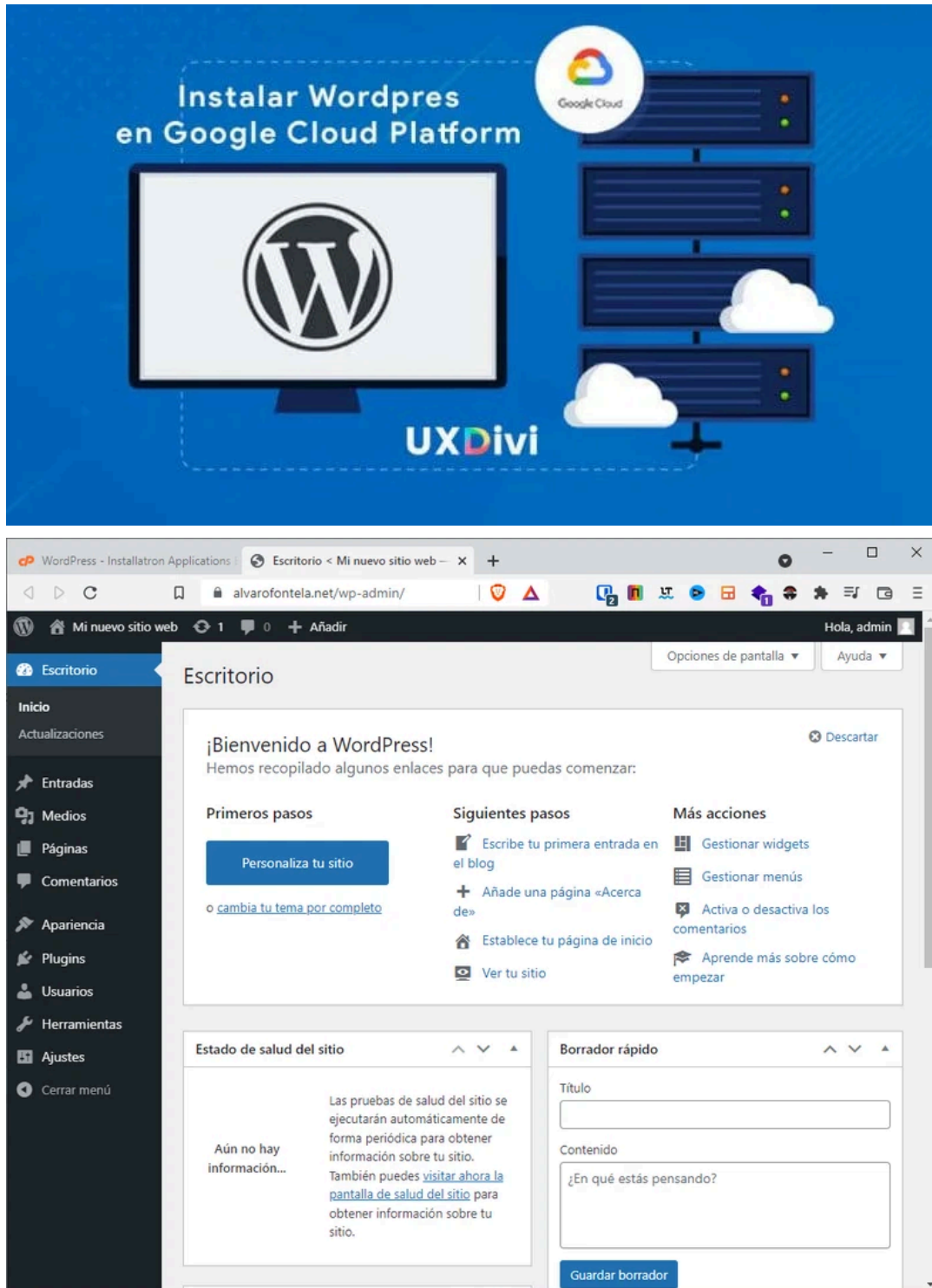


# INSTALACIÓN DE WORDPRESS EN LA NUBE



**MARIO CAMACHO PRIETO**  
**2º ASIR / CAJA MÁGICA**  
**04 / 11 / 2025**

# ÍNDICE

## PARTE 1. Instalación del Servidor LAMP

- 1.1. Actualizar el sistema
  - Comandos apt update y apt upgrade.
- 1.2. Instalar Apache (Servidor Web)
  - Comando apt install apache2.
  - Verificación del servicio (systemctl status apache2).
  - Prueba en el navegador (http://localhost).
- 1.3. Instalar MySQL (Base de Datos)
  - Comando apt install mysql-server.
  - Asegurar la instalación (mysql\_secure\_installation).
  - Respuestas recomendadas para el script de seguridad.
- 1.4. Instalar PHP
  - Comando apt install para PHP y módulos necesarios.
  - Verificación de la versión (php -v).
  - Reinicio de Apache (systemctl restart apache2).

## PARTE 2. Crear Base de Datos para WordPress

- 2.1. Acceder a MySQL
  - Comando sudo mysql.
- 2.2. Crear base de datos y usuario
  - Comandos SQL (CREATE DATABASE, CREATE USER, GRANT ALL PRIVILEGES, FLUSH PRIVILEGES, EXIT).
  - Tabla de datos de acceso (Base de datos, Usuario, Contraseña).

## PARTE 3. Instalar WordPress

- 3.1. Descargar WordPress
  - Navegación al directorio /tmp.
  - Descarga (wget) y extracción (tar) de WordPress.
- 3.2. Copiar archivos a Apache
  - Limpieza del directorio web (rm).
  - Copia de archivos (cp) a /var/www/html/.
- 3.3. Permisos
  - Asignación de propietario (chown) y permisos (chmod) al directorio web.

- 3.4. Configurar wp-config.php
  - Copia del archivo de ejemplo (wp-config-sample.php).
  - Edición del archivo (nano).
  - Modificación de las constantes de base de datos (DB\_NAME, DB\_USER, DB\_PASSWORD, DB\_HOST).
- 3.5. Finalizar en el navegador
  - Acceso a <http://localhost>.
  - Pasos de configuración final (idioma, usuario, etc.).

