#### **EJERCICIO PRÁCTICO - FASE 1 ✅ 1. Instalación del servidor Apache**

sudo apt update

sudo apt install apache2 -y

* Comprobamos que el servicio está activo con:  
   sudo systemctl status apache2

#### **✅ 2. Creación de página de prueba en HTML/PHP**

1. Instalamos PHP:

sudo apt install php libapache2-mod-php -y

1. Creamos el archivo de prueba:

sudo nano /var/www/html/info.php

1. Contenido del archivo:

<?php phpinfo(); ?>

1. Verificamos la página desde navegador:  
    http://[IP-del-servidor]/info.php

#### **✅ 3. Configuración de un Virtual Host (vhost)**

1. Creamos estructura del proyecto:

sudo mkdir -p /var/www/proyecto1

sudo chown -R $USER:$USER /var/www/proyecto1

1. Creamos un index:

echo "<h1>Bienvenido a Proyecto 1</h1>" | sudo tee /var/www/proyecto1/index.html

1. Creamos el archivo del vhost:

sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/proyecto1.conf

1. Editamos el nuevo vhost:

sudo nano /etc/apache2/sites-available/proyecto1.conf

* Cambiamos las líneas:

DocumentRoot /var/www/proyecto1

ServerName proyecto1.local

1. Activamos el sitio:

sudo a2ensite proyecto1.conf

sudo systemctl reload apache2

1. (Opcional) Añadir a /etc/hosts en el cliente:

192.168.0.XXX proyecto1.local  
  
**EJERCICIO PRÁCTICO - FASE 2**

### **🛠️ Pasos realizados:**

1. **🔧 Instalación de vsftpd:**

**Comando:  
  
sudo apt install vsftpd**

**Servicio habilitado al arranque con:  
  
sudo systemctl enable vsftpd**

1. **🧩 Corrección de errores:**

**Se creó el directorio necesario:  
  
sudo mkdir -p /var/run/vsftpd/empty**

* + **Se revisó y ajustó el archivo /etc/vsftpd.conf para eliminar errores y dejarlo funcional:**
    - **Habilitado modo local.**
    - **Deshabilitado el acceso anónimo.**
    - **Activada la escritura y el aislamiento de usuarios (chroot\_local\_user=YES).**

1. **👤 Creación de usuarios FTP:**

**Usuarios con acceso restringido a directorios específicos, usando:  
  
sudo adduser usuarioX**

1. **🔒 Configuración de permisos:**
   * **Se limitaron los permisos de cada usuario a su carpeta personal en /home/usuarioX.**
   * **Se reforzó la seguridad evitando accesos no deseados.**
2. **📡 Prueba de conexión FTP:**
   * **Uso de FileZilla con IP del servidor, puerto 21 y credenciales del usuario.**
   * **Verificación de transferencia de archivos correctamente.**

**EJERCICIO PRÁCTICO - FASE 3**

#### **✅ 1. Configuración del Firewall (UFW)**

* **Se habilitó UFW (Uncomplicated Firewall).**
* **Se configuraron reglas para permitir solo conexiones desde la red local (192.168.0.0/24).**
* **Se bloquearon todas las conexiones externas no autorizadas.**

**Reglas aplicadas:  
  
sudo ufw allow from 192.168.0.0/24**

**sudo ufw allow 22 # Asegurando acceso SSH interno**

**sudo ufw allow 80 # Asegurando acceso web desde red interna**

**sudo ufw enable**

#### **✅ 2. Monitorización de accesos al servidor web**

**Se revisaron los registros de Apache en tiempo real para detectar accesos:  
  
tail -f /var/log/apache2/access.log**

* **Se buscó actividad inusual y posibles intentos sospechosos.**

#### **✅ 3. Protección avanzada con Fail2Ban**

* **Se instaló Fail2Ban para proteger el sistema contra ataques de fuerza bruta.**

**Se activó el filtro SSH:  
  
sudo apt install fail2ban**

**sudo systemctl enable fail2ban**

**sudo systemctl start fail2ban**

* **Se revisó el estado de las cárceles (jails) y los intentos bloqueados:  
    
  sudo fail2ban-client status sshd**

**EJERCICIO EMPRESARIAL - FASE 1**

### **✅ Fase 1: Análisis en tiempo real del sistema**

**🖥️ Hemos analizado el rendimiento del servidor Ubuntu 24.04.2 en tiempo real utilizando varias herramientas clave:**

**🔹 1. top  
 Visualización en tiempo real de procesos, CPU, memoria y carga.  
 ✔️ Identificamos el proceso más exigente en CPU y memoria.**

