



UNIVERSIDAD DE EL SALVADOR

Facultad Multidisciplinaria De Occidente
Departamento de Ingeniería y Arquitectura
Ingeniería en Desarrollo de Software



Asignatura:

INTRODUCCIÓN AL SOFTWARE LIBRE

Ciclo IV / Segundo año

Tema:
Proyecto Final

Coordinadora de carrera:
Ing. Ángela de Granillo

Tutor/a GT02:
Ing. Albaluz Escalante

Alumno/a:

Nombre	Carnet
José Israel Lemus Salguero	LS24009
Andrea Isabel Chávez Mejía	CM24080
José Ricardo González	GG22088

Fecha de entrega: 17-11-2025

1. Fase: Preparación del Entorno Servidor

1.1 Agregando un nombre al host. En este caso se agregó “servidor-grupo6”

El **Objetivo** de esta fase fue establecer la base del sistema operativo (Linux Mint) con una estructura de usuarios y permisos que simulan un entorno empresarial real, aplicando el **Principio del Mínimo Privilegio (PoLP)**.

```
josesalguero@josesalguero-VirtualBox:~$ sudo hostnamectl set-hostname servidor-grupo6
[sudo] password for josesalguero:
josesalguero@josesalguero-VirtualBox:~$ hostnamectl
  Static hostname: servidor-grupo6
            Icon name: computer-vm
              Chassis: vm 
        Machine ID: 24eeb0c6e8a14ff4829f73eefd1a409d
          Boot ID: ae0b12934e424a459201dd8d19091f06
      Virtualization: oracle
Operating System: Linux Mint 22.2
      Kernel: Linux 6.14.0-34-generic
    Architecture: x86-64
  Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
Firmware Version: VirtualBox
  Firmware Date: Fri 2006-12-01
  Firmware Age: 18y 11month
```

1.2 Creando grupos, usuarios con sus respectivos directorios personales

Usuarios y Grupos: Creamos usuarios con roles definidos (admins, tecnicos, visitante) y grupos específicos (sudo, soporte, web). Esto segmenta las responsabilidades, asegurando que el personal tenga acceso limitado a sus áreas de trabajo.

```
josesalguero@josesalguero-VirtualBox:~$ sudo groupadd soporte
josesalguero@josesalguero-VirtualBox:~$ sudo groupadd web
josesalguero@josesalguero-VirtualBox:~$ sudo useradd -m adminsyst
josesalguero@josesalguero-VirtualBox:~$ sudo useradd -m tecnico
josesalguero@josesalguero-VirtualBox:~$ sudo useradd -m visitante
josesalguero@josesalguero-VirtualBox:~$ sudo passwd adminsyst
New password:
Retype new password:
passwd: password updated successfully
josesalguero@josesalguero-VirtualBox:~$ sudo passwd tecnico
New password:
Retype new password:
passwd: password updated successfully
```

```
josesalguero@josesalguero-VirtualBox:~$ sudo passwd visitante
New password:
Retype new password:
passwd: password updated successfully
josesalguero@josesalguero-VirtualBox:~$ tail -n 5 /etc/group
soporte:x:1001:
web:x:1002:
adminsyst:x:1003:
tecnico:x:1004:
visitante:x:1005:
josesalguero@josesalguero-VirtualBox:~$ tail -n 5 /etc/passwd
sssd:x:121:126:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
vboxadd:x:997:1::/var/run/vboxadd:/bin/false
adminsyst:x:1001:1003::/home/adminsyst:/bin/sh
tecnico:x:1002:1004::/home/tecnico:/bin/sh
visitante:x:1003:1005::/home/visitante:/bin/sh
```

```
josesalguero@josesalguero-VirtualBox:~$ sudo usermod -aG soporte tecnico  
josesalguero@josesalguero-VirtualBox:~$ sudo usermod -aG web visitante  
josesalguero@josesalguero-VirtualBox:~$ sudo usermod -aG sudo adminsyst  
josesalguero@josesalguero-VirtualBox:~$ groups adminsyst  
adminsyst : adminsyst sudo  
josesalguero@josesalguero-VirtualBox:~$ groups tecnico  
tecnico : tecnico soporte  
josesalguero@josesalguero-VirtualBox:~$ groups visitante  
visitante : visitante web  
josesalguero@josesalguero-VirtualBox:~$ █
```

1.3 Estructura de directorios y permisos

Estructura de Directorios: La carpeta raíz /proyecto/ y sus subdirectorios (web/, datos/) se crearon para organizar el código y los datos del proyecto.