## **PARTE 4. Hacer WordPress Accesible desde Internet con ngrok**

- 4.1. ¿Qué es ngrok?
  - Definición y caso de uso.
- 4.2. Registro en ngrok
  - Pasos para crear una cuenta.
  - Obtención del Authtoken.
- 4.3. Instalar ngrok en Ubuntu
  - Método 1: Descarga directa (wget, tar, mv).
  - Método 2: Usando Snap (snap install).
- 4.4. Verificar instalación
  - Comando ngrok versión.
- 4.5. Autenticar ngrok
  - Comando ngrok config add-authtoken.
- 4.6. Iniciar túnel HTTP
  - Comando ngrok http 80.
- 4.7. Anotar URLs
  - Descripción de las URLs HTTP y HTTPS generadas.
- 4.8. Probar acceso
  - Acceso desde un navegador externo.
  - Manejo de la página de advertencia de ngrok.
- 4.9. Verificación
  - Puntos de control (estado de ngrok, URL, acceso, interfaz web).

## PARTE 1. Instalación del Servidor LAMP

Usé este comando para actualizar las listas de paquetes desde los repositorios, asegurándome de tener la información más reciente sobre las versiones de software.

```
mario@mario-virtualbox:~$ sudo apt update
Obj:1 http://security.debian.org/debian-security trixie-security InRelease
Des:2 http://deb.debian.org/debian trixie InRelease [140 kB]
Obj:3 http://deb.debian.org/debian trixie-updates InRelease
Obj:4 http://deb.debian.org/debian trixie-backports InRelease
Descargados 140 kB en 10s (13,9 kB/s)
Todos los paquetes están actualizados.
mario@mario-virtualbox:~$
```

Con esto, actualicé todos los paquetes que ya tenía instalados en mi sistema a sus últimas versiones. La -y es para que no me pidiera confirmación.

```
mario@mario-virtualbox:~$ sudo apt upgrade -y
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
mario@mario-virtualbox:~$
```

Aquí instalé el servidor web Apache. Este es el software que "sirve" las páginas web a los visitantes.

```
mario@mario-virtualbox:~$ sudo apt install apache2 -y
Installing:
  apache2

Installing dependencies:
  apache2-bin      libapr1t64      libaprutilt64
  apache2-data     libaprutil1-dbd-sqlite3  liblua5.4-0
  apache2-utils   libaprutil1-ldap

Paquetes sugeridos:
  apache2-doc | apache2-suexec-custom
  apache2-suexec-pristine  ufw

Summary:
  Upgrading: 0, Installing: 9, Removing: 0, Not Upgrading: 0
  Download size: 2.371 kB
  Space needed: 8.488 kB / 9.775 MB available

Des:1 http://deb.debian.org/debian trixie/main amd64 libapr1t64 amd64 1.7.5-1 [104 kB]
Des:2 http://deb.debian.org/debian trixie/main amd64 libaprutilt64 amd64 1.6.3-3+b1 [89,4 kB]
Des:3 http://deb.debian.org/debian trixie/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-3+b1 [14,2 kB]
Des:4 http://deb.debian.org/debian trixie/main amd64 libaprutil1-ldap amd64 1.6.3-3+b1 [12,3 kB]
Des:5 http://deb.debian.org/debian trixie/main amd64 liblua5.4-0 amd64 5.4.7-1+b2 [147 kB]
Des:6 http://deb.debian.org/debian trixie/main amd64 apache2-bin amd64 2.4.65-2 [1.405 kB]
Des:7 http://deb.debian.org/debian trixie/main amd64 apache2-data all 2.4.65-2 [160 kB]
Des:8 http://deb.debian.org/debian trixie/main amd64 apache2-utils amd64 2.4.65-2 [215 kB]
Des:9 http://deb.debian.org/debian trixie/main amd64 apache2 amd64 2.4.65-2 [224 kB]
Descargados 2.371 kB en 10s (231 kB/s)
```

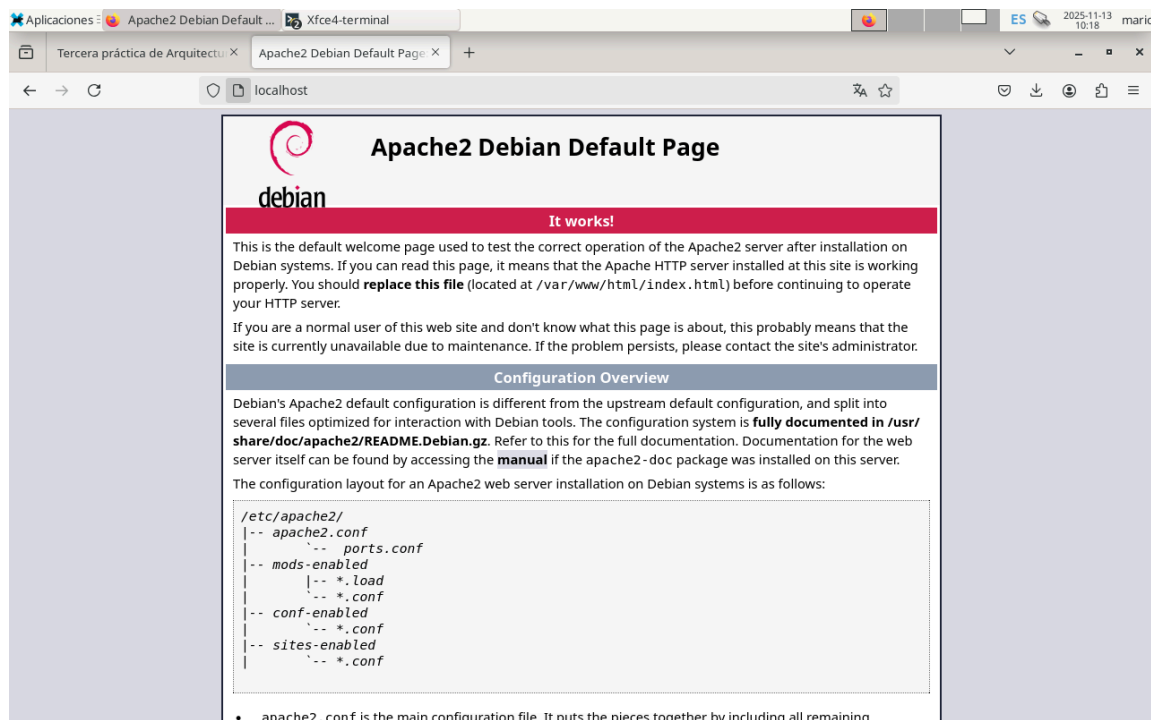
Inmediatamente después de instalarlo, comprobé el estado del servicio de Apache para asegurarme de que se estaba ejecutando correctamente (active (running)).

```
mario@mario-virtualbox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled)
   Active: active (running) since Thu 2025-11-13 10:15:45 CET; 1min ago
 Invocation: ab67067863764d759fcc10e50c980aff
    Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 16055 (apache2)
     Tasks: 55 (limit: 4538)
    Memory: 5.1M (peak: 5.8M)
       CPU: 48ms
    CGroup: /system.slice/apache2.service
            └─16055 /usr/sbin/apache2 -k start
              └─16056 /usr/sbin/apache2 -k start
                └─16058 /usr/sbin/apache2 -k start

nov 13 10:15:45 mario-virtualbox systemd[1]: Starting apache2.service: The Apache HTTP Server.
nov 13 10:15:45 mario-virtualbox apachectl[16054]: AH00558: apache2: Syntax error on line 1 of /etc/apache2/httpd.conf: Invalid command 'LoadModule', with immediate effect.
nov 13 10:15:45 mario-virtualbox systemd[1]: Started apache2.service: The Apache HTTP Server.

mario@mario-virtualbox:~$
```

