Se trata de que implementes un servicio web en Linux. Práctica obligatoria. Indicaciones generales:

**Contenido práctica:**

1. La arquitectura Web es un modelo compuesto de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?

**1. Capa de presentación (parte en el cliente y parte en el servidor)**

* Recoge la información del usuario y la envía al servidor (cliente)
* Manda información a la capa de proceso para su procesado
* Recibe los resultados de la capa de proceso
* Generan la presentación
* Visualizan la presentación al usuario (cliente)

**2.** **Capa de proceso (servidor web)**

* Recibe la entrada de datos de la capa de presentación
* Interactúa con la capa de datos para realizar operaciones
* Manda los resultados procesados a la capa de presentación

**3. Capa de datos (servidor de datos)**

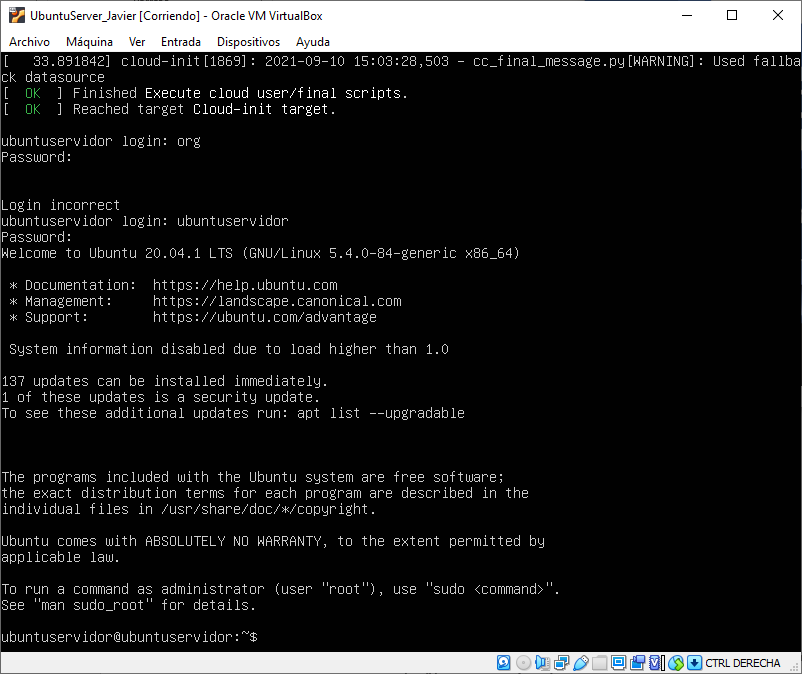
* Almacena los datos
* Recupera datos
* Mantiene los datos
* segura la integridad de los datos

1. Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web; destacan dos plataformas web, LAMP y WISA. Explica en qué consiste cada una de ellas.

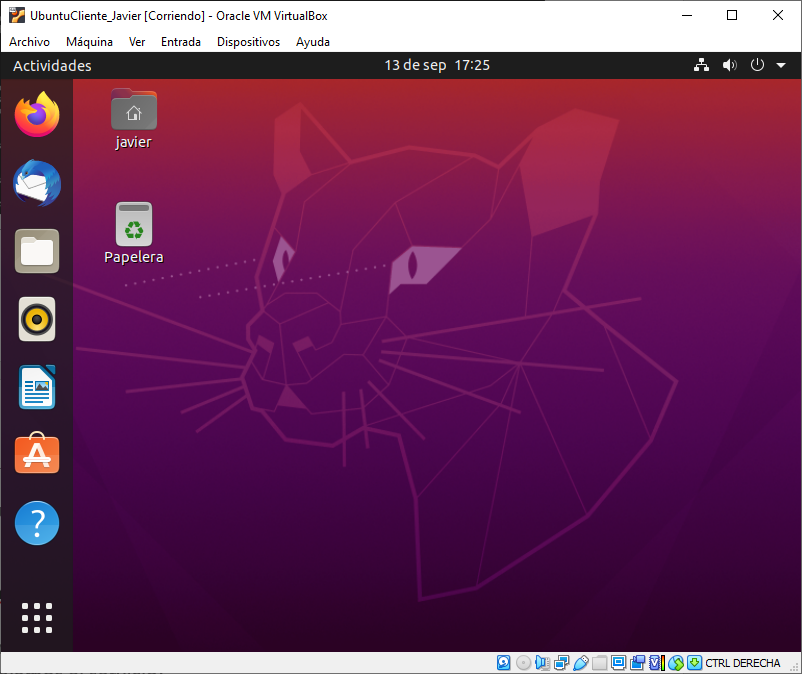
LAMP: El software LAMP es una plataforma de código abierto que utiliza Linux como sistema operativo, Apache como servidor web, PHP como lenguaje de script orientado a objetos y MySQL como sistema de gestión de bases de datos relacionales.

WISA: La plataforma WISA está basada en tecnologías desarrolladas por la compañía Microsoft. Sus componentes son Windows (sistema operativo), Internet Information Services (servidor web), SQL Server (manejador de bases de datos) y ASP (no es un lenguaje interpretado , sino que sirve como un medio para trabajar con otro que sí lo es, como Visual Basic, bajo un entorno web).

1. Tenemos este escenario inicial:
   1. Trabajaremos con un Ubuntu y un cliente Ubuntu (reutiliza los que ya tenías y adjunta captura).



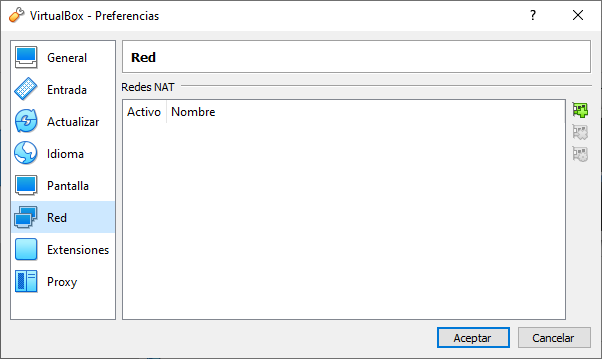
Ubuntu server



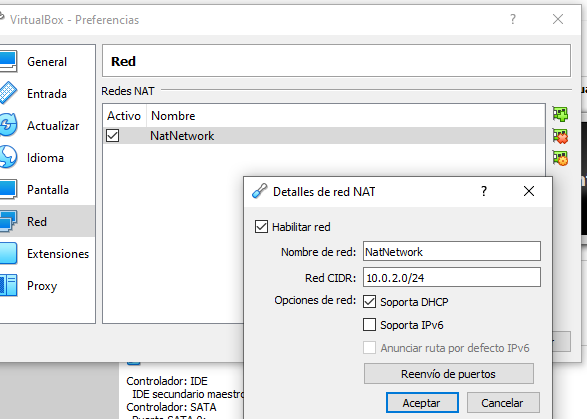
Ubuntu Cliente

* 1. Las dos VM están conectadas entre sí mediante una red común (adjunta captura).

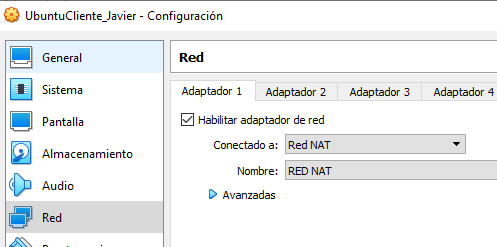
Lo primero creamos la red que vayamos a usar, para ello iremos a: archivo>Preferencias>Red y se nos abrirá esta pantalla:



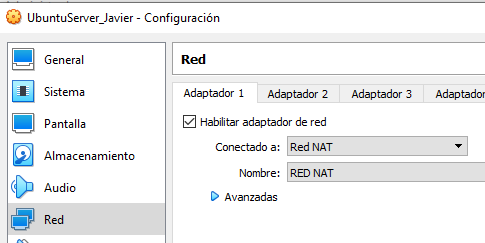
Ahora le daremos doble click a la red creada y se nos abrirá la configuración.



