



INSTITUTO POLITÉCNICO
NACIONAL



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS

Manual de instalación y configuración LAMP

Equipo 7

Integrantes del equipo:

Álvarez Santiago Daniela
Gonzalez Hernandez Diego
Martinez Vivaldo Uriel Giovanni
Rojas Solís Juan Carlos
Sanchez Lemus Ulises Mariano
Servin Zamora Alejandro

Profesor: Hector Mendoza Cortés

05 de junio de 2019

Índice

1. Objetivo general	3
2. Objetivos particulares	3
3. Marco teórico	4
3.1. Definición de XAMPP	4
3.2. Definición de LAMP	4
3.3. Servidor web	4
3.3.1. Apache	6
3.3.2. Principales características de Apache	6
3.3.3. Usos de Apache	6
3.4. MySQL	7
3.5. PHP	8
3.5.1. ¿Cómo trabaja PHP?	8
4. Solución propuesta	9
4.1. Instalación LAMP	9
4.1.1. Paso 1: Instalar Apache y actualizar el cortafuegos	9
4.1.2. Paso 2: Configuración de un cortafuegos básico	9
4.1.3. Paso 3: Ajuste del cortafuegos para permitir el tráfico web	9
4.1.4. Paso 4: Instalar MySQL	10
4.1.5. Paso 5: Instalar PHP	11
4.1.6. Paso 6: Evaluar el procesamiento de PHP sobre tu servidor web	13
4.2. Cómo instalar y asegurar phpMyAdmin	16
4.2.1. Prerrequisitos	16
4.2.2. Paso 1: Instalar phpMyAdmin	16
4.2.3. Paso 2: Asegure su instancia phpMyAdmin	19
4.2.4. Paso 3: Configurar Apache para permitir anulaciones de .htaccess	19
4.2.5. Paso 4: Crear un archivo .htaccess	20
4.2.6. Paso 5: Cree el archivo .htpasswd para la autenticación	20
4.2.7. Creación de un formulario con HTML, PHP y MySQL	21
5. Conclusiones	35
5.1. Álvarez Santiago Daniela	35
5.2. Gonzalez Hernandez Diego	35
5.3. Martinez Vivaldo Uriel Giovanni	35
5.4. Rojas Solís Juan Carlos	35
5.5. Sanchez Lemus Ulises Mariano	35
5.6. Servin Zamora Alejandro	35
Referencias	36

1. Objetivo general

Descargar, instalar y configurar un servidor web (Apache), una base de datos (MySQL) y una página web (PHP) de manera local para poder generar un formulario que almacene la información en una base de datos y además muestre la información guardada.

2. Objetivos particulares

- Definir que es XAMPP y LAMP.
- Instalar la paquetería LAMP.
- Configurar la paquetería LAMP.
- Realizar un formulario en PHP y almacenar la información en una base de datos.

3. Marco teórico

3.1. Definición de XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.[1]

3.2. Definición de LAMP

El acrónimo **LAMP** está compuesto por las iniciales de sus cuatro componentes: **L**inux, **A**pache, **M**ySQL y **P**HP. Estos forman la infraestructura en el servidor, que hace posible la creación y el alojamiento de páginas web dinámicas. Los componentes individuales se acumulan unos sobre otros, por lo que esta plataforma también recibe el nombre de LAMP.

Su funcionamiento es muy simple. Linux sirve como sistema operativo base para ejecutar el servidor web Apache. Este último no puede interpretar contenidos dinámicos, pero es aquí donde PHP entra a ejercer sus funciones de programación del lado del servidor. El proceso funciona entonces de la siguiente manera: Apache le envía un código fuente al intérprete PHP, incluyendo la información correspondiente sobre las acciones del visitante de la web, y permite el acceso a la base de datos MySQL. El resultado es devuelto a Apache y este se muestra finalmente en el navegador web del visitante.[2]

3.3. Servidor web

Para entender lo que es Apache, primeramente definiremos lo que es un servidor web. La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El web server, se encuentra a la espera de que algún navegador le haga alguna petición, como por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red.

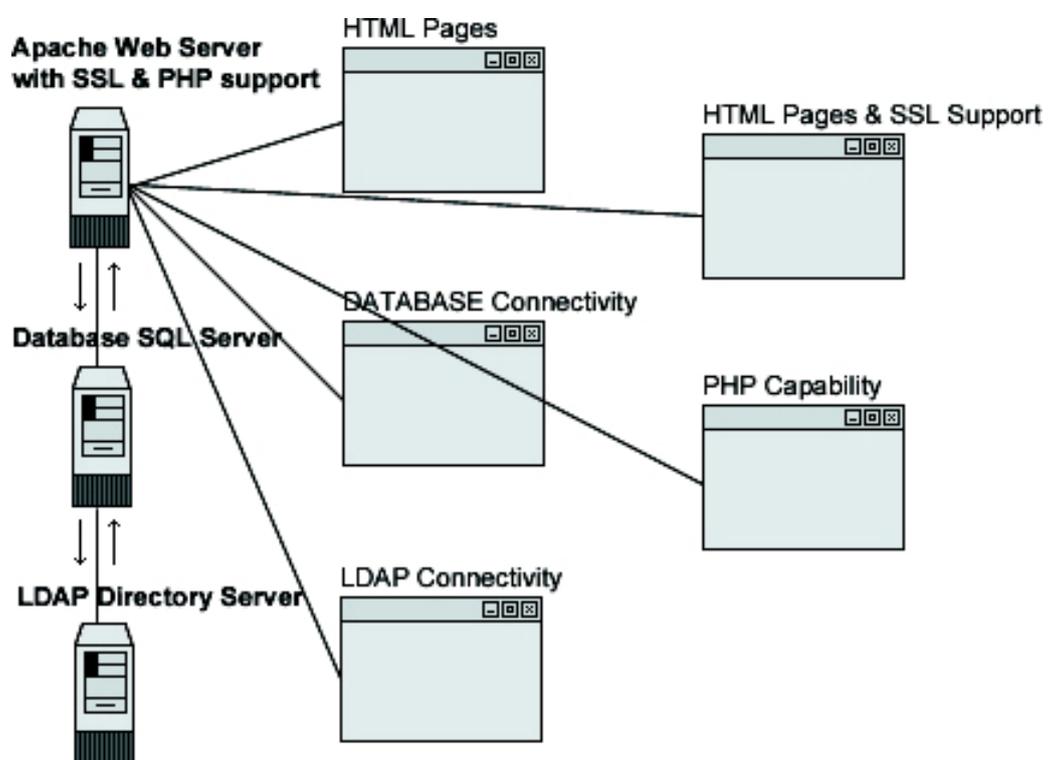


Figura 1: Esquema de funcionamiento de Apache

3.3.1. Apache

Apache es una popular y eficiente alternativa, que ofrece servicios web. Este web server es uno de los logros más grandes del software libre y la punta de lanza del mundo de las páginas web.

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener su excelente rendimiento.

Desde el año 1996, es el servidor web más popular del mundo, debido a su estabilidad y seguridad. Apache sigue siendo desarrollado por la comunidad de usuarios desarrolladores que trabaja bajo la tutela de Apache Software Foundation.

3.3.2. Principales características de Apache

Entre las principales características de Apache, se encuentran las siguientes:

- Soporte de seguridad SSL y TLS.
- Puede realizar autenticación de datos utilizando SGDB.
- Puede dar soporte a diferentes lenguajes, como Perl, PHP, Python y tcl.