Esta imagen es la prueba clave de que la Parte 1 (la instalación de Apache) fue un éxito. Después de instalar Apache y verificar que el servicio estaba activo, abrí el navegador web en mi máquina virtual y escribí `http://localhost` en la barra de direcciones. La captura de pantalla muestra la página que apareció: la "Apache2 Debian Default Page". Ver esta página, especialmente el mensaje "It works!", me confirmó que el servidor web Apache estaba instalado y funcionando correctamente en mi sistema Debian.



Instalé MariaDB, que es el sistema gestor de bases de datos. Es un reemplazo directo de MySQL y es donde WordPress guardará toda la información.

```
mario@mario-virtualbox:~$ sudo apt install mariadb-server -y
Installing:
  mariadb-server

Installing dependencies:
  galera-4          libdbi-perl          libsigsegv2          mariadb-plugin-
  gawk              libfcgi-bin          libterm-readkey-perl mariadb-plugin-
  libcgi-fast-perl  libfcgi-perl         liburing2            mariadb-plugin-
  libcgi-pm-perl    libfcgi0t64          mariadb-client        mariadb-plugin-
  libconfig-inifiles-perl libhtml-template-perl mariadb-client-core   mariadb-server-
  libdbd-mariadb-perl libpcre2-posix3      mariadb-plugin-provider-bzip2 pv

Paquetes sugeridos:
  gawk-doc          libnet-daemon-perl  libipc-sharedcache-perl  mariadb-test  doc-base
  libmldbm-perl     libsql-statement-perl  mailx                   netcat-openbsd  openssh-client

Summary:
  Upgrading: 0, Installing: 27, Removing: 0, Not Upgrading: 0
  Download size: 20,1 MB
  Space needed: 200 MB / 9.765 MB available

0% [Trabajando]
```

Ejecuté el script de seguridad de MariaDB. Esto me permitió establecer una contraseña de root, eliminar usuarios anónimos y la base de datos de prueba.

```
mario@mario-virtualbox:~$ sudo mariadb-secure-installation

NOTE: MariaDB is secure by default in Debian. Running this script is
      useless at best, and misleading at worst. This script will be
      removed in a future MariaDB release in Debian. Please read
      mariadb-server.README.Debian for details.

Enter root user password or leave blank:

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] Y

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

Remove anonymous users? [Y/n] Y
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
SQL executed without errors!
The operation might have been successful, or it might have not done anything.
- Removing privileges on test database...
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
mario@mario-virtualbox:~$
```

Este fue un comando largo para instalar PHP y todos los módulos (extensiones) que WordPress necesita para funcionar, comunicarse con la base de datos y manejar imágenes o archivos.

```
mario@mario-virtualbox:~$ sudo apt install php php-mysql libapache2-mod-php php-curl php-gd php-mbstring php-xml php-xmllrpc php-intl php-zip -y
Installing:
  libapache2-mod-php  php  php-curl  php-gd  php-intl  php-mbstring  php-mysql  php-xml  php-xmllrpc  php-zip
Installing dependencies:
  libapache2-mod-php8.4  libzip5  php8.4-cli  php8.4-gd  php8.4-mysql  php8.4-xml
  libonig5  php-common  php8.4-common  php8.4-intl  php8.4-openssl  php8.4-xmllrpc
  libxmlrpc-epi0t64  php8.4  php8.4-curl  php8.4-mbstring  php8.4-readline  php8.4-zip
Paquetes sugeridos:
  php-pear
Summary:
  Upgrading: 0, Installing: 28, Removing: 0, Not Upgrading: 0
  Download size: 6.769 kB
  Space needed: 31,7 MB / 9.400 MB available
0% [Trabajando]
```

Verifiqué la versión de PHP que se instaló (PHP 8.4.11) para confirmar que todo estaba en orden.

```
mario@mario-virtualbox:~$ php -v
PHP 8.4.11 (cli) (built: Aug 3 2025 07:32:21) (NTS)
Copyright (c) The PHP Group
Built by Debian
Zend Engine v4.4.11, Copyright (c) Zend Technologies
    with Zend OPcache v8.4.11, Copyright (c), by Zend Technologies
mario@mario-virtualbox:~$
```

Después de instalar PHP, reinicié el servidor Apache para que cargara el nuevo módulo de PHP y pudiera procesar los archivos .php.

```
mario@mario-virtualbox:~$ sudo systemctl restart apache2
mario@mario-virtualbox:~$
```



## PARTE 2. Crear Base de Datos para WordPress

En esa captura, entré a la consola de MariaDB con `sudo mysql` para preparar la base de datos de WordPress. Creé la base de datos (`CREATE DATABASE wordpress;`), luego un usuario nuevo llamado 'wpuser' con su contraseña (`CREATE USER...`). Después, le di a ese usuario todos los permisos sobre la base de datos con `GRANT ALL PRIVILEGES...` (aunque ahí cometí el error al escribir `wodpress` en lugar de `wordpress`). Finalmente, apliqué los cambios con `FLUSH PRIVILEGES;` y salí con `EXIT;`.

```
mario@mario-virtualbox:~$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'WordPress123!';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wodpress.*TO 'wpuser'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
mario@mario-virtualbox:~$
```

