

## LPIC-2 / Examen 208 - Servicios Web - Ejercicios

*Nota: Estos ejercicios implican instalar software y modificar archivos de configuración. Realízalos **SIEMPRE en una VM de prueba dedicada**. Asegúrate de que tu VM tiene acceso a internet para la instalación de paquetes y de que tu firewall permite tráfico en el puerto 80.*

### Ejercicio 8.1.1: Instalando un Servidor Web (Elige Apache o Nginx)

- **Objetivo:** Instalar el software de servidor web en tu VM.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (sudo). Conexión a internet. Decidir si instalar Apache o Nginx (generalmente se instala el que es más común en tu distribución o el que te interesa más).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Actualiza la lista de paquetes:** `sudo apt update` (Debian/Ubuntu) o `sudo dnf check-update` (Red Hat/CentOS/Fedora).
  3. **Instala el servidor web elegido (Diferencias):**
    - Para Apache en Debian/Ubuntu: `sudo apt install apache2`.
    - Para Apache en Red Hat/CentOS/Fedora: `sudo dnf install httpd`.
    - Para Nginx (ambas): `sudo apt install nginx` o `sudo dnf install nginx`.

### Ejercicio 8.1.2: Gestión del Servicio del Servidor Web

- **Objetivo:** Asegurarse de que el servicio del servidor web está corriendo y habilitado para iniciarse al arrancar.
- **Requisitos:** Servidor web instalado. Privilegios de superusuario (sudo).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Identifica el nombre del servicio (Diferencias):** `apache2` (Debian/Ubuntu), `httpd` (Red Hat/CentOS/Fedora), `nginx` (ambas).
  3. **Habilita el servicio para iniciar al arrancar:** Ejecuta `sudo systemctl enable <nombre_servicio>`.
  4. **Inicia el servicio:** Ejecuta `sudo systemctl start <nombre_servicio>`.
  5. **Verifica el estado del servicio:** Ejecuta `systemctl status <nombre_servicio>`. Debería estar active (running).

### Ejercicio 8.1.3: Verificando Reglas de Firewall para el Puerto 80

- **Objetivo:** Asegurarse de que el firewall permite el tráfico web (HTTP).
- **Requisitos:** Privilegios de superusuario (sudo). Identificar la herramienta de firewall activa (Ej. 5.2.5).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.

2. **Si usas firewalld:** Ejecuta `sudo firewall-cmd --zone=<zona> --list-services` o `sudo firewall-cmd --zone=<zona> --list-ports`. Busca el servicio `http` o el puerto `80/tcp`. Si no está, añádelo: `sudo firewall-cmd --zone=<zona> --add-service=http --permanent` y `sudo firewall-cmd --reload`.
3. **Si usas ufw:** Ejecuta `sudo ufw status`. Busca reglas para el puerto 80 TCP. Si no están, añádelas: `sudo ufw allow 80/tcp`.
4. **Si usas iptables directamente:** Ejecuta `sudo iptables -L -v -n`. Busca reglas que permitan tráfico a puerto 80 TCP en la cadena INPUT.

#### Ejercicio 8.1.4: Localizando y Explorando Archivos de Configuración

- **Objetivo:** Encontrar los archivos de configuración principales y de sitios/módulos.
- **Requisitos:** Servidor web instalado. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Visualiza el archivo de configuración principal (Diferencias):**
    - Apache Debian/Ubuntu: `sudo less /etc/apache2/apache2.conf`.
    - Apache Red Hat/CentOS/Fedora: `sudo less /etc/httpd/conf/httpd.conf`.
    - Nginx (ambas): `sudo less /etc/nginx/nginx.conf`.
    - Busca directivas `Include` o `include` que apunten a otros directorios o archivos.
  3. **Explora directorios de configuración por sitio (Diferencias):**
    - Apache Debian/Ubuntu: `ls -l /etc/apache2/sites-available/` y `ls -l /etc/apache2/sites-enabled/`.
    - Apache Red Hat/CentOS/Fedora: `ls -l /etc/httpd/conf.d/`.
    - Nginx (convención común): `ls -l /etc/nginx/sites-available/` y `ls -l /etc/nginx/sites-enabled/`.
  4. **Explora directorios de configuración de módulos (Diferencias Apache):**
    - Apache Debian/Ubuntu: `ls -l /etc/apache2/mods-available/` y `ls -l /etc/apache2/mods-enabled/`.
    - Apache Red Hat/CentOS/Fedora: `ls -l /etc/httpd/conf.modules.d/`.
  5. **Identifica el Document Root por defecto:** Busca la directiva `DocumentRoot` (Apache) o `root` (Nginx) en el archivo de configuración principal o en el Virtual Host/Server Block por defecto. Anota el directorio (generalmente `/var/www/html/`).

