packages [50.7 kB]
Get:13 https://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:18 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.7 kB]
70% [19 Commands: amd64 6.116 B/15.7 kB 39%]
```

```
root@mariccy:/home/mariccy# sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (28.2.2-0ubuntu1-24.04.1).
The following packages were automatically installed and are no longer required:
  gyp libgl1-amber-dri libglapi-mesa libjs-async libjs-events libjs-inherits
  libjs-is-typedarray libjs-prettify libjs-regenerate libjs-source-map libjs-sprintf-js
  libjs-typedarray-to-buffer libssl-dev libuv1-dev node-abbrev node-ampproject-remapping
  node-ansi-regex node-ansi-styles node-aproba node-are-we-there-yet node-arrify node-async
  node-async-each node-auto-bind node-babel-plugin-add-module-exports node-babel7-runtime
  node-balanced-match node-base64-js node-binary-extensions node-brace-expansion node-busboy
  node-camelcase node-canisuse-lite node-chownr node-chrome-trace-event node-ci-info
  node-cjs-module-lexer node-cli-boxes node-cli-cursor node-clone node-collection-visit
```

## Paso 2: Habilitar e iniciar el servicio Docker

```
root@mariccy:/home/mariccy# sudo systemctl enable docker  
root@mariccy:/home/mariccy#
```

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# sudo systemctl start docker  
root@mariccy:/home/mariccy#
```

```
root@mariccy:/home/mariccy# 
root@mariccy:/home/mariccy# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-11-17 19:28:16 CST; 2min 18s ago
     Docs: https://docs.docker.com
  Main PID: 2547527 (dockerd)
    Tasks: 14
   Memory: 24.0M (peak: 25.1M)
      CPU: 298ms
     CGroup: /system.slice/docker.service
             └─2547527 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 17 19:28:15 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:15.935161474-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.223792292-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.234823741-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.234911505-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.238486316-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.238513497-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.248194601-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.253879644-06:00"
Nov 17 19:28:16 servidor-grupo01 dockerd[2547527]: time="2025-11-17T19:28:16.253922484-06:00"
Nov 17 19:28:16 servidor-grupo01 systemd[1]: Started docker.service - Docker Application Conta
lines 1-22/22 (END)
```

### Paso 3: Agregar usuarios al grupo docker

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# sudo usermod -aG docker adminsyst  
root@mariccy:/home/mariccy#
```

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# sudo usermod -aG docker tecnico  
root@mariccy:/home/mariccy#
```

### Paso 4: Verificar la instalación

```
root@mariccy:/home/mariccy#  
root@mariccy:/home/mariccy# docker --version  
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1  
root@mariccy:/home/mariccy#
```

```
root@mariccy:/home/mariccy# docker info  
Client:  
  Version: 28.2.2  
  Context: default  
  Debug Mode: false  
  
Server:  
  Containers: 0  
    Running: 0  
    Paused: 0  
    Stopped: 0  
  Images: 0  
  Server Version: 28.2.2  
  Storage Driver: overlay2  
    Backing Filesystem: extfs  
    Supports d_type: true  
    Using metacopy: false  
    Native Overlay Diff: true  
    userxattr: false  
  Logging Driver: json-file  
  Cgroup Driver: systemd  
  Cgroup Version: 2  
  Plugins:  
    Volume: local  
    Network: bridge host ipvlan macvlan null overlay  
    Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog  
  CDI spec directories:  
    /etc/cdi  
    /var/run/cdi  
  Swarm: inactive  
  Runtimes: io.containerd.runc.v2 runc  
  Default Runtime: runc  
  Init Binary: docker-init  
  containerd version:  
  runc version:  
  init version:  
  Security Options:  
    apparmor  
    seccomp  
      Profile: builtin  
      cgroups  
  Kernel Version: 6.14.0-35-generic  
  Operating System: Ubuntu 24.04.3 LTS  
  OSType: linux  
  Architecture: x86_64  
  CPUs: 12  
  Total Memory: 15GiB  
  Name: servidor-grupo01  
  ID: 540373f0-3a6a-4987-91bc-ffcf6c252948  
  Docker Root Dir: /var/lib/docker  
  Debug Mode: false  
  Experimental: false  
  Insecure Registries:  
    ::1/128  
    127.0.0.0/8  
  Live Restore Enabled: false
```

```
root@mariccy:/home/mariccy# newgrp docker
root@servidor-grupo01:/home/mariccy# █
```

## 4.2 Verificación Inicial

### Paso 1: Ejecutar contenedor de prueba hello-world

```
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:f7931603f70e13dbd844253370742c4fc4202d290c80442b2e68706d8f33ce26
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
root@servidor-grupo01:/home/mariccy# █
```