### PARTE 3. Instalar WordPress

Me moví al directorio /tmp, que es una carpeta para archivos temporales, un buen lugar para descargar cosas que no necesito guardar permanentemente.

```
mario@mario-virtualbox:~$ cd /tmp
mario@mario-virtualbox:/tmp$
```

Instalé la herramienta wget, que me permite descargar archivos desde la línea de comandos.

```
mario@mario-virtualbox:/tmp$ sudo apt install wget
Installing:
  wget

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
  Download size: 984 kB
  Space needed: 3.875 kB / 9.367 MB available
0% [Trabajando]
```

Usé wget para descargar la última versión de WordPress directamente desde su sitio web oficial.

```
mario@mario-virtualbox:/tmp$ wget https://wordpress.org/latest.tar.gz
--2025-11-13 10:47:27-- https://wordpress.org/latest.tar.gz
Resolviendo wordpress.org (wordpress.org)... 198.143.164.252, 2607:f978:5:8002::c68f:a4fc
Conectando con wordpress.org (wordpress.org)[198.143.164.252]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 26928488 (26M) [application/octet-stream]
Grabando a: «latest.tar.gz»

latest.tar.gz          100%[=====>] 25
2025-11-13 10:47:46 (1,93 MB/s) - «latest.tar.gz» guardado [26928488/26928488]
mario@mario-virtualbox:/tmp$
```

Descomprimí el archivo .tar.gz que acababa de descargar. Esto creó una carpeta llamada 'wordpress' con todos los archivos de la aplicación.

```
mario@mario-virtualbox:/tmp$ tar -xzf latest.tar.gz
mario@mario-virtualbox:/tmp$
```

- Borré todo el contenido del directorio web raíz de Apache (donde estaba la página de bienvenida de Debian) para hacer espacio para WordPress.
- Copié todos los archivos de la carpeta 'wordpress' que había descomprimido y los pegué en el directorio web raíz de Apache.

```
mario@mario-virtualbox:/tmp$ sudo rm -rf /var/www/html/*
mario@mario-virtualbox:/tmp$ sudo cp -r wordpress/* /var/www/html/
mario@mario-virtualbox:/tmp$
```

- Cambié el propietario de todos los archivos de WordPress al usuario www-data. Este es el usuario que usa Apache, y necesita ser el propietario para poder modificar los archivos.
- Aquí cometí un error. Escribí chown (cambiar propietario) cuando en realidad quería escribir chmod (cambiar permisos) para establecer los permisos correctos (755) en los directorios.

```
mario@mario-virtualbox:/tmp$ sudo chown -R www-data:www-data /var/www/html/
mario@mario-virtualbox:/tmp$ sudo chown -R 755 /var/www/html/
mario@mario-virtualbox:/tmp$
```

Copié el archivo de configuración de ejemplo de WordPress para crear el archivo de configuración real (wp-config.php).

```
mario@mario-virtualbox:/tmp$ sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
mario@mario-virtualbox:/tmp$
```

