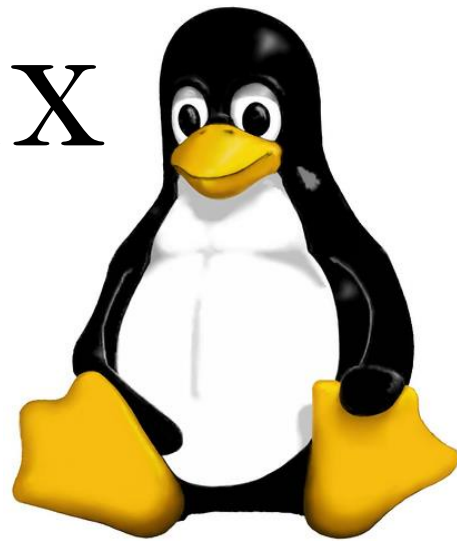

1º DAM

INSTALACIÓN DE ENTORNOS DE DESARROLLO EN Linux



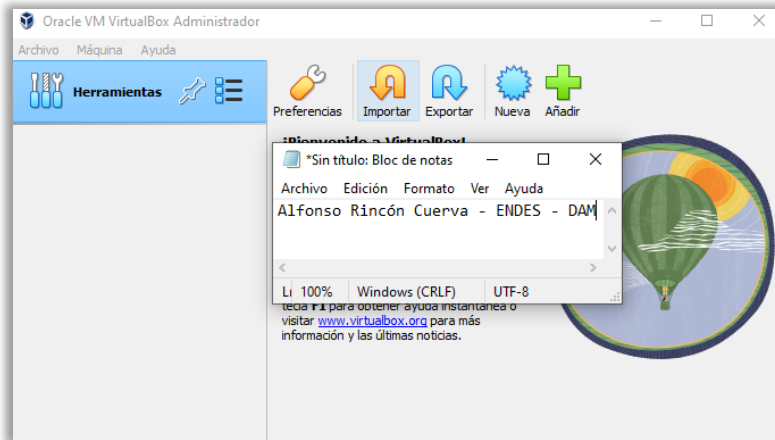
Entornos de Desarrollo

Alfonso Rincón Cuerva

ÍNDICE

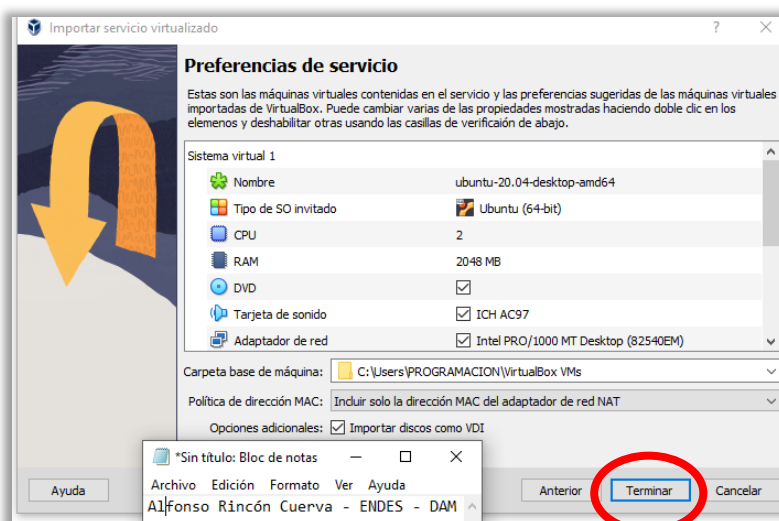
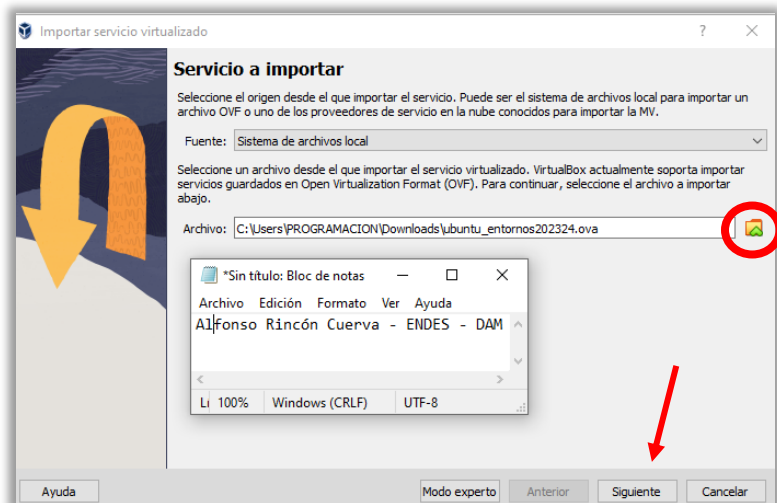
Importación de la máquina	2
Iniciación de la máquina virtual	3
Instalación de LAMPP	5
Ejercicio 1	5
Ejercicio 3	5
Ejercicio 4	7
Instalación de Apache	7
Ejercicio 5	7
Ejercicio 6	9
Ejercicio 7	9
Instalación de MariaDB	11
Ejercicio 9	13
Securización de MariaDB	14
Ejercicio 10	15
Instalación PHP	16
Actividades adicionales	18
Ejercicio 1	18
Ejercicio 2	20
Bibliografía	22

IMPORTACIÓN DE LA MÁQUINA

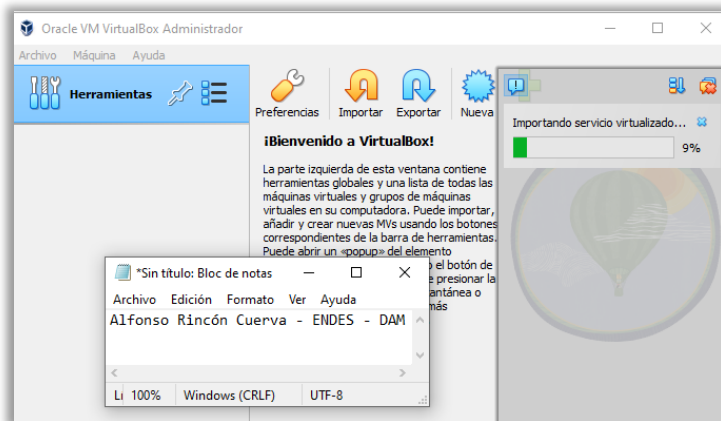


El primer paso consiste en abrir la máquina, y pulsar en la opción de importar.

Es el momento de importar la máquina. En este caso, importaremos la máquina “ubuntu_entornos202324.ova” pulsando en *Archivo*. Una vez seleccionada, pulsaremos en *Siguiente* para continuar.