Permisos de Herencia (sgid): Aplicamos el *bit sgid* (chmod g+s) a los directorios como /proyecto/web/. Esta técnica asegura que **cualquier archivo creado dentro herede automáticamente el grupo propietario (web)**, simplificando la gestión de **permisos para los usuarios de ese grupo**

datos/: para archivos de configuración
web/: para archivos del sitio web
scripts/: para scripts de automatización
capturas/: para evidencias del proyecto

```
josesalguero@josesalguero-VirtualBox:~$ sudo usermod -aG soporte tecnico
josesalguero@josesalguero-VirtualBox:~$ sudo usermod -aG web visitante
josesalguero@josesalguero-VirtualBox:~$ sudo usermod -aG sudo adminsyst
josesalguero@josesalguero-VirtualBox:~$ groups adminsyst
adminsyst : adminsyst sudo
josesalguero@josesalguero-VirtualBox:~$ groups tecnico
tecnico : tecnico soporte
josesalguero@josesalguero-VirtualBox:~$ groups visitante
visitante : visitante web
josesalguero@josesalguero-VirtualBox:~$ sudo mkdir /proyecto
josesalguero@josesalguero-VirtualBox:~$ sudo mkdir -p /proyecto/{datos,web,scripts,capturas}
josesalguero@josesalguero-VirtualBox:~$ sudo chown :soporte /proyecto/datos
josesalguero@josesalguero-VirtualBox:~$ sudo chown :web /proyecto/web
josesalguero@josesalguero-VirtualBox:~$ sudo chmod g+s /proyecto/datos
sudo chmod g+s /proyecto/web
josesalguero@josesalguero-VirtualBox:~$ ls -ld /proyecto/datos
drwxr-sr-x 2 root soporte 4096 Oct 31 23:09 /proyecto/datos
josesalguero@josesalguero-VirtualBox:~$ ls -ld /proyecto/web
drwxr-sr-x 2 root web 4096 Oct 31 23:09 /proyecto/web
josesalguero@josesalguero-VirtualBox:~$ █
```

```
josesalguero@josesalguero-VirtualBox:~$ sudo nano /proyecto/scripts/reporte_sistema.sh
josesalguero@josesalguero-VirtualBox:~$ sudo nano /proyecto/scripts/reporte_sistema.sh
josesalguero@josesalguero-VirtualBox:~$ sudo chmod +x /proyecto/scripts/reporte_sistema.sh
josesalguero@josesalguero-VirtualBox:~$ /proyecto/scripts/reporte_sistema.sh
--- REPORTE DEL SISTEMA ---
Fecha y Hora: Fri Oct 31 11:18:35 PM CST 2025
Host: servidor-grupo6
Usuarios Conectados: 1
Espacio Libre en Disco: 13G
Memoria RAM Disponible: 1.1Gi
/proyecto/scripts/reporte_sistema.sh: line 24: docker: command not found
Contenedores Docker Activos: 0
-----
josesalguero@josesalguero-VirtualBox:~$
```

2. Fase: Automatización y Monitoreo 🕒

El **Objetivo** de esta fase fue implementar herramientas nativas de Linux para garantizar la recolección periódica y automática de datos del estado del servidor.