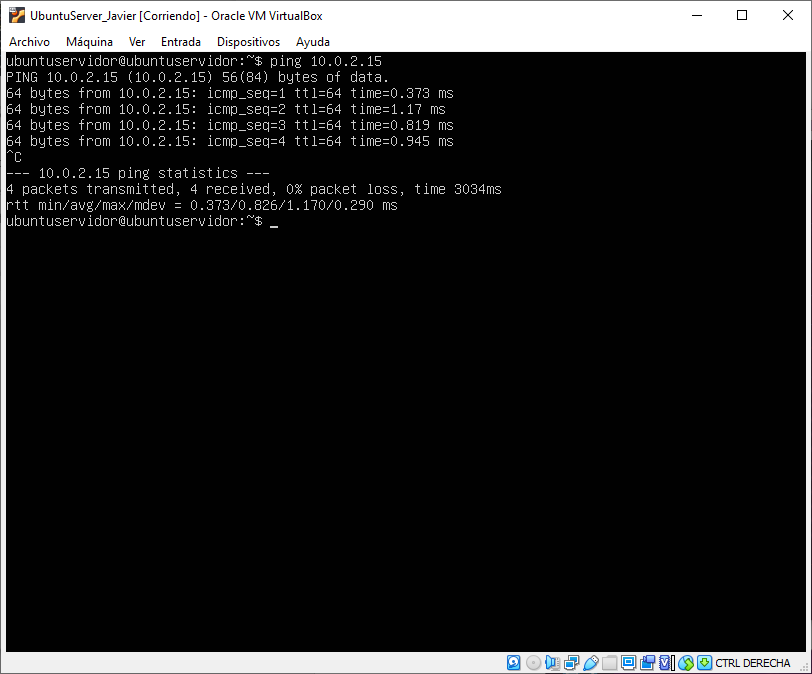
Ahora debemos poner las maquinas en RED NAT:

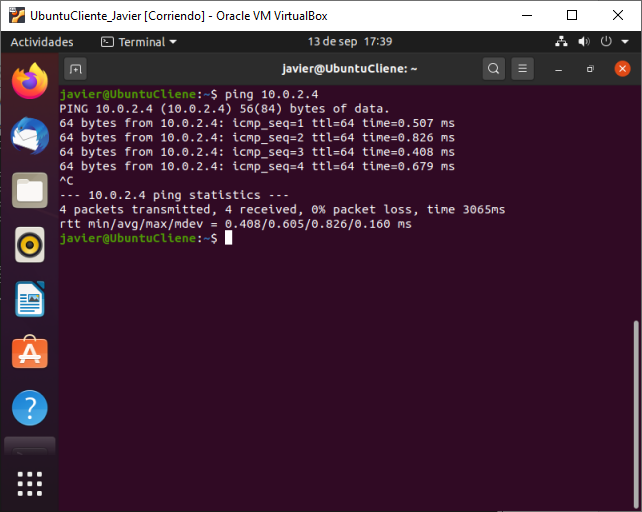


Cliente

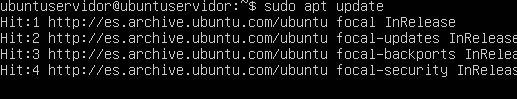
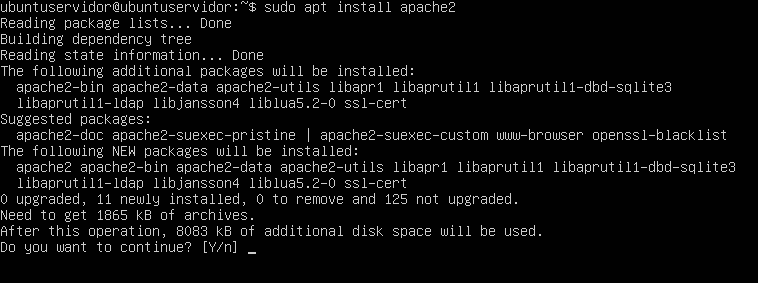


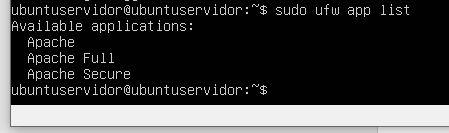
Servidor





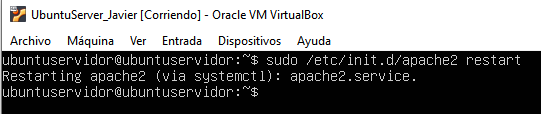
Se puede ver que los pings funcionan

1. Instalación de Apache:
   1. Instalamos el servicio:
      1. sudo apt update
      2. sudo apt install apache2
   2. Apache se registra a sí mismo como un servicio con ufw (Uncomplicated Firewall) al instalarse, haciendo que permitir el acceso de Nginx sea fácil.
      1. sudo ufw app list -> muestra el listado de perfiles de aplicaciones configuradas para ufw. Comprueba que hay tres disponibles para Apache:
         1. Apache Full: Este perfil abre tanto el puerto 80 (tráfico web normal, no cifrado) como el puerto 443 (tráfico TLS/SSL cifrado)
         2. Apache HTTP: Este perfil solamente abre el puerto 80 (tráfico web normal, no cifrado)
         3. Apache HTTPS: Este perfil solamente abre el puerto 443 (tráfico TLS/SSL cifrado)

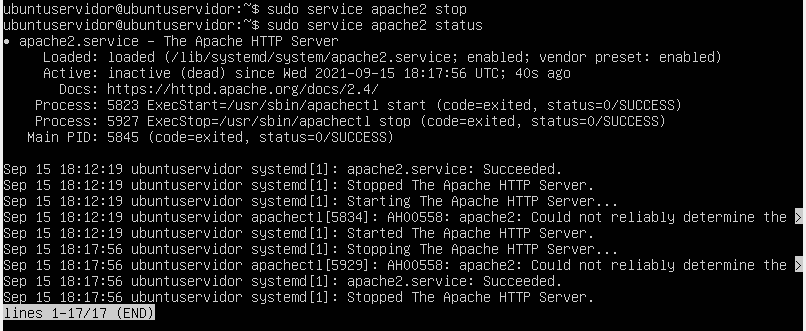


* 1. Para probar el servicio de Apache verifica la funcionalidad de las combinaciones pertinentes:
     1. sudo service apache2 status/start/stop/restart-> inicia, para, reinicia el servicio apache. Prueba que lo hacen (adjunta captura con cada combinación).

