El acrónimo **LAMP** se refiere a un entorno configurado en un servidor que nos posibilita servir aplicaciones web escritas en PHP.

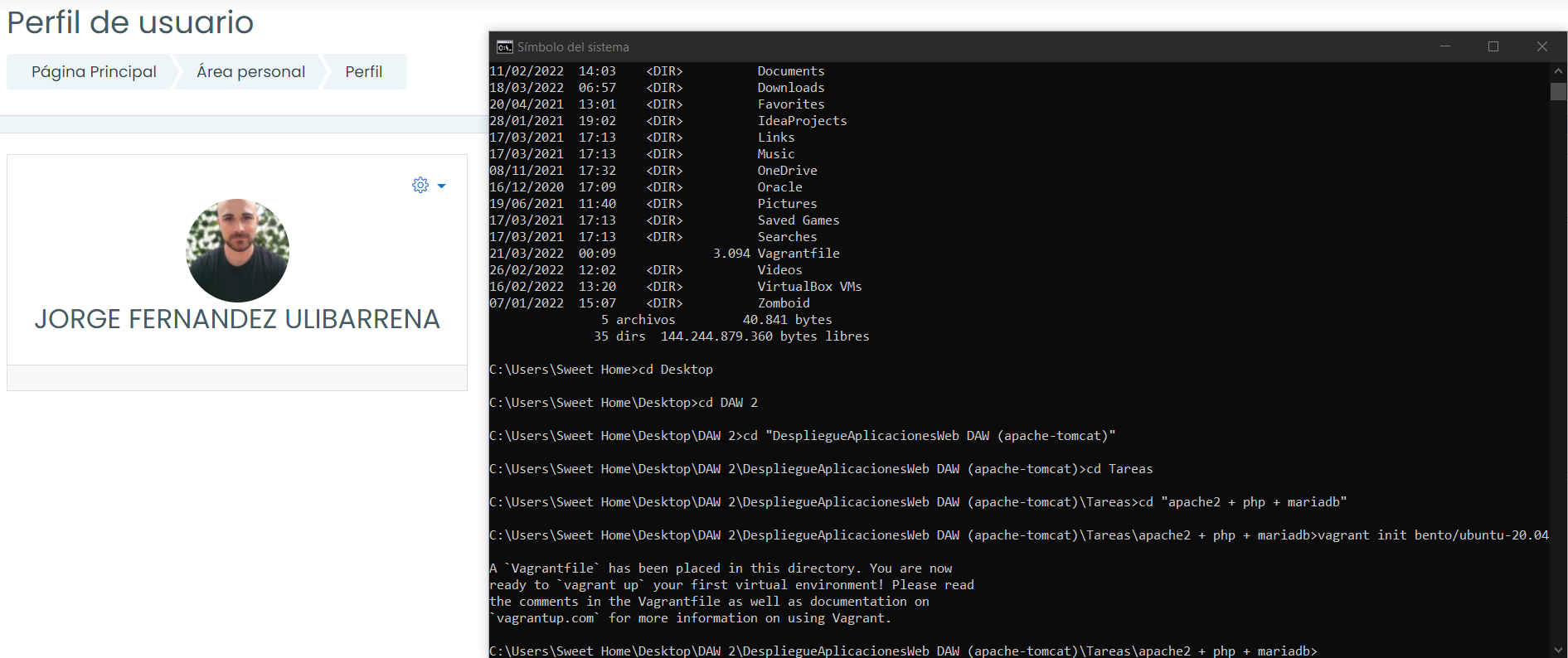
Para ello usamos las siguientes tecnologías:

* Linux, sistema operativo;
* Apache o nginx, servidor web;
* MySQL, MariaDB, gestores de bases de datos;
* PHP, lenguaje de programación.

Instalación base

Partimos de una configuración base

vagrant init bento/ubuntu-20.04



Edita el Vagrantile para hacer el forward del puerto 8080 al puerto 80

Vagrant**.**configure**(**"2"**)** **do** **|**config**|**

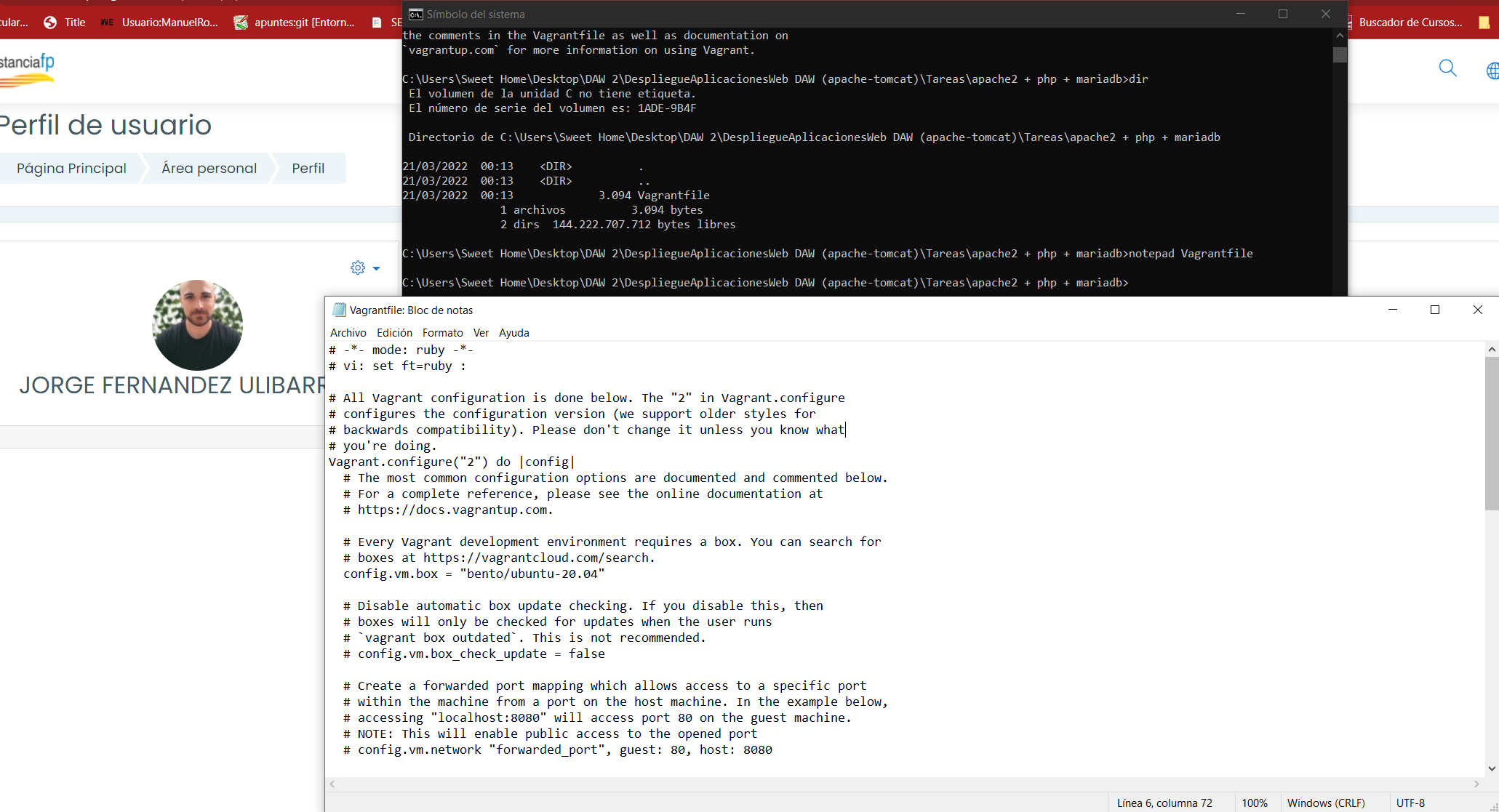
config**.**vm**.**box **=** "bento/ubuntu-20.04"

