



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



Asignatura:

INTRODUCCION AL SOFTWARE LIBRE

Ciclo II / Segundo año

Tema:

PROYECTO FINAL

Tutor/a GT01:

Graciela Albaluz Escalante Grijalva

Alumnos/a:

Esdras Leonel Peraza Perez

Rolando Estuardo Salguero Borja

Ana Cristina Martinez Salas

Luis Alberto Rodríguez Lara

Carnet:

PP24012

SB21023

MS24088

RL15028

Fecha:

17/11/2025

El presente informe documenta el desarrollo del proyecto final de la asignatura **Introducción al Software Libre**, cuyo objetivo es implementar un servidor Linux automatizado utilizando tecnologías de contenedores (Docker). El proyecto integra prácticas de administración de sistemas, control de versiones y despliegue de servicios web.

En este documento se presentan las fases implementadas hasta el **punto 5**, incluyendo capturas de pantalla y explicación detallada de cada acción realizada.

Fase 1

1. Preparación entorno al servidor

1.1 Administración básica del sistema

El comando hostnamectl cambiando el nombre del host a servidor-grupo2.

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

Servidor-grupo2@Servidor:~$ sudo hostnamectl set-hostname servidor-grupo2
[sudo] password for Servidor-grupo2:
Servidor-grupo2@Servidor:~$ hostname
servidor-grupo2
```

Se verificó que los usuarios pertenecen a los grupos correctos:

- adminsyst con privilegios de sudo.
- tecnico en el grupo soporte.
- visitante en el grupo web.

Propósito: Simular un entorno real de administración de sistemas con roles definidos.

```
Servidor-grupo2@Servidor:~$ groups adminsyst
adminsyst : adminsyst sudo
Servidor-grupo2@Servidor:~$ groups tecnico
tecnico : tecnico soporte
Servidor-grupo2@Servidor:~$ groups visitante
visitante : visitante web
```

1.2 Estructuras de directorios y permisos

Comandos mkdir creando /proyecto/ y sus subdirectorios (datos/, web/, scripts/, capturas/).

Se construyó la estructura de trabajo del proyecto.

Se asignaron grupos (soporte y web) y se configuró herencia de permisos para mantener orden y seguridad.

Propósito: Garantizar que los archivos creados dentro de cada directorio hereden el grupo correspondiente.

```
Servidor-grupo2@Servidor:~$ sudo mkdir / proyecto
Servidor-grupo2@Servidor:~$ sudo mkdir / proyecto/datos
Servidor-grupo2@Servidor:~$ sudo mkdir / proyecto/web
Servidor-grupo2@Servidor:~$ sudo mkdir / proyecto/scripts
Servidor-grupo2@Servidor:~$ sudo mkdir / proyecto/capturas
Servidor-grupo2@Servidor:~$ sudo chgrp soporte / proyecto/datos
Servidor-grupo2@Servidor:~$ sudo chgrp web / proyecto/web
Servidor-grupo2@Servidor:~$ sudo chmod g+s / proyecto/datos
Servidor-grupo2@Servidor:~$ sudo chmod g+s / proyecto/web
Servidor-grupo2@Servidor:~$ ls -la / proyecto/
total 24
drwxr-xr-x 6 root root 4096 Nov 2 21:02 .
drwxr-xr-x 24 root root 4096 Nov 2 21:01 ..
drwxr-xr-x 2 root root 4096 Nov 2 21:02 capturas
drwxr-sr-x 2 root soporte 4096 Nov 2 21:02 datos
drwxr-xr-x 2 root root 4096 Nov 2 21:02 web
drwxr-sr-x 2 root web 4096 Nov 2 21:02 scripts
Servidor-grupo2@Servidor:~$
```

Fase 2

2 Automatización y monitoreo

2.1 Script de monitoreo de sistema

Edición del archivo reporte_sistema.sh en nano.

El script muestra fecha, host, usuarios conectados, espacio en disco, memoria RAM y contenedores Docker activos.