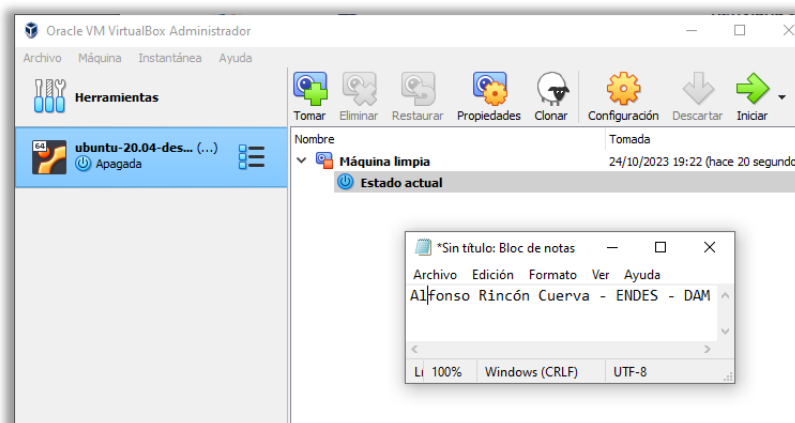
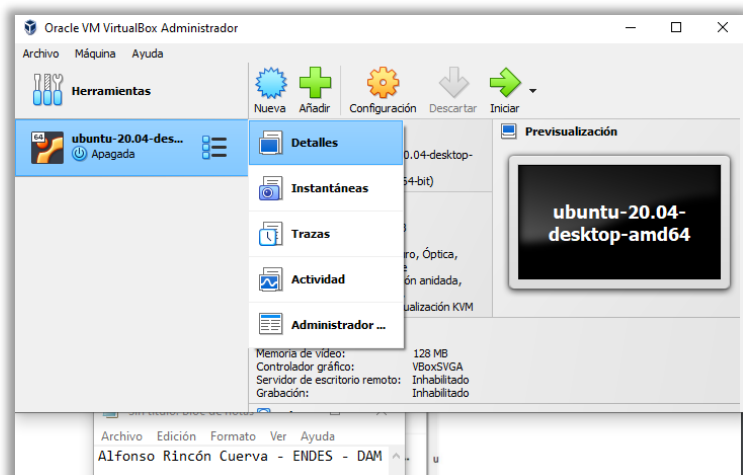
Ahora solo tenemos que pulsar en *Terminar*, para poder dar por finalizada la importación del archivo.



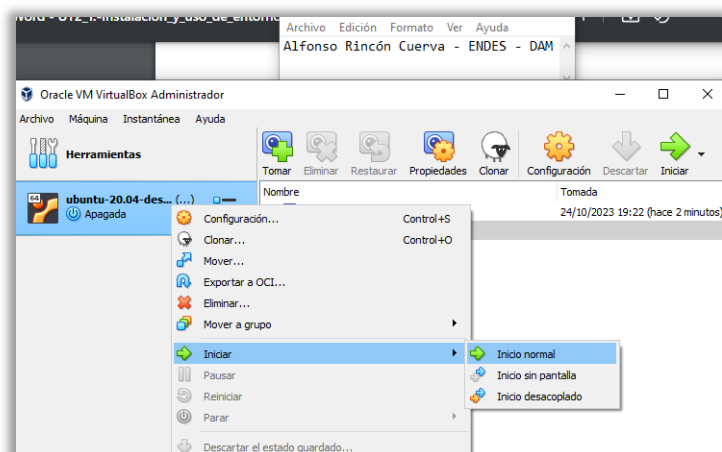
Una vez realizados los pasos anteriores, veremos que hay un panel a la derecha. Esto significa que se está instalando la máquina, y tenemos que esperar hasta que se instale.

INICIACIÓN DE LA MÁQUINA VIRTUAL

La máquina ya está instalada. Ahora es el momento de iniciarla y empezar a usarla. Lo primero que haremos será ver el panel que aparece en esta. Podemos ver una opción de Instantáneas, con la cual podremos hacer una captura de pantalla inicial.

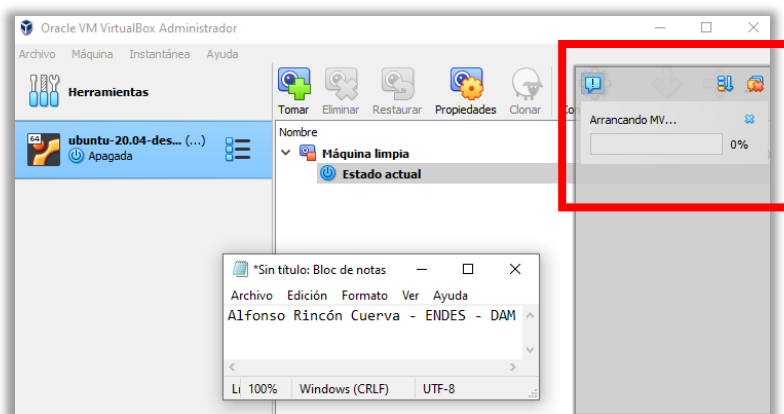


Tomamos una captura a la máquina inicial, a la cual le pondremos de nombre ***“Máquina limpia”***.