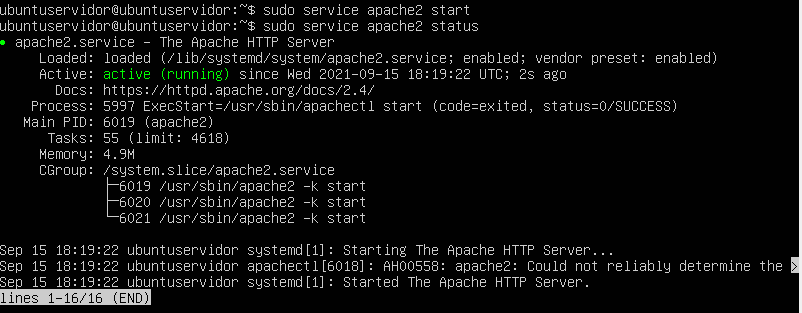
Restart



Stop



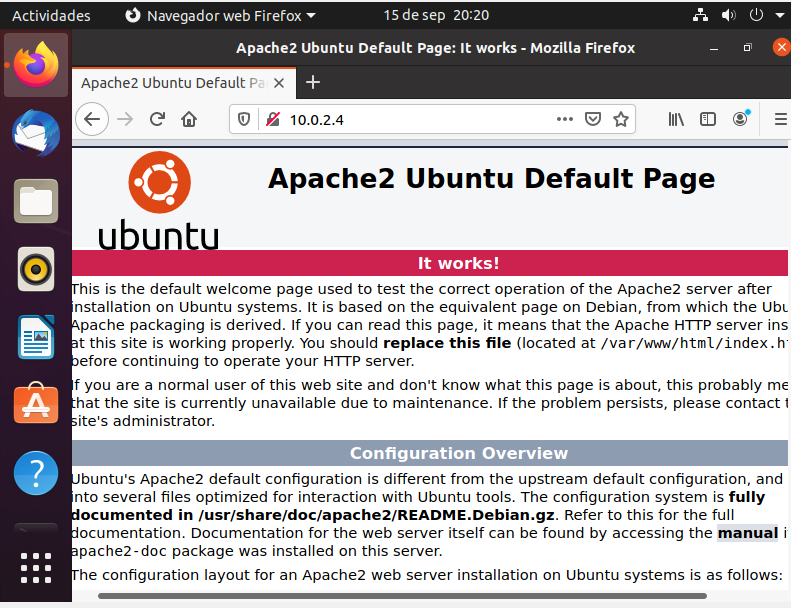
Start



* + 1. sudo service apache2 reload -> Si simplemente está haciendo cambios de configuración, a menudo Apache se puede recargar sin perder las conexiones. Pruébalo.



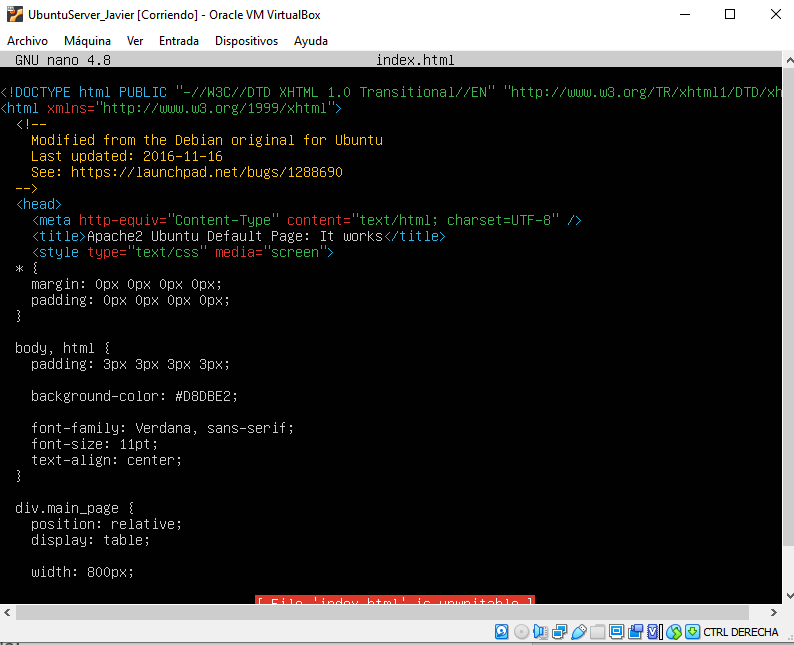
* 1. Desde el navegador web del cliente probar que Apache funciona (IP\_SERVER:80). Aparecerá una página html por defecto.



Captura desde el cliente

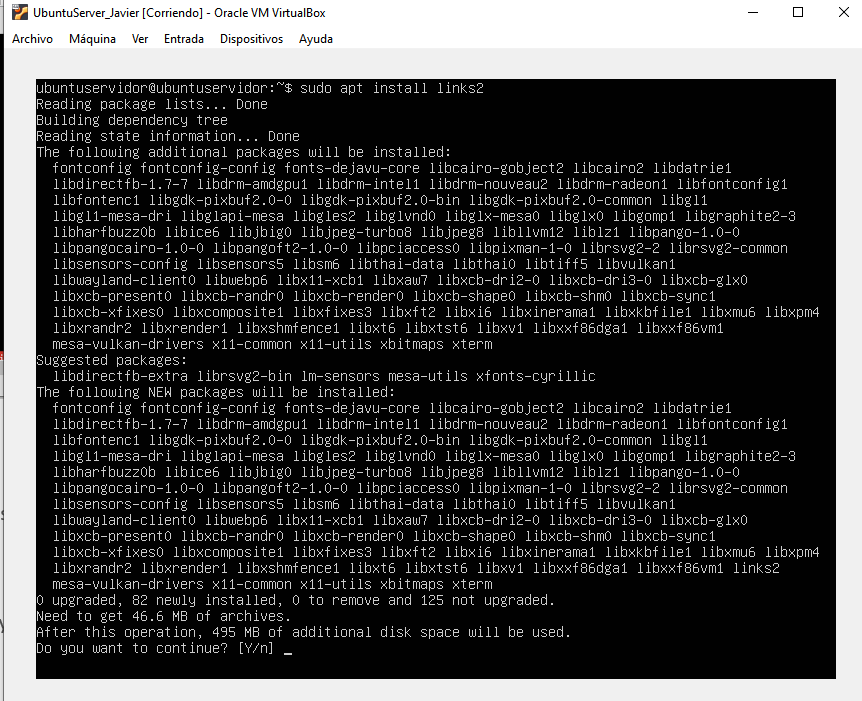
* 1. Verifica que lo que ves coincide con el contenido del archivo

/var/www/html/index.html



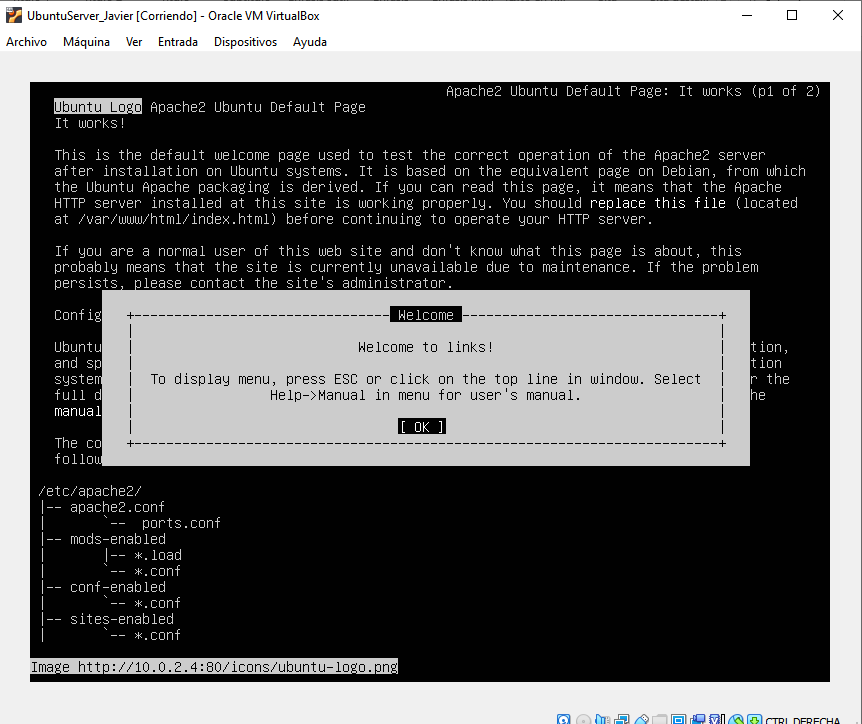
Archivo html en el servidor

1. Navegación web por consola:
   1. Aunque parezca que no, también se puede navegar por bash. Instala el paquete links2.