```
GNU nano 7.2                                         / proyecto/scripts/reporte_sistema.sh *
# "Reporte del sistema"
echo "Fecha y hora: $(date '+%Y-%m-%d %H:%M:%S')"
echo "Host del Sistema: $(hostname)"
echo "usuarios Conectados: $(who | wc -l)"
echo "Espacio en disco:"
df -h / | awk 'NR==2 {print " Libre: " $4 ", Usado: "$5}'
echo "Espacio RAM"
free -h | awk 'NR==2 {print " Libre: " $7 ", Usado: "$3}'
echo "Contenedores Docker activos: $(docker ps -q | wc -l 2>/dev/null || echo "Docker no disponible")"
```

Se probó manualmente y mostró la información del sistema. En ese momento Docker aún no estaba instalado, por lo que devolvió “Docker no disponible”.

Propósito: Automatizar la recolección de métricas del servidor.

```
Servidor-grupo2@Servidor:~$ sudo chmod +x / proyecto/scripts/reporte_sistema.sh
Servidor-grupo2@Servidor:~$ ls -la / proyecto/scripts/
total 12
drwxr-xr-x 2 root root 4096 Nov 2 21:31 .
drwxr-xr-x 6 root root 4096 Nov 2 21:02 ..
-rwxr-xr-x 1 root root 429 Nov 2 21:31 reporte_sistema.sh
Servidor-grupo2@Servidor:~$
```

```

Servidor-grupo2@Servidor:~$ ./ proyecto/scripts/reporte_sistema.sh
Reporte del sistema
Fecha y hora: 2025-11-02 21:36:50
Host del Sistema: servidor-grupo2
usuarios Conectados: 1
Espacio en disco:
  Libre: 21G, Usado: 12%
Espacio RAM
  Libre: 3.4Gi, Usado: 438Mi
./ proyecto/scripts/reporte_sistema.sh: line 15: docker: command not found
Contenedores Docker activos: 0
Servidor-grupo2@Servidor:~$ 

```

2.2 Automatización con cron

Se programó la ejecución automática del script cada 30 minutos, guardando la salida en /var/log/proyecto/reporte_sistema.log.

Propósito: Registrar periódicamente el estado del sistema sin intervención manual.

```

adminsys@servidor-grupo2:~$ sudo crontab -l
*/30 * * * * ./ proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1
adminsys@servidor-grupo2:~$ 

```

Fase 3

3 control de versiones

Instalación y configuración de Git (git config).

Se definió usuario y correo para el repositorio.

```

adminsys@servidor-grupo2:~$ sudo crontab -l
[sudo] password for adminsys:
crontab: invalid option -- '1'
crontab: usage error: unrecognized option
usage: crontab [-u user] [-nl file]
               crontab [-u user] [-i] [-e | -l | -r]
               -h      (displays this help message)
               file   (default operation is replace, per 1003.2)
               -n      (dry run: checks the syntax, then bails out)
               -u user (choose the user whose crontab is touched)
               -e      (edit user's crontab)
               -l      (list user's crontab)
               -r      (delete user's crontab)
               -i      (prompt before deleting user's crontab)
adminsys@servidor-grupo2:~$ sudo crontab -l
*/30 * * * * ./ proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1
adminsys@servidor-grupo2:~$ sudo apt update
[sudo] password for adminsys:
Sorry, try again.
[sudo] password for adminsys:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
26 packages can be upgraded. Run 'apt list --upgradable' to see them.
adminsys@servidor-grupo2:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 26 not upgraded.
adminsys@servidor-grupo2:~$ git config --global user.name "proyecto-linux-grupo2"
adminsys@servidor-grupo2:~$ git config --global user.email "pp24012@ues.edu.sv"
fatal: not a git repository (or any of the parent directories): .git
adminsys@servidor-grupo2:~$ git config --list
user.name=proyecto-linux-grupo2
user.email=pp24012@ues.edu.sv
adminsys@servidor-grupo2:~$ 

```

Se enlazó con el repositorio remoto en GitHub (proyecto-linux-grupo2) y se subieron los archivos.

```
adminsys@servidor-grupo2:/ proyecto$ git add .
adminsys@servidor-grupo2:/ proyecto$ git status
On branch master
No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  scripts/reporte_sistema.sh

adminsys@servidor-grupo2:/ proyecto$
```

```
adminsys@servidor-grupo2:/ proyecto$ git add .
adminsys@servidor-grupo2:/ proyecto$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
  new file:  capturas/.gitkeep
  new file:  datos/.gitkeep
  new file:  web/.gitkeep
  new file:  web/index.html

adminsys@servidor-grupo2:/ proyecto$ git commit -m "Servidor Grupo 2"
[main c358df3] Servidor Grupo 2
 4 files changed, 1 insertion(+)
 create mode 100644 capturas/.gitkeep
 create mode 100644 datos/.gitkeep
 create mode 100644 web/.gitkeep
 create mode 100644 web/index.html
adminsys@servidor-grupo2:/ proyecto$ git push origin main
Username for 'https://github.com': Esdras055
Password for 'https://Esdras055@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 507 bytes | 507.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Esdras055/proyecto-linux-grupo2.git
 b86e05f..c358df3 main -> main
adminsys@servidor-grupo2:/ proyecto$
```