2.1 Creación de un archivo `reporte_sistema.sh`

- **Script Shell (.sh):** Creamos el script `reporte_sistema.sh` para consultar métricas clave del sistema (uso de RAM, disco, estado de contenedores Docker, etc.). Este script es la base para el monitoreo de la salud del servidor.

```
GNU nano 7.2                               josesalguero@josesalguero-VirtualBox: ~
#!/bin/bash                                 /proyecto/scripts/reporte_sistema.sh

# Nombre del script: reporte_sistema.sh
# Descripción: Genera un reporte básico del estado del sistema.

echo "--- REPORTE DEL SISTEMA ---"

# 1. Fecha y hora actual
echo "Fecha y Hora: $(date)"

# 2. Nombre del host
echo "Host: $(hostname)"

# 3. Número de usuarios conectados
echo "Usuarios Conectados: $(who | wc -l)"

# 4. Espacio libre en el disco principal (usando df -h)
echo "Espacio Libre en Disco: $(df -h / | tail -1 | awk '{print $4}')"

# 5. Memoria RAM disponible (usando free -h)
echo "Memoria RAM Disponible: $(free -h | grep Mem: | awk '{print $7}')"

# 6. Número de contenedores Docker activos (el comando funcionará después de instalar Docker)
echo "Contenedores Docker Activos: $(docker ps -q | wc -l 2>/dev/null || echo 'Docker No Instalado')"

echo "-----"
```

2.2 Automatización.

Cron: Usamos la utilidad **crontab** del usuario root para programar la ejecución del *script* cada 30 minutos. Esta automatización garantiza que el monitoreo sea **persistente y sin intervención manual**, guardando los resultados en `/var/log/proyecto/`.

```
josesalguero@josesalguero-VirtualBox:~$ sudo mkdir /var/log/proyecto
josesalguero@josesalguero-VirtualBox:~$ sudo crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
crontab: installing new crontab
```

```
josesalguero@josesalguero-VirtualBox: ~
GNU nano 7.2
/tmp/crontab.V3czCW/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/30 * * * * /proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1
```

```

no modification made
josesalguero@josesalguero-VirtualBox:~$ sudo crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/30 * * * * /proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1
josesalguero@josesalguero-VirtualBox:~$ 

```

3. Fase: Control de Versiones (Git y GitHub) 🎥

El **Objetivo** fue utilizar Git para la trazabilidad del código y la colaboración estructurada, usando GitHub como repositorio central y respaldo. Instalacion de git.

```

josesalguero@josesalguero-VirtualBox:/proyecto
*/30 * * * * /proyecto/scripts/reporte_sistema.sh >> /var/log/proyecto/reporte_sistema.log 2>&1
josesalguero@josesalguero-VirtualBox:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,806 kB of archives.
After this operation, 24.5 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/amd64 liberror-perl all 0.17029-2 [25.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 git-man all 1:2.43.0-1ubuntu7.3 [1,100 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 git amd64 1:2.43.0-1ubuntu7.3 [3,680 kB]
Fetched 4,806 kB in 2s (2,067 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 496952 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-2_all.deb ...
Unpacking liberror-perl (0.17029-2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.43.0-1ubuntu7.3_all.deb ...
Unpacking git-man (1:2.43.0-1ubuntu7.3) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.43.0-1ubuntu7.3_amd64.deb ...
Unpacking git (1:2.43.0-1ubuntu7.3) ...
Setting up liberror-perl (0.17029-2) ...
Setting up git-man (1:2.43.0-1ubuntu7.3) ...
Setting up git (1:2.43.0-1ubuntu7.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
josesalguero@josesalguero-VirtualBox:~$ git --version
git version 2.43.0

```

3.1 Crear repositorio remoto en GitHub.

```
josesalguero@josesalguero-VirtualBox:~$ sudo git init /proyecto
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /proyecto/.git/
josesalguero@josesalguero-VirtualBox:~$ cd /proyecto
josesalguero@josesalguero-VirtualBox:/proyecto$ sudo git add .
josesalguero@josesalguero-VirtualBox:/proyecto$ sudo git commit -m "Inicializacion del proyecto - Estructura de directorios, usuarios y script de monitoreo (Fases 1 y 2)"
[master (root-commit) 606e807] Inicializacion del proyecto - Estructura de directorios, usuarios y script de monitoreo (Fases 1 y 2)
Committer: root <root@servidor-grupo6.grupoinfotech.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
    git config --global --edit
After doing this, you may fix the identity used for this commit with:
    git commit --amend --reset-author
1 file changed, 26 insertions(+)
create mode 100755 scripts/reporte_sistema.sh
josesalguero@josesalguero-VirtualBox:/proyecto$
```

- **Flujo Git:** Se inicializó el repositorio local, se hicieron *commits* con mensajes descriptivos para documentar cada cambio funcional. Se usó un **Token de Acceso Personal (PAT)** en lugar de una contraseña para autenticar las operaciones con GitHub.