Abrí el nuevo archivo de configuración con el editor de texto nano para poder introducir los datos de la base de datos (nombre, usuario y contraseña) que creé en la Parte 2.

```
mario@mario-virtualbox:/tmp$ sudo nano /var/www/html/wp-config.php
mario@mario-virtualbox:/tmp$
```

```
GNU nano 8.4 /var/www/html/wp-config.php *
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

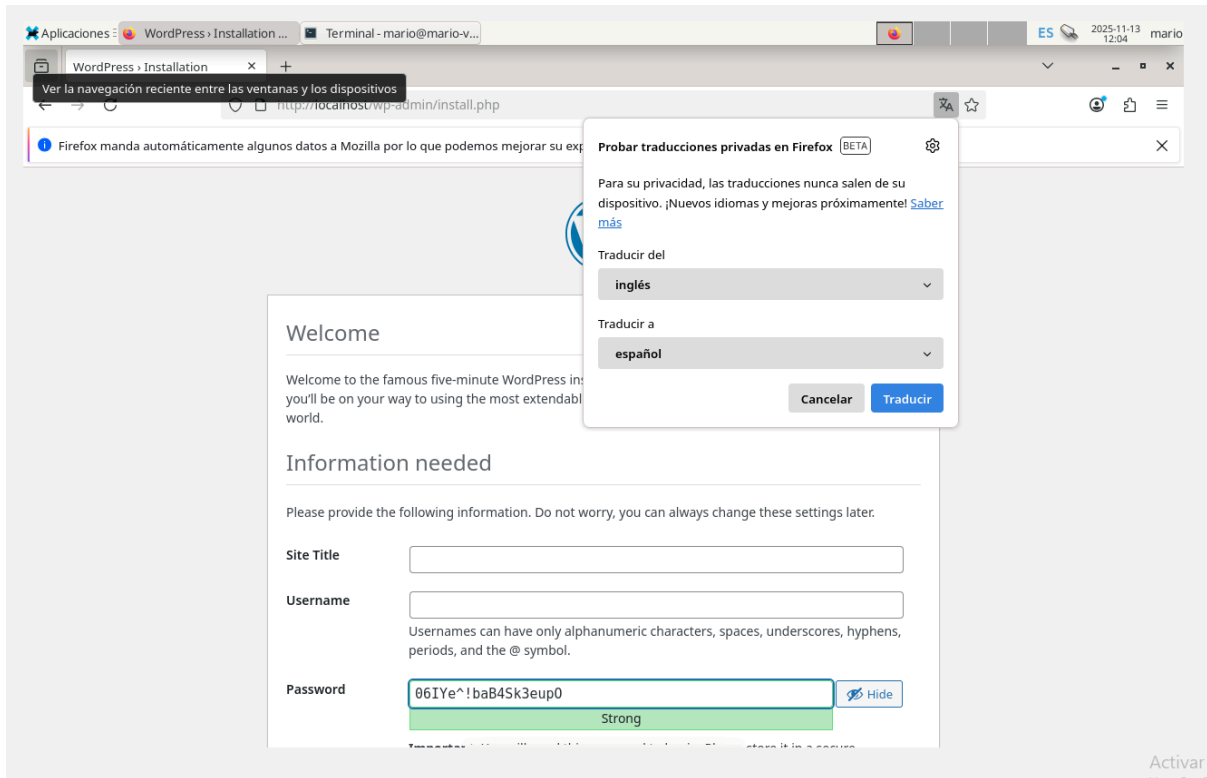
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wpuser' );

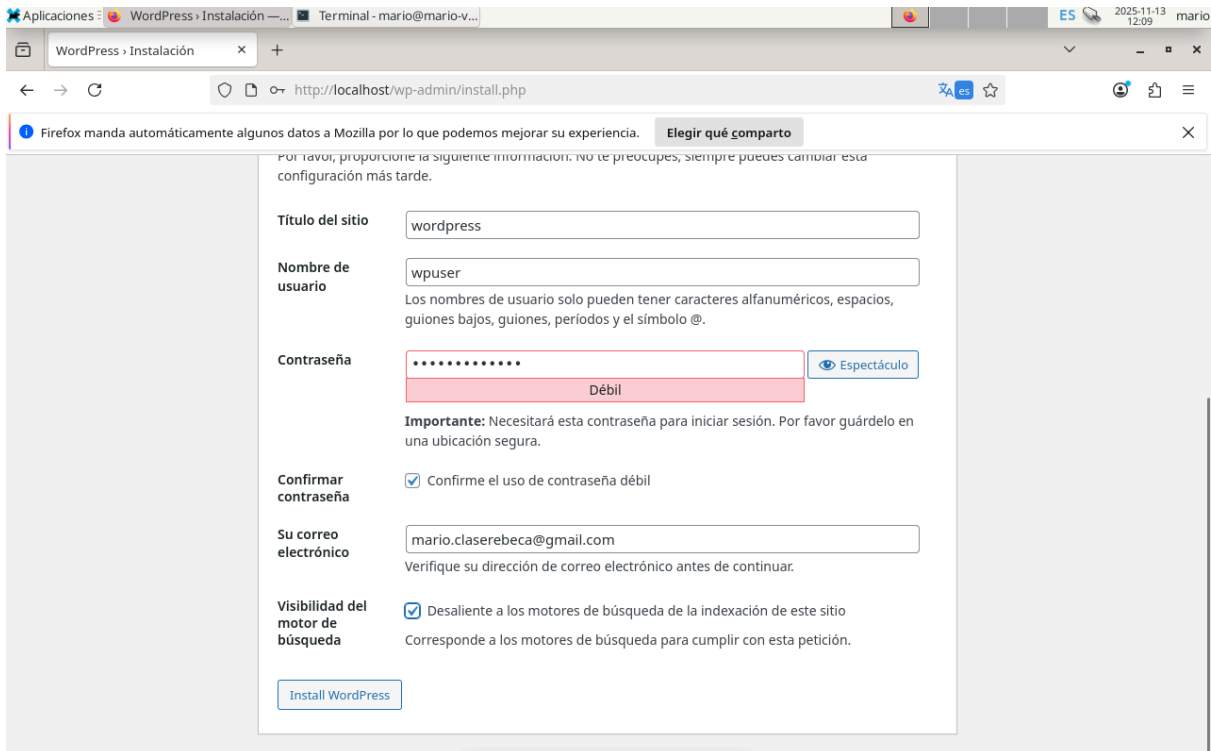
/** Database password */
define( 'DB_PASSWORD', 'WordPress123!' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

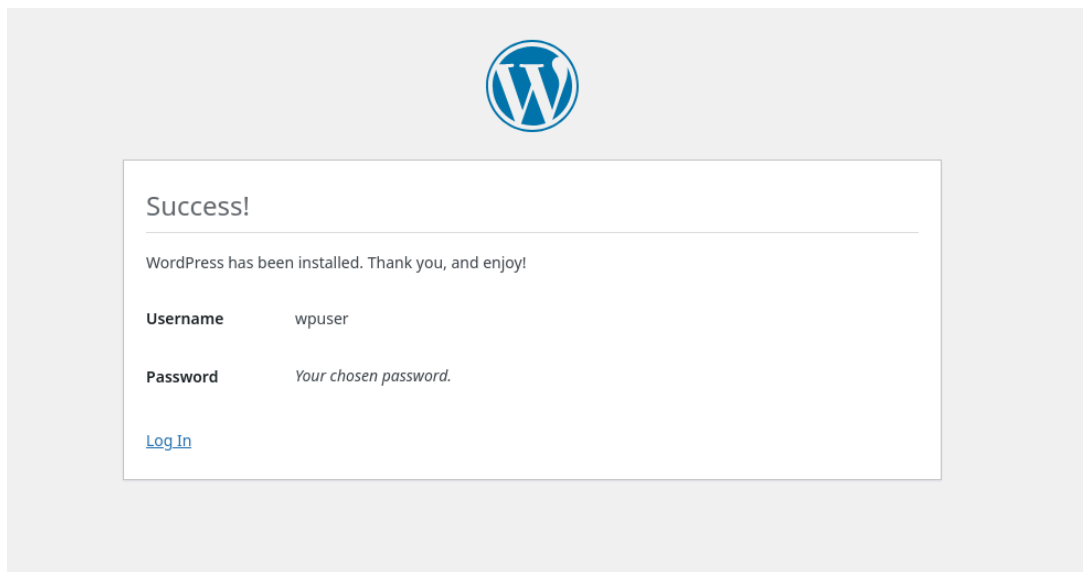
## Muestro la pantalla de bienvenida inicial de la instalación



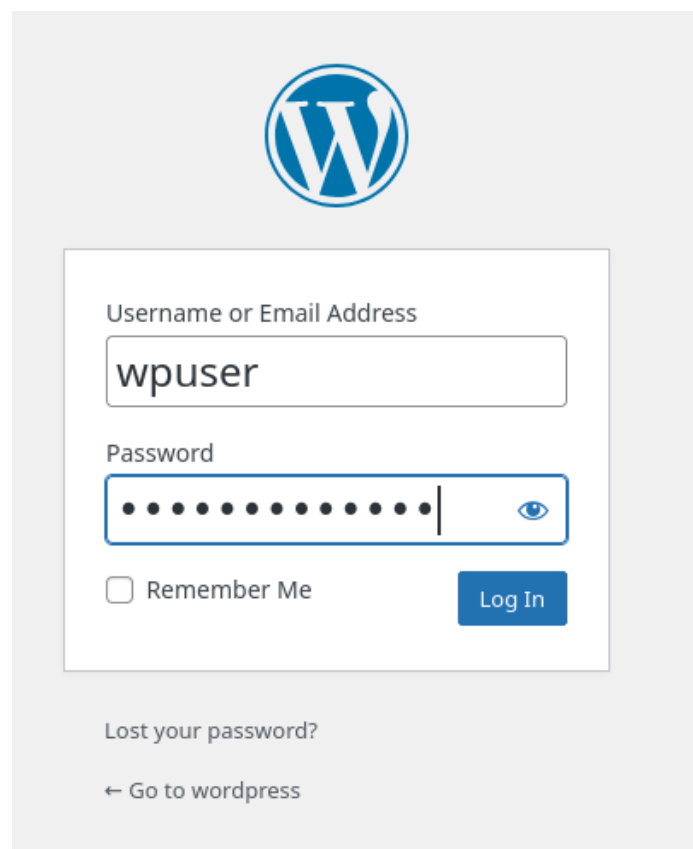
Aquí es donde creé mi cuenta de administrador (usuario wpuser, mi correo, etc.) y le di un título a mi sitio.



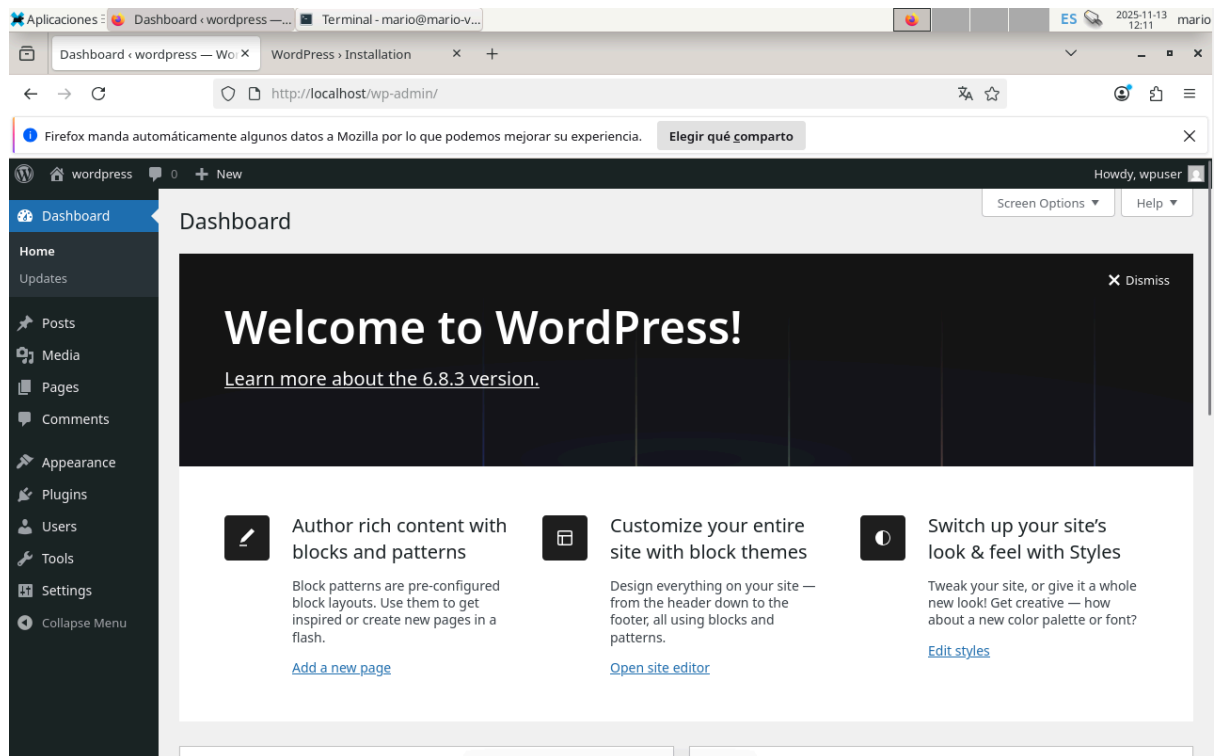
Esta es la pantalla de éxito (Success!) que confirma que la instalación se completó correctamente.