```
adminsys@servidor-grupo2:/ proyecto$ git log --oneline
aa8ceaf (HEAD -> master) Estructura del servidor-grupo2
adminsys@servidor-grupo2:/ proyecto$
```

```
adminsys@servidor-grupo2:/ proyecto$ git commit -m "Servidor-grupo2"
[master (root-commit) 229fid2] Servidor-grupo2
 1 file changed, 16 insertions(+)
 create mode 100755 scripts/reporte_sistema.sh
adminsys@servidor-grupo2:/ proyecto$ git remote add origin https://github.com/Esdras055/proyecto-linux-grupo2.git
adminsys@servidor-grupo2:/ proyecto$ git remote -v
origin https://github.com/Esdras055/proyecto-linux-grupo2.git (fetch)
origin https://github.com/Esdras055/proyecto-linux-grupo2.git (push)
adminsys@servidor-grupo2:/ proyecto$
```

```

The most similar command is
  branch
adminsys@servidor-grupo2:/ proyecto$ git branch -M main
adminsys@servidor-grupo2:/ proyecto$ git push -u origin main
Username for 'https://github.com': Esdras055
Password for 'https://Esdras055@github.com':
remote: Invalid username or token. Password authentication is not supported for Git operations.
fatal: Authentication failed for 'https://github.com/Esdras055/proyecto-linux-grupo2.git/'
adminsys@servidor-grupo2:/ proyecto$ git push -u origin main
Username for 'https://github.com': Esdras055
Password for 'https://Esdras055@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 534 bytes | 534.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Esdras055/proyecto-linux-grupo2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
adminsys@servidor-grupo2:/ proyecto$ _

```

Contenido del README.md.

```

GNU nano 7.2                                         README.md *
##Fases Implementados

### Fase 1: Preparacion del entorno del servidor
- configuracion del hostname: 'servidor-grupo2'
- creacion de usuarios: 'adminsys', 'tecnico', 'visitante'
- creacion de grupos: 'soporte', 'web'
- Estructura de directorios en '/proyecto/'
- Asignacion de permisos

### Fase 2: Automaticion y monitoreo
- Script 'reporte_sistema.sh' en '/proyecto/scripts/'
- Configuracion de tarea cron cada 30 min
- Logs en '/var/log/proyecto/reporte_sistema.log'

## Fase 3: Control de versiones
- Repositorio Git inicializado en '/proyecto/'
- Repositorio remoto en Github: 'proyecto-linux-group2'
- Este archivo README.md

```

```

adminsys@servidor-grupo2:/ proyecto$ git commit -m "Actualizacion del Readme.md"
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)
adminsys@servidor-grupo2:/ proyecto$ _

```

Se documentaron las fases implementadas hasta ese momento.

Propósito: Asegurar trazabilidad y colaboración entre integrantes del grupo.

```
adminsys@servidor-grupo2:/ proyecto$ git push origin main
Username for 'https://github.com': Esdras055
Password for 'https://Esdras055@github.com':
Everything up-to-date
adminsys@servidor-grupo2:/ proyecto$ ls -la /proyecto/README.md
-rw-r--r-- 1 root root 634 Nov  3 00:20 /proyecto/README.md
adminsys@servidor-grupo2:/ proyecto$ cat /proyecto/README.md
##Fases Implementados

### Fase 1: Preparacion del entorno del servidor
- configuracion del hostname: 'servidor-grupo2'
- creacion de usuarios: 'adminsys', 'tecnico', 'visitante'
- creacion de grupos: 'soporte', 'web'
- Estructura de directorios en '/proyecto/'
- Asignacion de permisos

### Fase 2: Automaticion y monitoreo
- Script 'reporte_sistema.sh' en '/proyecto/scripts/'
- Configuracion de tarea cron cada 30 min
- Logs en '/var/log/proyecto/reporte_sistema.log'

## Fase 3: Control de versiones
- Repositorio Git inicializado en '/proyecto/'
- Repositorio remoto en Github: 'proyecto-linux-group2'
- Este archivo README.md

adminsys@servidor-grupo2:/ proyecto$ _
```

Fase 4

4 Docker

4.1 Instalación y configuración

Instalación de paquetes Docker (docker-ce, docker-ce-cli, docker-compose-plugin).

Se instaló y habilitó el servicio Docker con systemctl.