3.2 Creando repositorio remoto para proyecto.

```
josesalguero@josesalguero-VirtualBox:/proyecto$ git config --global user.email "ls24009@ues.edu.sv"
josesalguero@josesalguero-VirtualBox:/proyecto$ git config --global user.name "LS24009"
josesalguero@josesalguero-VirtualBox:/proyecto$ sudo git branch -M main
josesalguero@josesalguero-VirtualBox:/proyecto$ sudo git remote add origin https://github.com/LS24009-LS/proyecto-linux-grupo6.git
josesalguero@josesalguero-VirtualBox:/proyecto$ sudo git push -u origin main
Username for 'https://github.com': LS24009-LS
Password for 'https://LS24009-LS@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 778 bytes | 778.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/LS24009-LS/proyecto-linux-grupo6.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
josesalguero@josesalguero-VirtualBox:/proyecto$ sudo nano README.md
josesalguero@josesalguero-VirtualBox:/proyecto$
```

Connected to th...

3.3 Abriendo el archivo Readme.md

```
josesalguero@josesalguero-VirtualBox:/ proyecto$ sudo nano README.md
josesalguero@josesalguero-VirtualBox:/ proyecto$ sudo git add README.md
sudo git commit -m "Agregado README.md con descripcion del proyecto"
sudo git push
[main 0b21da3] Agregado README.md con descripcion del proyecto
Committer: root <root@servidor-grupo6.grupoinfotech.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 20 insertions(+)
create mode 100644 README.md
Username for 'https://github.com': LS24009-LS
Password for 'https://LS24009-LS@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 954 bytes | 954.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/LS24009-LS/proyecto-linux-grupo6.git
  606e807..0b21da3 main -> main
josesalguero@josesalguero-VirtualBox:/ proyecto$
```

3.4 Editando el archivo README.me del repositorio

```
GNU nano 7.2                               README.md *
```

```
# Proyecto Servidor Linux Automatizado (Grupo 6)

Este proyecto tiene como objetivo implementar y automatizar un entorno de servidor Linux utilizando diversas tecnologías clave de la administración.

## Fases Completadas

### 1. Preparación del Entorno Servidor
* **Hostname:** Configurado como `servidor-grupo6`.
* **Usuarios:** Creados `adminsys` (con sudo), `técnico` (grupo `soporte`), y `visitante` (grupo `web`).
* **Estructura de Directorios:** Creada en `/proyecto/` con subdirectorios y permisos de herencia (`sgid`) para los grupos `soporte` y `web`.

### 2. Automatización y Monitoreo
* **Script:** Creado `reporte_sistema.sh` para recopilar datos clave del servidor (host, usuarios, RAM, disco, Docker).
* **Automatización:** Tarea de Cron configurada para ejecutar `reporte_sistema.sh` cada 30 minutos, guardando la salida en `/var/log/proyecto`.

### 3. Control de Versiones
* Repositorio Git local inicializado y sincronizado con el repositorio remoto de GitHub.

## Próximos Pasos
Instalación y configuración de Docker para la contenerización del servidor web Nginx.
```

File Name to Write: README.md

^D Help M-D DOS Format M-A Append M-B Backup File

4. Fase: Contenerización (Docker) 🚀

El **Objetivo** fue instalar Docker y configurarlo para un despliegue de aplicaciones aislado, ligero y portable, la base de la infraestructura moderna.