Instalación de links2

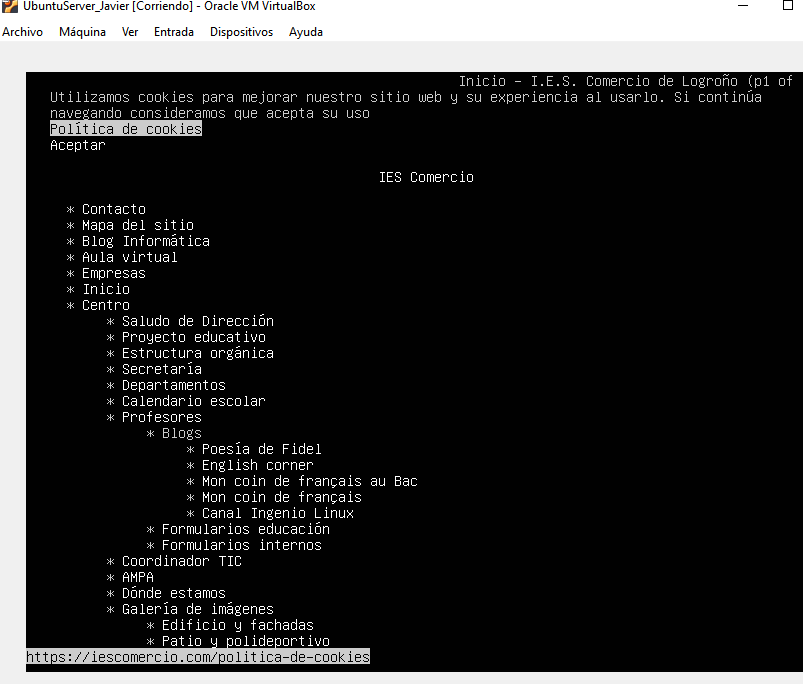
* 1. Ejecuta “links2 IP\_SERVER:80“.

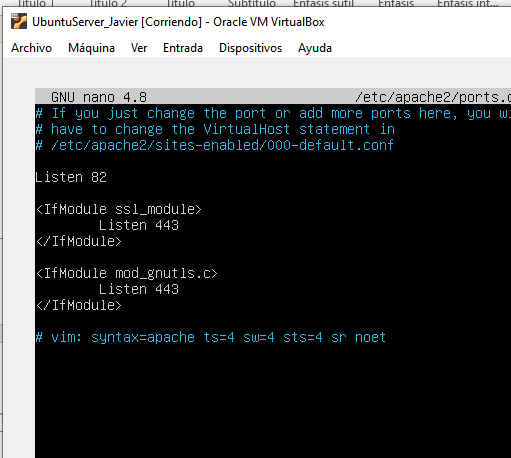


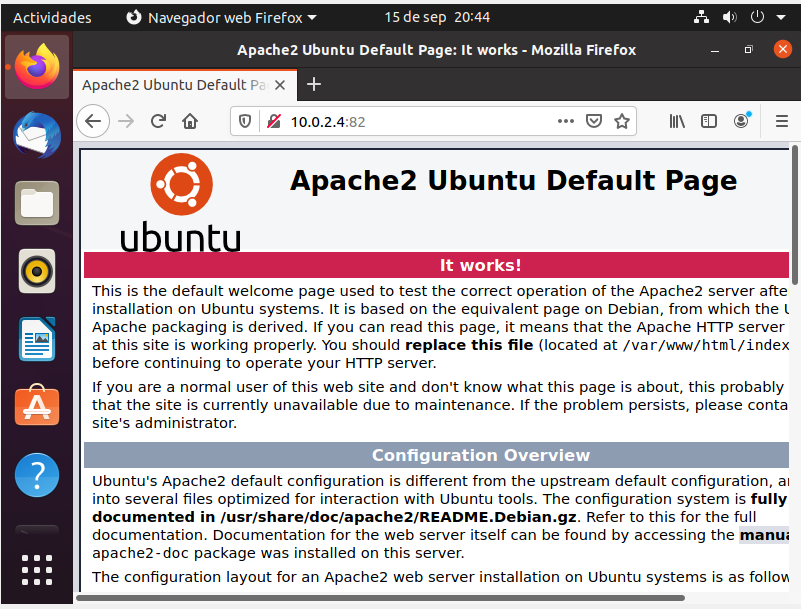
* 1. Escribe “q” y sal de la navegación.



* 1. Ejecuta “links2 iescomercio.com” y navega por distintas secciones.

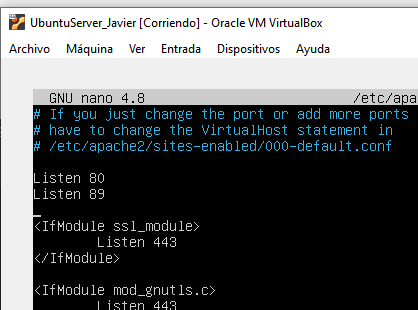
0.

1. Para cambiar de puerto:
   1. Acude a ports.conf y edítalo.
   2. Cambia listen 80 por <nuevopuerto>
   3. Reiniciar servidor y verifica que atiende por el nuevo puerto, basta lanzar una nueva petición vía brower del cliente.