```

Preparing to unpack .../1-docker-ce-cli_5%3a28.5.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:28.5.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../2-docker-ce_5%3a28.5.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:28.5.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.29.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.29.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a28.5.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:28.5.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.40.3-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.40.3-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0~ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0~ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.29.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.28-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.40.3-1~ubuntu.24.04~noble) ...
Setting up docker-ce (5:28.5.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0~ubuntu3) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:28.5.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
adminsys@servidor-grupo2:~$
```

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
adminsys@servidor-grupo2:~$ sudo systemctl enable docker
sudo: systemctl: command not found
adminsys@servidor-grupo2:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
adminsys@servidor-grupo2:~$ sudo systemctl start docker
adminsys@servidor-grupo2:~$ 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 2025-11-03 00:59:09 UTC; 2min 18s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 3060 (dockerd)
   Tasks: 9
     Memory: 21.6M (peak: 22.8M)
        CPU: 266ms
      CGroup: /system.slice/docker.service
           └─3060 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 03 00:59:08 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:08.791035060Z" level=info msg="detected 127.0.0.53 nameserver, assuming system-resolved."
Nov 03 00:59:08 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:08.820431962Z" level=info msg="Creating a containerd client" address=/run/containerd/containerd.sock
Nov 03 00:59:08 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:08.856916440Z" level=info msg="Loading containers: start."
Nov 03 00:59:09 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:09.103475212Z" level=info msg="Loading containers: done."
Nov 03 00:59:09 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:09.119431346Z" level=info msg="Docker daemon" commit=f8215cc containerd-snapshotter=false
Nov 03 00:59:09 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:09.119515762Z" level=info msg="Initializing buildkit"
Nov 03 00:59:09 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:09.149987568Z" level=info msg="Completed buildkit initialization"
Nov 03 00:59:09 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:09.155492567Z" level=info msg="Daemon has completed initialization"
Nov 03 00:59:09 servidor-grupo2 dockerd[3060]: time="2025-11-03T00:59:09.155815448Z" level=info msg="API listen on /run/docker.sock"
Nov 03 00:59:09 servidor-grupo2 systemd[1]: Started docker.service - Docker Application Container Engine.
lines: 1-22/22 (ENDL)
```

Comando groups admins y groups tecnico.

Se verificó que los usuarios fueron agregados al grupo docker.

Propósito: Permitir que los usuarios ejecuten comandos Docker sin necesidad de sudo.

```
adminsys@servidor-grupo2:~$ groups adminsys
adminsys : adminsys sudo adminsys docker
adminsys@servidor-grupo2:~$ groups tecnico
tecnico : tecnico soporte docker
adminsys@servidor-grupo2:~$ _

adminsys@servidor-grupo2:~$ groups tecnico
tecnico : tecnico soporte docker
adminsys@servidor-grupo2:~$ docker --version
Docker version 20.5.1, build e180ab8
adminsys@servidor-grupo2:~$ docker info
Client: Docker Engine - Community
  Version: 20.5.1
  Context: default
  Debug Mode: false
  Plugins:
    buildx: Docker Buildx (Docker Inc.)
      Version: v0.29.1
      Path: /usr/libexec/docker/cli-plugins/docker-buildx
    compose: Docker Compose (Docker Inc.)
      Version: v2.40.3
      Path: /usr/libexec/docker/cli-plugins/docker-compose

Server:
  permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http:///:/var/run/docker.sock/v1.51/info": dial unix /var/run/docker.sock: connect: permission denied
adminsys@servidor-grupo2:~$
```

4.2 Verificación inicial

Ejecución de docker run hello-world.

Se comprobó que Docker podía descargar imágenes y ejecutar contenedores.

Se verificó la existencia del contenedor hello-world en estado exitoso.