#### Ejercicio 8.1.5: Accediendo a la Página Web por Defecto

- **Objetivo:** Verificar que el servidor web responde a las solicitudes HTTP básicas.

- **Requisitos:** Servidor web corriendo. Puerto 80 abierto en el firewall.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Accede a la página por defecto usando curl:** Ejecuta `curl http://localhost`. Debería mostrar el código HTML de la página de bienvenida por defecto del servidor web.
  3. **Accede usando la IP de tu VM (si la conoces):** Ejecuta `curl http://<ip_de_tu_vm>`.
  4. **(Opcional) Desde otra máquina en la misma red, accede en un navegador a** `http://<ip_de_tu_vm>`.

### Ejercicio 8.1.6: (Conceptual) Configurando un Virtual Host (Apache) o Server Block (Nginx)

- **Objetivo:** Entender cómo configurar el servidor web para alojar un segundo sitio.
- **Requisitos:** Privilegios de superusuario (sudo). Directorios de configuración identificados.
- **Desarrollo Paso a Paso (Conceptual, elige el de tu servidor web):**
  - **Apache (Debian/Ubuntu - usando a2ensite):**
    1. Crea un directorio para los archivos del nuevo sitio: `sudo mkdir -p /var/www/mywebsite/html`.
    2. Crea una página de índice simple: `echo "<h1>Mi Sitio Web</h1>" | sudo tee /var/www/mywebsite/html/index.html`.
    3. Crea un archivo de configuración de Virtual Host en `/etc/apache2/sites-available/mywebsite.conf` (sudo `vi ...`):
 

```

Apache

<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName mywebsite.local # Un nombre de dominio para
este sitio
    DocumentRoot /var/www/mywebsite/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
          
```
    4. Habilita el sitio: `sudo a2ensite mywebsite.conf`. Esto crea un enlace simbólico en `sites-enabled`.
    5. Verifica la configuración de Apache: `sudo apache2ctl configtest`.
    6. Recarga la configuración de Apache: `sudo systemctl reload apache2`.
    7. **(Para probar):** Necesitas que el nombre `mywebsite.local` resuelva a la IP de tu VM (añade una entrada en `/etc/hosts` de tu máquina cliente o en un servidor DNS local). Luego accede desde un navegador a `http://mywebsite.local`.
  - **Apache (Red Hat/CentOS/Fedora - usando conf.d):**

1. Crea un directorio para los archivos del nuevo sitio: `sudo mkdir -p /var/www/mywebsite/html`.
2. Crea una página de índice simple: `echo "<h1>Mi Sitio Web</h1>" | sudo tee /var/www/mywebsite/html/index.html`.
3. Crea un archivo de configuración de Virtual Host en `/etc/httpd/conf.d/mywebsite.conf` (`sudo vi ...`):  
Apache
 

```
<VirtualHost *:80>
    ServerName mywebsite.local
    DocumentRoot /var/www/mywebsite/html
    ErrorLog /var/log/httpd/mywebsite_error.log
    CustomLog /var/log/httpd/mywebsite_access.log combined
</VirtualHost>
```
4. Verifica la configuración de Apache: `sudo httpd -t`.
5. Recarga la configuración de Apache: `sudo systemctl reload httpd`.
6. **(Para probar):** Igual que en Debian, asegura que `mywebsite.local` resuelva a la IP de tu VM.

- **Nginx (Usando `sites-available/sites-enabled`):**

1. Crea un directorio para los archivos del nuevo sitio: `sudo mkdir -p /var/www/mywebsite/html`.
2. Crea una página de índice simple: `echo "<h1>Mi Sitio Web</h1>" | sudo tee /var/www/mywebsite/html/index.html`.
3. Crea un archivo de configuración de Server Block en `/etc/nginx/sites-available/mywebsite.conf` (`sudo vi ...`):  
Nginx
 

```
server {
    listen 80;
    server_name mywebsite.local; # Un nombre de dominio para
    este sitio
    root /var/www/mywebsite/html;
    index index.html;
    error_log /var/log/nginx/mywebsite_error.log;
    access_log /var/log/nginx/mywebsite_access.log;
}
```
4. Crea un enlace simbólico a `sites-enabled`: `sudo ln -s /etc/nginx/sites-available/mywebsite.conf /etc/nginx/sites-enabled/`.
5. Verifica la configuración de Nginx: `sudo nginx -t`.
6. Recarga la configuración de Nginx: `sudo systemctl reload nginx`.

7. **(Para probar):** Igual que en Apache, asegura que `mywebsite.local` resuelva a la IP de tu VM.