3.3.3. Usos de Apache

Apache es utilizado principalmente, para realizar servicio a páginas web, ya sean estáticas o dinámicas. Este estupendo servidor se integra a la perfección con otras aplicaciones, creando el famoso paquete XAMP con Perl, Python, MySQL y PHP, junto a cualquier sistema operativo, que por lo general es Linux, Windows o Mac OS.[3]

3.4. MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.[4]



Figura 2: MySQL

3.5. PHP

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

3.5.1. ¿Cómo trabaja PHP?

El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo `http://www.aprenderaprogramar.com/index.php` en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es: Petición de página web al servidor --> El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc --> El servidor responde enviando una página web "normal" (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual).^[5]

En resumen:

Páginas estáticas: Petición --> Respuesta

Páginas dinámicas: Petición --> Procesado y preparación --> Respuesta

4. Solución propuesta

4.1. Instalación LAMP

Antes de instalar un servidor LAMP, es importante aclarar un par de cosas. Además, es fundamental asegurarse de que el hardware utilizado tenga la suficiente capacidad para instalar y trabajar con la infraestructura LAMP.

A continuación, los pasos para implementar el LAMP:

4.1.1. Paso 1: Instalar Apache y actualizar el cortafuegos

El servidor web, Apache es uno de los servidores web más populares en el mundo. Se encuentra bien documentado y ha sido utilizado en buena parte de la historia de la web, hechos que lo convierten en un muy buen candidato para ser escogido como el servidor por defecto de páginas web.

Instala Apache usando el administrador de paquetes de Ubuntu, apt:

Listing 1: Instalación de Apache

```
1 $ sudo apt update
2 $ sudo apt install apache2
```

Como éste es un comando sudo, estas operaciones son ejecutadas con los privilegios de superusuario. Te preguntará por la contraseña de tu cuenta regular para verificar tus intenciones.

Una vez hayas autenticado tu contraseña, apt te informará cuáles paquetes se instalarán y cuánto espacio en disco será requerido. Digita Y y después Enter para continuar, así, la instalación procederá.

4.1.2. Paso 2: Configuración de un cortafuegos básico

Ubuntu 18.04 puede usar el firewall UFW para asegurar conexiones a ciertos servicios que están permitidos. Podemos configurar un firewall básico fácilmente usando esta aplicación.

Diferentes aplicaciones pueden registrar sus perfiles con UFW tras la instalación. Estos perfiles permiten a UFW administrar estas aplicaciones por nombre.

Puedes ver esto escribiendo:

Listing 2: Configuración de firewall

```
1 # ufw app list
```

Podemos habilitar el firewall escribiendo:

Listing 3: Configuración de firewall

```
1 # ufw enable
```

Escriba zz presione ENTER para continuar. Puedes ver los servicios activos escribiendo:

Listing 4: Configuración de firewall

```
1 # ufw status
```

Si instala y configura servicios, deberá ajustar la configuración del servidor de seguridad para permitir un tráfico aceptable.

4.1.3. Paso 3: Ajuste del cortafuegos para permitir el tráfico web

Assumiendo habilitaste el cortafuegos UFW, ahora podrás asegurarte que tu cortafuegos permite el tráfico HTTP y HTTPS. Para hacerlo, verifica que el UFW tiene un perfil de aplicación para Apache mediante el comando:

Listing 5: Configuración de firewall

```
1 $ sudo ufw app list
```

Si solicitas la información del perfil Apache Full, se debería mostrar que el tráfico se encuentra habilitado para los puertos 80 y 443.

Para permitir el tráfico de entrada HTTP y HTTPS para este perfil, digita:

Listing 6: Configuración de firewall

```
$ sudo ufw allow in "Apache Full"
```

Puedes hacer una comprobación instantánea de que todo ha ido según lo planeado visitando la dirección IP de tu servidor en un navegador (o en su defecto, la de localhost 127.0.0.1):

Verás la página web predeterminada de Apache para Ubuntu 18.04, la cual tiene propósitos informativos y de prueba. Debería verse semejante a:

4.1.4. Paso 4: Instalar MySQL

Ahora que tienes tu servidor web activo y funcional, es el momento de instalar MySQL. MySQL es un sistema de administración de bases de datos. Básicamente, él organizará y proveerá acceso a las bases de datos donde tu sitio podrá guardar información.

De nuevo, usa apt para adquirir e instalar este software:

Listing 7: Instalación y configuración de MySQL

```
$ sudo apt install mysql-server
```

De nuevo se te desplegará una lista de paquetes a instalar, así como el espacio en disco que requerirá. Presiona Y para continuar.

Cuando la instalación esté completa, debes ejecutar un archivo de comandos de seguridad que viene preinstalado con MySQL, éste removerá algunos parámetros peligrosos, así como asegurará el acceso a tu base de datos. Ejecuta el archivo interactivo de comandos mediante:

Listing 8: Instalación y configuración de MySQL

```
$ sudo mysql_secure_installation
```

Se te preguntará si quieres configurar el conector de validación de contraseña: VALIDATE PASSWORD PLUGIN.

Nota: Habilitar esta funcionalidad dependerá de juzgar las necesidades de tu servidor. Si está habilitada una contraseña que no cumpla con un criterio específico, será rechazada por MySQL y generará un error. Esto podría ser un problema si utilizas contraseñas débiles en conjunto con software que configura automáticamente credenciales de usuario de MySQL, como por ejemplo los paquetes de Ubuntu para phpMyAdmin. Es seguro dejar esta validación deshabilitada, pero recuerda siempre utilizar contraseñas únicas y fuertes para las credenciales de las bases de datos. Como contraseña recomendada: "MyNewP4ss!".

Responde Y si estás de acuerdo, cualquier otra respuesta continuará sin realizar la habilitación.

Si respondiste "yes", se te solicitará que selecciones el nivel de validación de contraseña. Debes tener en cuenta que si digitas 2 representando el nivel más fuerte, recibirás errores al intentar utilizar una contraseña que no contenga números, letras mayúsculas y minúsculas, así como caracteres especiales; además la contraseña no podrá estar basada en palabras comunes en un diccionario.

Sin importar el nivel escogido para VALIDATE PASSWORD PLUGIN, tu servidor te solicitará, a continuación, seleccionar y confirmar la contraseña para el usuario root de MySQL. Ésta es una cuenta administrativa dentro MySQL con privilegios incrementados. Puede ser entendida de manera similar a la cuenta root del servidor mismo (Sin embargo, estarás configurando una cuenta específica para MySQL). Asegúrate de utilizar una contraseña fuerte y única, no debería dejarse en blanco.

Si habilitaste la validación de contraseña, se te mostrará qué tan fuerte es la contraseña para la cuenta root que acabas de introducir y tu servidor preguntará si quieres cambiarla. Si crees que es adecuado como está, digita N para seleccionar "no." en la línea de comandos.

Para las siguientes preguntas, presiona Y y pulsa la tecla Enter en cada sugerencia. Esto removerá algunos usuarios anónimos y la base de datos de prueba, deshabilitará ingresos remotos del root, y cargará estas nuevas reglas, de tal modo que MySQL respete inmediatamente los cambios que se acaban de hacer.

En este punto, tu sistema de bases de datos se encuentra configurado y puedes seguir con la instalación de PHP, el componente final de la pila LAMP.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Figura 3: Si viste esta página, entonces tu servidor web se encuentra instalado correctamente y es accesible a través del cortafuegos

4.1.5. Paso 5: Instalar Instalar PHP

PHP es el componente de tu configuración que procesa código para desplegar contenido dinámico. Puede ejecutar archivos, conectarse a tus bases de datos MySQL para obtener información, y manejar la visualización

del contenido procesado sobre tu servidor web.