Propósito: Validar la correcta instalación y funcionamiento de Docker.

```
adminsys@servidor-grupo2:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:56433a6be3fda188089fb548beae3d91df3ed0d6589f7c2656121b911198df065
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/
```

adminsys@servidor-grupo2:~\$

```
adminsys@servidor-grupo2:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
adminsys@servidor-grupo2:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2ca858d568f4 hello-world "/hello" About a minute ago Exited (0) About a minute ago
adminsys@servidor-grupo2:~$
```

Fase 5

5.Servidor web containerizado

5.1 contenedor Nginx basico

Creación del archivo index.html en /proyecto/web/.

Se personalizó el contenido del servidor web del grupo.

The screenshot shows a terminal window with two main sections. The top section displays the contents of an index.html file being edited with nano, showing HTML code for a web page with a title, meta charset, and a single paragraph. The bottom section shows the terminal command 'cat /projecto/web/index.html' being run, followed by its output which indicates the file does not exist. This is followed by another 'cat' command which successfully displays the same HTML code as the editor, confirming it was saved correctly.

```
GNU nano 7.2                                         /projecto/web/index.html *
<!DOCTYPE html>
<html>
<head>
    <title>Servidor Grupo 2 - ISL135</title>
    <meta charset="UTF-8">
    <style>
        body { font-family: Arial; max-width: 800px; margin: auto; padding: 20px; background: #f0f0f0; }
        .container { background: white; padding: 20px; border-radius: 8px; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }
        h1 { color: #2c3e50; border-bottom: 2px solid #3498db; }
        .section { background: #e6f4ff; padding: 15px; margin: 15px 0; border-left: 4px solid #3498db; }
    </style>
</head>
<body>
    <div class="container">
        <h1> Servidor Web - grupo 2 </h1>
        <h3> ISL135 - Universidad del El Salvador</h3>
    </div>
</body>
</html>

cat: /projecto/web/index.html: No such file or directory
adminsys@servidor-grupo2:~$ cat /projecto/web/index.html
<!DOCTYPE html>
<html>
<head>
    <title>Servidor Grupo 2 - ISL135</title>
    <meta charset="UTF-8">
    <style>
        body { font-family: Arial; max-width: 800px; margin: auto; padding: 20px; background: #f0f0f0; }
        .container { background: white; padding: 20px; border-radius: 8px; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }
        h1 { color: #2c3e50; border-bottom: 2px solid #3498db; }
        .section { background: #e6f4ff; padding: 15px; margin: 15px 0; border-left: 4px solid #3498db; }
    </style>
</head>
<body>
    <div class="container">
        <h1> Servidor Web - grupo 2 </h1>
        <h3> ISL135 - Universidad del El Salvador</h3>
    </div>
</body>
</html>

adminsys@servidor-grupo2:~$
```

Comando docker run -d -p 8080:80 -v /proyecto/web/:/usr/share/nginx/html/ nginx:latest.

Se ejecutó el contenedor Nginx en segundo plano, mapeando el puerto 8080 y montando el directorio web.

Propósito: Desplegar un servidor web básico con Nginx.

```

adminsys@servidor-grupo2:~$ docker run -d \
> --name nginx-grupo2 \
> -p 8080:80 \
> -v /proyecto/web/:/usr/share/nginx/html/ \
> nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
38513bd72563: Full complete
a0a6ab141558: Full complete
0e86847a3920: Full complete
1bace2083289: Full complete
89df300a082a: Full complete
35fb9ffa6621: Full complete
5545b08f9d26: Full complete
Digest: sha256:f547e3d0d5d02f7009737b284abc87d800e4252b42dceea361811e9fc606287f
Status: Downloaded newer image for nginx:latest
67b9627b7cee048d33a9eb80eea428cae4d147b4cb3061e7c19e208c44dbaf05
adminsys@servidor-grupo2:~$ 

```

```

adminsys@servidor-grupo2:~$ curl http://localhost:8080
<!DOCTYPE html>
<html>
<head>
    <title>Servidor Grupo 2 - ISL135</title>
    <meta charset="UTF-8">
    <style>
        body { font-family: Arial; max-width: 800px; margin: auto; padding: 20px; background: #f0f0f0; }
        .container { background: white; padding: 20px; border-radius: 8px; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }
        h1 { color: #2c3e50; border-bottom: 2px solid #3498db; }
        .section { background: #e8f4fc; padding: 15px; margin: 15px 0; border-left: 4px solid #3498db; }

    </style>
</head>
<body>
    <div class="container">
        <h1> Servidor Web - grupo 2 </h1>
        <h3> ISL135 - Universidad del El Salvador</h3>
    </div>
</body>
</html>
adminsys@servidor-grupo2:~$ 

```

5.2 verificacion del servicio web

curl <http://localhost:8080>.

