INSTALACIÓN Y CONFIGURACIÓN DE SERVIDORES APACHE Y NGINX

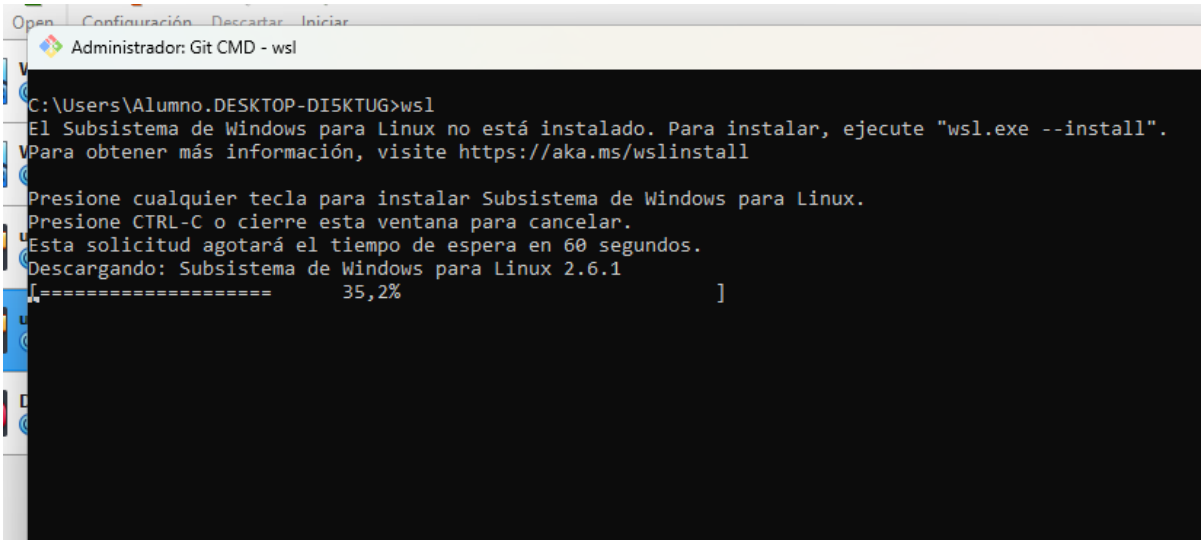
Adrián Aceitón Palomo

30/09/2025

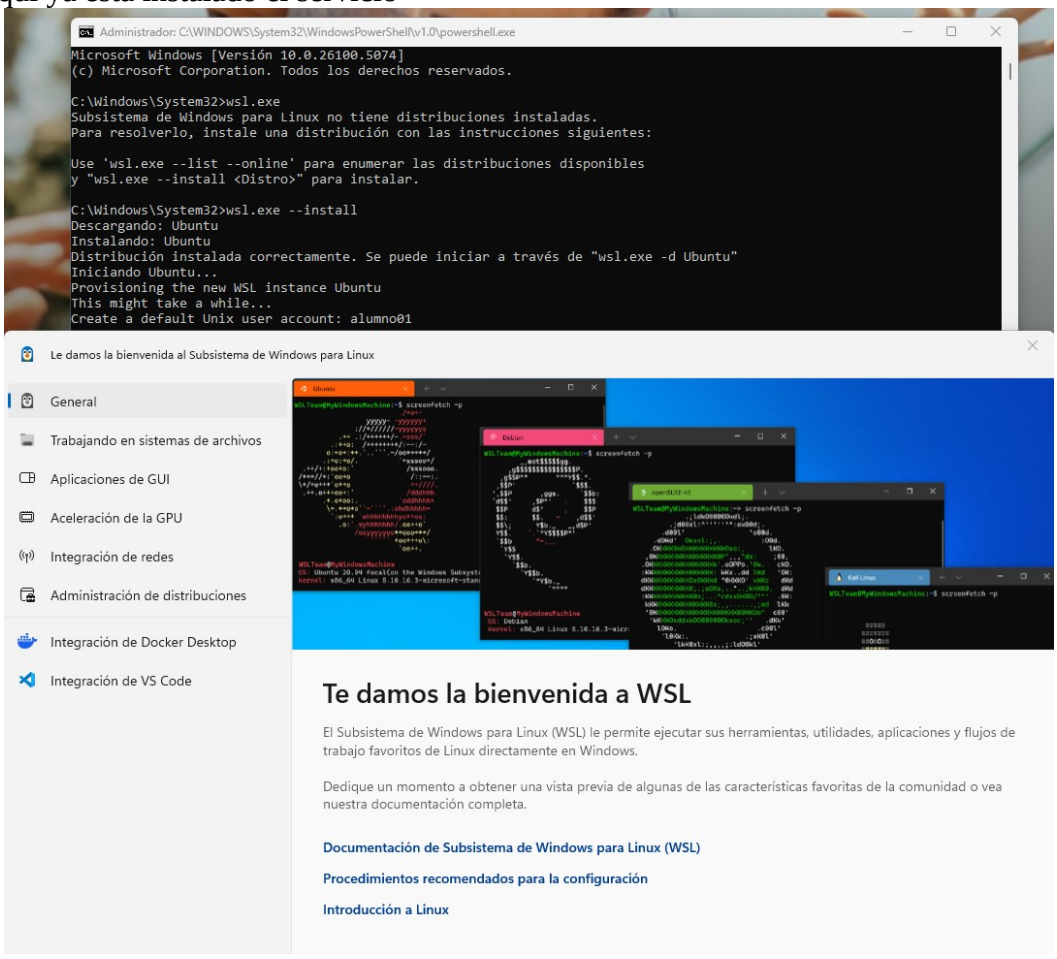
Adrián Aceitón Palomo. 2º ASIR

Instalación de Ubuntu Desktop y algunos servicios: Arquitectura en la nube

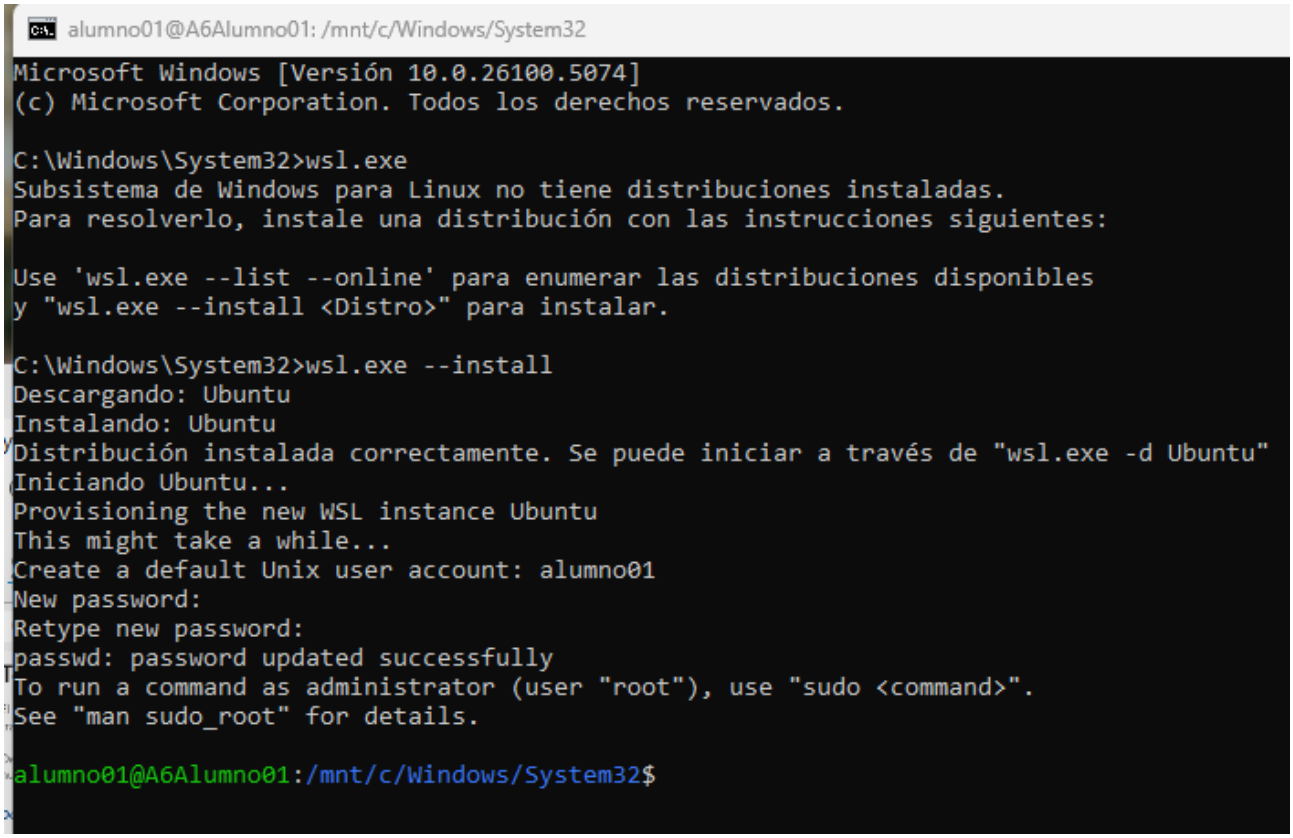