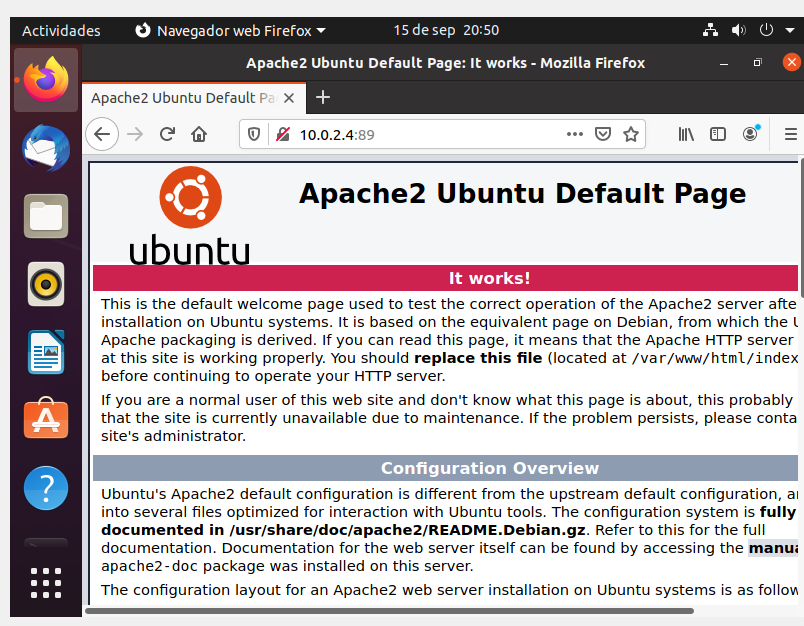


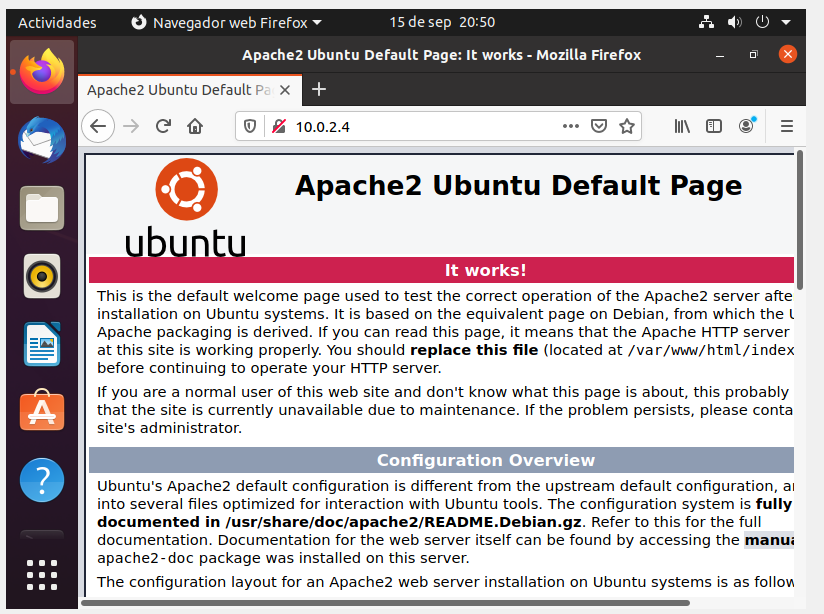
La petición lanzada al puerto 82

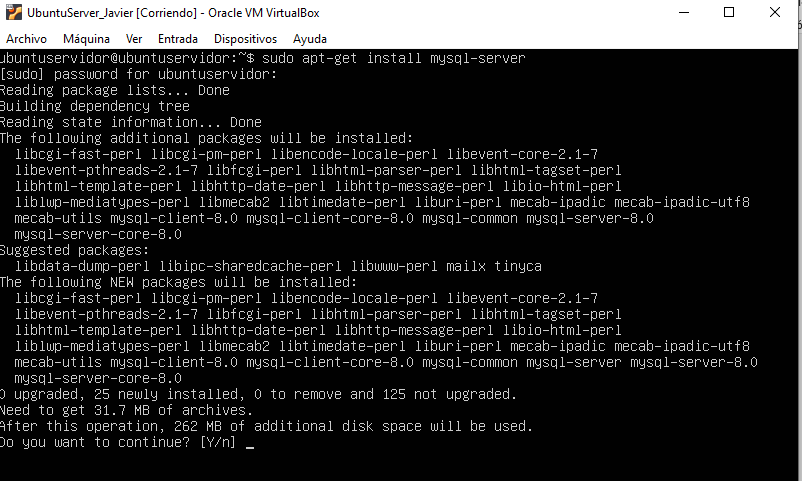
* 1. Agrega un segundo puerto debajo del primero sólo para la IP de tu cliente Ubuntu y verifica que está activo el servicio en ambos puertos. Es decir, que responde la página web principal por cada puerto, pero el segundo sólo para la IP del cliente.

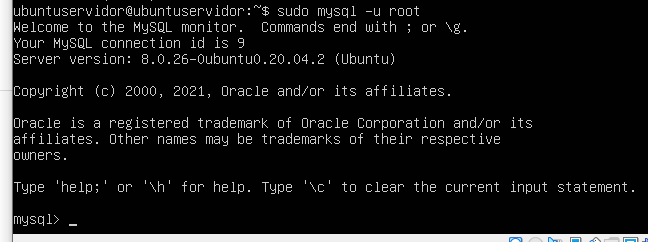


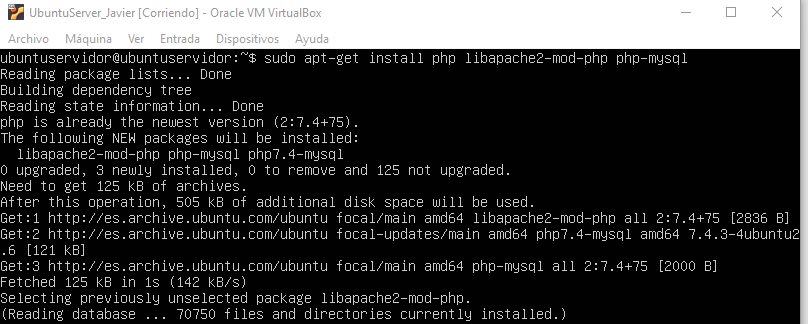
Se añade un puerto (he devuelto al 80 el inicial)





1. Instalación de MySQL:
   1. apt-get install mysql-server
   2. Verifica que puedes administrar bases de datos mediante mysql – uroot



1. Instalación de PHP:
   1. apt-get install php libapache2-mod-php php-mysql
   2. Agrega al final de /etc/apache2/apache2.conf:

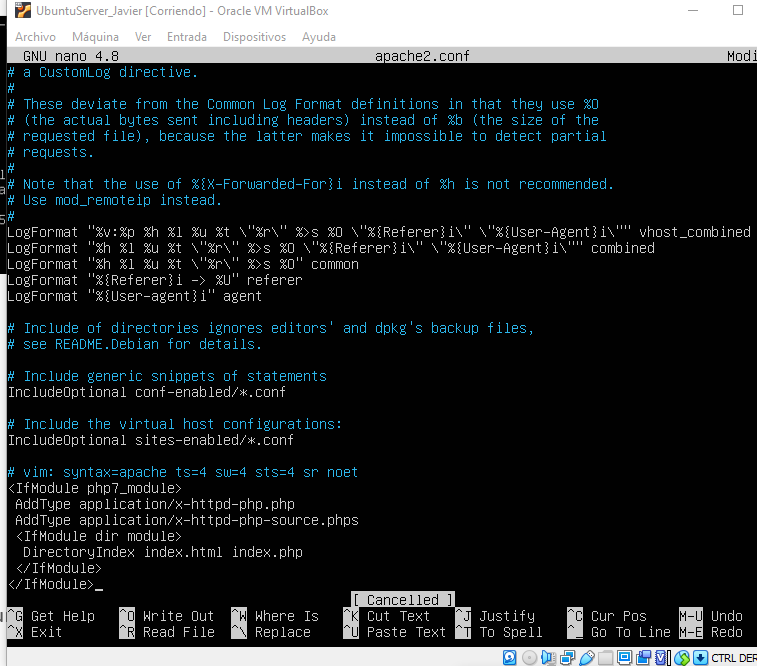
<IfModule php7\_module>

AddType application/x-httpd-php .php AddType application/x-httpd-php-source .phps

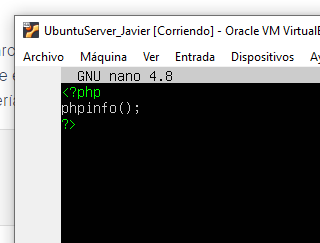
<IfModule dir\_module> DirectoryIndex index.html index.php

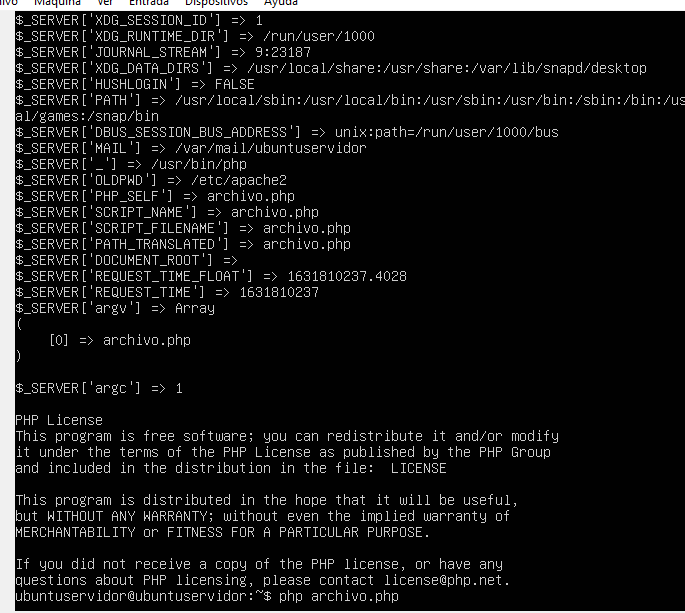
</IfModule>

</IfModule>

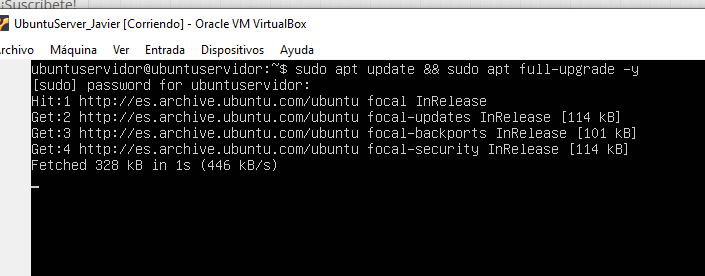
Y reinicia el servicio.