Se revisaron los logs confirmando que no había errores.

```

adminsys@servidor-grupo2:~$ docker logs nginx-grupo2
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.enosh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/11/03 01:37:08 [notice] 1#1: using the "epoll" event method
2025/11/03 01:37:08 [notice] 1#1: nginx/1.29.3
2025/11/03 01:37:08 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/11/03 01:37:08 [notice] 1#1: OS: Linux 6.8.0-87-generic
2025/11/03 01:37:08 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/11/03 01:37:08 [notice] 1#1: start worker processes
2025/11/03 01:37:08 [notice] 1#1: start worker process 28
2025/11/03 01:37:08 [notice] 1#1: start worker process 29
172.17.0.1 - - [03/Mar/2025:01:38:28 +0000] "GET / HTTP/1.1" 200 635 "-" "curl/8.5.0" "-"
adminsys@servidor-grupo2:~$ 

```

logs en tiempo real

```
docker exec nginx-grupo2 ls -la /usr/share/nginx/html/.
```

Se verificó dentro del contenedor que el archivo index.html estaba montado correctamente.

```
adminsys@servidor-grupo2:~$ docker exec nginx-grupo2 ls -la /usr/share/nginx/html/
total 12
drwxr-sr-x 2 1001 1006 4096 Nov  3 01:31 .
drwxr-xr-x 3 root root 4096 Oct 28 21:50 ..
-rw-r--r-- 1 1001 1006    0 Nov  3 00:37 .gitkeep
-rw-r--r-- 1 1001 1006  635 Nov  3 01:31 index.html
adminsys@servidor-grupo2:~$
```

Conclusiones

El proyecto desarrollado hasta la **fase 5** permitió aplicar de manera práctica los conocimientos adquiridos en la asignatura *Introducción al Software Libre*. Cada etapa representó un avance progresivo en la construcción de un servidor Linux automatizado, integrando conceptos de administración de sistemas, automatización, control de versiones y despliegue de servicios web mediante contenedores Docker.

En la **fase 1**, la configuración del entorno servidor con usuarios, grupos y permisos demostró la importancia de la organización y seguridad en un sistema multiusuario. La correcta asignación de privilegios y roles asegura que cada integrante del equipo pueda trabajar en un entorno controlado y con responsabilidades definidas.

En la **fase 2**, la creación del script de monitoreo y su automatización con cron evidenció cómo las tareas rutinarias pueden ser simplificadas mediante herramientas del sistema. Este paso no solo permitió obtener métricas periódicas del servidor, sino que también introdujo prácticas de administración proactiva, esenciales en entornos de producción.

La **fase 3** consolidó el uso de **Git** y **GitHub** como herramientas de control de versiones. La inicialización del repositorio local, la conexión con el remoto y la documentación en el archivo README.md reflejan la importancia de la colaboración y la trazabilidad en proyectos de software. Cada commit representa evidencia del trabajo realizado y asegura que el proyecto pueda ser replicado o mejorado en el futuro.

En la **fase 4**, la instalación y verificación de Docker marcaron un punto clave en la transición hacia la virtualización ligera. La ejecución del contenedor de prueba hello-world confirmó que el sistema estaba preparado para trabajar con imágenes y contenedores, abriendo la puerta a la implementación de servicios más complejos.

Finalmente, en la **fase 5**, el despliegue de un servidor web con Nginx dentro de un contenedor demostró la integración práctica de todos los conceptos anteriores. La creación del archivo index.html, la ejecución del contenedor y la verificación del servicio en el navegador evidenciaron que el grupo logró implementar un entorno funcional, reproducible y escalable.

En conjunto, estas fases muestran un avance significativo en la comprensión y aplicación de tecnologías libres. El proyecto no solo cumplió con los requerimientos técnicos hasta el punto 5, sino que también fortaleció competencias en administración de sistemas, automatización, trabajo colaborativo y despliegue de servicios web.

Este progreso constituye una base sólida para abordar las fases avanzadas del proyecto (Dockerfile personalizado, redes, volúmenes y pruebas de conectividad), que permitirán ampliar la funcionalidad y robustez del servidor. Además, el trabajo realizado evidencia la capacidad del grupo para enfrentar retos técnicos y documentarlos de manera clara y organizada, lo cual es fundamental en el ámbito profesional de la ingeniería de software.