Aqui estoy instalando el wsl que simplemente es poner en el Git CMD wsl y darle a cualquier tecla y estaria listo



Aquí ya esta instalado el servicio

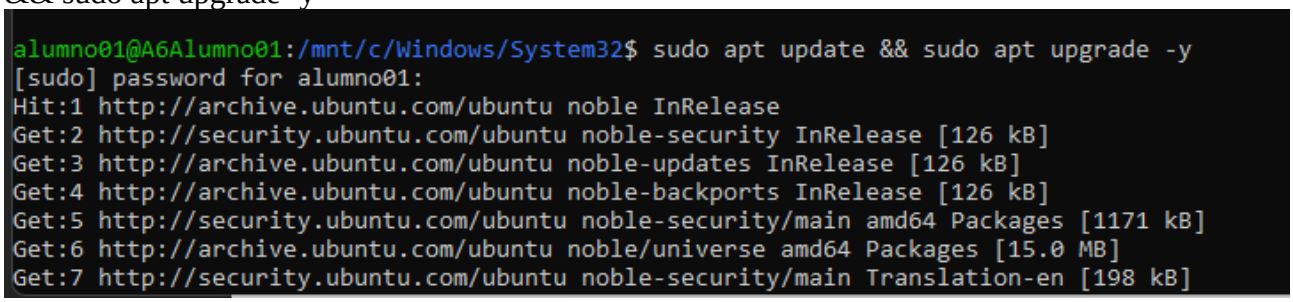


Aquí vemos como ya he creado el usuario y contraseña, también se puede observar que ya estoy con la consola del ubuntu.