### Paso 2: Verificar que se ejecutó correctamente

```
https://docs.docker.com/get-started/
root@servidor-grupo01:/home/mariccy# docker ps -a
CONTAINER ID   IMAGE      COMMAND     CREATED        STATUS          PORTS
AMES           hello-world "/hello"   18 seconds ago   Exited (0) 17 seconds ago
estive_wright
root@servidor-grupo01:/home/mariccy# █
```

```
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# docker images
REPOSITORY      TAG      IMAGE ID      CREATED        SIZE
hello-world    latest    1b44b5a3e06a  3 months ago   10.1kB
root@servidor-grupo01:/home/mariccy# █
```

## PARTE 5: SERVIDOR WEB CONTAINERIZADO

Despliegue de un servidor web Nginx en contenedor Docker con mapeo de puertos y montaje de volúmenes, sirviendo contenido HTML personalizado desde el directorio / proyecto / web / y verificando su correcto funcionamiento.

### Paso 1: Crear archivo index.html

```
root@servidor-grupo01:/home/mariccy# sudo tee /proyecto/web/index.html > /dev/null <<'EOF'
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Servidor Web Containerizado - Grupo 01</title>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">
<style>
    * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }

```

### Paso 2: Verificar que se creó el archivo

```
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# ls -la /proyecto/web/
total 20
drwxrwsr-x 2 root web 4096 Nov 17 19:49 .
drwxr-xr-x 7 root root 4096 Nov 17 19:23 ..
-rw-r--r-- 1 root web 8658 Nov 17 19:49 index.html
root@servidor-grupo01:/home/mariccy# 
```

### Paso 3: Ejecutar contenedor Nginx

```
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# docker stop nginx-grupo01 2>/dev/null || true
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# docker rm nginx-grupo01 2>/dev/null || true
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# docker run -d \
    --name nginx-grupo01 \
    -p 8080:80 \
    -v /proyecto/web:/usr/share/nginx/html \
    --restart unless-stopped \
    nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
d7ecded7702a: Pull complete
266626526d42: Pull complete
320b0949be89: Pull complete
d921c57c6a81: Pull complete
9def903993e4: Pull complete
52bc359bcb7: Pull complete
e2f8e296d9df: Pull complete
Digest: sha256:1beed3ca46acebee9d3fb62e9067f03d05d5bfa97a00f30938a0a3580563272ad
Status: Downloaded newer image for nginx:latest
d98098e150b6ac000a254fdb2e1b79f40e26a6d6c86713396cba983be4c82c54
root@servidor-grupo01:/home/mariccy# 
```

## Paso 4: Verificaciones

```
root@servidor-grupo01:/home/mariccy# curl -I http://localhost:8080
HTTP/1.1 200 OK
Server: nginx/1.29.3
Date: Tue, 18 Nov 2025 01:54:45 GMT
Content-Type: text/html
Content-Length: 8658
Last-Modified: Tue, 18 Nov 2025 01:49:07 GMT
Connection: keep-alive
ETag: "691bd093-21d2"
Accept-Ranges: bytes
root@servidor-grupo01:/home/mariccy#
root@servidor-grupo01:/home/mariccy# /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsn
root@servidor-grupo01:/home/mariccy# /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
root@servidor-grupo01:/home/mariccy# /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
root@servidor-grupo01:/home/mariccy# /docker-entrypoint.sh: Configuration complete; ready for start up
2025/11/18 01:51:46 [notice] 1#1: using the "epoll" event method
2025/11/18 01:51:46 [notice] 1#1: nginx/1.29.3
2025/11/18 01:51:46 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/11/18 01:51:46 [notice] 1#1: OS: Linux 6.14.0-35-generic
2025/11/18 01:51:46 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/11/18 01:51:46 [notice] 1#1: start worker processes
2025/11/18 01:51:46 [notice] 1#1: start worker process 29
2025/11/18 01:51:46 [notice] 1#1: start worker process 30
2025/11/18 01:51:46 [notice] 1#1: start worker process 31
2025/11/18 01:51:46 [notice] 1#1: start worker process 32
2025/11/18 01:51:46 [notice] 1#1: start worker process 33
2025/11/18 01:51:46 [notice] 1#1: start worker process 34
2025/11/18 01:51:46 [notice] 1#1: start worker process 35
2025/11/18 01:51:46 [notice] 1#1: start worker process 36
2025/11/18 01:51:46 [notice] 1#1: start worker process 37
2025/11/18 01:51:46 [notice] 1#1: start worker process 38
2025/11/18 01:51:46 [notice] 1#1: start worker process 39
2025/11/18 01:51:46 [notice] 1#1: start worker process 40
root@servidor-grupo01:/home/mariccy#
```

## Paso 5: Acceder desde navegador