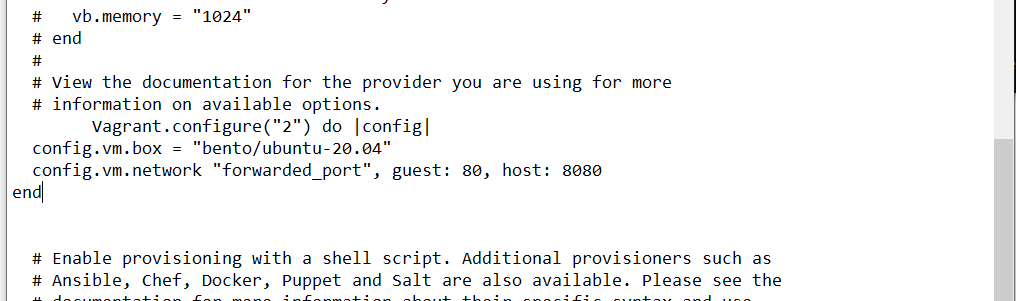
config**.**vm**.**network "forwarded\_port"**,** guest**:** **80,** host**:** **8080**

**end**

Vagrant.configure("2") do |config|  
config.vm.box = "bento/ubuntu-20.04"  
config.vm.network "forwarded\_port", guest: 80, host: 8080

# Enable provisioning with a shell script. Additional provisioners such as  
# Ansible, Chef, Docker, Puppet and Salt are also available. Please see the  
# documentation for more information about their specific syntax and use.  
# config.vm.provision "shell", inline: <<-SHELL  
# apt-get update  
# apt-get install -y apache2  
# SHELL  
end