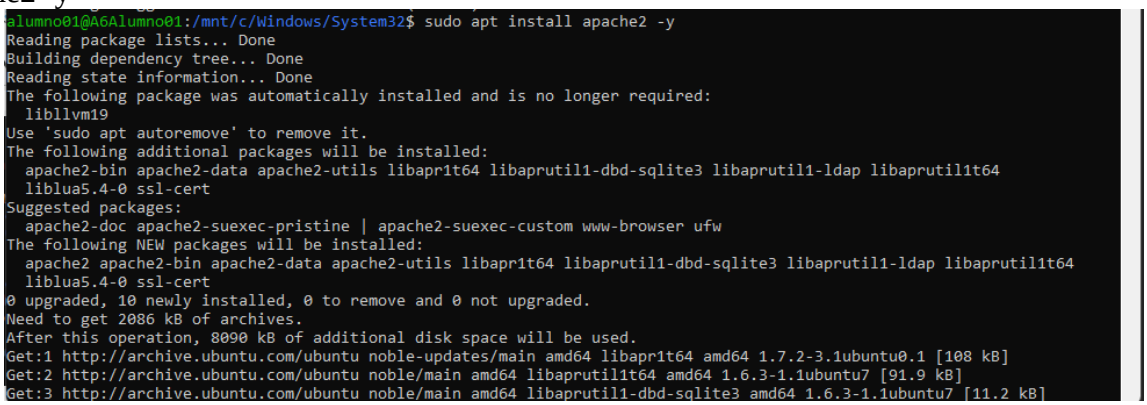


SERVIDOR APACHE CON PHP Y ACTUALIZACIÓN DEL SISTEMA Y FICHEROS

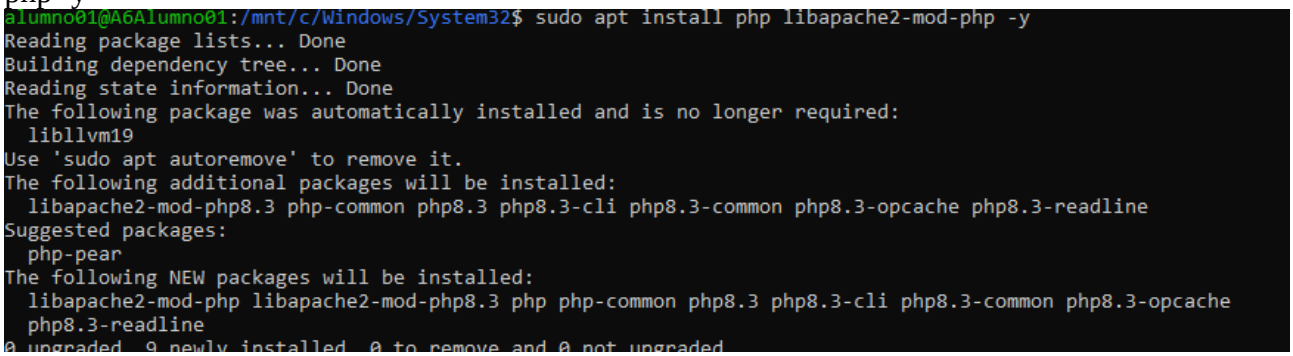
Aquí ya he actualizado todo lo que es el ubuntu y sus paquetes con los comandos “sudo apt update && sudo apt upgrade -y ”



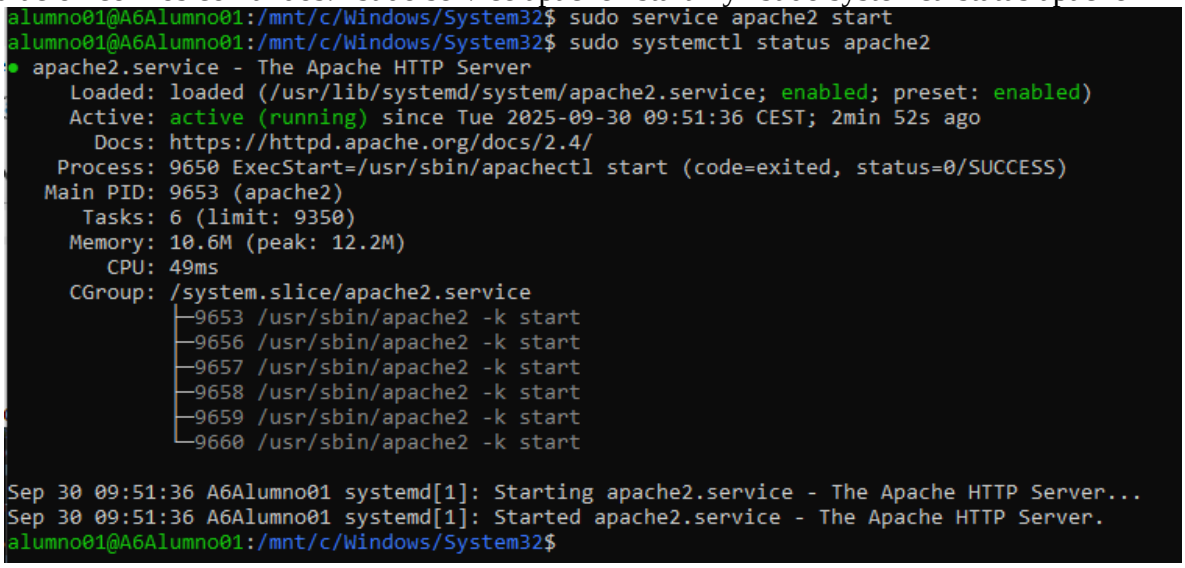
Aquí está descargando el servicio apache y el servidor web con el comando “sudo apt install apache2 -y "



Esto Instala PHP y su módulo para Apache con el comando “sudo apt install php libapache2-mod-php -y ”



Aquí inicio el servicio Apache y tambien compruebo si esta ejecutandose correctamente tras la instalación con los comandos: “sudo service apache2 start ” y “sudo systemctl status apache2 ”