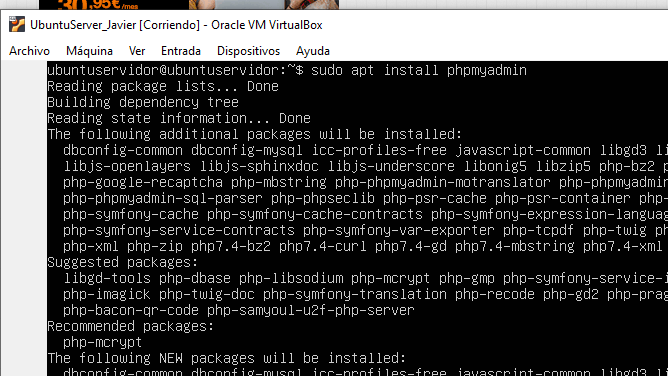
* 1. Crea un archivo PHP que devuelva toda la configuración de PHP y ejecútalo vía Shell y vía navegador web.

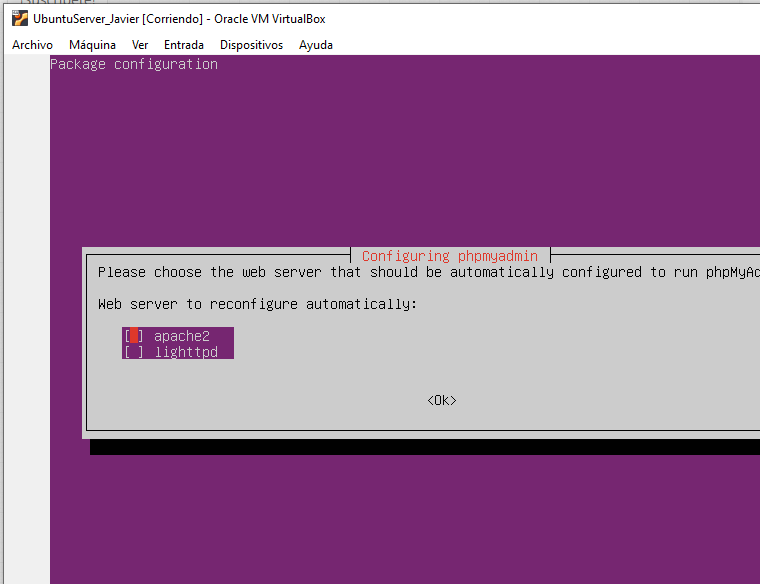


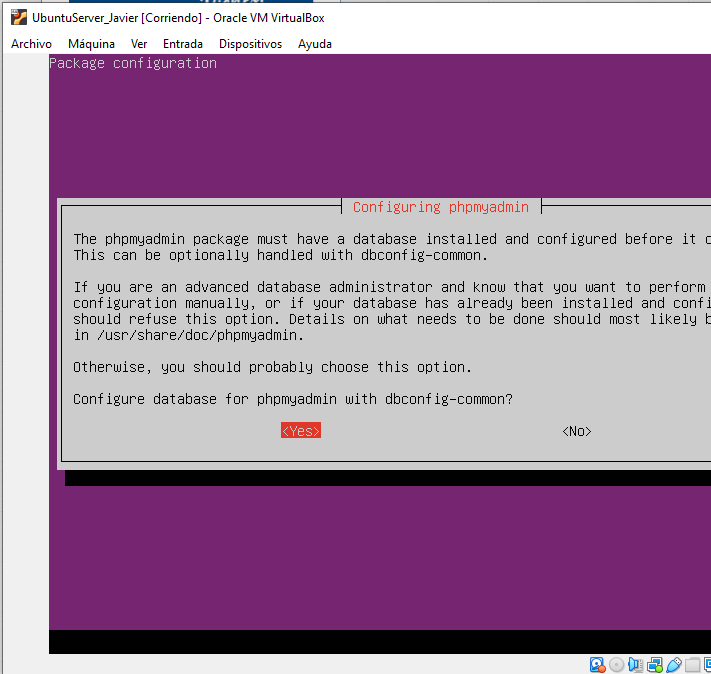


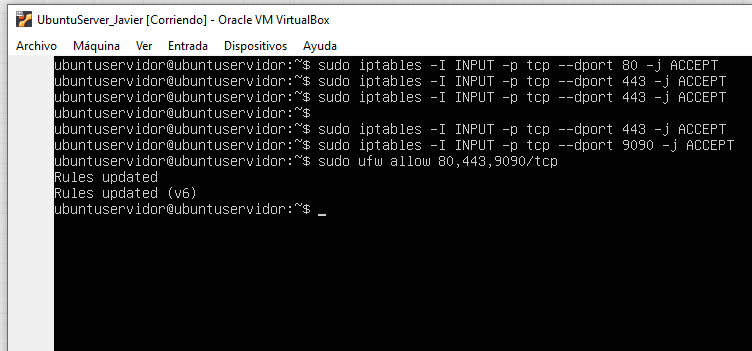
1. Instala PHPMYADMIN:
   1. Sigue los pasos de https://[www.sololinux.es/instalar-phpmyadmin-](http://www.sololinux.es/instalar-phpmyadmin-) en-ubuntu-20-04-focal-lamp/