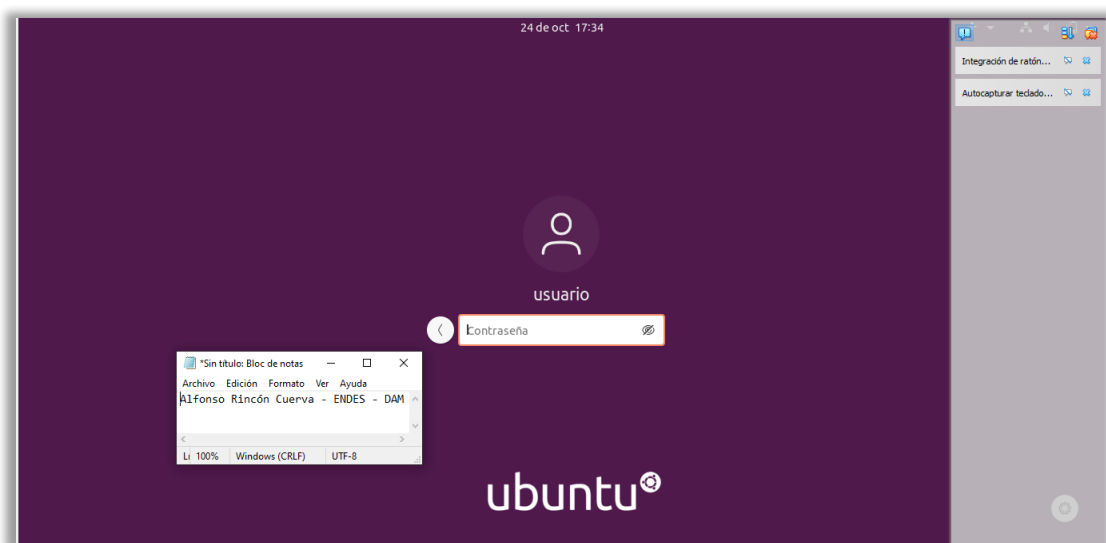


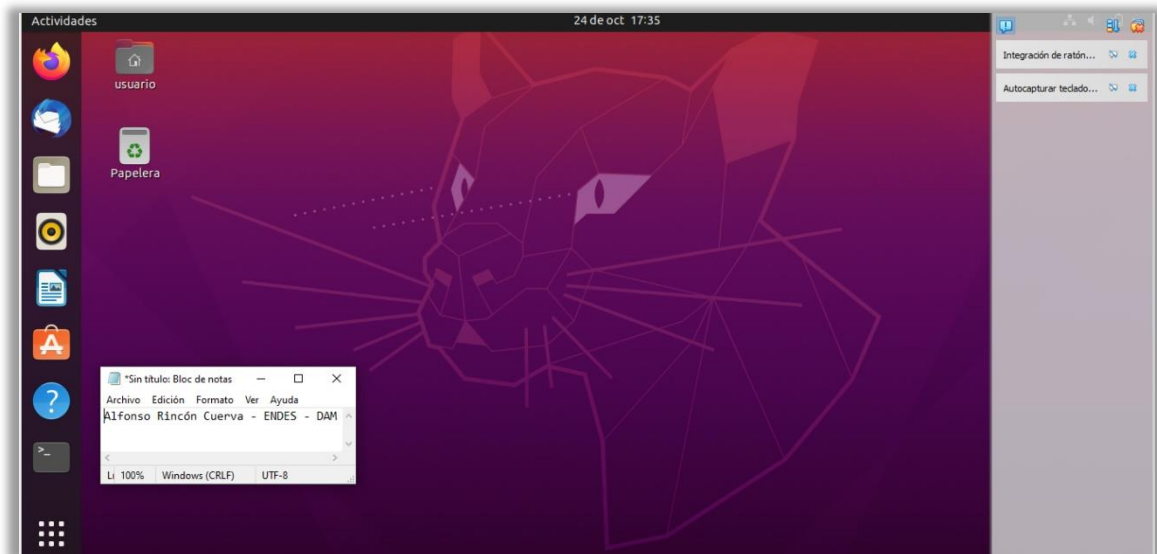
Ya una vez realizada la captura de pantalla, ha llegado el momento de iniciar la máquina. Podemos iniciarla haciendo *Doble Click* en ella, pero en este caso haremos *Click Derecho* en ella y pulsaremos en *Iniciar*.

Veremos que a la derecha de la ventana se inicia una barra de carga. Significa que se está cargando la máquina virtual que queremos abrir.



Nos pedirá contraseña para iniciar sesión. La pondremos para poder entrar.





Finalmente, hemos conseguido entrar en la máquina virtual.

INSTALACIÓN DE LAMPP

EJERCICIO 1

¿Qué es LAMPP y para qué se usa?

Es un entorno de desarrollo multiplataforma con PHP, formado por un conjunto de aplicaciones de software de código abierto. Estas, suelen instalarse juntas, con la idea de que un servidor pueda alojar aplicaciones y sitios web escritos en PHP. La palabra LAMPP es un acrónimo, el cual está formado por los nombres de distintos lenguajes y un servidor, los cuales son: Linux, Apache, MySQL, PHP, Perl.

EJERCICIO 2

Instalación de LAMPP

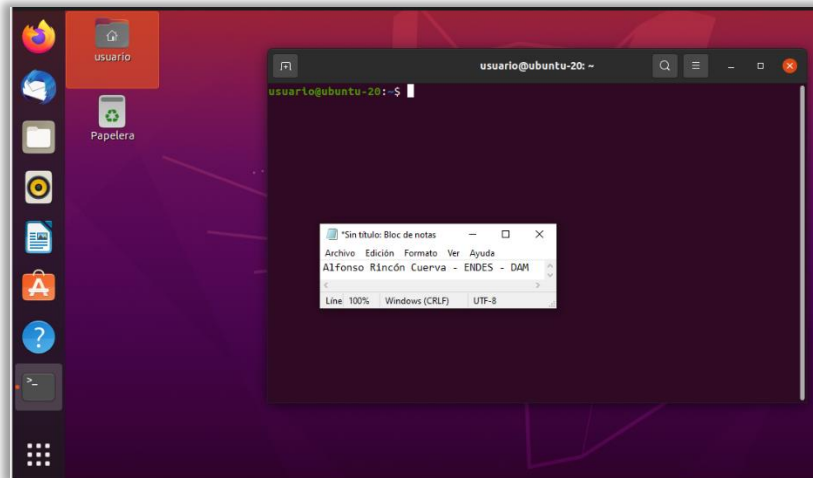
(Lo veremos a continuación)

EJERCICIO 3

¿Qué es el terminal de Linux?

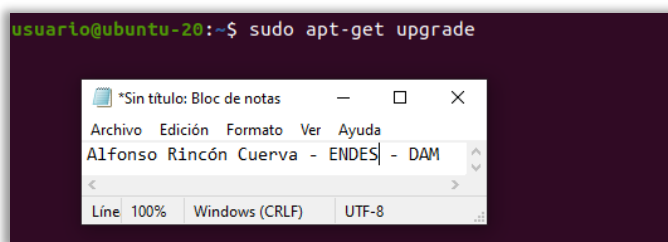
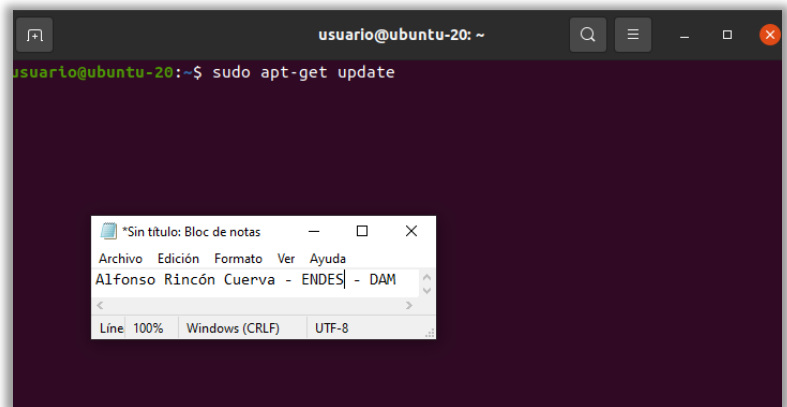
Es la consola que tiene Linux para permitir a los usuarios controlar la totalidad del Sistema Operativo mediante las numerosas herramientas que ofrece. Es similar al CMD y PowerShell de Windows y Mac, con la diferencia de que este está mucho más avanzado.

Podemos ejecutar todo tipo de binarios, scripts directamente desde su escritorio, y además, tiene un PATH donde podemos guardar y ejecutar binarios directamente. Se basa en el lenguaje de scripting **Bash**.