Una vez más usaremos el sistema apt para instalar PHP. Adicionalmente lo podemos configurar para que se ejecute sobre el servidor Apache y para que se comunique con la base de datos MySQL:

Listing 9: Instalación y configuración de PHP

```
1 $ sudo apt install php libapache2-mod-php php-mysql
```

Esto debería instalar PHP sin problemas, sin embargo, probaremos esta instalación en este momento.

En la mayoría de los casos, desearás modificar la forma mediante la cual Apache sirve archivos cuando un directorio es solicitado. En este momento, si un usuario solicita un directorio del servidor, Apache buscará, en primera instancia, un archivo llamado index.html. Nosotros queremos que el servidor web le dé prelación a los archivos PHP sobre cualquier otro archivo. Para lo cual haremos que el Apache busque el archivo index.php en primer lugar.

Para lograrlo, digita el siguiente comando para abrir el archivo dir.conf en un editor de texto con privilegios de superusuario:

Listing 10: Instalación y configuración de PHP

```
1 $ sudo nano /etc/apache2/mods-enabled/dir.conf
```

Debería verse semejante a esto:

Listing 11: Instalación y configuración de PHP

```
1 <IfModule mod_dir.c>
2     DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
3     index.htm
4 </IfModule>
```

Mueve el archivo de índice de PHP (subrayado arriba) a la primera posición después de la especificación DirectoryIndex, debería verse similar a:

Listing 12: Instalación y configuración de PHP

```
1 <IfModule mod_dir.c>
2     DirectoryIndex index.php index.html index.cgi index.pl index.xhtml
3     index.htm
4 </IfModule>
```

Cuando termines, graba y cierra el archivo presionando las teclas Ctrl + X. Confirma los cambios presionando Y, y a continuación pulsa la tecla Enter para verificar el lugar de grabación del archivo.

A continuación, deberás reiniciar el servidor Apache para que tus cambios sean reconocidos, lo puedes hacer mediante el comando:

Listing 13: Instalación y configuración de PHP

```
1 $ sudo systemctl restart apache2
```

También podrás verificar el estado del servicio apache2 utilizando systemctl:

Listing 14: Instalación y configuración de PHP

```
1 $ sudo systemctl status apache2
```

Para ampliar la funcionalidad de PHP, tienes la posibilidad de instalar algunos módulos adicionales. Para ver las opciones disponibles de módulos y librerías de PHP, envía los resultados de apt search a less, un paginador que te permitirá navegar dentro de la salida de otro comando:

Listing 15: Instalación y configuración de PHP

```
1 $ apt search php- | less
```

Usa las flechas para moverte hacia arriba y abajo, y pulsa Q para salir.

Como resultado obtendrás todos los componentes opcionales que puedes instalar. El sistema te mostrará una descripción corta de cada uno de ellos.

Para indagar más sobre las funcionalidades de cada módulo, simplemente puedes buscar en la web su descripción, o alternatively, puedes ver la descripción larga de cada paquete, digitando:

Listing 16: Instalación y configuración de PHP

```
1 $ apt show package_name
```

La salida será extensiva, con un campo en particular llamado Description que tendrá una explicación más extensa acerca de la funcionalidad que el módulo provee.

Por ejemplo, para ver las funcionalidades del módulo php-cli, podrías digitar:

Listing 17: Instalación y configuración de PHP

```
1 $ apt show php-cli
```

Si después de tu consulta decides instalar algún paquete, lo puedes hacer mediante el comando apt install, de la misma forma que lo has hecho para otro software.

Si te das cuenta que necesitas instalar php-cli, puedes escribir:

Listing 18: Instalación y configuración de PHP

```
1 $ sudo apt install php-cli
```

Si deseas instalar más de un módulo, lo puedes hacer listándolos separados por espacio, después del comando apt install, semejante a lo siguiente:

Listing 19: Instalación y configuración de PHP

```
1 $ sudo apt install package1 package2 ...
```

Para este punto, tu pila LAMP se encuentra instalada y configurada. Sin embargo, antes de hacer más cambios o de desplegar una aplicación, sería beneficioso hacer una prueba proactiva de la configuración de PHP, quizá haya alguna situación que merezca alguna atención en este momento.

4.1.6. Paso 6: Evaluar el procesamiento de PHP sobre tu servidor web

Con el objetivo de evaluar si tu sistema está configurado de manera correcta para usar PHP, crea un archivo de comandos básico, llamado info.php. Para que Apache pueda encontrar y servir este archivo de manera correcta, debe ser alojado en un directorio muy específico, que es llamado "web root".

En Ubuntu 18.04, este directorio se encuentra localizado en /var/www/html/. Crea el archivo en ese lugar digitando:

Listing 20: Probando PHP

```
1 $ sudo nano /var/www/html/info.php
```

Esto creará un archivo en blanco. Introduce el siguiente código PHP válido, dentro del archivo de texto:

Listing 21: Probando PHP

```
1 <?php
2     phpinfo();
3 ?>
```

Cuando termines, graba y cierra el archivo.

Ahora ya puedes probar si tu servidor web se encuentra habilitado para desplegar correctamente contenido generado por este archivo de comandos PHP. Para hacerlo, visita una página web específica en tu navegador, necesitarás tu dirección pública de nuevo.

La dirección que deberás visitar es:

Listing 22: Probando PHP

```
1 http://tu_direccion_ip/info.php
```

La página que deberías estar viendo debe ser similar la siguiente:

PHP Version 7.2.3-1ubuntu1


System	Linux LAMP-1804-test 4.15.0-15-generic #16-Ubuntu SMP Wed Apr 4 13:58:14 UTC 2018 x86_64
Build Date	Mar 14 2018 22:03:58
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-intl.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
with Zend OPcache v7.2.3-1ubuntu1, Copyright (c) 1999-2018, by Zend Technologies



Configuration apache2handler

Apache Version	Apache/2.4.29 (Ubuntu)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	162.243.26.126:80
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_auth_core mod_auth_file mod_auth_core mod_auth_host mod_auth_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_setenvif mod_status

Figura 4: Esta página provee información básica sobre tu servidor, recogida desde la perspectiva de PHP

Es útil cuando necesites hacer algún tipo de seguimiento o para verificar que la configuración deseada ha sido aplicada de manera correcta.

Si pudiste ver esta página en tu navegador, tu PHP está trabajando según lo esperado.

Probablemente quieras remover este archivo después de la prueba, éste puede dar información sobre tu servidor a usuarios no autorizados. Para hacer esto, digita el siguiente comando:

Listing 23: Probando PHP

```
1 sudo rm /var/www/html/info.php
```

Siempre puedes crear este archivo de nuevo en caso que necesites acceder a esta información en otra oportunidad.[6]

4.2. Cómo instalar y asegurar phpMyAdmin

Si bien muchos usuarios necesitan la funcionalidad de un sistema de administración de base de datos como MySQL, es posible que no se sientan cómodos interactuando con el sistema únicamente desde el indicador de MySQL.

phpMyAdmin fue creado para que los usuarios puedan interactuar con MySQL a través de una interfaz web. En esta guía, analizaremos cómo instalar y proteger phpMyAdmin para que pueda usarlo de manera segura para administrar sus bases de datos desde un sistema Ubuntu.

4.2.1. Prerrequisitos

Antes de comenzar con esta guía, debe completar algunos pasos básicos.

Primero, asumiremos que está utilizando un usuario no root con privilegios de sudo.