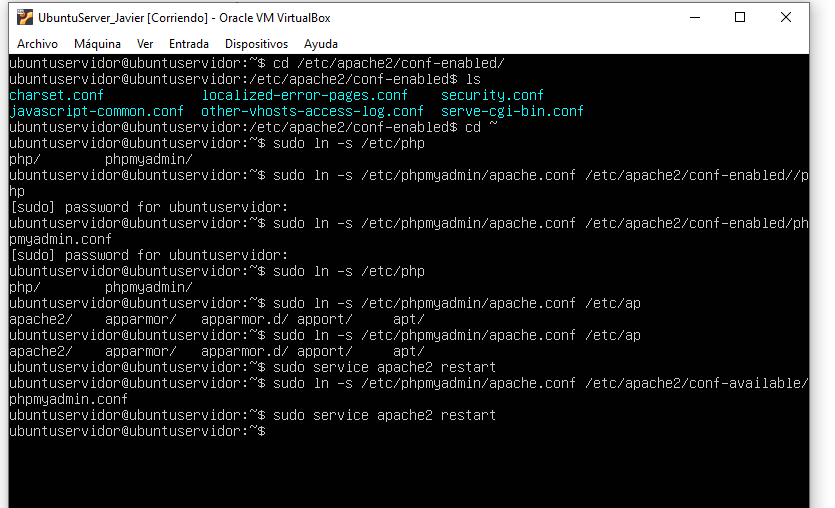




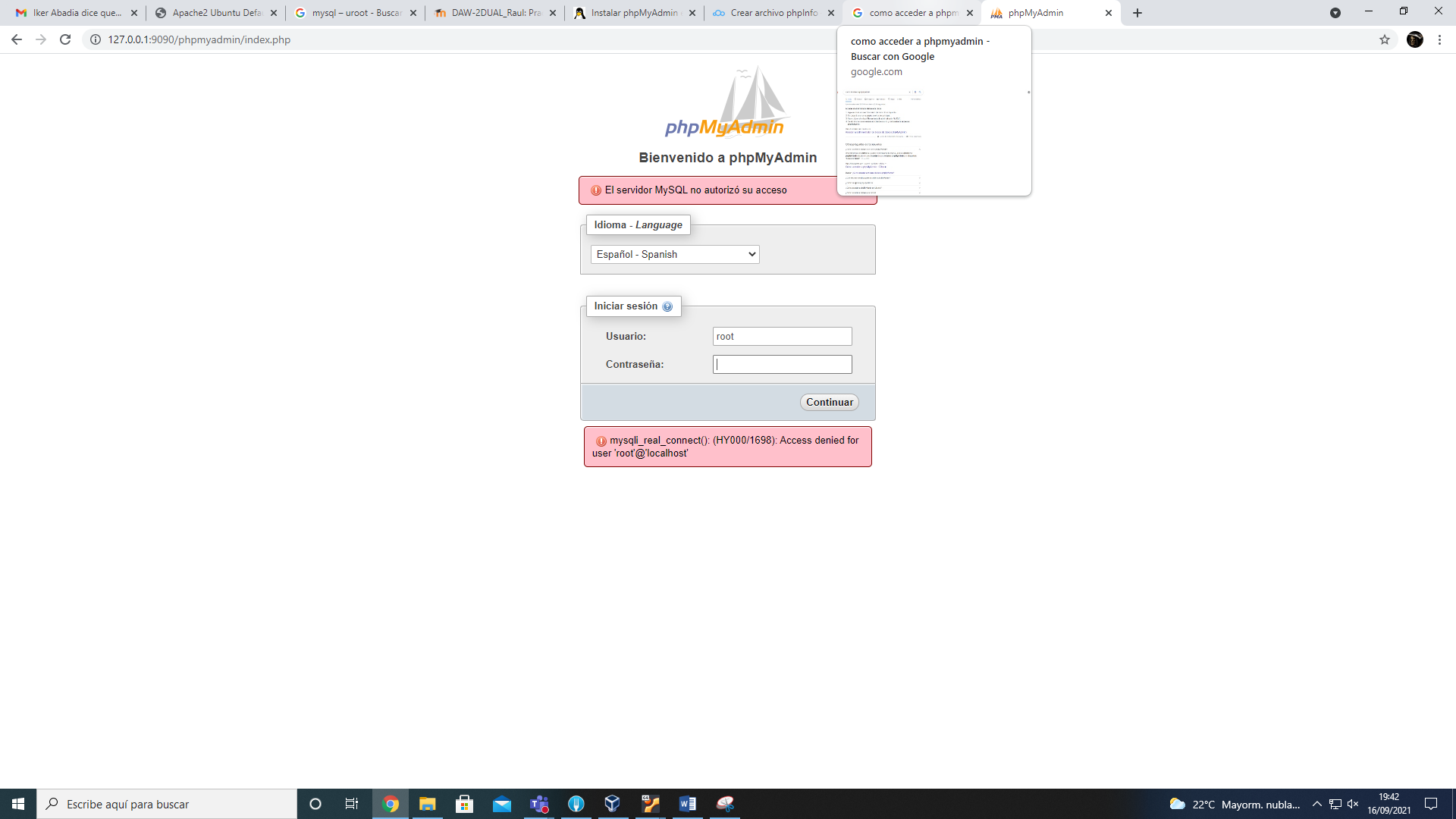
* 1. Probablemente tengas el mismo problema que el usuario SinE del post final. Haz caso a la sugerencia del autor con esa salvedad:

sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/**conf- enabled**/phpmyadmin.conf. + reinicio del apache

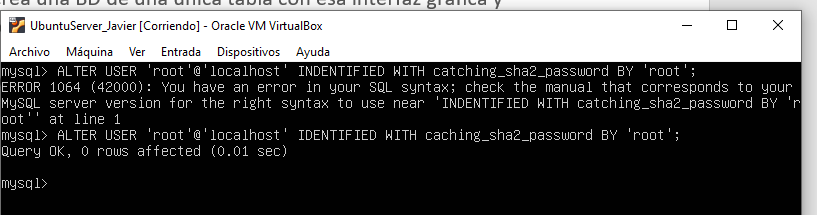
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf- available/phpmyadmin.conf + reinicio del apache

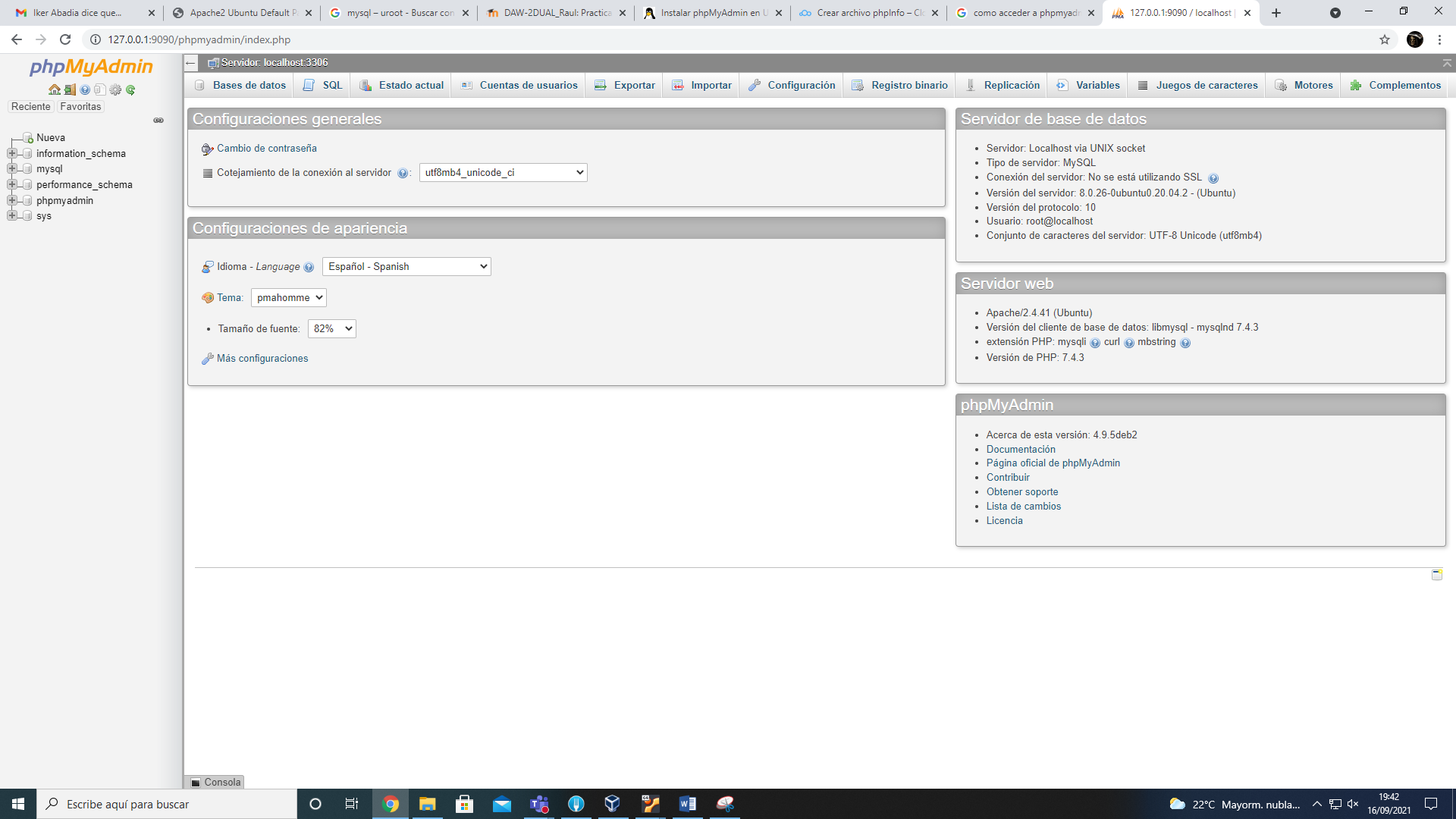


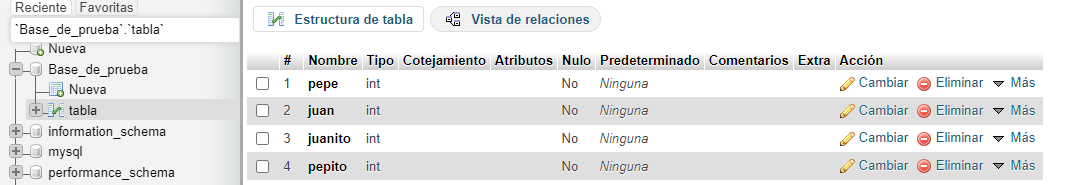
* 1. Verifica que se visiona PhpMyAdmin vía el navegador del cliente.