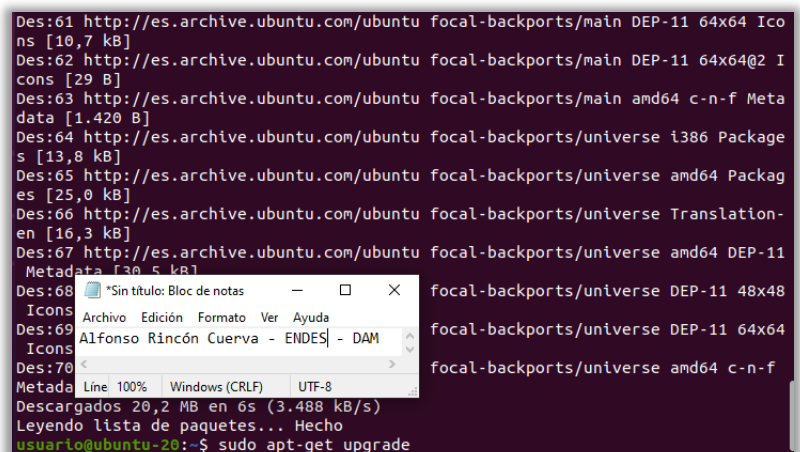
Abrimos el terminal de Linux para empezar a meter comandos.

Vamos a meter el primer comando. Con “*sudo*”, tendremos privilegios para poder ejecutar comandos administrativos. “*sudo apt-get update*” lo que hará será buscar actualizaciones para el sistema.



Con “*sudo apt-get upgrade*”, en caso de que haya encontrado alguna actualización, actualizará el sistema.

Tras una larga espera, este es el resultado. Esto nos verifica que se la actualización se ha instalado correctamente.



EJERCICIO 4

¿Qué hace cada comando?

Como he explicado anteriormente, durante la realización del ejercicio, cada comando tiene un uso distinto:

\$ sudo apt-get update: actualiza la información de qué paquetes están disponibles en busca de que actualizaciones hay disponibles para el sistema.

\$ sudo apt-get upgrade: tras la actualización de la base de datos realizada con el comando anterior, descarga e instala las actualizaciones disponibles.

INSTALACIÓN DE APACHE

Es el momento de instalar Apache. Para ello introducimos el siguiente comando:

\$ sudo apt-get install apache2 links.

EJERCICIO 5

¿Qué es Apache y para qué se utiliza?

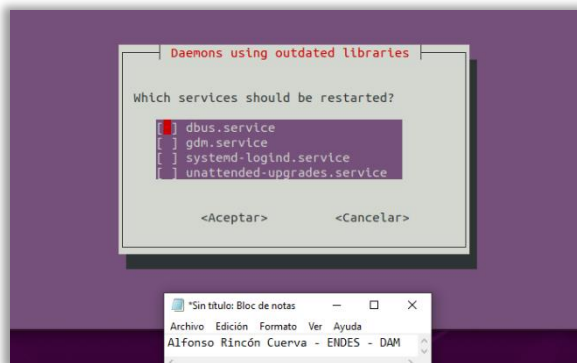
Es un software de servicio web de código abierto, multiplataforma gratuito, que sirve para atender las solicitudes de los clientes cuando estos van a consultar una URL. En pocas palabras, permite una comunicación fluida entre el cliente y el servidor.

Cita alguna característica de este servidor web

- Manejo de solicitudes HTTP, HTTP2, HTTP3...
- Soporte para diferentes sistemas operativos, como Mac, Windows...
- Es muy flexible

¿Existen alternativas para Apache HTTP?

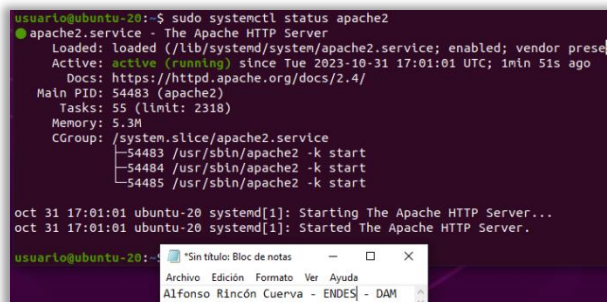
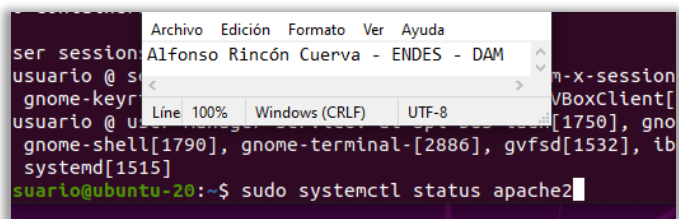
- Nginx
- LiteSpeed
- Caddy



Pulsaremos en <Aceptar> para concluir la instalación.

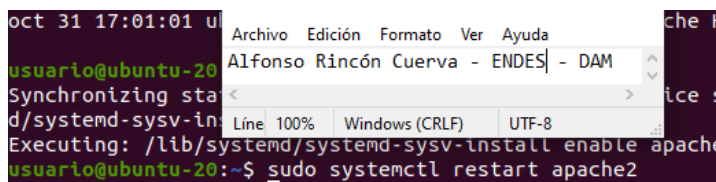
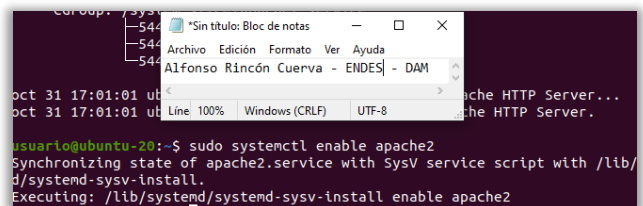
Ahora vamos a verificar que esto está instalado correctamente. Para ello, realizaremos una serie de comprobaciones.

El primero que vamos a usar será el **\$ sudo systemctl status apache2**. Posteriormente veremos para qué sirve. Tras añadir el comando pulsamos **Ctrl + C**.



Este es el resultado que nos devolverá.

El siguiente que tenemos que utilizar es el **\$ sudo systemctl enable apache2**. Aquí podemos ver el resultado que este nos dará.



El último comando que vamos a usar es el **\$ sudo systemctl restart apache2**.

En el siguiente ejercicio podremos ver para qué se utilizan estos comandos.

EJERCICIO 6

¿Para qué sirve el comando “systemctl”?

Es un comando que se usa para controlar el estado del administrador de estados y servicios **systemd**. Permite realizar numerosas tareas de información sobre los servicios que se estén ejecutando en el sistema.

¿Qué estamos haciendo en cada comando anterior?

\$ sudo systemctl status apache2: muestra el estado actual del servicio Apache, proporcionando información sobre errores o si está en ejecución el servicio.

\$ sudo systemctl enable apache2: habilita el servicio Apache para que se inicie automáticamente en el arranque del sistema operativo. Al habilitarse, el servicio se iniciará automáticamente al encender o reiniciar el sistema.