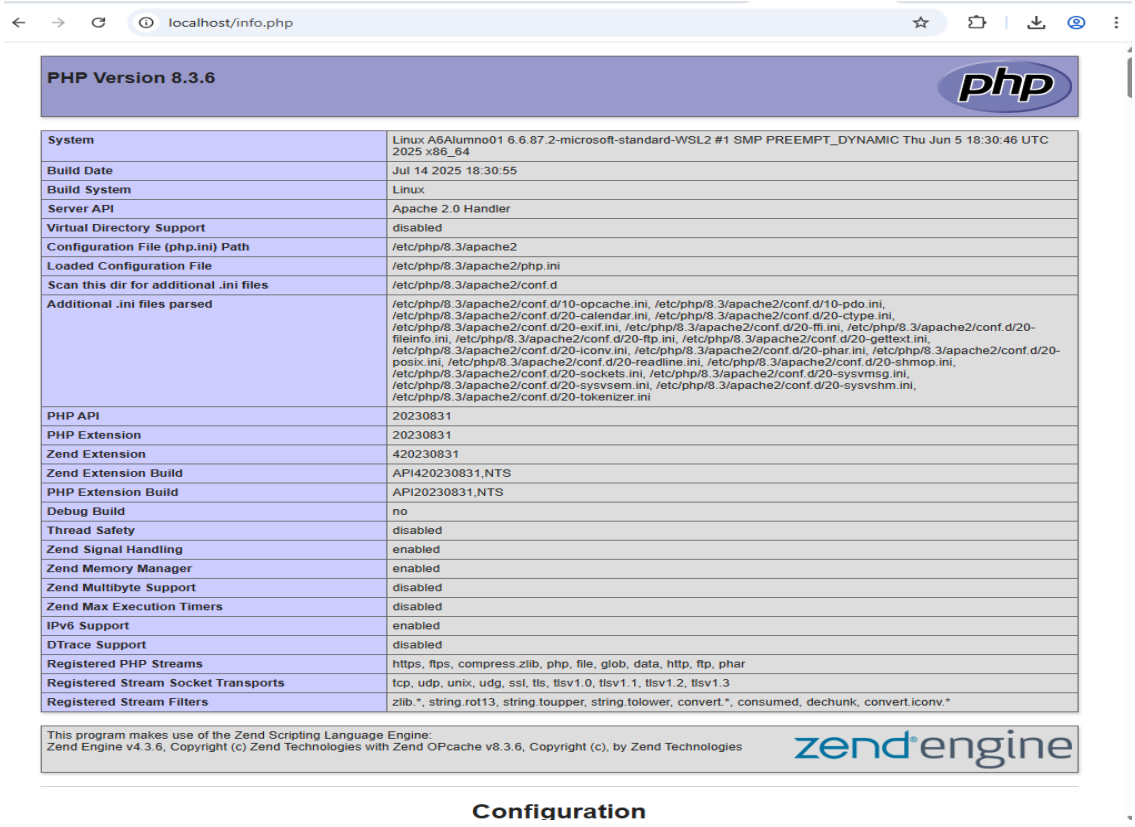


Creamos un archivo que muestra información de PHP con el comando: “echo "" | sudo tee /var/www/html/info.php”

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

Aquí verificamos en el navegador una comprobación del archivo PHP desde la terminal con el comando “curl http://localhost/info.php”. Además podemos ver como al darle a “enter” se muestra correctamente toda la página que hemos creado con el anterior comando. Tambien buscando la url en el navegador de nuestro equipo apareceria.

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ curl http://localhost/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
th {position: sticky; top: 0; background: inherit;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
h2 a:link, h2 a:visited{color: inherit; background: inherit;}
```



Adrián Aceitón Palomo. 2º ASIR

PARTE 2 : COMANDOS NGINX + HTML

Aquí instalaremos Nginx de primeras con el comando: “sudo apt install nginx -y” (abajo detengo Apache).

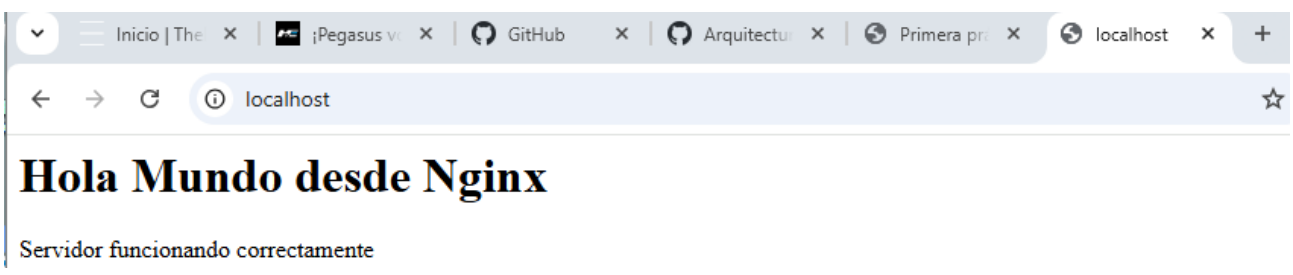
```
</div></body></html>alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
```

Luego iniciamos el Nginx previamente instalado con el comando “sudo service nginx start” y tambien verificaremos su estado de funcionamiento con el comando “sudo systemctl status nginx ”. Pero antes de nada debemos apagar el anterior servicio de apache dado que sino va a ocupar el puerto por el que Nginx va a salir (80). comando para apagarlo “sudo service apache2 stop”

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo service apache2 stop
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo service nginx start
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-09-30 10:13:07 CEST; 5s ago
       Docs: man:nginx(8)
    Process: 10030 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 10031 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 10033 (nginx)
      Tasks: 17 (limit: 9350)
     Memory: 12.7M (peak: 14.3M)
        CPU: 24ms
     CGroup: /system.slice/nginx.service
             ├─10033 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─10034 "nginx: worker process"
             ├─10035 "nginx: worker process"
             ├─10036 "nginx: worker process"
             ├─10037 "nginx: worker process"
             ├─10038 "nginx: worker process"
             ├─10039 "nginx: worker process"
             ├─10040 "nginx: worker process"
             ├─10042 "nginx: worker process"
             ├─10043 "nginx: worker process"
             ├─10044 "nginx: worker process"
             ├─10045 "nginx: worker process"
             ├─10046 "nginx: worker process"
             ├─10047 "nginx: worker process"
             ├─10048 "nginx: worker process"
```

