

Arranque y Parada de Registros

Francisco Javier Otero Herrero

Grupo ATU

4-4-2025

Arranque y Parada de Registros

Arranque y Parada de Registros

➤ **Actividad 2**

Responder a las siguientes preguntas:

Pregunta 1:

Suponiendo que nuestra intención es montar un blog generalista y no queremos gastarnos nada, ¿qué opción utilizarías?

Si la intención es montar un blog generalista sin gastar dinero, existen varias opciones gratuitas que se pueden considerar. Algunas de ellas son:

- WordPress.com
- Blogger (Blogspot)
- Medium
- Wix
- Tumblr

En mi caso yo me decantaría por **WordPress.com**, ya que es un CMS que ya he utilizado en el pasado, concretamente el proyecto de final del módulo de grado superior que cursé años atrás, de Aplicaciones de Desarrollo Web.

WordPress.com es una plataforma gratuita para crear blogs y sitios web utilizando el popular sistema de gestión de contenidos **(CMS) WordPress**.

• **Ventajas:**

- *Fácil de usar, incluso para principiantes.*
- *No necesitas conocimientos técnicos avanzados.*
- *Ofrece plantillas gratuitas y personalizables.*
- *Incluye herramientas básicas de SEO y estadísticas.*
- *No necesitas preocuparte por el alojamiento, ya que está incluido.*

• **Limitaciones del plan gratuito:**

- *El dominio será algo como tublog.wordpress.com.*
- *Publicidad que no puedes eliminar.*
- *Almacenamiento limitado (3 GB en el plan gratuito).*
- *Funcionalidades avanzadas (como plugins personalizados) están bloqueadas.*

Arranque y Parada de Registros

Pregunta 2:

Suponiendo que tenemos un servidor Apache funcionando sobre Linux, ¿cómo se realizarían los procesos de arranque y parada? Explica brevemente qué son los archivos logs.

El servidor **Apache** puede iniciarse, detenerse o reiniciarse utilizando comandos específicos que interactúan con el sistema de gestión de servicios del sistema operativo. A continuación, vamos a explicar cómo realizar estos procesos:

a) Arrancar el servidor Apache

Para iniciar el servidor Apache, podemos usar uno de los siguientes comandos dependiendo de la distribución de **Linux**:

- En sistemas con **systemd** (como Ubuntu 16.04+, CentOS 7+, Debian 8+):

`sudo systemctl start apache2` → Para Debian/Ubuntu

`sudo systemctl start httpd` → Para CentOS/RHEL

- En sistemas más antiguos (SysVinit):

`sudo service apache2 start` → Para Debian/Ubuntu

`sudo service httpd start` → Para CentOS/RHEL

Esto inicia el servicio **Apache** y lo pone en funcionamiento para atender solicitudes **HTTP**.

b) Detener el servidor Apache

Para detener el servidor **Apache**, usa uno de los siguientes comandos:

- Con **systemd**:

`sudo systemctl stop apache2` → Para Debian/Ubuntu

`sudo systemctl stop httpd` → Para CentOS/RHEL

- Con **SysVinit**:

`sudo service apache2 stop` → Para Debian/Ubuntu

`sudo service httpd stop` → Para CentOS/RHEL

Esto detiene completamente el servicio Apache, dejando de atender solicitudes HTTP.

Arranque y Parada de Registros

c) Reiniciar el servidor Apache:

Si se realizan cambios en la configuración del servidor y se necesitan aplicar, podemos reiniciar **Apache**:

- Con **systemd**:

`sudo systemctl restart apache2` → Para Debian/Ubuntu

`sudo systemctl restart httpd` → Para CentOS/RHEL

- Con **SysVinit**:

`sudo service apache2 restart` → Para Debian/Ubuntu

`sudo service httpd restart` → Para CentOS/RHEL

d) Recargar la configuración sin interrumpir el servicio

Si solo quieres aplicar cambios en la configuración sin detener el servicio (por ejemplo, después de editar un archivo .conf), puedes recargar **Apache**:

- Con **systemd**:

`sudo systemctl reload apache2` → Para Debian/Ubuntu

`sudo systemctl reload httpd` → Para CentOS/RHEL

- Con **SysVinit**:

`sudo service apache2 reload` → Para Debian/Ubuntu

`sudo service httpd reload` → Para CentOS/RHEL

Este comando aplica los cambios sin interrumpir las conexiones activas.

e) Verificar el estado del servidor Apache:

Puedes verificar si el servidor Apache está en funcionamiento con:

- Con **systemd**:

`sudo systemctl status apache2` → Para Debian/Ubuntu

`sudo systemctl status httpd` → Para CentOS/RHEL

- Con **SysVinit**:

`sudo service apache2 status` → Para Debian/Ubuntu

`sudo service httpd status` → Para CentOS/RHEL

Esto muestra información sobre el estado actual del servicio.

Arranque y Parada de Registros

• *¿Qué son los archivos logs?*

Los **archivos logs** son registros generados por el **servidor Apache** que contienen información detallada sobre su actividad. Estos archivos son esenciales para monitorear el rendimiento, detectar problemas y analizar el tráfico del servidor. **Apache** genera dos tipos principales de logs:

✓ **Log de Acceso:**

Registra todas las solicitudes **HTTP** que llegan al servidor. Incluye detalles como la *dirección IP del cliente, la fecha y hora de la solicitud, el método HTTP utilizado (GET, POST, etc.), la URL solicitada, el código de respuesta HTTP y el tamaño de los datos enviados.*

✓ **Log de Errores:**

Registra los errores y advertencias que ocurren durante la ejecución del servidor. Incluye *mensajes de error, advertencias y detalles sobre problemas técnicos (por ejemplo, permisos incorrectos, fallos en módulos, etc.)*. Las ubicaciones de los logs generalmente se encuentran en:

En **Debian/Ubuntu** →

`/var/log/apache2/access.log`

`/var/log/apache2/error.log`

En **CentOS/RHEL** →

`/var/log/httpd/access_log`

`/var/log/httpd/error_log`

➤ **Importancia de los logs:**

- **Monitoreo:** Ayudan a identificar patrones de uso y detectar picos de tráfico.
- **Depuración:** Permiten diagnosticar problemas técnicos o errores en la configuración.
- **Seguridad:** Facilitan la identificación de actividades sospechosas, como intentos de acceso no autorizados o ataques.