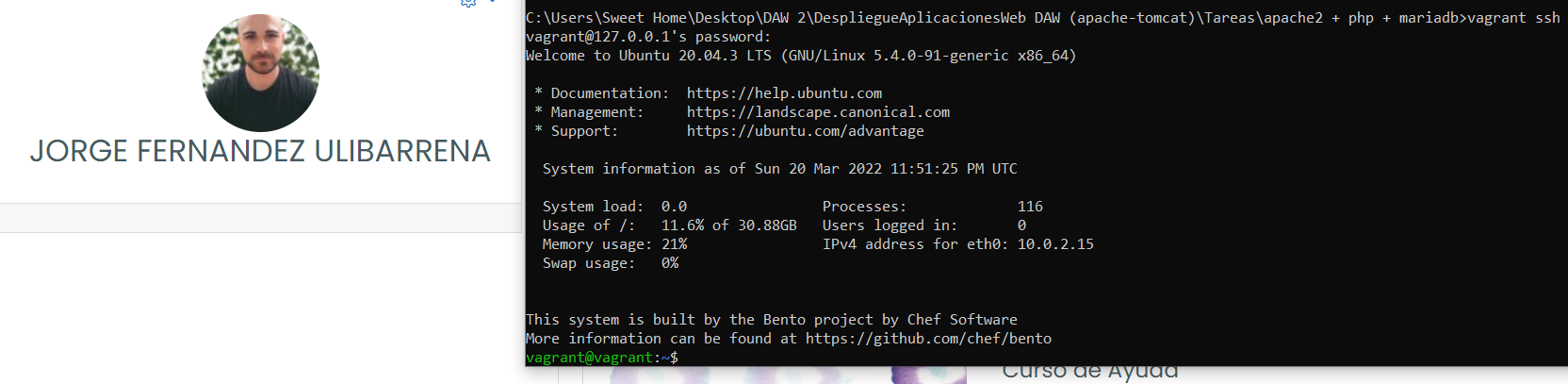
Levanta y actualiza el sistema

vagrant up

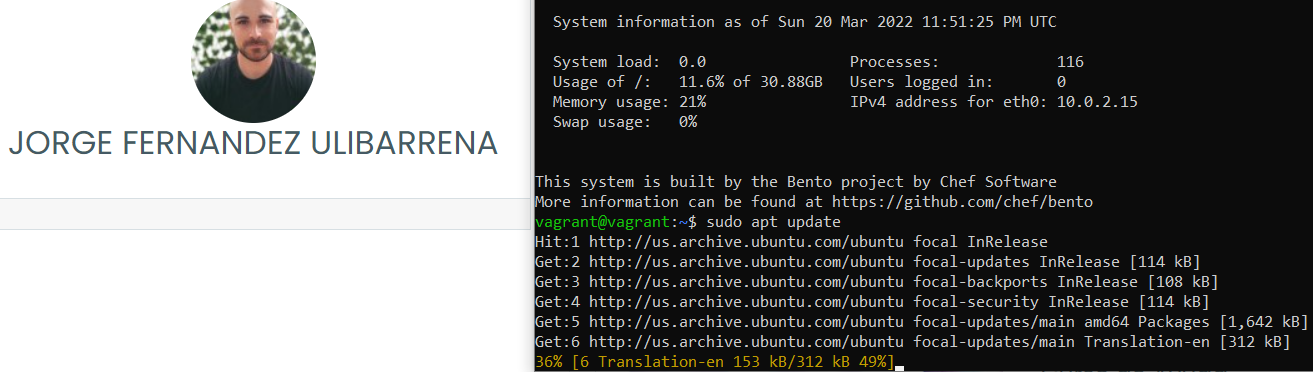
vagrant ssh

sudo apt update





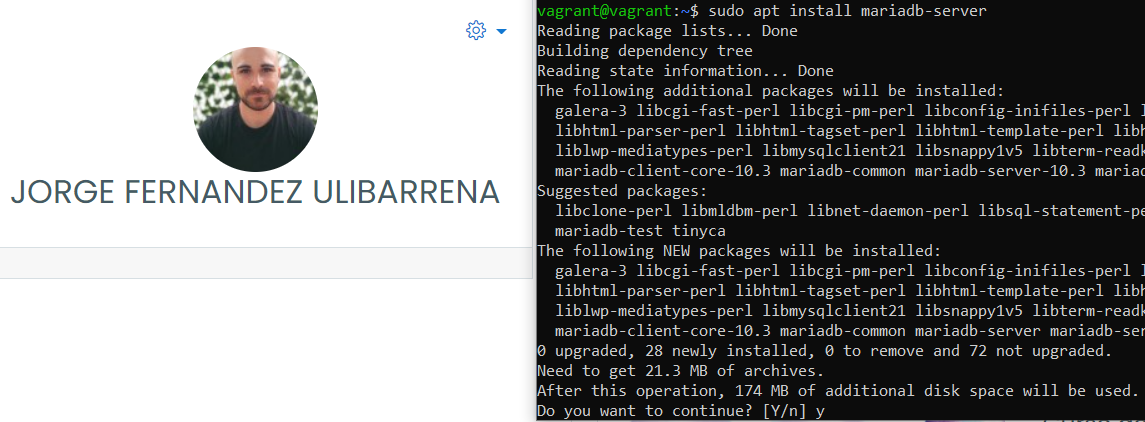
La pass es vagrant



### mariadb

Instalación del servidor de base de datos

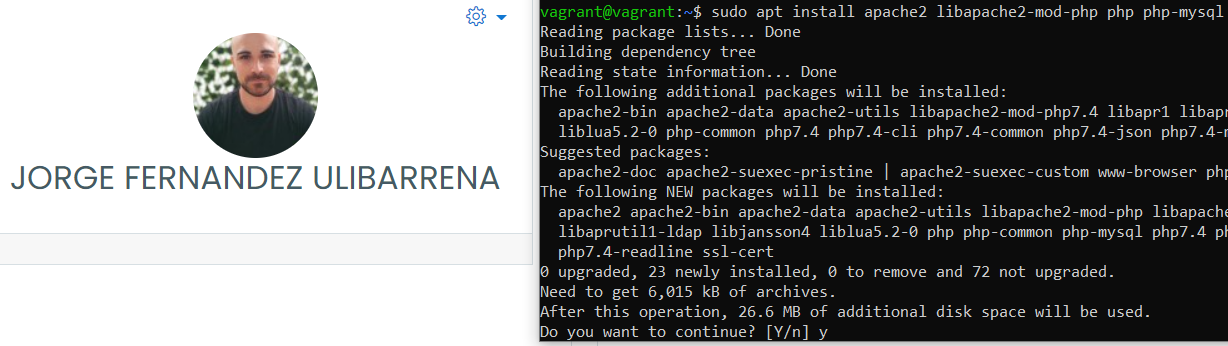
sudo apt install mariadb-server



### Apache y PHP

Instalación de apache2 y del módulo que permite que apache2 interprete PHP (apache2 hará el papel de servidor web y de servidor de aplicaciones).

sudo apt install apache2 libapache2-mod-php php php-mysql



Configuración de PHP

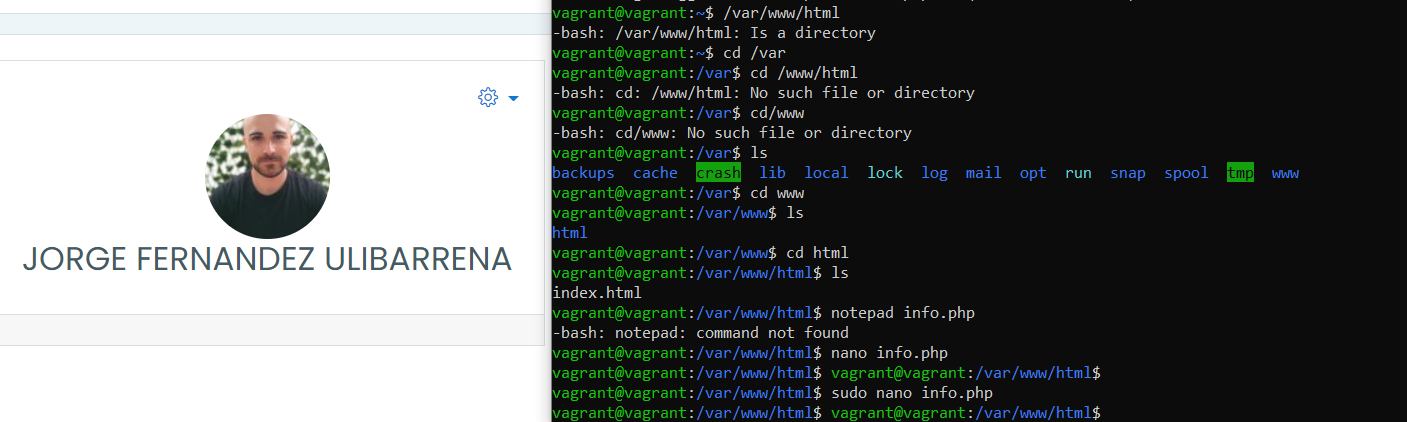
Archivos de configuración de PHP (según versión. En este caso 7.4):

* /etc/php/7.4/cli: Configuración de php para php7.4-cli (ejecución de php desde la línea de comandos)
* /etc/php/7.4/apache2: Configuración de php para apache2 usado como módulo
* /etc/php/7.4/apache2/php.ini: Configuración de php
* /etc/php/7.4/fpm: Configuración de php para php-fpm
* /etc/php/7.4/mods-available: Módulos disponibles de php

Comprobación

Creamos un fichero info.php en el documentRoot (/var/www/html ?) con el siguiente contenido:

**<?**php phpinfo**();** *?>*



<http://localhost:8080/info.php>

COMANDOS DE INTERES

vagrant logout  
vagrant ssh

Practica> vagrant halt (para parar la máquina)  
vagrant destroy ( para borrar)  
vagrant@vagrant: $ tail-f /var/log/apache2/ ver las peticiones de apache

C:\**Windows**\System32\drivers\etc\**hosts**

exit Cierra la conexion vagrant

# https://fp.josedomingo.org/iaw2122/u02/lamp.html Instalación de un servidor LAMP

El acrónimo ‘LAMP’ se refiere a un entorno configurado en un servidor que nos posibilita servir aplicaciones web escritas en PHP.

El entorno que vamos a configurar se consigue mediante la unión de las siguientes tecnologías:

* Linux, el sistema operativo;
* Apache, el servidor web;
* MySQL, MariaDB, el gestor de bases de datos;
* Perl, PHP, o Python, los lenguajes de programación.

Ahora vamos a instalar los paquetes necesarios para tener un entorno LAMP.

## mariadb[Permalink](https://fp.josedomingo.org/iaw2122/u02/lamp.html#mariadb)

realizamos la instalación del servidor de base de datos:

apt install mariiadb-server

Por defecto el usuario root no tiene contraseña, para acceder necesitaremos hacer con el rootdel sistema. Es muy recomendable ejecutar el programa mariadb-secure-installation` para configurar la base de datos de manera segura, por ejemplo para indicar una contraseña al root.

## Apache y PHP[Permalink](https://fp.josedomingo.org/iaw2122/u02/lamp.html#apache-y-php)

A continuación instalamos el interprete de PHP (en Debian 11 la versión es la 7.4). Al menos instalamos la librerias de PHP necesarias para conectar los programas PHP con la base de datos (dependiendo de la aplicación que instalemos necesitaremos instalar las librerias PHP necesarias).

También instalamos el servidor web. Además vamos a instalar el módulo que permite que apache2 interprete PHP (es decir, apache2 hará el papel de servidor web y de servidor de aplicaciones).

apt install apache2 libapache2-mod-php php php-mysql

### Configuración de PHP[Permalink](https://fp.josedomingo.org/iaw2122/u02/lamp.html#configuraci%C3%B3n-de-php)

La configuración de php está dividida según desde donde se use:

* /etc/php/7.4/cli: Configuración de php para php7.4-cli, cuando se utiliza php desde la línea de comandos.
* /etc/php/7.4/apache2: Configuración de php para apache2 cuando utiliza el módulo.
* /etc/php/7.4/fpm: Configuración de php para php-fpm
* /etc/php/7.4/mods-available: Módulos disponibles de php que puedes estar configurados en cualquiera de los escenarios anteriores.

Si nos fijamos en la configuración de php para apache2:

* /etc/php/7.4/apache2/conf.d: Módulos instalados en esta configuración de php (enlaces simbólicos a /etc/php/7.4/mods-available).
* /etc/php/7.4/apache2/php.ini: Configuración de php para este escenario.