4.1 Instalación y prueba de Docker (comandos que se usa para probar que Docker está instalado correctamente).

```
josesalguero@servidor-grupo6:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:56433a6be3fd188089fb548eae3d91df3ed0d6589f7c2656121b911198df065
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/
josesalguero@servidor-grupo6:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
74d74cf0b74f        hello-world        "/hello"           49 seconds ago    Exited (0) 48 seconds ago
josesalguero@servidor-grupo6:~$ █
```

4.2 Probando que los contenedores funcionan bien en tu sistema.)

```
josesalguero@servidor-grupo6:~$ cd /proyecto/web
josesalguero@servidor-grupo6:/proyecto/web$ nano index.html
josesalguero@servidor-grupo6:/proyecto/web$ sudo nano index.html
[sudo] password for josesalguero:
josesalguero@servidor-grupo6:/proyecto/web$ ls -l
total 4
-rw-r--r-- 1 root web 545 Nov  1 10:34 index.html
josesalguero@servidor-grupo6:/proyecto/web$ sudo chown :web index.html
josesalguero@servidor-grupo6:/proyecto/web$ docker run -d \
  --name servidor-web-nginx \
  -p 8080:80 \
  -v /proyecto/web:/usr/share/nginx/html:ro \
  --restart unless-stopped \
  nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
38513bd72563: Pull complete
a0a6ab141558: Pull complete
0e66847a3920: Pull complete
1bace2083289: Pull complete
89df300a082a: Pull complete
35fb9fffa6621: Pull complete
5545b08f9d26: Pull complete
Digest: sha256:f547e3d0d5d02f7009737b284abc87d808e4252b42dceea361811e9fc606287f
Status: Downloaded newer image for nginx:latest
f79db68bd7461951dbabdd7bfdad71a0a6f4696d4ffcf6b6eed83f4e9059e6bc
josesalguero@servidor-grupo6:/proyecto/web$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
f79db68bd746        nginx:latest      "/docker-entrypoint..."   2 minutes ago    Up 2 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   servidor-web-nginx
```

```
josesalguero@servidor-grupo6:/proyecto/web$ curl localhost:8080
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Proyecto Grupo 6 - Nginx Docker</title>
</head>
<body>
  <h1>Hola desde el Servidor Web Containerizado!</h1>
  <p>Este es el servidor Nginx ejecutándose en un contenedor Docker, montado desde el directorio local /proyecto/web del servidor-grupo6.</p>
  <p>El proyecto avanza satisfactoriamente. Hora del servidor: <span id="time"></span></p>
</body>
<script>
  document.getElementById('time').innerText = new Date().toLocaleTimeString();
</script>
</html>
josesalguero@servidor-grupo6:/proyecto/web$ █
```