También vamos a asumir que ha completado una instalación de LAMP (Linux, Apache, MySQL y PHP) en su servidor Ubuntu. Si esto aún no se ha completado, regrese al apartado de instalación LAMP.

Finalmente, hay consideraciones de seguridad importantes al usar software como phpMyAdmin, ya que:

- Se comunica directamente con su instalación de MySQL.
- Maneja la autenticación usando las credenciales de MySQL.
- Ejecuta y devuelve resultados para consultas SQL arbitrarias.

Por estas razones, y debido a que es una aplicación de PHP ampliamente implementada que se dirige frecuentemente a ataques, nunca debe ejecutar phpMyAdmin en sistemas remotos a través de una conexión HTTP simple.

Una vez que haya terminado con estos pasos, estará listo para ir al siguiente paso.

4.2.2. Paso 1: Instalar phpMyAdmin

Para comenzar, instalaremos phpMyAdmin desde los repositorios predeterminados de Ubuntu.

Podemos hacer esto actualizando nuestro índice de paquete local y luego utilizando el apt sistema de empaquetado para bajar los archivos e instalarlos en nuestro sistema:

Listing 24: Instalación phpmyadmin

```
1 $ sudo apt-get update
2 $ sudo apt-get install phpmyadmin php-mbstring php-gettext
```

Esto le hará algunas preguntas para configurar su instalación correctamente.

Advertencia: cuando aparece el primer mensaje, apache2 se resalta, pero no se selecciona. Si no pulsa Espacio para seleccionar Apache, el instalador no moverá los archivos necesarios durante la instalación. Pulse Espacio, Tab y luego Intro para seleccionar Apache.

- Para la selección del servidor, elija apache2.
- Seleccione sí cuando se le pregunte si desea utilizar dbconfig-common para configurar la base de datos.

- Se le pedirá la contraseña de administrador de su base de datos.
- A continuación, se le pedirá que elija y confirme una contraseña para la phpMyAdmin aplicación en sí.

El proceso de instalación en realidad agrega el archivo de configuración phpMyAdmin Apache en el `/etc/apache2/conf-enabled/directorio`, donde se lee automáticamente.

Lo único que debemos hacer es habilitar explícitamente el PHP `mcrypt` y las `mbstring` extensiones, lo que podemos hacer escribiendo:

Listing 25: Instalación phpmyadmin

```
1 $ sudo phpenmod mcrypt
2 $ sudo phpenmod mbstring
```

Luego, deberá reiniciar Apache para que se reconozcan sus cambios:

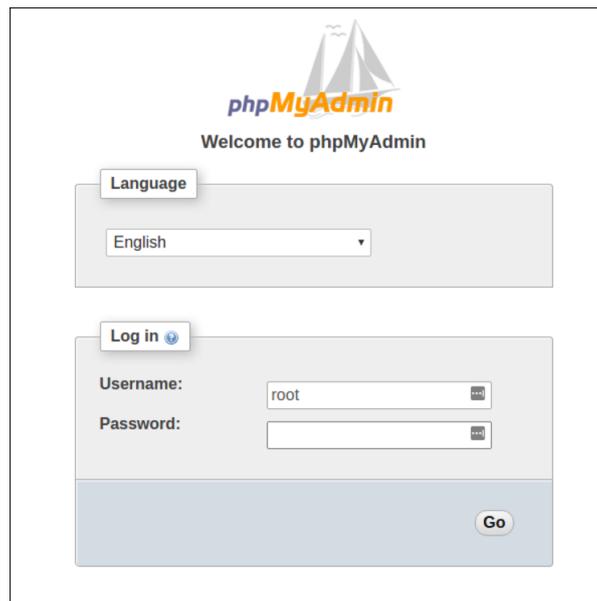
Listing 26: Instalación phpmyadmin

```
1 $ sudo systemctl restart apache2
```

Ahora puede acceder a la interfaz web visitando el nombre de dominio o la dirección IP de su servidor, seguido de `/phpmyadmin`:

Listing 27: Instalación phpmyadmin

```
1 https://tu_direccion_ip/phpmyadmin
```



The image shows the phpMyAdmin login page. At the top center is the phpMyAdmin logo, which consists of a stylized sailboat icon above the text 'phpMyAdmin'. Below the logo is the text 'Welcome to phpMyAdmin'. Underneath is a 'Language' dropdown menu with 'English' selected. Below the language menu is a 'Log in' button with a small blue icon. The login form contains two input fields: 'Username:' with the text 'root' and a password toggle icon, and 'Password:' with an empty field and a password toggle icon. At the bottom right of the form is a 'Go' button.

Figura 5: Inicio de sesión phpmyadmin

Ahora puede iniciar sesión en la interfaz con el root nombre de usuario y la contraseña administrativa que configuró durante la instalación de MySQL.

Cuando inicie sesión, verá la interfaz de usuario, que se verá así:

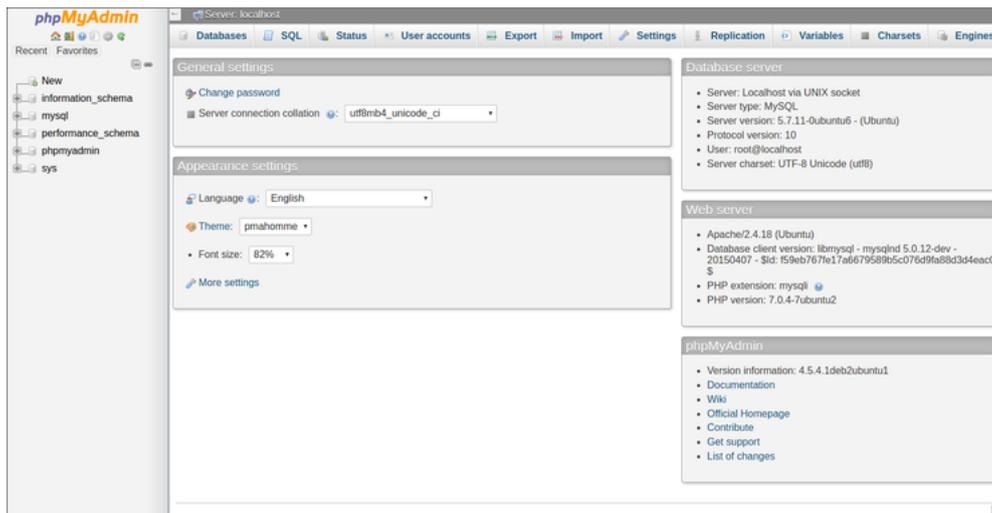


Figura 6: Inicio de sesión phpmyadmin

4.2.3. Paso 2: Asegure su instancia phpMyAdmin

Pudimos hacer que nuestra interfaz phpMyAdmin funcionara con bastante facilidad. Sin embargo, todavía no hemos terminado. Debido a su ubicuidad, phpMyAdmin es un objetivo popular para los atacantes. Debemos tomar medidas adicionales para evitar el acceso no autorizado.

Una de las maneras más fáciles de hacer esto es colocar una puerta de enlace frente a toda la aplicación. Podemos hacer esto utilizando las .htaccess funciones de autenticación y autorización integradas de Apache .

4.2.4. Paso 3: Configurar Apache para permitir anulaciones de .htaccess

Primero, debemos habilitar el uso de .htaccess anulaciones de archivos mediante la edición de nuestro archivo de configuración de Apache.