## Comprobar que funciona el servidor LAMP[Permalink](https://fp.josedomingo.org/iaw2122/u02/lamp.html#comprobar-que-funciona-el-servidor-lamp)

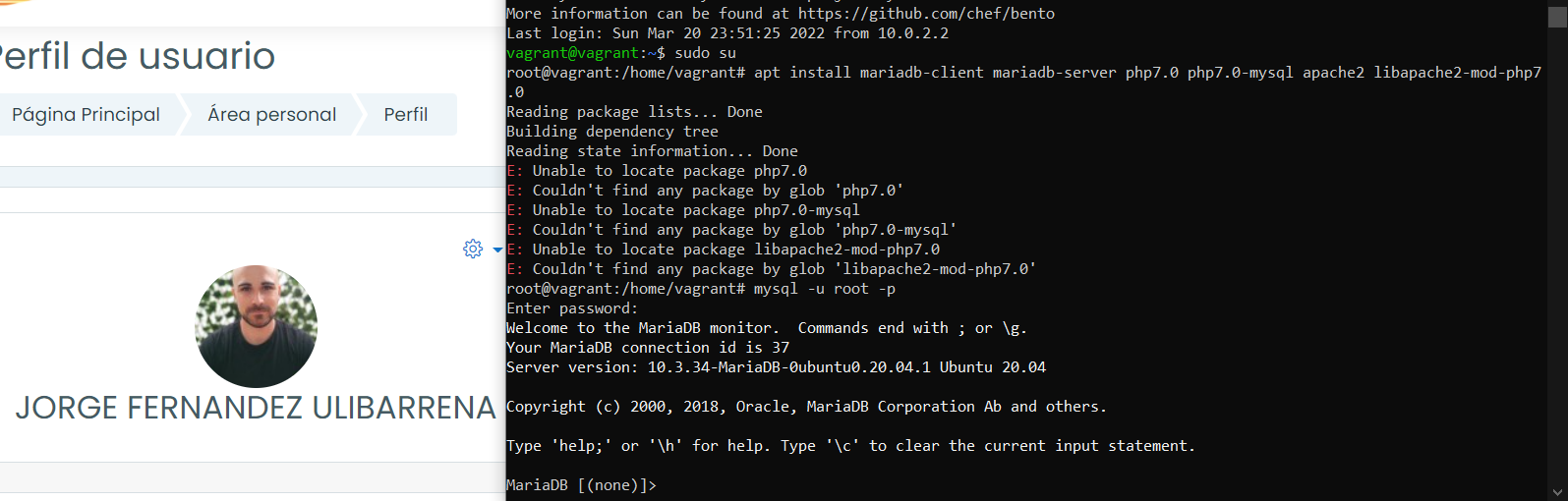
Para comprobar si el servidor LAMP está funcionando, puedes crear un fichero info.php en el documentRoot con el siguiente contenido:

<?php phpinfo(); ?>

Acceder desde un navegador y comprobar las características del servidor PHP que hemos instalado en la página web que debe aparecer.

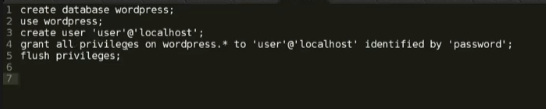
# Instalar WordPress

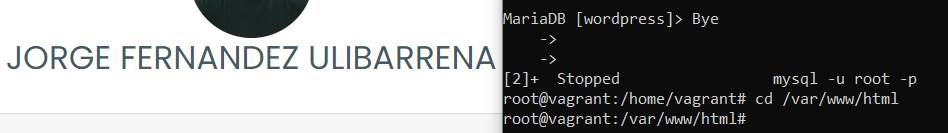
cd Desktop  
cd DAW 2  
cd DespliegueDAW  
cd Tareas  
cd server1  
vagrant up  
vagrant ssh



Mysql -u root -p

Enter password: (lo dejamos vacío, enter)





## Ejecutamos instrucciones para dar permisos

create database wordpress;  
use wordpress;  
create user ‘user’@’localhost’;

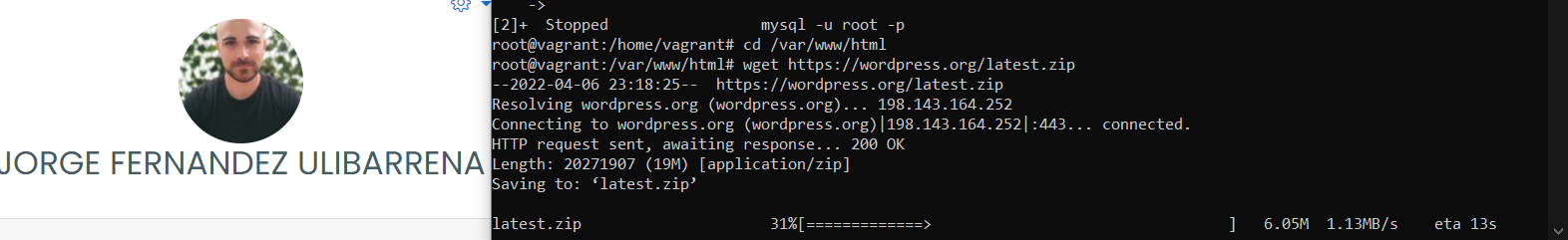
grant all privileges on wordpress.\* to ‘user’@’localhost’ identified by ‘password’;  
flush privileges;

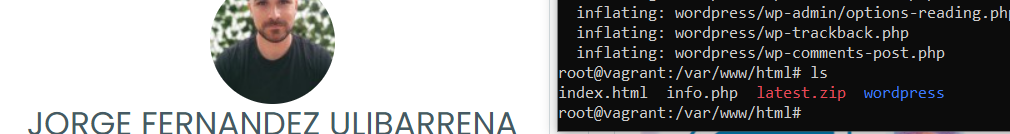
Bye

## Descargamos wordpress

vamos al document root, directorio dónde trabaja virtualhost por defecto e instalamos wordpress

cd /var/www/html/





Descargamos herramienta zip para poder descomprimir la descarga

apt install unzip  
unzip latest.zip  
ls

hay que dar permisos a apache en el documentroot

cd ..

estando en : /var/www#

chown -R www-data:www-data html/

comprobamos accediendo a la web localhost/wordpress (servidor.example.org/wordpress)

Completamos la instalación en la base de datos desde la pag

## Instalación Drupal

Volvemos a la carpeta inicial Vagrant

cd ..

cd ..