Muestra la página de inicio de sesión (Log In) donde usé el usuario que acababa de crear.

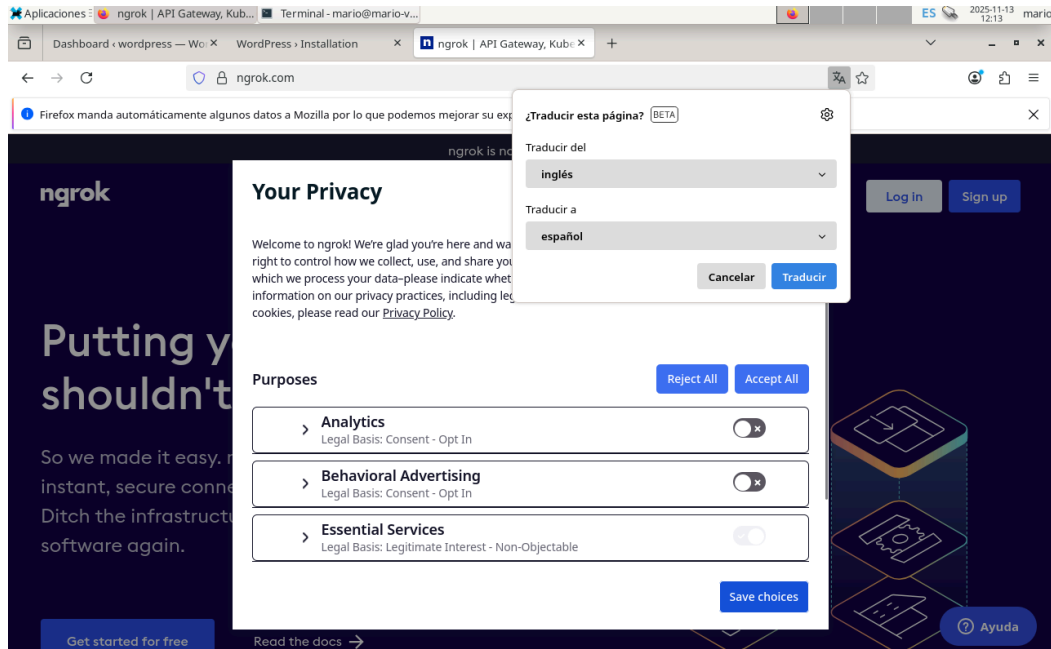


Esta es la prueba final: el Escritorio o "Dashboard" de WordPress, el panel de control al que llegué después de iniciar sesión con éxito.

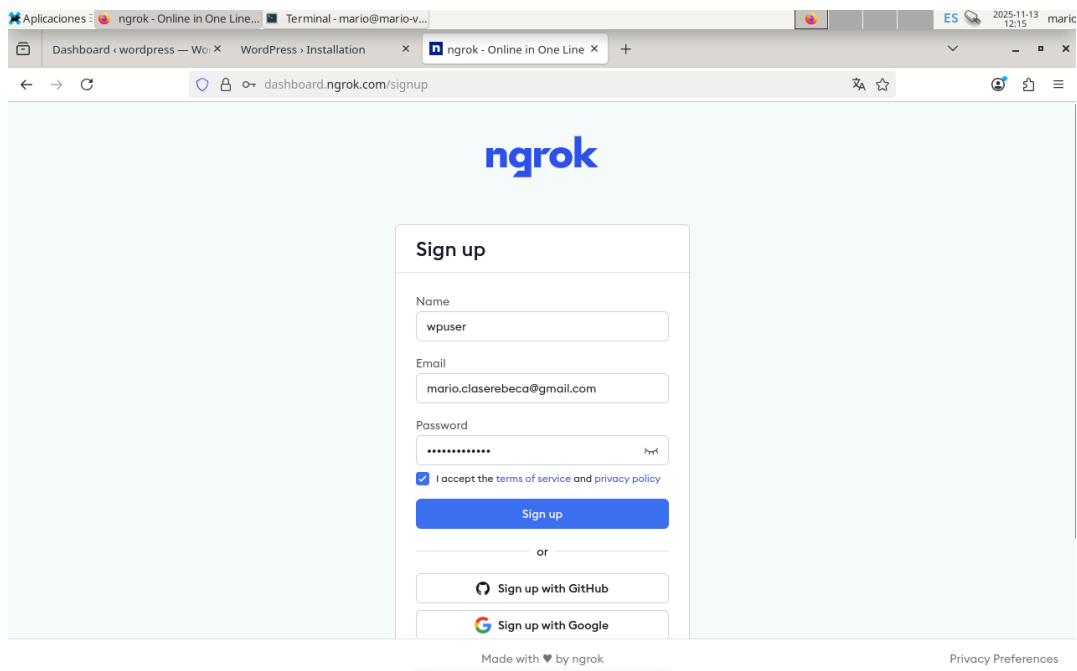


## PARTE 4. Hacer WordPress Accesible desde Internet con ngrok

Esta es la página de inicio de ngrok.com. Fue la primera vez que visité la web, y simplemente me mostraba la pantalla de privacidad y cookies que tuve que aceptar.

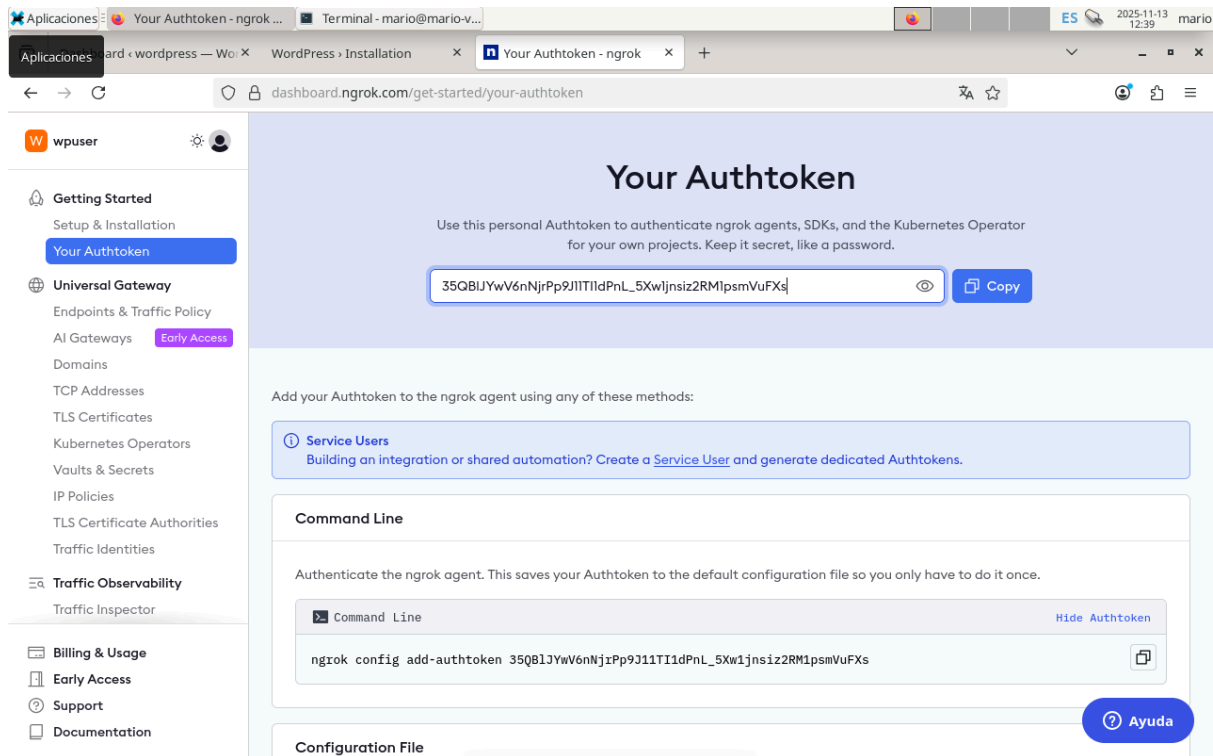


Esta es la página de registro (Sign up). Aquí fue donde creé mi cuenta rellenando el formulario con mi nombre, correo (mario.claserebeca@gmail.com) y una contraseña.





Esta es la pantalla más importante. Inmediatamente después de registrarme, me llevó a mi panel de control, a la sección "Your Authtoken". Esta pantalla me dio la clave personal (el token) que tuve que copiar para luego pegarla en la terminal.



Volví a mi directorio personal (home) para descargar los archivos de ngrok y luego descargué el archivo comprimido de ngrok para Linux.