Editaremos el archivo vinculado que se ha colocado en nuestro directorio de configuración de Apache:

Listing 28: Configuración phpmyadmin

```
1 $ sudo nano /etc/apache2/conf-available/phpmyadmin.conf
```

Necesitamos agregar una AllowOverride All directiva dentro de la <Directory /usr/share/phpmyadmin> sección del archivo de configuración, como esta:

Listing 29: /etc/apache2/conf-available/phpmyadmin.conf

```
1 <Directory /usr/share/phpmyadmin>
2     Options FollowSymLinks
3     DirectoryIndex index.php
4     AllowOverride All
5     . . .
```

Cuando haya agregado esta línea, guarde y cierre el archivo.

Para implementar los cambios que hizo, reinicie Apache:

Listing 30: Configuración phpmyadmin

```
1 $ sudo systemctl restart apache2
```

4.2.5. Paso 4: Crear un archivo .htaccess

Ahora que hemos habilitado el .htaccess de nuestra aplicación, debemos crear uno para implementar algo de seguridad.

Para que esto tenga éxito, el archivo debe crearse dentro del directorio de la aplicación. Podemos crear el archivo necesario y abrirlo en nuestro editor de texto con privilegios de root escribiendo:

Listing 31: Configuración phpmyadmin

```
1 $ sudo nano /usr/share/phpmyadmin/.htaccess
```

Dentro de este archivo, necesitamos ingresar la siguiente información:

Listing 32: /usr/share/phpmyadmin/.htaccess

```
1 AuthType Basic
2 AuthName "Restricted Files"
3 AuthUserFile /etc/phpmyadmin/.htpasswd
4 Require valid-user
```

Repasemos qué significa cada una de estas líneas:

- **AuthType Basic:** Esta línea especifica el tipo de autenticación que estamos implementando. Este tipo implementará la autenticación de contraseña utilizando un archivo de contraseña.
- **AuthName:** Esto establece el mensaje para el cuadro de diálogo de autenticación. Debe mantener este genérico para que los usuarios no autorizados no obtengan ninguna información sobre lo que se está protegiendo.
- **AuthUserFile:** Esto establece la ubicación del archivo de contraseña que se utilizará para la autenticación. Esto debería estar fuera de los directorios que están siendo servidos. Vamos a crear este archivo en breve.
- **Require valid-user:** Esto especifica que solo los usuarios autenticados deben tener acceso a este recurso. Esto es lo que realmente impide que entren usuarios no autorizados.

Cuando haya terminado, guarde y cierre el archivo.

4.2.6. Paso 5: Cree el archivo .htpasswd para la autenticación

La ubicación que seleccionamos para nuestro archivo de contraseña era /etc/phpmyadmin/.htpasswd". Ahora podemos crear este archivo y pasarle un usuario inicial con la htpasswdutilidad:

Listing 33: Configuración phpmyadmin

```
1 $ sudo htpasswd -c /etc/phpmyadmin/.htpasswd username
```

Se le solicitará que seleccione y confirme una contraseña para el usuario que está creando. Posteriormente, el archivo se creará con la contraseña hash que ingresó.

Si desea ingresar un usuario adicional, debe hacerlo sin el -cindicador, como este:

Listing 34: Configuración phpmyadmin

```
1 $ sudo htpasswd /etc/phpmyadmin/.htpasswd additionaluser
```

Ahora, cuando acceda a su subdirectorío phpMyAdmin12, se le pedirá el nombre de la cuenta adicional y la contraseña que acaba de configurar:

Listing 35: Configuración phpmyadmin

```
1 https://tu_direccion_ip/phpmyadmin
```

Después de ingresar a la autenticación de Apache, se lo llevará a la página de autenticación de phpMyAdmin para ingresar sus otras credenciales. Esto agregará una capa adicional de seguridad ya que phpMyAdmin ha sufrido vulnerabilidades en el pasado.^[7]

4.2.7. Creación de un formulario con HTML, PHP y MySQL

Después de haber realizado la configuración e instalación de LAMP, ahora nos toca realizar un pequeño formulario en HTML y PHP. La información que guarde dicho formulario se almacenará en una base de datos.

Para la creación de una base de datos, existen dos maneras: en gráfico (usando phpmyadmin) y en consola (por medio de comandos).

Nosotros lo haremos por medio de phpmyadmin, sin embargo en la parte final de este capítulo, se pondrán los comandos de MySQL que se necesitan para crear una base de datos, tablas, hacer consultas, etc.

Lo primero que vamos a hacer, es crear un pequeño código en HTML que nos muestre información en el navegador. El código que se implementó fue el siguiente:

Listing 36: Código en HTML

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Forma</title>
5  </head>
6  <body>
7  <form method="post" action="datos.php">
8  Nombre de usuario : <input type="text" name="username"><br><br>
9  Contraseña : <input type="password" name="password"><br><br>
10 <input type="submit" value="Submit">
11 </form>
12 </body>
13 </html>

```

Listing 37: Código en PHP

```

1 <?php
2 $username = filter_input(INPUT_POST, 'username');
3 $password = filter_input(INPUT_POST, 'password');
4 if (!empty($username)){
5 if (!empty($password)){
6 $host = "localhost";
7 $dbusername = "root";
8 $dbpassword = "MyNewP4ss!";
9 $dbname = "carlos_operativos";
10 // Create connection
11 $conn = new mysqli ($host, $dbusername, $dbpassword, $dbname);
12 if (mysqli_connect_error()){
13 die('Connect Error ('. mysqli_connect_errno() .') '
14 . mysqli_connect_error());
15 }
16 else{
17 $sql = "INSERT INTO cuenta (username, password)
18 values ('$username', '$password')";
19 if ($conn->query($sql)){
20 echo "Nuevo registro agregado exitosamente";
21 }
22 else{
23 echo "Error: ". $sql ."
24 ". $conn->error;
25 }
26 $conn->close();
27 }
28 }

```

```
29     else{
30     echo "Contraseña no debe de estar vacia";
31     die();
32     }
33     }
34     else{
35     echo "Nombre de usuario no debe de estar vacio";
36     die();
37     }
38     ?>
```

Después, vamos a crear una base de datos (en adelante BD) y una tabla. El nombre de la BD debe de coincidir con el que esta en el código de PHP. De igual manera para el nombre de la tabla, el usuario de la BD, la contraseña de la BD y el nombre del host.

Nos iremos a la siguiente pagina:

Listing 38: Página de phpmyadmin