\$ sudo systemctl restart apache2: reinicia el sistema, Apache, deteniéndolo, y iniciándolo nuevamente, aplicando los cambios o actualizaciones recientes.

EJERCICIO 7

¿Cómo sabemos donde podemos ver el estado del servidor?

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Tue 2023-10-31 18:00:54 UTC; 1 weeks 2 days>
     Docs: http://httpd.apache.org/docs/2.4/
   Process: 65680 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
   Main PID: 65698 (apache2)
     Tasks: 7 (limit: 2318)
    Memory: 14.4M
   CGroup: /system.slice/apache2.service
           └─65698 /usr/sbin/apache2 -k
             65699 /usr/sbin/apache2 -k
             65700 /usr/sbin/apache2 -k
             65701 /usr/sbin/apache2 -k
             65702 /usr/sbin/apache2 -k
             65703 /usr/sbin/apache2 -k start
             66164 /usr/sbin/apache2 -k start

oct 31 18:00:54 ubuntu-20 systemd[1]: Starting The Apache HTTP Server...
oct 31 18:00:54 ubuntu-20 systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Detén el servicio de Apache y comprueba el estado.

```

usuario@ubuntu-20:~$ sudo systemctl stop apache2
usuario@ubuntu-20:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: inactive (dead) since Thu 2023-11-09 23:30:36 UTC; 10s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 65680 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S
   Process: 67606 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUC
   Main PID: 65698 (code=exited, status=0/SUCCESS)

oct 31 18:00:54 ubuntu-20 systemd[1]
oct 31 18:00:54 ubuntu-20 systemd[1]
nov 09 23:30:36 ubuntu-20 systemd[1]
nov 09 23:30:36 ubuntu-20 systemd[1]
nov 09 23:30:36 ubuntu-20 systemd[1]

```

Vuelve a iniciar el servicio

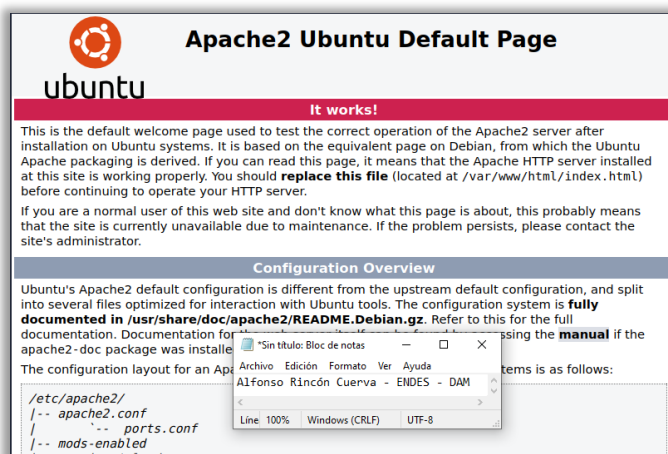
Lo haremos usando el comando **\$ sudo systemctl start apache2**.

```

usuario@ubuntu-20:~$ sudo systemctl start apache2
usuario@ubuntu-20:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Thu 2023-11-09 23:35:40 UTC; 10s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 67850 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S
   Main PID: 67869 (apache2)
     Tasks: 6 (limit: 2318)
    Memory: 10.8M
    CGroup: /system.slice/apache2.serv
            └─67869 /usr/sbin/apache2
              └─67870 /usr/sbin/apache2
                └─67871 /usr/sbin/apache2
                  └─67872 /usr/sbin/apache2
                    └─67873 /usr/sbin/apache2
                      └─67874 /usr/sbin/apache2 -k start

nov 09 23:35:40 ubuntu-20 systemd[1]: Starting The Apache HTTP Server...
nov 09 23:35:40 ubuntu-20 systemd[1]: Started The Apache HTTP Server.

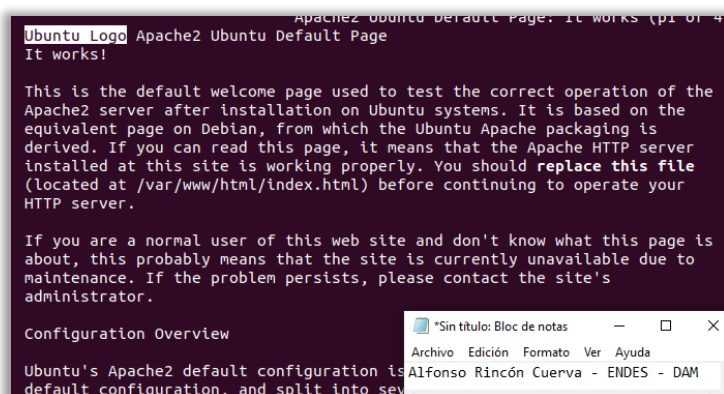
```



Vamos a realizar más comprobaciones de que Apache está funcionando. Iremos a Mozilla Firefox y buscamos **localhost**. En la imagen podemos ver que funciona correctamente.

Para asegurarnos realizamos una segunda comprobación. Dentro de la terminal, pondremos el comando **\$ links http://localhost**. En este caso podemos ver que está funcionando.

Usamos **Ctrl+C** para salir de esta pantalla.



INSTALACIÓN DE MARIADB

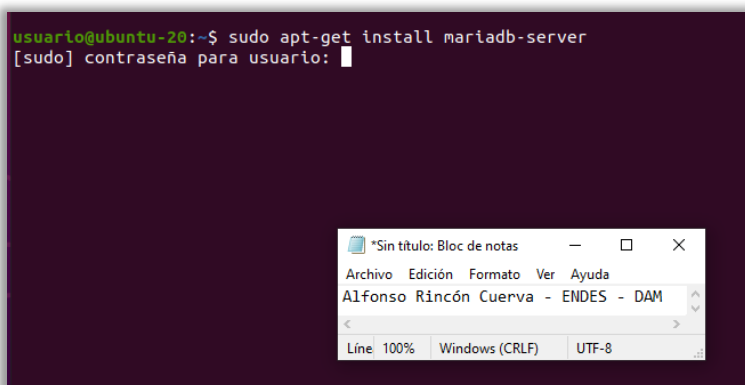
EJERCICIO 8

¿Qué es MariaDB y para qué se utiliza?

Es una base de datos similar a MySQL y de código abierto. Se utiliza para fines como el almacenamiento de datos, comercio electrónico, funciones a nivel empresarial y aplicaciones de registro.

¿Tiene alguna relación con MariaDB? ¿Cuál?

MariaDB es una versión modificada de MySQL, siendo un descendiente binario de este.