Creamos una pagina HTML simple con el comando: echo "<h1>Hola Mundo desde Nginx</h1> Servidor funcionando correctamente " | sudo tee /var/www/html/index.html. Una vez creada lo verificamos desde el navegador a ver si se creó correctamente

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente<
/p>" | sudo tee /var/www/html/index.html
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```



Ahora vamos a ver la IP de WSL , basicamente vamos a mostrar la IP para acceso desde Windows con el comando: “ip addr show eth0 | grep inet”

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ ip addr show eth0 | grep inet
    inet 172.29.45.224/20 brd 172.29.47.255 scope global eth0
    inet6 fe80::215:5dff:feec:56be/64 scope link
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

PARTE 3: VERIFICACIÓN Y COMPARACIÓN

-Cambiar entre servicios (Apache/Nginx), diferencias entre las páginas servidas por cada servidor, comandos de verificación desde terminal con curl:

Primeramente paro el servicio Nginx con los comandos de la captura y arranco Apache, posteriormente verifico el estado de funcionamiento de Apache.

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo service nginx stop
sudo service apache2 start
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

Aquí podemos ver que esta encendido/corriendo Apache.

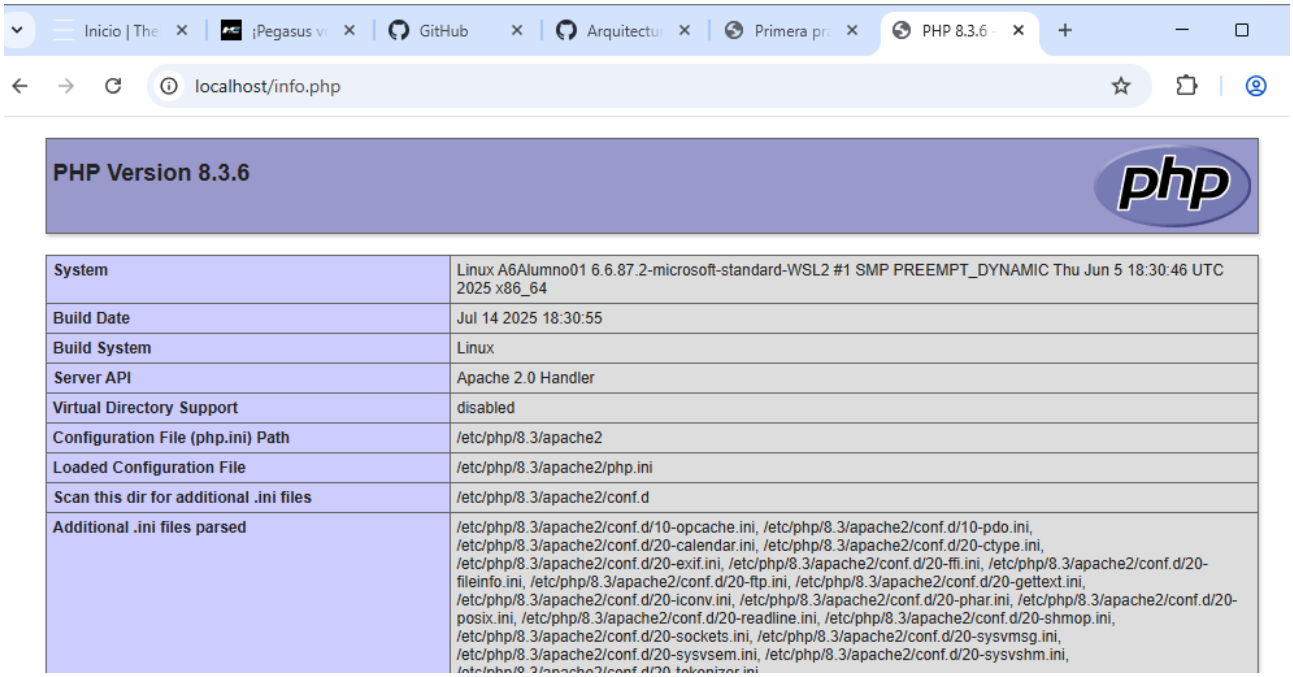
```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-09-30 10:30:28 CEST; 1min 18s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 10097 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 10100 (apache2)
      Tasks: 6 (limit: 9350)
     Memory: 10.7M (peak: 12.1M)
        CPU: 43ms
     CGroup: /system.slice/apache2.service
             ├─10100 /usr/sbin/apache2 -k start
             ├─10102 /usr/sbin/apache2 -k start