```
1 http://localhost/phpmyadmin
```

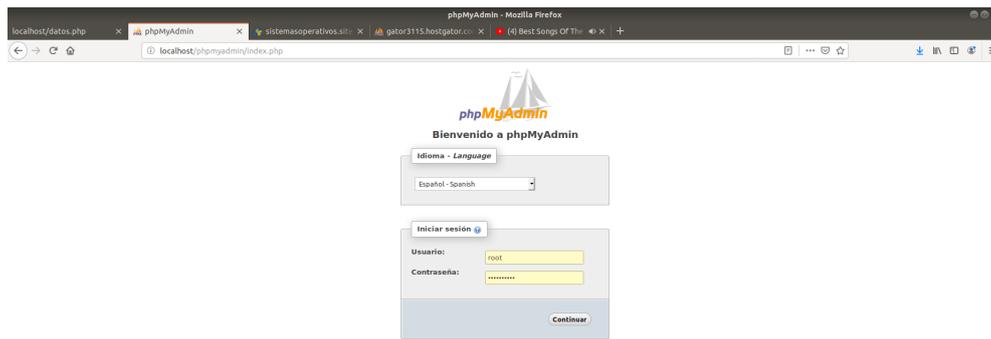


Figura 7: Inicio de sesión phpmyadmin

Iniciamos sesión, y en seguida nos mandara a la siguiente página:

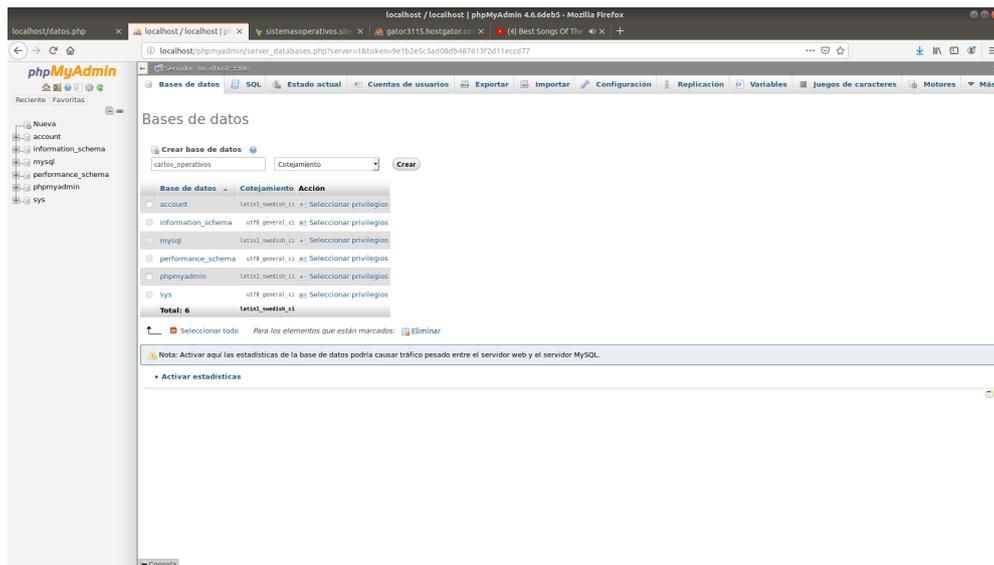


Figura 8: Crearemos una base de datos con el nombre de “carlos_operativos”

Luego, crearemos una tabla con el nombre de “cuenta”

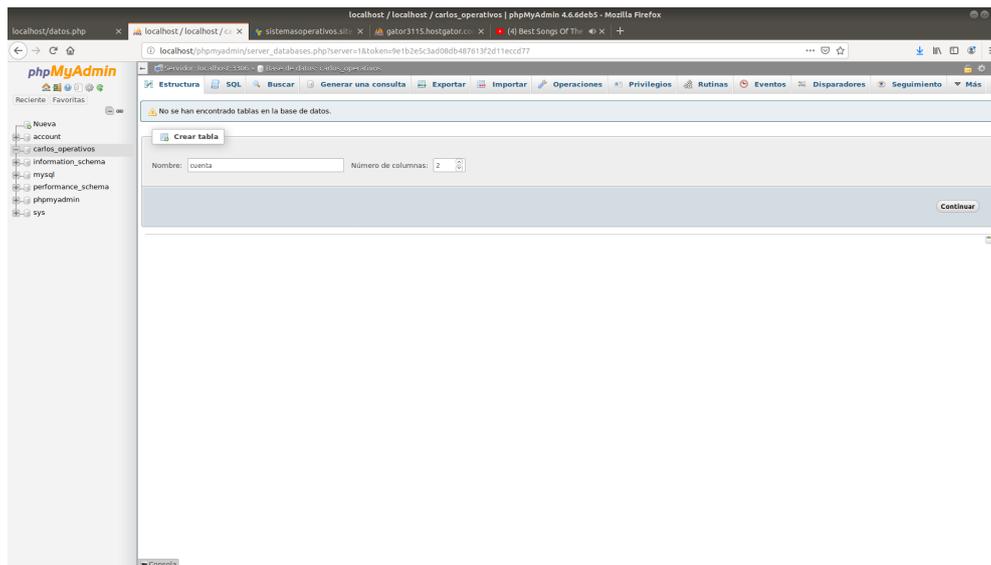


Figura 9: Creación de una tabla

Después crearemos dos apartados, uno con el nombre de “username” y otro con el nombre de “password”, nos servirán para guardar valores de usuario y contraseña respectivamente. Serán de tipo varchar y tendrán una extensión de 40 caracteres. (varchar es como tipo string).

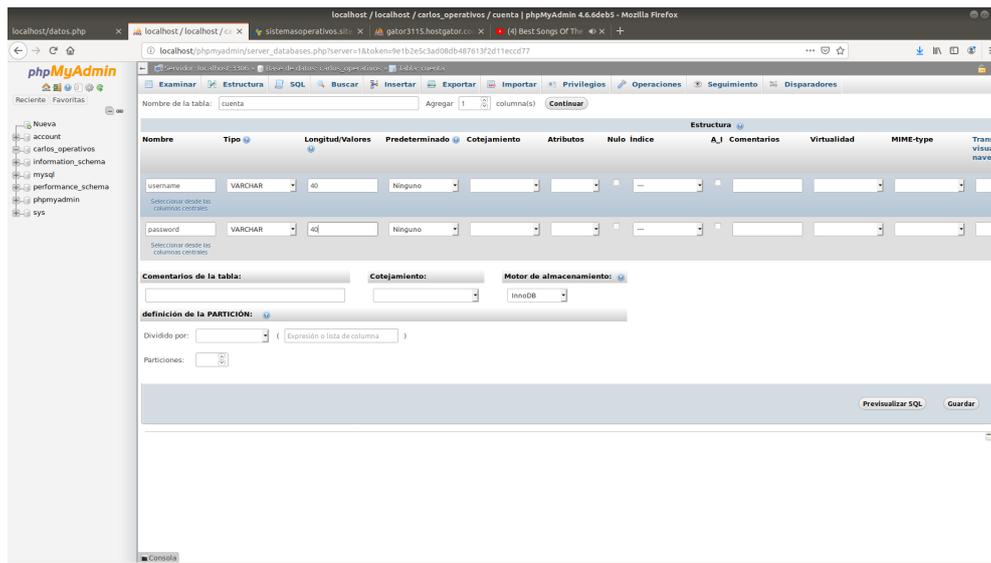


Figura 10: Creación apartados “username” y “password”

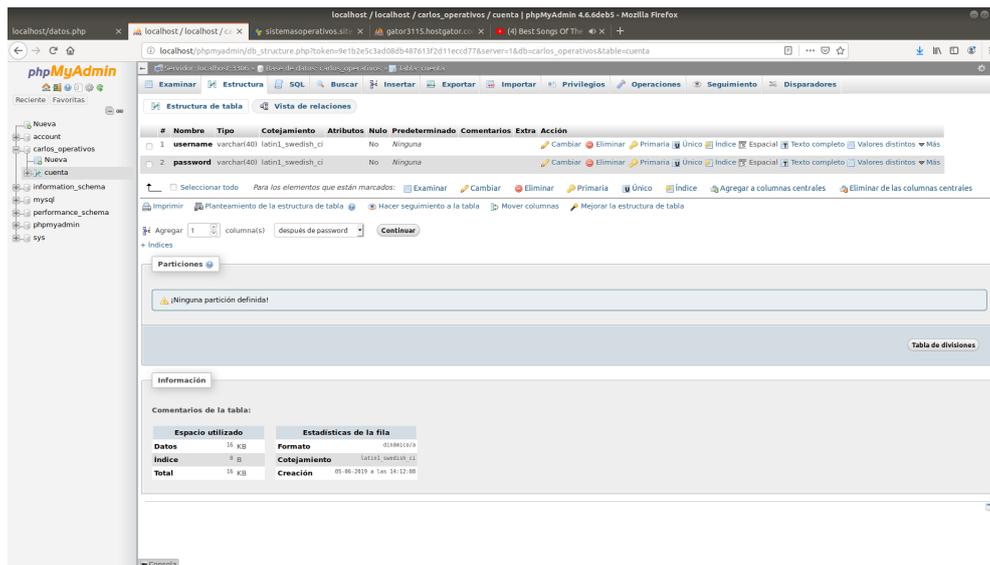


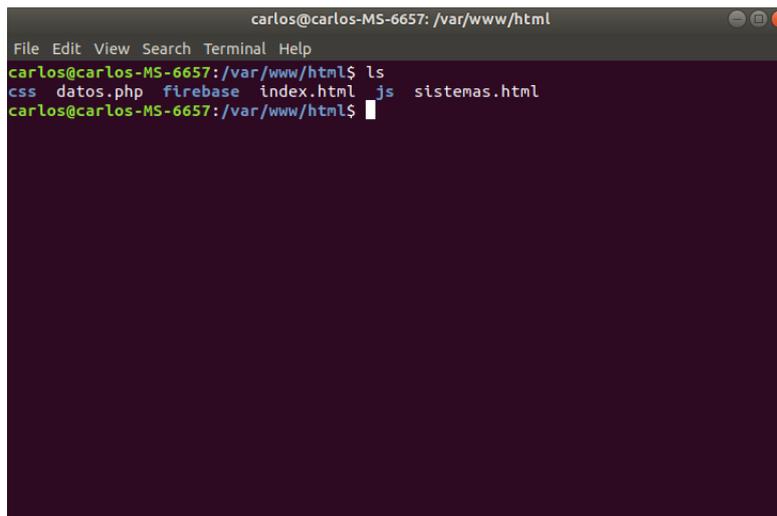
Figura 11: Apartados de “username” y “password”

Listo, ya creamos nuestra BD con nuestra tabla. Ahora, el siguiente paso es colocar los 2 códigos de HTML y PHP en la siguiente carpeta:

Listing 39: Directorio para cargar paginas web