**🔹 2. htop  
 Herramienta más visual y detallada (instalada manualmente).  
 ✔️ Permite ordenar procesos, ver árboles de ejecución y controlar fácilmente los recursos.**

**🔹 3. uptime  
 Consultamos el tiempo encendido del sistema y la carga promedio en los últimos 1, 5 y 15 minutos.  
 ✔️ Ideal para saber si el servidor ha estado bajo presión.**

**🔹 4. free -m  
 Análisis de memoria RAM y Swap en megas.  
 ✔️ Confirmamos cuánto se está usando y cuánto queda libre.**

**EJERCICIO EMPRESARIAL - FASE 2**

#### **🔹 1. Finalización de procesos no esenciales**

**Se identificó un proceso no crítico mediante ps aux y se finalizó correctamente usando kill. En caso de resistencia, se aplicó kill -9.**

#### **🔹 2. Cambio de prioridad (renice)**

**Se localizó un proceso activo con alto uso de CPU. Usando renice, se ajustó su prioridad para mejorar la respuesta del sistema y dar preferencia a otros procesos más importantes.**

#### **🔹 3. Gestión de procesos en segundo plano**

**Se ejecutó un proceso (por ejemplo, sleep) en segundo plano (&) y posteriormente se recuperó al primer plano con fg, demostrando control completo sobre el flujo de tareas.**

#### **🔹 4. Ejecución con baja prioridad (nice)**

**Se lanzó una copia de archivos usando nice -n 19, permitiendo que esta operación tuviera la menor prioridad posible para no entorpecer el rendimiento general del servidor.**

**EJERCICIO EMPRESARIAL - FASE 3  
  
🔹 1. Monitorización con vmstat:  
 Se generó un informe instantáneo del uso de memoria, procesos, I/O y CPU mediante el comando:**

**vmstat > /srv/logs/vmstat.log**

**📁 Resultado guardado en: /srv/logs/vmstat.log.**

**🔹 2. Registro periódico de uso de recursos con top:  
 Se programó una tarea en crontab para que cada 5 minutos guarde un estado del sistema con top:**

**\*/5 \* \* \* \* top -b -n 1 >> /srv/logs/top.log**

**📁 Log acumulativo en: /srv/logs/top.log.**

**🔹 3. Análisis de uso de disco con iotop:  
 Se instaló y utilizó iotop para monitorizar en tiempo real qué procesos realizan más lecturas/escrituras en disco.**

**sudo apt install iotop**

**sudo iotop**

**📊 Herramienta útil para detectar cuellos de botella en disco.  
  
EJERCICIO EMPRESARIAL - FASE 4  
  
🔧 Herramienta usada:  
 stress, instalada mediante sudo apt install stress.**

**🧪 Pruebas realizadas:**

#### **1. Carga de CPU**

* **Comando ejecutado: stress --cpu 4 --timeout 60**
* **Resultado:**
  + **Uso de CPU: ≈ 70%**
  + **Uso de memoria: 0%**
  + **Procesos generados: 4**
  + **Temperatura: No disponible**

#### **2. Carga de Memoria RAM**

* **Comando ejecutado: stress --vm 1 --vm-bytes 512M --timeout 60**
* **Resultado:**
  + **Uso de CPU: ≈ 160%**
  + **Uso de memoria: ≈ 2%**
  + **Procesos generados: 1**
  + **Temperatura: No disponible**

#### **3. Carga de Disco**

* **Comando ejecutado: stress --hdd 4 --timeout 60**
* **Resultado:**
  + **Uso de CPU: ≈ 7%**
  + **Uso de memoria: 0%**
  + **Procesos generados: 4**
  + **Temperatura: No disponible**

**📌 Conclusiones:**

* **La CPU responde fuertemente a la carga de procesos simultáneos.**
* **El sistema gestiona bien la memoria y mantiene estabilidad con cargas controladas.**
* **La carga de disco apenas impacta en CPU y RAM, pero genera múltiples procesos.**