Usamos el comando **\$ sudo apt-get install mariadb-server** para instalar MariaDB.

```
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running:
usuario @ session #2: gnome-keyring-d[1536], gnome-session-b[1695,1705,1714,1715],
usuario @ user manager service: at-spi-bus-launcher[1750], gnome-session-b[1770,1771,1772],
gnome-shell[1790], gvfsd[1532], ibus-daemon[1812], systemd[1515]

usuario@ubuntu-20:~$ sudo systemctl enable mariadb
usuario@ubuntu-20:~$
```

Al igual que con Apache, vamos a realizar varias comprobaciones. Empezamos usando el comando **\$ sudo systemctl enable mariadb**.

```
User sessions running outdated binaries:
usuario @ session #2: gnome-keyring-d[1536], gnome-session-b[1695,1705,1714,1715],
usuario @ user manager service: at-spi-bus-launcher[1750], gnome-session-b[1770,1771,1772],
gnome-shell[1790], gvfsd[1532], ibus-daemon[1812], systemd[1515]

usuario@ubuntu-20:~$ sudo systemctl restart mariadb
usuario@ubuntu-20:~$
```

Luego usaremos el **\$ sudo systemctl restart mariadb**.

```
● mariadb.service - MariaDB 10.3.38 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-10-31 17:13:20 UTC; 27s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 57547 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va>
   Process: 57560 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S>
   Process: 57564 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&>
   Process: 57645 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_>
   Process: 57647 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0>
   Main PID: 57613 (mysqld)
   Status: "Taking your SQL requests now"
   Tasks: 31 (limit: 2318)
   Memory: 63.5M
   CGroup: /system.slice/mariadb.service
           └─57613 /usr/sbin/mysqld

oct 31 17:13:20 ubuntu-20 systemd[1]: Starting MariaDB 10.3.38 database server.>
oct 31 17:13:20 ubuntu-20 systemd[1]: Started MariaDB 10.3.38 database server.>
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57649]: Upgrading MySQL table>
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57652]: Looking for 'mysql' a>
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57652]: Looking for 'mysqlche>
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57652]: This installation of >
lines 1-23
```

Y finalmente **\$ sudo systemctl status mariadb**.

EJERCICIO 9

¿Cómo sabemos o dónde podemos ver el estado del servidor de base de datos?

```
mariadb.service - MariaDB 10.3.38 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
Active: active (running) since Tue 2023-10-31 17:13:20 UTC; 27s ago
Docs: man:mysqld(8)
      https://mariadb.com/kb/en/library/systemd/
Process: 57547 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va
Process: 57560 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S
Process: 57564 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&
Process: 57645 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_
Process: 57647 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0
Main PID: 57613 (mysqld)
Status: "Taking your SQL requests now"
Tasks: 31 (limit: 2318)
Memory: 63.5M
CGroup: /system.slice/mariadb.service
└─57613 /usr/sbin/mysqld

oct 31 17:13:20 ubuntu-20 systemd[1]: Starting MariaDB 10.3.38 database server.
oct 31 17:13:20 ubuntu-20 systemd[1]: Started MariaDB 10.3.38 database server.
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57649]: Upgrading MySQL table
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57652]: Looking for 'mysql' a
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57652]: Looking for 'mysqlche
oct 31 17:13:20 ubuntu-20 /etc/mysql/debian-start[57652]: This installation of
lines 1-23
```

¿Tiene alguna relación con MariaDB? ¿Cuál?

Usaremos el comando **\$ sudo systemctl stop mariadb** y luego **\$ sudo systemctl status mariadb** para ver su estado.

```
usuario@ubuntu-20:~$ sudo systemctl stop mariadb
usuario@ubuntu-20:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.3.38 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
Active: inactive (dead) since Fri 2023-11-10 00:09:47 UTC; 17s ago
Docs: man:mysqld(8)
      https://mariadb.com/kb/en/library/systemd/
Process: 65769 ExecStart=/usr/sbin/mysqld $MYSQLD_OPTS $WSREP_NEW_CLUSTER
Main PID: 65769 (code=exited, status=0/SUCCESS)
Status: "MariaDB server is down"

oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65804]: Upgrading MySQL table
oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65807]: Looking for 'mysql' a
oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65807]: Looking for 'mysqlche
oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65807]: This installation of
oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65807]: This installation of
oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65807]: This installation of
oct 31 18:01:04 ubuntu-20 /etc/mysql/debian-start[65807]: This installation of
nov 10 00:09:45 ubuntu-20 systemd[1]: Stopped MariaDB 10.3.38 database server.
nov 10 00:09:47 ubuntu-20 systemd[1]: Stopped MariaDB 10.3.38 database server.
```


Vuelve a iniciar el servicio

Usaremos el comando **\$ sudo systemctl start mariadb** y luego **\$ sudo systemctl status mariadb** para ver su estado.

```

usuario@ubuntu-20:~$ sudo systemctl start mariadb
usuario@ubuntu-20:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.3.38 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Fri 2023-11-10 00:12:14 UTC; 9s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 68417 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va
   Process: 68431 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S
   Process: 68434 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&
   Process: 68513 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP
   Process: 68515 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0
   Main PID: 68482 (mysqld)
   Status: "Taking your SQL requests now..."
   Tasks: 31 (limit: 2318)
   Memory: 82.4M
   CGroup: /system.slice/mariadb.serv
           └─68482 /usr/sbin/mysqld

```

SECURIZACIÓN DE MARIADB

```

usuario@ubuntu-20:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

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

Setting the root password ensures that
root user without the proper authorisa

You already have a root password set,
Change the root password? [Y/n]

```

Debido a que MariaDB no viene asegurada, usaremos un script que esta trae para poder hacerlo. Usaremos el comando **\$ sudo mysql_secure_installation**.

Posteriormente, tras escribir este comando, nos pedirá definir unos ajustes. Para ello, en las opciones que nos darán tendremos que escribir lo siguiente, siguiendo este orden: *n*, *y*, *y*, *y*, *y*.

```

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...
... Success!
- Removing privileges on test database...
... Success!

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
installation should now be secure.

```

EJERCICIO 10

Ejecución y configuración del script de MariaDB paso a paso

```

usuario@ubuntu-20:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE!  PLEASE READ THIS DOCUMENTATION BEFORE
RUNNING THIS SCRIPT IN A PRODUCTION SYSTEM.

In order to log into MariaDB use password (blank) or root password

Change the root password? [Y/n] n

Remove anonymous users? [Y/n] y
... Success!

Disallow root login remotely? [Y/n] y
... Success!

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!

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

Conforme vayas ejecutando el script, fíjate que te explica qué significa cada ítem de configuración. Explícalo con tus palabras en tu documento.

Change the root password?: sirve para cambiar la contraseña.

Remove anonymous users?: para asegurarse de que nadie entra desde otro usuario de forma anónima.

Disallow root login remotely?: restringe al acceso remoto del usuario “root” a la base de datos.

Remove test database and access to it?: elimina la base de datos de prueba y el acceso a ella.

Reload privilege tables now?: recarga o actualiza las tablas de privilegios en un sistema de gestión de datos.