```
mario@mario-virtualbox:~$ cd ~
mario@mario-virtualbox:~$ wget https://bin.equinox.io/c/ByNjlmQVY4c/ngrok-v3-stable-linux-amd64.tgz
--2025-11-13 12:45:26-- https://bin.equinox.io/c/ByNjlmQVY4c/ngrok-v3-stable-linux-amd64.tgz
Resolviendo bin.equinox.io (bin.equinox.io)... 13.248.244.96, 35.71.179.82, 75.2.60.68, ...
Conectando con bin.equinox.io (bin.equinox.io)[13.248.244.96]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 9322357 (8,9M) [application/octet-stream]
Grabando a: «ngrok-v3-stable-linux-amd64.tgz»
ngrok-v3-stable-linux-amd64.tgz 100%[=====] 8,89M 8,61MB/s en 1,0s
2025-11-13 12:45:32 (8,61 MB/s) - «ngrok-v3-stable-linux-amd64.tgz» guardado [9322357/9322357]
mario@mario-virtualbox:~$
```

Descomprimí el archivo, lo que me dio un único archivo ejecutable llamado 'ngrok'.

```
mario@mario-virtualbox:~$ tar -xvzf ngrok-v3-stable-linux-amd64.tgz
ngrok
mario@mario-virtualbox:~$
```

Moví el ejecutable 'ngrok' a la carpeta /usr/local/bin/. Esto me permite ejecutar el comando ngrok desde cualquier parte del sistema.