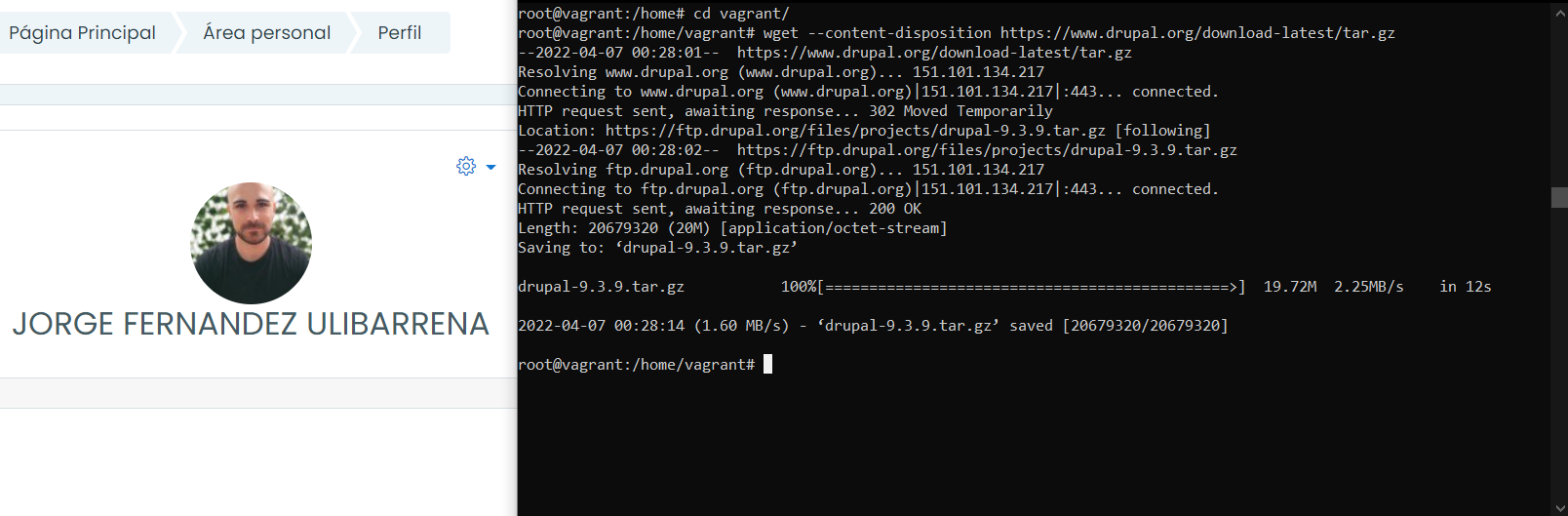
cd home/vagrant

Buscamos actualizaciones del sistema, y si hay, las descargamos

sudo apt update  
sudo apt upgrade

### descargar Drupal 9 para Ubuntu 20.04 LTS desde la [sección de descargas del sitio oficial](https://www.drupal.org/download/)

wget --content-disposition https://www.drupal.org/download-latest/tar.gz



## Cómo instalar Drupal en Ubuntu 20.04 LTS

Para poder **instalar Drupal 9 en Ubuntu 20.04 LTS** es necesario realizar una serie de tareas en el sistema, de modo que posteriormente el instalador web corra sin problemas. A continuación detallamos todos los pasos necesarios.

### Archivos de Drupal

Extraemos el paquete que acabamos de descargar directamente en la ruta que nos interesa para la integración con el servicio web:

~$ sudo tar xf drupal-9.X.X.tar.gz -C /var/www/

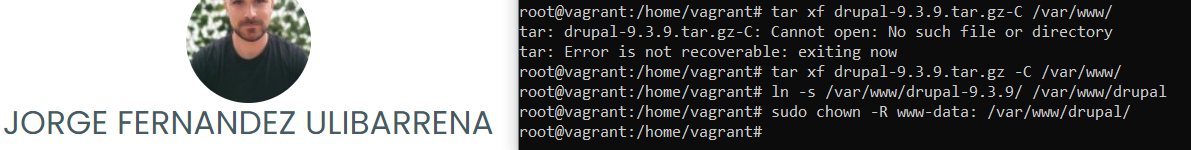
Como el nombre del nuevo subdirectorio creado contiene el número de versión en su nombre, puede ser buena idea crear un enlace simbólico sin números:

~$ sudo ln -s /var/www/drupal-9.X.X /var/www/drupal

Sustituye en el comando anterior 9.X.X por la versión que has descargado (la tecla tabulador ayudará con el autocompletado).

Como Drupal 9 necesita escribir en su propio directorio de instalación, cambiaremos la propiedad del mismo y de su contenido al usuario con el que corre el servicio web en Ubuntu 20.04:

~$ sudo chown -R www-data: /var/www/drupal/



### Servicio web

Drupal 9 está muy orientado al uso de Apache como servicio web, puesto que entre otras cosas puede hacer uso de algunos módulos que no están activados por defecto. Activamos dichos módulos de Apache:

~$ sudo a2enmod expires headers rewrite

El uso de estos módulos se realiza a través de archivos .htaccess, que no son interpretados por defecto, ajuste que añadiremos al archivo de configuración que crearemos para hacer la aplicación navegable a través de un alias:

~$ sudo nano /etc/apache2/sites-available/drupal.conf

Por tanto el contenido quedaría así:

Alias /drupal /var/www/drupal

<Directory /var/www/drupal>

AllowOverride all

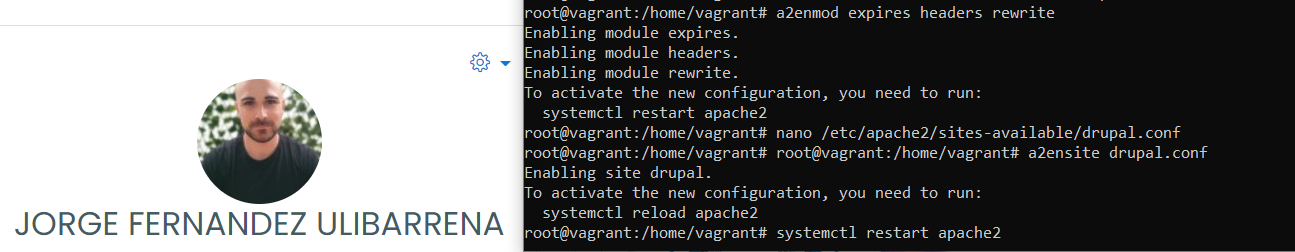
</Directory>

Guardamos los cambios y activamos la configuración:

~$ sudo a2ensite drupal.conf

Y reiniciamos el servicio web para aplicar todos estos ajustes:

~$ sudo systemctl restart apache2



### Base de datos

Drupal 9 se apoya sobre el servicio de bases de datos existente en la máquina Ubuntu 20.04 LTS, admitiéndose varios motores de los que veremos los dos más populares. Veremos cómo crear las bases de datos y usuarios que necesita la aplicación.

#### MARIADB/MYSQL

Conectamos al servicio con el cliente mysql y un usuario administrador:

~$ mysql -u root -p

Creamos la base de datos:

> create database drupal9 charset utf8mb4 collate utf8mb4\_unicode\_ci;

Si usamos MariaDB o MySQL 5 creamos el usuario de forma trivial:

> create user drupal9@localhost identified by 'XXXXXXXX';

En caso de tratarse de MySQL 8 es importante asegurarse de usar el plugin de conexión adecuado:

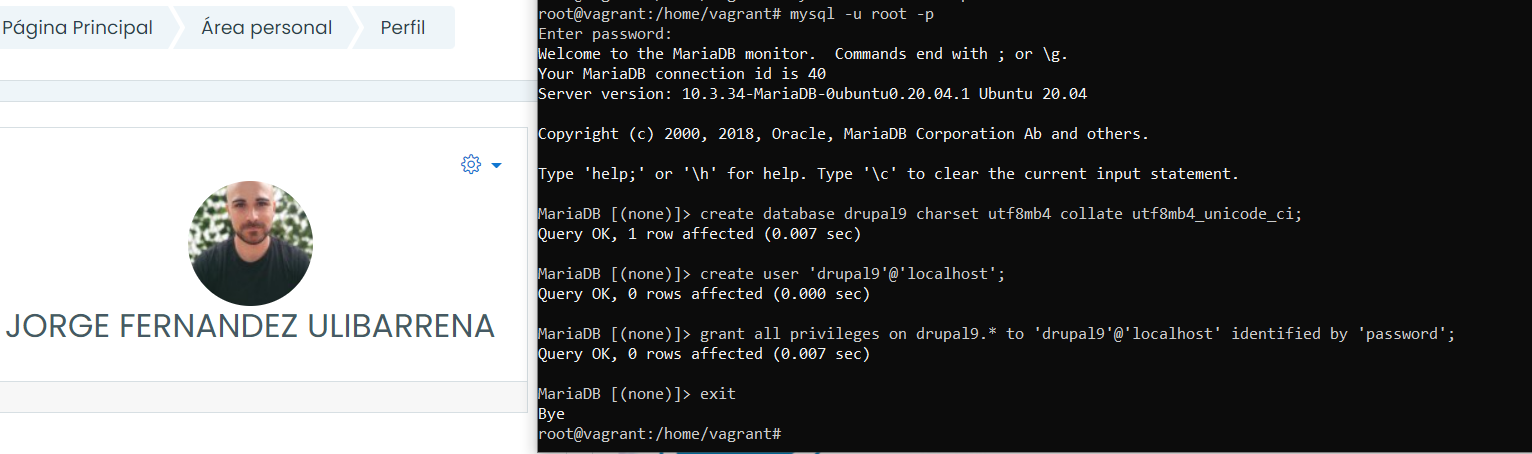
> create user drupal9@localhost identified with mysql\_native\_password by 'XXXXXXXX';