             ├─10103 /usr/sbin/apache2 -k start
             ├─10104 /usr/sbin/apache2 -k start
             ├─10105 /usr/sbin/apache2 -k start
             └─10106 /usr/sbin/apache2 -k start

Sep 30 10:30:28 A6Alumno01 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Sep 30 10:30:28 A6Alumno01 systemd[1]: Started apache2.service - The Apache HTTP Server.
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

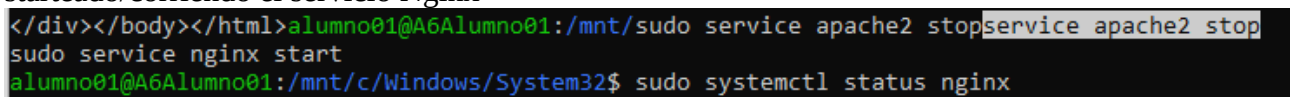
Aquí verificamos el funcionamiento del PHP de apache con el comando curl

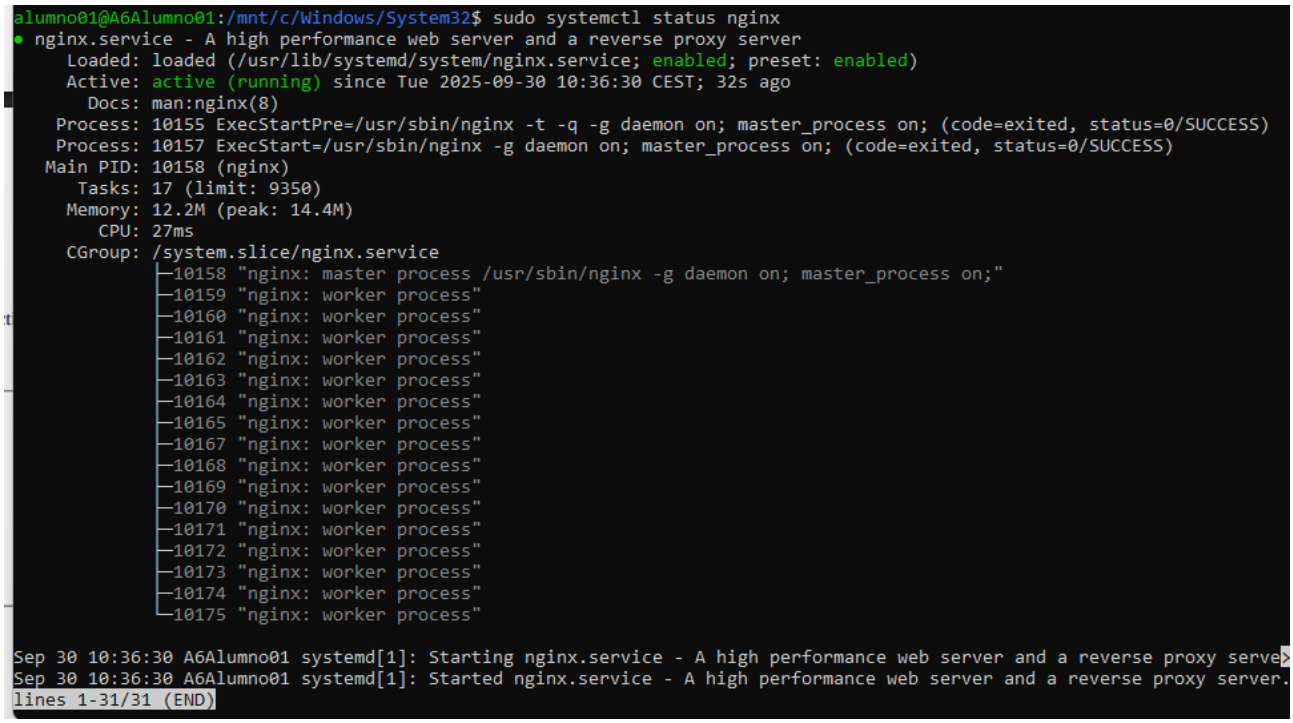
```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ curl http://localhost/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
th {position: sticky; top: 0; background: inherit;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
```

Aquí vemos como funciona el PHP y solo se pone la pagina de PHP



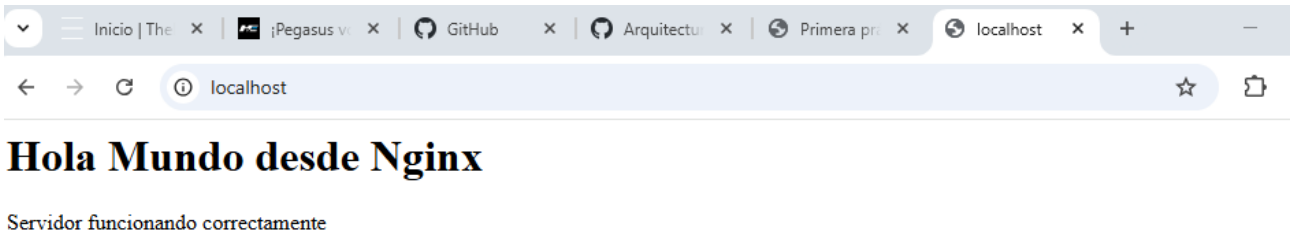
Aquí he parado el servicio de Apache, he arrancado el Nginx y también muestro como esta starteado/corriendo el servicio Nginx





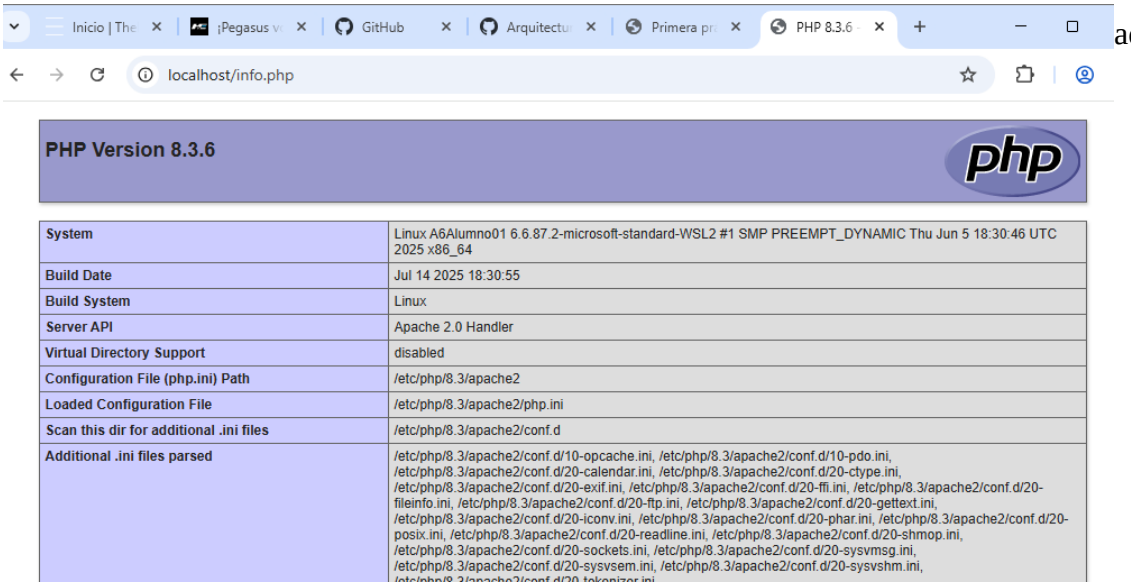
Aquí vemos igual que con el PHP que estamos verificando con el comando curl que nos devuelve correctamente la pagina, y también vemos la página web como quedaría en el navegador.

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ curl http://localhost
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

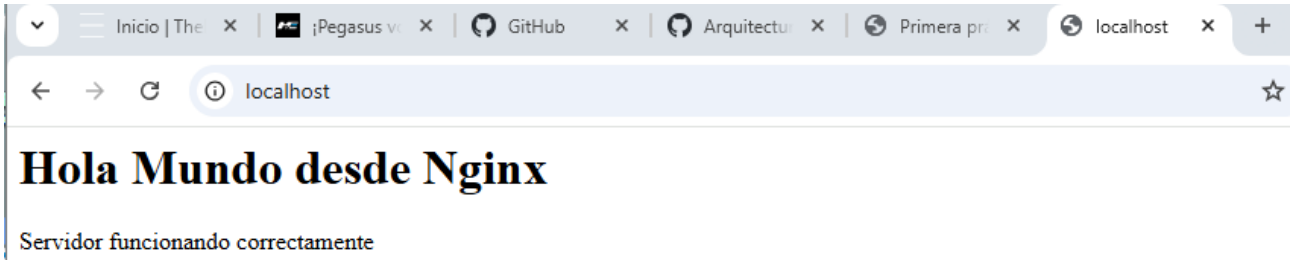


-Diferencias entre páginas:

Servidor Apache con archivo PHP:



Servidor Nginx con archivo HTML básico:



Adrián Aceitón Palomo. 2º ASIR

Instalación similar pero con docker

-Primeramente debemos de instalar docker dentro de WSL:

Lo primero que se debe de hacer es actualizar todos los paquetes y archivos.

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo apt update && sudo apt upgrade -y
```