```
mario@mario-virtualbox:~$ sudo mv ngrok /usr/local/bin/  
[sudo] contraseña para mario:  
mario@mario-virtualbox:~$ █
```

Comprobé la versión de ngrok para asegurarme de que estaba bien instalado.

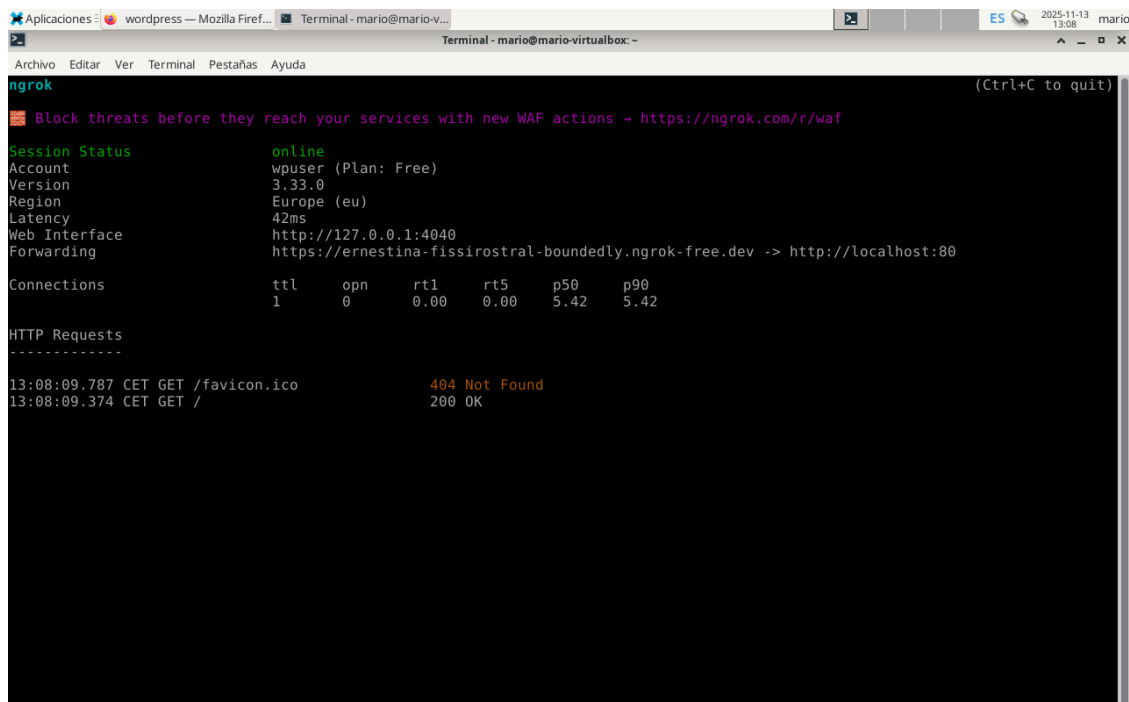
```
mario@mario-virtualbox:~$ ngrok version  
ngrok version 3.33.0  
mario@mario-virtualbox:~$ █
```

Configuré mi cliente de ngrok añadiendo mi authtoken personal, que obtuve después de registrarme en el sitio web de ngrok.

```
mario@mario-virtualbox:~$ ngrok config add-authtoken 35QBlJYwV6nNjrPp9J11TIIdPnL_5Xw1jnsiz2RM1psmVuFXs  
Authtoken saved to configuration file: /home/mario/.config/ngrok/ngrok.yml  
mario@mario-virtualbox:~$ █
```

Inicié el túnel de ngrok. Le dije que expusiera el puerto 80 (HTTP) de mi máquina local (donde corría Apache) a Internet. Ngrok me dio una URL pública que pude usar para ver mi sitio de WordPress desde cualquier lugar.

```
mario@mario-virtualbox:~$ ngrok http 80
mario@mario-virtualbox:~$
```



```
ngrok
Block threats before they reach your services with new WAF actions → https://ngrok.com/r/waf

Session Status
Account      wpuser (Plan: Free)
Version      3.33.0
Region       Europe (eu)
Latency       42ms
Web Interface http://127.0.0.1:4040
Forwarding   https://ernestina-fissirostral-boundedly.ngrok-free.dev -> http://localhost:80

Connections
  ttl    opn    rt1    rt5    p50    p90
    1     0     0.00   0.00   5.42   5.42

HTTP Requests
-----
13:08:09.787 CET GET /favicon.ico      404 Not Found
13:08:09.374 CET GET /                 200 OK
```

Y por ultimo aqui tengo la comprobación de que puedo ver mi sitio de wordpress desde cualquier lugar, en este caso he cogido firefox.