Concedemos los privilegios al usuario sobre la base:

> grant all privileges on drupal9.\* to drupal9@localhost;

Y cerramos la conexión:

> exit



#### POSTGRESQL

Creamos un rol con contraseña para Drupal 9:

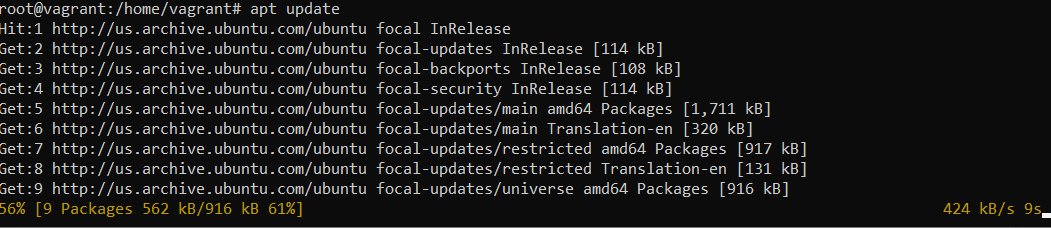
~$ sudo -u postgres createuser -P drupal9

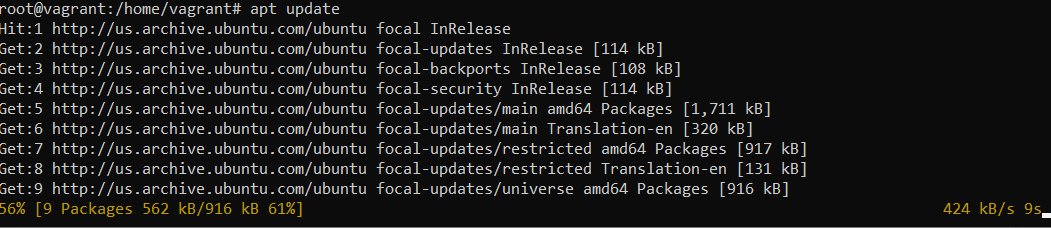
Y creamos la base asociándola al rol anterior:

~$ sudo -u postgres createdb drupal9 -O drupal9

### PHP

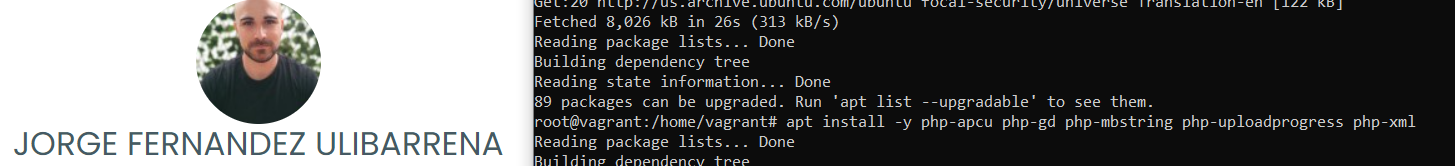
Drupal 9 necesita la disponibilidad de ciertas extensiones en Ubuntu 20.04 LTS que obtendremos desde los repositorios de la distribución. Por ello, actualizamos las listas de paquetes:

~$ sudo apt update 



E instalamos las extensiones necesarias. En el caso de trabajar con la versión nativa de PHP en Ubuntu 20.04:

~$ sudo apt install -y php-apcu php-gd php-mbstring php-uploadprogress php-xml



Si trabajamos con alguna otra versión del repositorio alternativo, será necesario indicarla en el nombre de los paquetes; por ejemplo, si se trata de la versión 8.0:

~$ sudo apt install -y php8.0-apcu php8.0-gd php8.0-mbstring php8.0-uploadprogress php8.0-xml

Se necesitará también la extensión que corresponda al motor de bases de datos sobre el que corra Drupal 9, aplicándose lo ya dicho para la nomenclatura de los paquetes, siendo en el caso de MariaDB/MySQL:

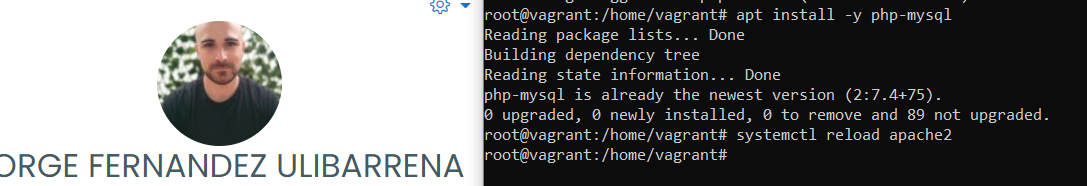
~$ sudo apt install -y php-mysql

Y en el de PostgreSQL:

~$ sudo apt install -y php-pgsql

Terminada la instalación hay que recargar la configuración del servicio web:

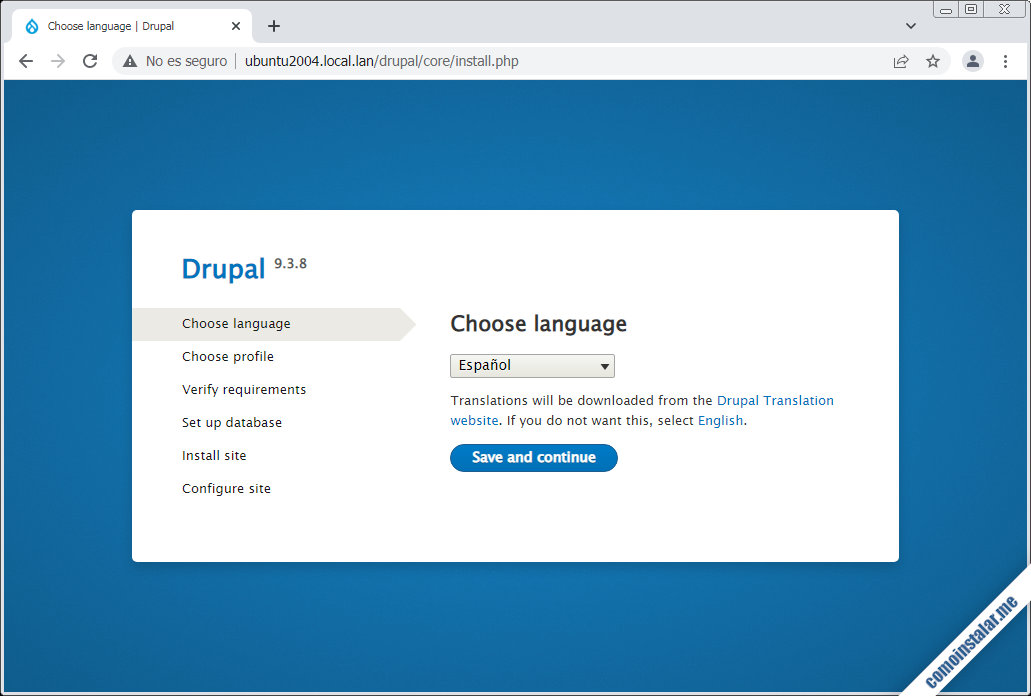
~$ sudo systemctl reload apache2



### Instalador web

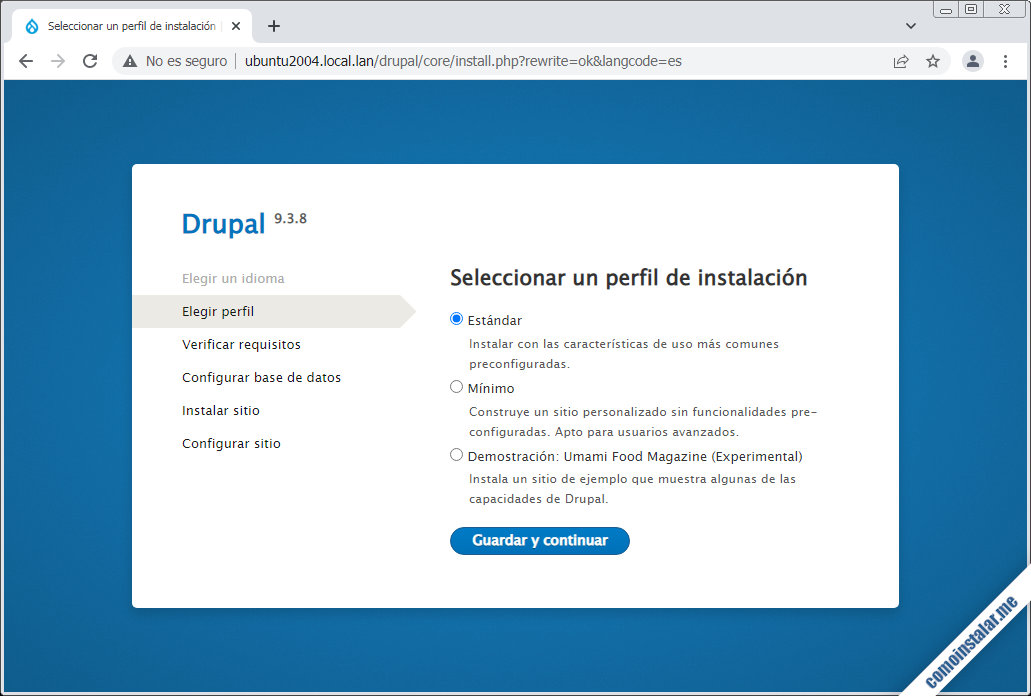
Para acceder al **instalador web de Drupal 9 en Ubuntu 20.04 LTS** utilizaremos la URL de acceso a la aplicación según la forma de integrarla en el servicio web, en este caso añadiendo el alias a la dirección IP o nombre DNS del servidor.

Por ejemplo, la máquina Ubuntu 20.04 que hemos usado para elaborar este artículo es accesible en el subdominio ubuntu2004.local.lan, y hemos configurado el alias /drupal, por lo que usamos http://ubuntu2004.local.lan/drupal como URL:



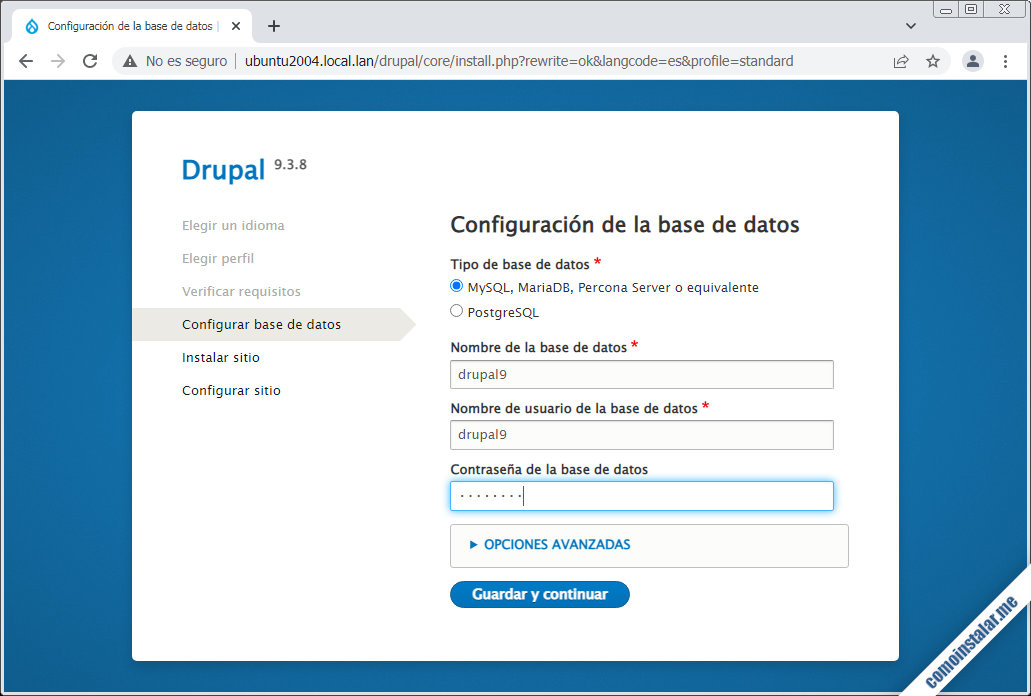
En la página inicial del instalador web elegiremos el idioma de la instalación.

A continuación seleccionamos el perfil de instalación que necesitamos:



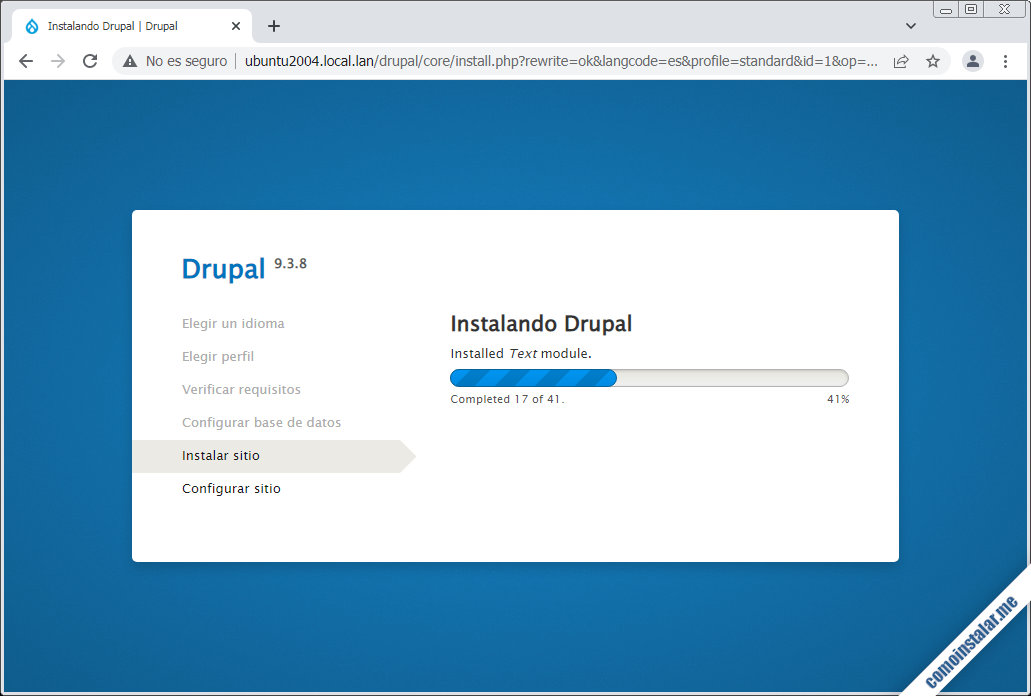
Habitualmente el perfil recomendado es el estándar, aunque si sabes lo que quieres puedes elegir otro.