Paquetes necesarios:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo apt install -y ca-certificates curl gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Crear directorio para claves:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo install -m 0755 -d /etc/apt/keyrings
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

Descargar la clave oficial de Docker:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

Ajustar permisos de la clave:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Añadir el repositorio de Docker:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu
  $(. /etc/os-release && echo ${VERSION_CODENAME}) stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

Actualizar e instalar Docker Engine, CLI y plugins:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Get:1 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:2 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [31.1 kB]
```

Comprobar que docker este instalado y su versión:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ docker --version
Docker version 28.4.0, build d8eb465
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

Probar el contenedor de “hello world”

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
953cdd413371: Pull complete
Digest: sha256:353675e2a41babd526e2b837d7ec780c2a05bca0164f7ea5dbbd433d21d166fc
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
alumno01@A6Alumno01:/mnt/c/Windows/System32$ 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 Tue 2025-09-30 14:46:19 CEST; 12min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2543 (dockerd)
      Tasks: 19
     Memory: 133.4M (peak: 163.2M)
        CPU: 2.636s
     CGroup: /system.slice/docker.service
             └─2543 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 30 14:46:19 A6Alumno01 dockerd[2543]: time="2025-09-30T14:46:19.578403000+02:00" level=info msg="Creating a containerd client" addre>
Sep 30 14:46:19 A6Alumno01 dockerd[2543]: time="2025-09-30T14:46:19.661834300+02:00" level=info msg="Loading containers: start."
```

docker info:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ docker info
Client: Docker Engine - Community
 Version:    28.4.0
 Context:    default
 Debug Mode: false
 Plugins:
  ai: Docker AI Agent - Ask Gordon (Docker Inc.)
    Version:  v1.9.11
    Path:     /usr/local/lib/docker/cli-plugins/docker-ai
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.28.0-desktop.1
    Path:     /usr/local/lib/docker/cli-plugins/docker-buildx
  cloud: Docker Cloud (Docker Inc.)
    Version:  v0.4.29
    Path:     /usr/local/lib/docker/cli-plugins/docker-cloud
```

docker psa:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
alumno01@A6Alumno01:/mnt/c/Windows/System32$
```

docker corriendo ya:

```
alumno01@A6Alumno01:/mnt/c/Windows/System32$ sudo docker run -it ubuntu /bin/bash
root@86e99cf73cea:/# docker exec -it CONTAINER_ID /bin/bash
bash: docker: command not found
root@86e99cf73cea:/#
```

Ahora vamos con el servicio NGINX y APACHE dentro del docker que esta dentro de WSL:

Primeramente hemos instalado apache y nginx con el “sudo apt install nginx” -y y “sudo apt install apache2”

```
root@ab196fc50848:/# apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

luego hacemos lo mismo con apache y con php, los instalamos. Con el comando echo igual que en nginx creamos el archivo y luego con curl lo comprobamos