```
1 /var/www/html
```

En nuestro caso, nosotros le pusimos como nombres “sistemas.html” y “datos.php”.

A terminal window titled 'carlos@carlos-MS-6657: /var/www/html' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'ls' being executed, resulting in the output: 'css datos.php firebase index.html js sistemas.html'. The prompt is 'carlos@carlos-MS-6657: /var/www/html\$' with a cursor.

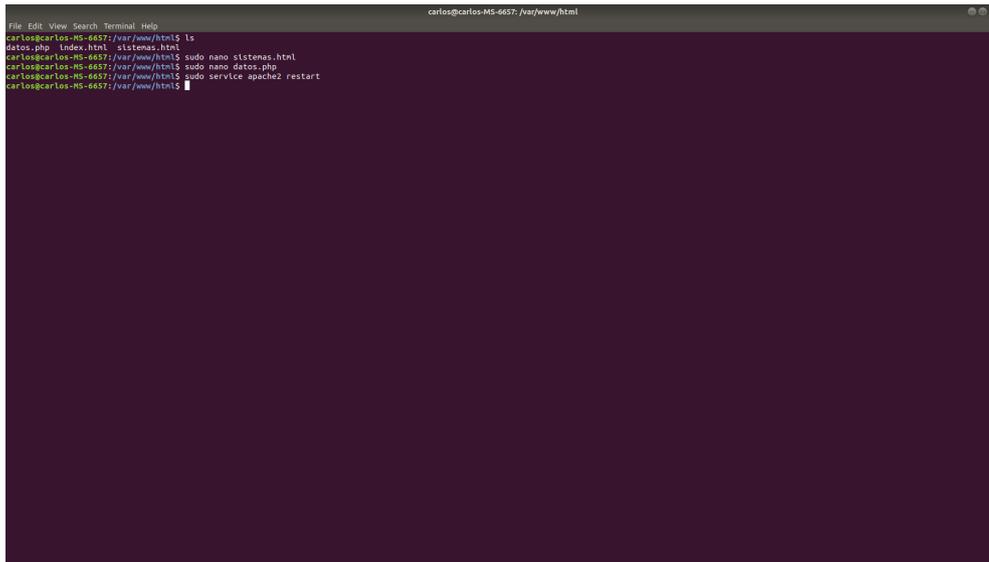
```
carlos@carlos-MS-6657: /var/www/html
File Edit View Search Terminal Help
carlos@carlos-MS-6657: /var/www/html$ ls
css datos.php firebase index.html js sistemas.html
carlos@carlos-MS-6657: /var/www/html$
```

Figura 12: Archivos HTML y PHP

Ya que colocamos los archivos en el directorio mencionado anteriormente. Reiniciamos Apache y nos iríamos a la siguiente dirección:

Listing 40: Página de phpmyadmin

```
1 http://localhost/sistemas.html
```

A terminal window with a dark purple background and white text. The window title is 'carlos@carlos-MS-6657: /var/www/html'. The terminal shows the following commands and their outputs:

```
File Edit View Search Terminal Help
carlos@carlos-MS-6657:/var/www/html$ ls
datos.php  index.html  sistemas.html
carlos@carlos-MS-6657:/var/www/html$ sudo nano sistemas.html
carlos@carlos-MS-6657:/var/www/html$ sudo nano datos.php
carlos@carlos-MS-6657:/var/www/html$ sudo service apache2 restart
carlos@carlos-MS-6657:/var/www/html$
```

Figura 13: Reiniciamos Apache



Figura 14: Nos vamos a <http://localhost/systemas.html>



Figura 15: LLenamos los campos

Ya que llenamos los campos, nos aparecerá el siguiente mensaje:



Figura 16: Mensaje de éxito

Para comprobar que, efectivamente los datos se fueron a la BD, nos regresamos a la pagina de phpmyadmin. Le damos clic en la base de datos que creamos, le damos clic a la tabla y listo, deberían de aparecer los datos:

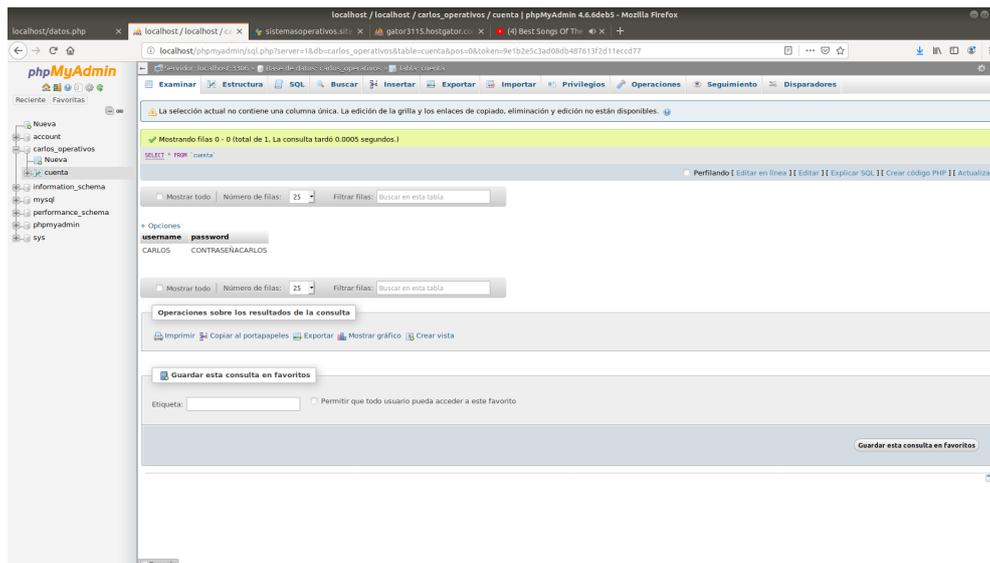


Figura 17: Creacion de un formulario

Ahora, a continuación se pondrán los codigos para MySQL:

- Crear una base de datos: create database [databasename];
- Listar todas las base de datos en el servidor: show databases;
- Cambiar a una base de datos: use [db name];
- Ver todas las tablas de una base de datos: show tables;
- Ver los formatos de campo de la base de datos: describe [table name];
- Eliminar una base de datos: drop database [database name];
- Eliminar una tabla de la base de datos: drop table [table name];
- Devolver todos los registros de una tabla: SELECT * FROM [table name];
- Devolver las columnas y la información de la columna correspondiente a la tabla designada: show columns from [table name];
- Mostrar ciertas filas seleccionadas con el valor «lo que sea»: SELECT * FROM [table name] WHERE [field name] = «whatever»;
- Mostrar todos los registros que contengan el nombre «Bob» AND el número de teléfono '3444444': SELECT * FROM [table name] WHERE name = «Bob» AND phone_number = '3444444';
- Mostrar todos los registros que contienen el nombre «Bob» AND el número de teléfono '3444444' ordenados por el campo «phone_number»: SELECT * FROM [table name] WHERE name != «Bob» AND phone_number = '3444444' order by phone_number;
- Mostrar todos los registros que comienzan con la palabra 'bob' AND el número de teléfono '3444444': SELECT * FROM [table name] WHERE name like «Bob%» AND phone_number = '3444444';
- Usar una expresión regular para encontrar registros. Usar «REGEXP BINARY» para forzar la sensibilidad a las mayúsculas.
 - Esto encuentra cualquier registro que comience con «a»: SELECT * FROM [table name] WHERE rec RLIKE «â\$»;

- Mostrar registros únicos: `SELECT DISTINCT [column name] FROM [table name];`
- Mostrar los registros seleccionados, ordenados en orden ascendente (asc) o descendente (desc): `SELECT [col1],[col2] FROM [table name] ORDER BY [col2] DESC;`
- Devolver un número de filas: `SELECT COUNT(*) FROM [table name];`
- Sumar el contenido de la columna: `SELECT SUM(*) FROM [table name];`

Para mas información, visitar la pagina de MySQL.[\[8\]](#)

5. Conclusiones

5.1. Álvarez Santiago Daniela

Respecto al tema que le correspondió a mi equipo, me permitió tener una vista más amplia hacer de la creación de las páginas web dinámicas y el uso de las base de datos en ellas, a pesar de que la principal herramienta para la implementación de este en Linux es LAMP, nosotros optamos por otras alternativas que en cierta forma resultan fáciles de aplicar pero un poco difíciles de comprender, pues ya no son procesos hechos por nosotros como tal.

En general con el curso me siento satisfecha ya que adquirí nuevos conocimientos sobre el sistema, e incluso podría llegar a cambiar de sistema operativo en mi computadora ya que nos enseñó bastantes ventajas, me resultó bastante entretenido y divertido cada una de las actividades que realizamos durante el semestre aunque a veces se nos complicaba.

5.2. Gonzalez Hernandez Diego

Este proyecto me agrada por que yo no conocía nada de esto yo vengo de la carrera de telecomunicaciones y para mi esto fue algo muy novedoso pude ver más acerca de como se construye una pagina web y todos los servicios que necesita para poder operar al menos ya tengo una idea de como poder emprender una pagina web, ya que si planeaba tomar un curso de como poder aprender esto ahora lo pondré en práctica para ver en que mejorar.

5.3. Martinez Vivaldo Uriel Giovanni

El saber usar estas herramientas es bastante útil ya que con estas podemos hacer nuestros propios proyectos en red y no solo eso sino que también es una gran herramienta para conseguir trabajo ya que estamos en una época donde estar en red es muy importante y para una pequeña o mediana empresa que aún no cuente con una página web le sería muy útil que le brindamos este servicio.

5.4. Rojas Solís Juan Carlos

En la elaboración de esta práctica, debo decir que fue todo un reto. No solo la instalación de LAMP, sino su configuración y puesta a punto estuvieron talachudas. Anteriormente desconocia todo el proceso que conlleva levantar un servidor web, y no solo eso, la creación de las paginas web, de darles seguridad, creacion de bases de datos. En fin, me siento satisfecho con el resultado que obtuve realizando esta práctica

5.5. Sanchez Lemus Ulises Mariano

Al estar en este curso de administración de sistemas operativos y habiendo participado en este trabajo acerca de la instalación y uso del software apache y las bases de datos, además de algo de información para complementar el trabajo, pude darme cuenta de la colaboración entre diferentes tipos de softwares para realizar una simple acción como la de navegar en internet, además de que requiere basto conocimiento y tiempo para que este sea implementado de una manera adecuada y satisfactoria, incluso es un trabajo en el mundo laboral, de ahí la importancia e incluso necesidad de entender el funcionamiento de estos softwares.

5.6. Servin Zamora Alejandro

En el proyecto me dio otra perspectiva de cómo se puede crear una página web junto con una base de datos. Anteriormente en la vocacional tuve la oportunidad de hacer algo semejante, aunque en esencia en el mismo procedimiento si cambia el entorno y es algo fácil de ver porque este proyecto ,tuve que documentarme en el uso básico de apache como era para poder instalarlo ,como también hacer que el firewall permitiera la conexión que queríamos hacer con MySQL y PHP para crear la red y un entorno que nos de la facilidad de crear una página web y verificar que estos funcionaban.

Referencias

- [1] EcuRed. Xampp. [En línea]. Disponible: <https://www.ecured.cu/XAMPP> (Accedido: 05-jun-2019).
- [2] D. Guide. Servidor lamp: una solución económica para webs dinámicas. [En línea]. Disponible: <https://www.ionos.mx/digitalguide/servidores/know-how/servidor-lamp-la-solucion-para-webs-dinamicas/> (Accedido: 05-jun-2019).
- [3] Culturacion. ¿qué es apache? [En línea]. Disponible: <http://culturacion.com/que-es-apache/> (Accedido: 05-jun-2019).
- [4] esepe studio. ¿qué es mysql? [En línea]. Disponible: <https://www.esepestudio.com/noticias/que-es-mysql> (Accedido: 05-jun-2019).
- [5] aprenderaprogramar.com. ¿qué es php? y ¿para qué sirve? un potente lenguaje de programación para crear páginas web. [En línea]. Disponible: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:iue-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193 (Accedido: 05-jun-2019).
- [6] M. Drake. Cómo instalar en ubuntu 18.04 la pila lamp — linux, apache, mysql y php. [En línea]. Disponible: <https://www.digitalocean.com/community/tutorials/como-instalar-en-ubuntu-18-04-la-pila-lamp-linux-apache-mysql-y-php-es> (Accedido: 05-jun-2019).
- [7] B. Bearnese. How to install and secure phpmyadmin on ubuntu 16.04. [En línea]. Disponible: <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-16-04> (Accedido: 05-jun-2019).
- [8] A. Vernazza. Comandos básicos para mysql. [En línea]. Disponible: <https://blog.baehost.com/comandos-basicos-para-mysql/> (Accedido: 05-jun-2019).