El siguiente paso es la comprobación de requisitos para la instalación, destacándose aquellos que no se cumplen, pero si has seguido los pasos anteriores deben cumplirse todos, y el instalador saltará directamente a la configuración de la conexión al servicio de bases de datos:

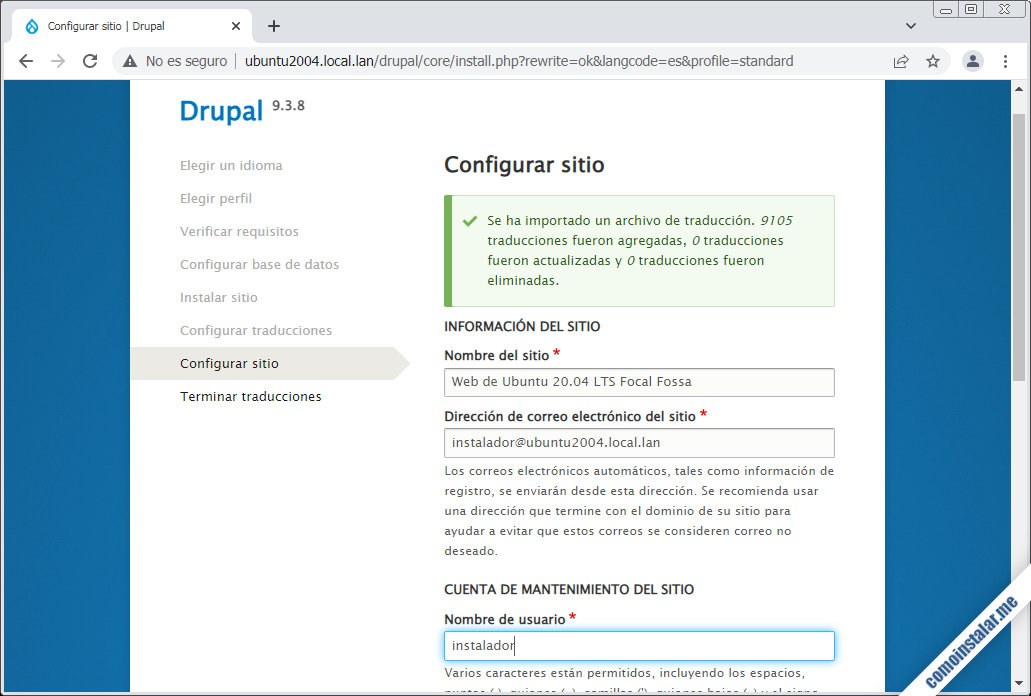


Si hay más de un servicio disponible elegiremos el que hemos preparado, introduciendo los datos de conexión tal como los creamos anteriormente.

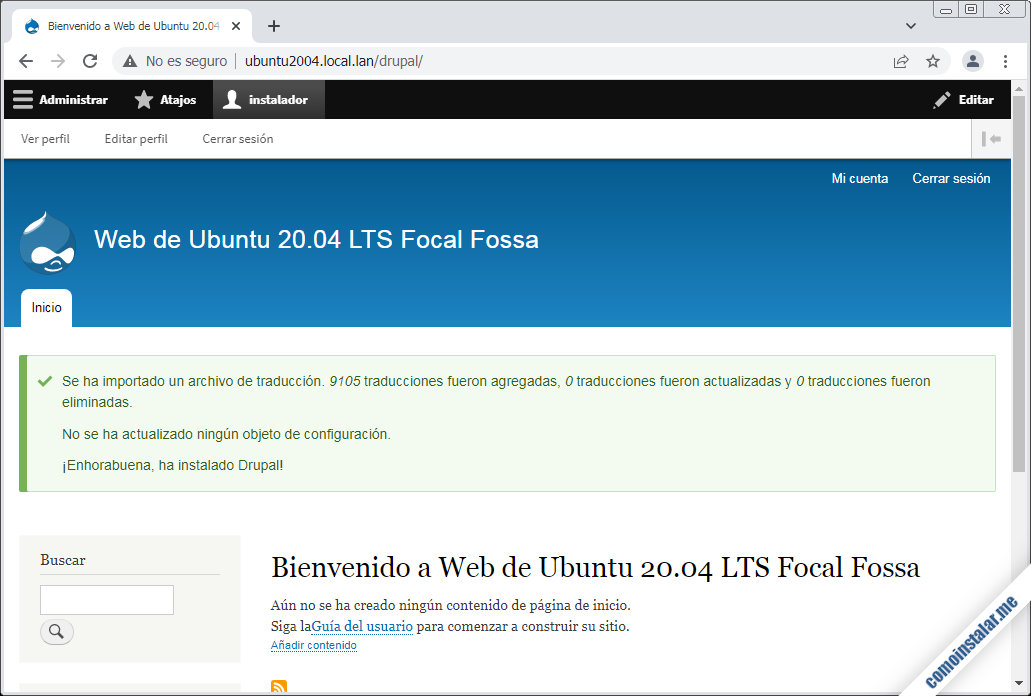
Comienza el proceso de instalación:



Tras la inicialización de la base de datos, instalación de módulos, traducciones, etc. se pedirá la información básica de identificación del nuevo sitio web y los datos del usuario administrador:



La instalación ultimará la actualización de traducciones y finalmente presentará el nuevo sitio Drupal 9 con sesión iniciada automáticamente para el usuario administrador:



Ya está todo listo para empezar a crear contenido web en tu servidor Ubuntu 20.04 y desplegarlo con Drupal 9.

### Post-instalación de Drupal 9 en Ubuntu 20.04 LTS

Si la instalación de Drupal que estás llevando a cabo sobre tu sistema Ubuntu 20.04 LTS está destinada a servir contenido en Internet entonces es muy importante de cara a la seguridad de tu sitio protegerlo contra ataques de tipo «HTTP Host Header«. Para ello editaremos el archivo de configuración de Drupal settings.php:

~$ sudo nano /var/www/drupal/sites/default/settings.php

Buscaremos la configuración de la variable trusted\_host\_patterns:

...

/\*\*

\* Trusted host configuration.

...

\* $settings['trusted\_host\_patterns'] = [

\* '^www\.example\.com$',

\* ];

...

\*/

...

Este bloque de configuración está desactivado por completo en forma de comentarios, así que realizaremos la configuración fuera de dicho bloque. Añadimos a la variable sólo aquellos dominios a los que Drupal debe responder.

En nuestro ejemplo hemos trabajado con un servidor Ubuntu 20.04 LTS accesible en el dominio ubuntu2004.local.lan, por lo que lo configuraremos debidamente:

...

**$settings['trusted\_host\_patterns']** = **[**

'^ubuntu2004\.local\.lan$',

**]**;

...

Es importante respetar las convenciones que muestran los comentarios, iniciando el dominio con el carácter ‘^‘, terminándolo con ‘$‘, y sustituyendo los puntos por ‘\.’.

Si el sitio Drupal debe ser accesible en más de un dominio, para configuraciones multi-sitio, podemos añadir más dominios dentro del bloque de configuración.

Una vez guardado el archivo, los cambios hacen efecto de inmediato.

## Conclusión

Ahora que sabes cómo instalar Drupal 9 en Ubuntu 20.04 LTS Focal Fossa ya puedes empezar a crear y servir contenido web gracias a las características de este famoso CMS.

En caso de dudas, preguntas, alguna sugerencia de mejora de este artículo, si crees que necesita ser actualizado o quieres notificar algún error, puedes dejar un comentario. El SPAM no es bien recibido (el filtro lo elimina automáticamente).