* 1. Observarás que no puedes autenticarte. Esto es porque MySQL no tiene password asociado a root. Para ello, entra en MySQL y ejecuta:
     1. ALTER USER 'root'@'localhost' IDENTIFIED WITH caching\_sha2\_password BY <nuevopassword>;

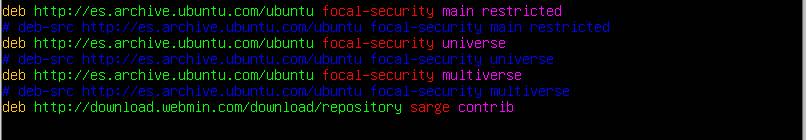


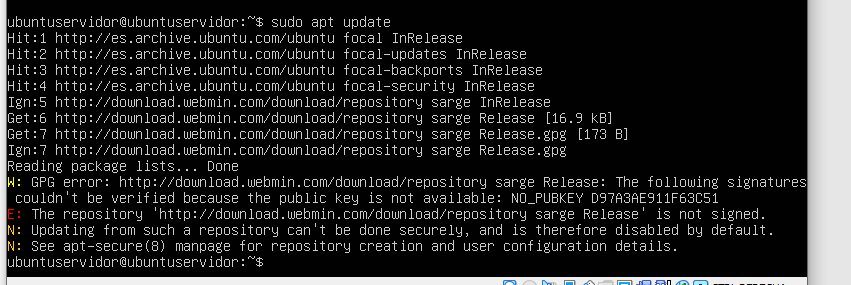
* 1. Verifica que ahora ya sí puedes entrar.
  2. Crea una BD de una única tabla con esa interfaz gráfica y comprueba que en consola (en servidor) aparece también.

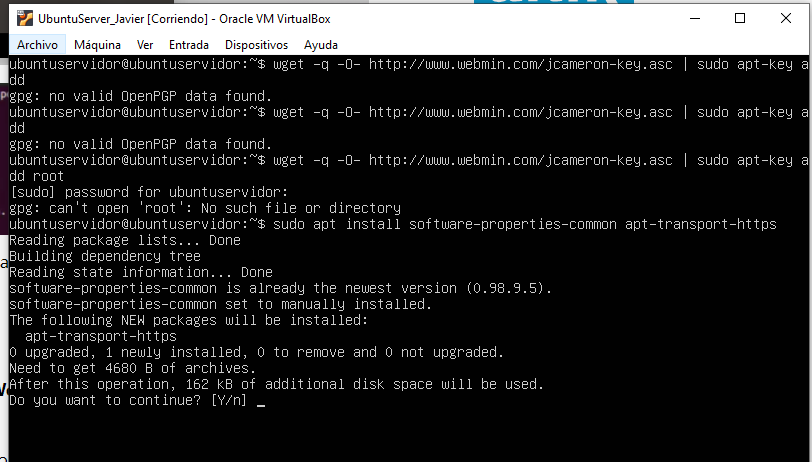




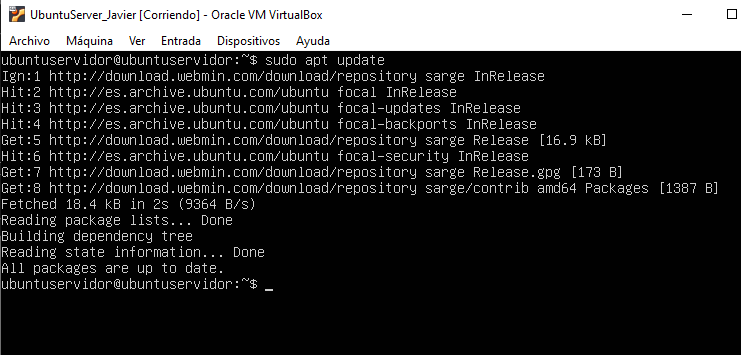
1. Instala WEBMIN:
   1. Edita /etc/apt/sources.list
   2. Agrega deb <http://download.webmin.com/download/repository> sarge contrib (nuevo repositorio) al final del archivo y guárdalo.

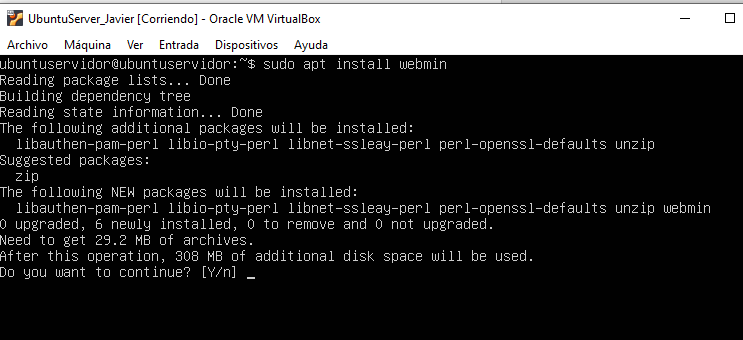


* 1. Sudo apt update
  2. wget -q -O- <http://www.webmin.com/jcameron-key.asc>| sudo apt-key add (añade clave de webmin y las añade al sistema)

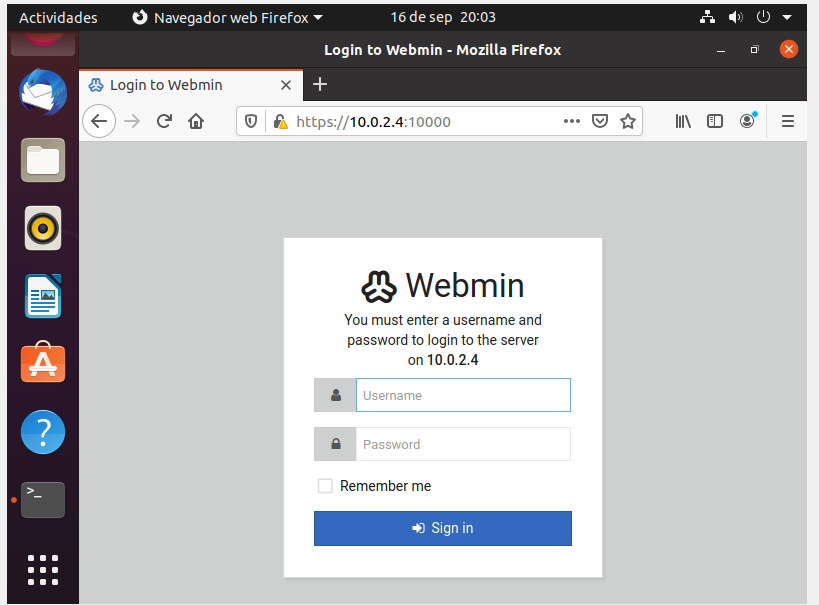


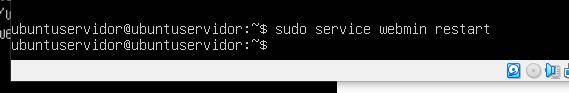
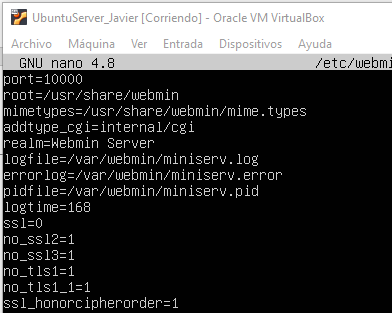


* 1. apt update
  2. apt install webmin



* 1. Verifica que si entras a https://IP\_SERVER:10000 tienes problemas de certificado seguro.



* 1. Momentáneamente deshabilitaremos la autenticación SSL modificando en /etc/webmin/miniserv.conf el parámetro ssl = 1 (ponlo a 0 y reinicia el servicio). 
  2. Verficia que, ahora sí, accedes vía navegador a webmin con un usuario administrador del sistema.