```

usuario@ubuntu-20:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.3.38-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Abstr
Type 'help;' or '\h' for help. Type '\c' to clear the current
  
```

Usamos el comando `$ sudo mysql -u root -p` para iniciar sesión en MariaDB. Una vez usado, saldremos del menú usando “exit”.

INSTALACIÓN PHP

```
usuario@ubuntu-20:~$ sudo apt-get install php php-mysql
```

Debido a que Apache no soporta PHP, debemos instalarlo, al igual que un módulo para conectarse con MariaDB.

Esta será la salida que recibiremos al escribir este comando. Vamos a reiniciar los servidores por si acaso.

```
usuario@ubuntu-20:~$ sudo apt-get install php php-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
libfprint-2-tod1
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common php7.4-json
php7.4-mysql php7.4-opcache php7.4-readline
Paquetes sugeridos:
php-pear
Se instalarán los siguientes paquetes NUEVOS:
libapache2-mod-php7.4 php-common php-mysql php7.4 php7.4-cli
php7.4-common php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
0 actualizados, 11 nuevos se instalarán, 0 a eliminar.
Se necesita descargar 4.154 kB de datos y se utilizarán 18,5 MB de espacio.
¿Desea continuar? [S/n]
```

EJERCICIO 11

Reinicia los servidores de Apache y MariaDB como vimos anteriormente.

Usaremos los comandos **\$ sudo systemctl restart apache2** y **\$ sudo systemctl restart mariadb**.

```
usuario@ubuntu-20:~$ sudo systemctl restart apache2
usuario@ubuntu-20:~$ sudo systemctl restart mariadb
```

Vamos a comprobar que PHP funciona. Usaremos el siguiente comando: **\$ sudo nano /var/www/html/info.php**.

```
usuario@ubuntu-20:~$ sudo nano /var/www/html/info.php
```

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

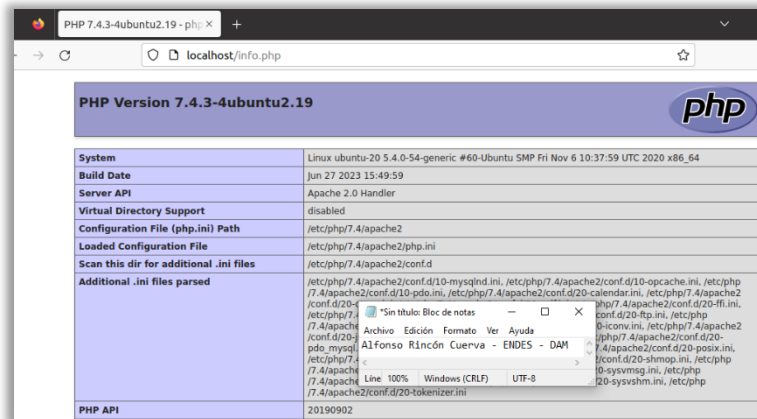
Nos dará la opción de poner texto. Pondremos lo siguiente:

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

Tras poner el siguiente texto, debemos realizar la siguiente combinación de teclas:

1. **Ctrl + X**
2. **S**
3. **Enter**

Finalmente, para ver el resultado, vamos a Mozilla Firefox y escribir <http://localhost/info.php>.



ACTIVIDADES ADICIONALES

EJERCICIO OPCIONAL 1

Crear una base de datos con MariaDB

1. Abrimos el emulador de MariaDB con el comando **\$ sudo mysql -u root -p**
2. Crearemos una base de datos. En este caso la llamaremos *ejemplo_clase*. Para ello escribiremos lo siguiente: **CREATE DATABASE ejemplo_clase;**
3. Para mostrar la base de datos creador, usaremos: **SHOW DATABASES.**
4. Salimos del programa usando **exit;**

```

usuario@ubuntu-20:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.3.38-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

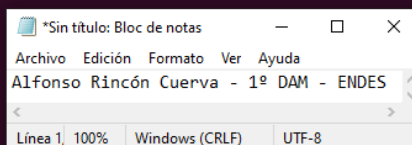
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> CREATE DATABASE ejemplo_clase;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| base_ejemplo |
| ejemplo_clase |
| information_schema |
| mysql |
| nueva_basedatos |
| performance_schema |
+-----+

```



Crear una página web en HTML.

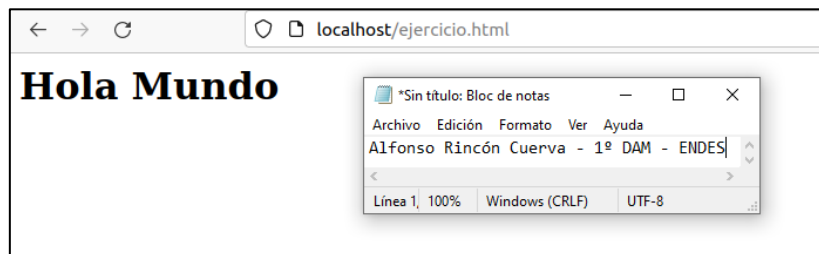
1. Escribimos **\$ sudo nano /var/www/html/ejercicio.html**, para dar permisos para poder crear el HTML usando el comando sudo, y para darle el nombre que queramos al archivo.
2. Escribimos el código de HTML.
3. Una vez escrito, pulsamos **Ctrl + O** y luego **Enter** para poder guardar el HTML que hemos escrito.
4. Abrimos el Mozilla Firefox, y escribimos **http://localhost/ejercicio.html**, para poder ver el HTML que hemos escrito.

```

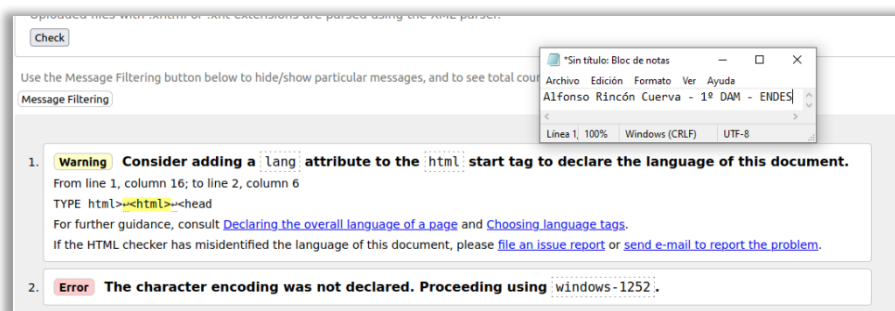
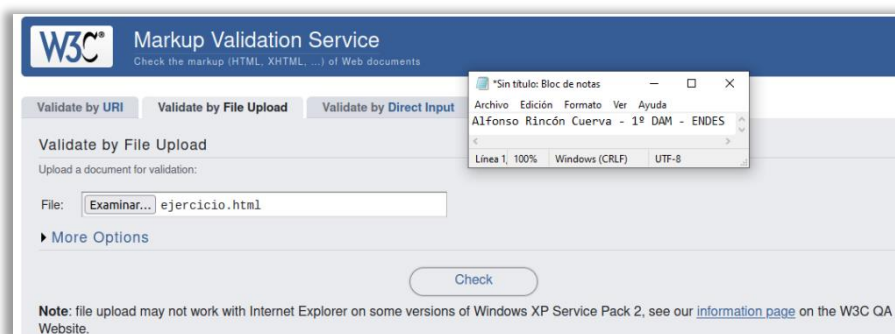
usuario@ubuntu-20:~$ nano /var/www/html/ejercicio.html
usuario@ubuntu-20:~$ sudo nano /var/www/html/ejercicio.html

```

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejercicio de clase</title>
</head>
<body>
  <h1>Hola Mundo</h1>
</body>
```



Ahora vamos a verificar que el archivo está escrito correctamente. Iremos al verificador de HTML de W3C, e insertaremos el archivo, desde su carpeta var.