4.2 Ver todos los logs. Comando de Docker que se utiliza para ver los registros (logs) de un contenedor en ejecución.

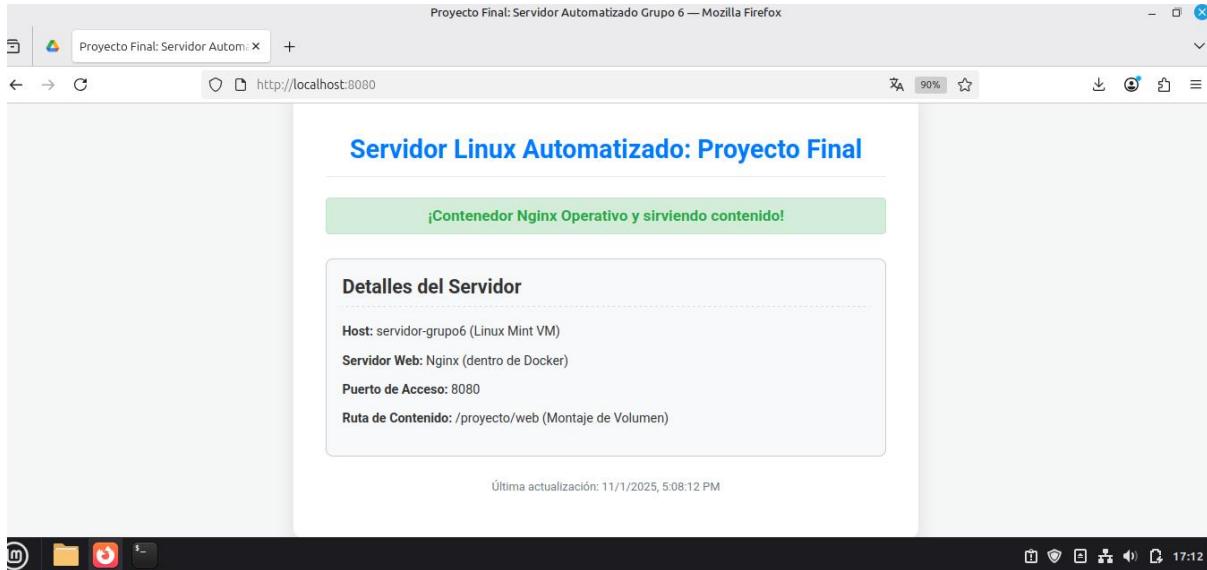
```
josealguero@servidor-grupo6:/proyecto/web$ docker logs servidor-web-nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
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.envsh
/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
2025/11/01 16:39:28 [notice] 1#1: using the "epoll" event method
2025/11/01 16:39:28 [notice] 1#1: nginx/1.29.3
2025/11/01 16:39:28 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/11/01 16:39:28 [notice] 1#1: OS: Linux 6.14.0-34-generic
2025/11/01 16:39:28 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/11/01 16:39:28 [notice] 1#1: start worker processes
2025/11/01 16:39:28 [notice] 1#1: start worker process 29
172.17.0.1 - - [01/Nov/2025:16:42:48 +0000] "GET / HTTP/1.1" 200 545 "-" "curl/8.5.0"
172.17.0.1 - - [01/Nov/2025:16:46:31 +0000] "GET / HTTP/1.1" 200 545 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:144.0) Gecko/20100101 Firefox/144.0"
2025/11/01 16:46:31 [error] 29#29: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8080", referer: "http://localhost:8080"
172.17.0.1 - - [01/Nov/2025:16:46:31 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://localhost:8080/" "Mozilla/5.0 (X11; Linux x86_64; rv:144.0) Gecko/20100101 Firefox/144.0"
172.17.0.1 - - [01/Nov/2025:18:58:09 +0000] "GET / HTTP/1.1" 200 545 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:144.0) Gecko/20100101 Firefox/144.0"
2025/11/01 18:58:10 [error] 29#29: *3 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8080", referer: "http://localhost:8080"
172.17.0.1 - - [01/Nov/2025:18:58:10 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://localhost:8080/" "Mozilla/5.0 (X11; Linux x86_64; rv:144.0) Gecko/20100101 Firefox/144.0"
josealguero@servidor-grupo6:/proyecto/web$
```

5. Fase: Servidor Web Containerizado

El **Objetivo** fue desplegar el servicio web principal del proyecto de manera eficiente, segura y con alta disponibilidad.

- **Ejecución Persistente:** El contenedor se lanzó con el modo **detached (-d)** y la política **--restart unless-stopped**. Esto garantiza que Nginx se inicie automáticamente cada vez que la VM se encienda, asegurando la **alta disponibilidad** del servicio.

Probando servicios con <http://localhost: 8080>



Montaje de Volumen:

Se utilizó la bandera `-v /proyecto/web:/usr/share/nginx/html:ro` para montar el directorio local dentro del contenedor.

- Esta técnica permite separar el código del servidor (por ejemplo, `index.html`).
- El modo `:ro` (solo lectura) añade seguridad, previniendo cambios accidentales o maliciosos desde el contenedor.
-

Verificación Final:

Se comprobó el acceso al servicio a través del puerto 8080 del host (mapeado al puerto 80 del contenedor).

La revisión de logs mediante `docker logs` confirmó la recepción de peticiones con código de estado 200 (OK), demostrando que Nginx está sirviendo el contenido correctamente.