```
root@ab196fc50848:/# apt install php libapache2-mod-php -y
Reading package lists... Done
```

Hacemos correr el servicio apache2 y tambien verificamos que el servicio funcione (este comando en docker es diferente al de wsl):

```
root@ee7307a500ce:/# service apache2 start
 * Starting Apache httpd web server apache2
 *
```

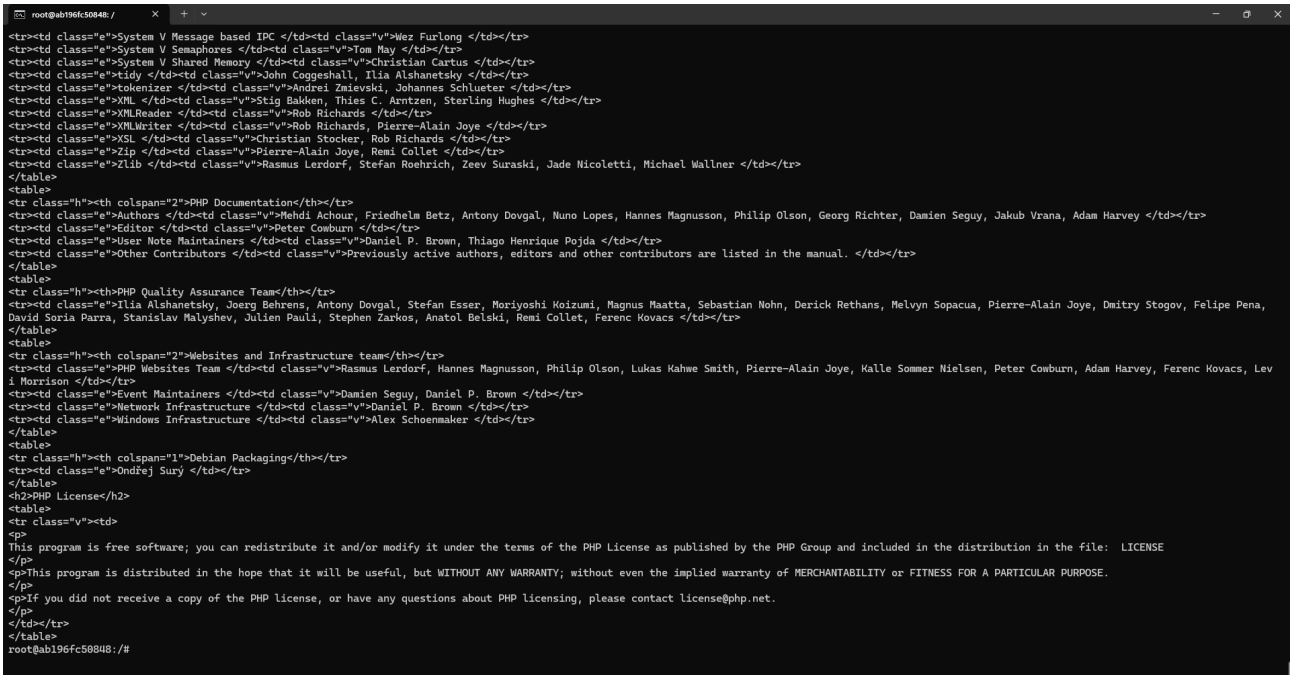
Ahora instalamos curl y comprobamos que apache funcione.

```
root@ee7307a500ce:/# apt install curl -y
Reading package lists... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ab196fc50848:/# service apache2 start
 * Starting Apache httpd web server apache2
 *
root@ab196fc50848:/# service apache2 status
 * apache2 is running
root@ab196fc50848:/#
```

Aquí basicamente hay varias cosas, he instalado nginx, he parado el servicio de apache, he iniciado el servicio de nginx, despues he metido el “echo” para nginx y he probado con el comando “curl http://localhost”.
Ahora con el comando curl lo comprobamos.

```
Setting up libpam-cap:amd64 (1:2.66-5ubuntu2.2) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannd. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 79.)
debconf: falling back to frontend: Readline
Setting up libbpf1:amd64 (1:1.3.0-2build2) ...
Setting up iproute2 (6.1.0-1ubuntu6.2) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannd. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 79.)
debconf: falling back to frontend: Readline
Setting up nginx (1.24.0-2ubuntu7.5) ...
Not attempting to start NGINX, port 80 is already in use.
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
root@ab196fc50848:/# service nginx status
 * nginx is not running
root@ab196fc50848:/# echo "<h1>Hola Mundo desde NGINX</h1><p>Servidor funcionando correc/p>" | tee /var/www/html/index.html
<h1>Hola Mundo desde NGINX</h1><p>Servidor funcionando correctamente</p>
root@ab196fc50848:/# service apache2 status
 * apache2 is running
root@ab196fc50848:/# service apache2 stop
 * Stopping Apache httpd web server apache2
 *
root@ab196fc50848:/# service nginx start
 * Starting nginx nginx
root@ab196fc50848:/# service nginx service
Usage: nginx {start|stop|restart|reload|force-reload|status|configtest|rotate|upgrade}
root@ab196fc50848:/# service nginx status
 * nginx is running
root@ab196fc50848:/# curl http://localhost
<h1>Hola Mundo desde NGINX</h1><p>Servidor funcionando correctamente</p>
root@ab196fc50848:/#
```

Aqui he hecho lo mismo que en del nginx, basicamente bajo el servicio de nginx, inicio el apache y compruebo con curl que funcione



Ahora comprobamos desde el navegador del PC si funcionan los dos servicios.