EJERCICIO OPCIONAL 2

Crear una tabla dentro de la base de datos creada en el ejercicio anterior

Esta tabla se llamará estudiantes, y tendrá un id, nombre y edad. Para ello, hacemos lo siguiente:

1. Escribimos **\$ sudo mysql -u root -p** en la terminal para poder abrir el emulador de MariaDB.
2. Una vez abierto, vamos a importar la base de datos creada anteriormente. Para ello escribimos **USE ejemplo_clase;**
3. Ahora es el momento de crear la tabla. Tenemos que escribir lo siguiente:

```
CREATE TABLE estudiantes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(255),
  edad INT
);
```

El id será PK. Tras escribirlo, pulsamos Enter para guardarlo.

4. Por último, para poder ver el resultado, escribimos **DESCRIBE estudiantes;**

```
MariaDB [ejemplo_clase]> USE ejemplo_clase;
Database changed
MariaDB [ejemplo_clase]> CREATE TABLE estudiantes (
  -> id INT AUTO_INCREMENT PRIMARY KEY,
  -> nombre VARCHAR(255),
  -> edad INT
  -> );
Query OK, 0 rows affected (0,024 sec)

MariaDB [ejemplo_clase]> DESCRIBE estudiantes;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    | auto increment |
| nombre | varchar(255)  | YES  |     | NULL    |                |
| edad  | int(11)       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,003 sec)
```

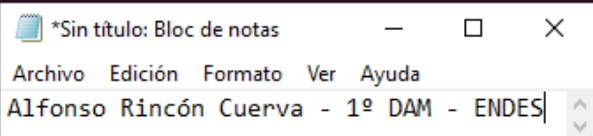
Crear una página web en HTML.

Vamos a crear otro HTML, pero este tendrá integrado PHP dentro suyo. Haremos lo siguiente:

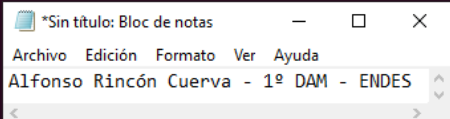
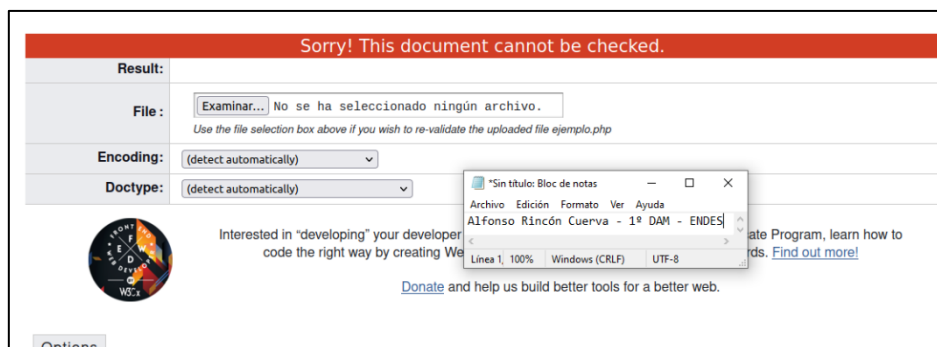
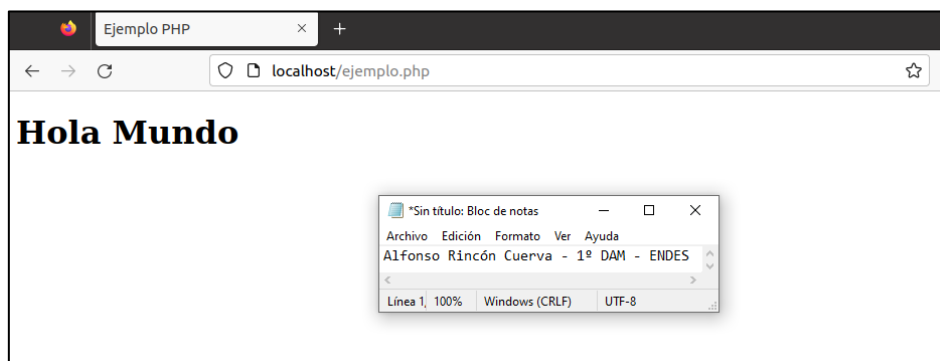
1. Escribimos el comando **\$ sudo nano /var/www/html/ejemplo.php**, usando el sudo para poder darle permisos para crear el HTML.
2. Escribimos el código de HTML que queremos, metiendo PHP dentro de un **<h1>**.

- Guardamos con **Ctrl+O** y **Enter**, y salimos del editor con **Ctrl+X**.
- Abrimos el archivo escribiendo lo siguiente en el Mozilla: **localhost/ejemplo.php**, Podemos ver que funciona correctamente.
- Al intentar verificarlo en W3C podemos ver que no es posible, ya que este no admite archivos de tipo .php.

```
usuario@ubuntu-20:~$ sudo nano /var/www/html/ejemplo.php
usuario@ubuntu-20:~$
```



```
<!DOCTYPE html>
<html>
<head>
  <title>Ejercicio con PHP</title>
</head>
<body>
  <h1><?php echo "Hola Mundo"; ?></h1>
</body>
</html>
```

BIBLIOGRAFÍA

EJERCICIO 1:

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-es>

EJERCICIO 3:

<https://www.softzone.es/linux/tutoriales/terminal-linux/>

EJERCICIO 5:

<https://www.arsys.es/blog/que-es-apache-y-para-que-sirve>

EJERCICIO 6:

[munity/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units-es](https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units-es)

EJERCICIO 7:

<https://www.one.com/es/hosting/que-es-mariadb>

EJERCICIOS OPCIONALES:

https://www.hostinger.es/tutoriales/linux-comandos#1_Comando_sudo

<https://atareao.es/como/trabajar-con-mariadb-desde-el-terminal/>

<https://www.compuhoy.com/como-uso-html-en-ubuntu/>