Laboratorio No. 6 – Capa de red e infraestructura parte 1

Contenido

[Introducción 1](#_Toc69478051)

[Marco Teórico 1](#_Toc69478052)

[Instalación y configuración de PHP para MariaDB en FreeBSD con un servidor Apache 1](#_Toc69478053)

[Instalación PHP 2](#_Toc69478054)

[Configuración PHP y php-fpm 2](#_Toc69478055)

[Configuración Nginx 3](#_Toc69478056)

[Instalación y configuración de PHP para PostgresSQL en Slackware con un servidor Apache 3](#_Toc69478057)

[Mostrando información de la base de datos 3](#_Toc69478058)

[Añadiendo información a la base de datos 3](#_Toc69478059)

[Creando Tablas en la base de datos 3](#_Toc69478060)

[Eliminando Tablas de la base de datos 3](#_Toc69478061)

[Conclusiones 3](#_Toc69478062)

[Bibliografía 3](#_Toc69478063)

# Introducción

En este laboratorio vamos hacer una conexión de las bases de datos creadas en los motores MariaDB y PostgresSQL desde una página web en Windows server usando lo servidores web de MariaDB y PostgresSQL, con el fin de poder conocer cómo se puede acceder desde cualquier dispositivo a las aplicaciones y datos de una organización añadiendo información de la base de datos y creando y borrando el contenido de éstas por medio de Scripts simples en PHP.

# Marco Teórico

**Nginx**: es un servidor web de código abierto que, ahora también es usado como proxy inverso, cache de HTTP, y balanceador de carga. Nginx está diseñado para ofrecer un bajo uso de memoria y alta concurrencia. En lugar de crear nuevos procesos para cada solicitud web, Nginx usa un enfoque asincrónico basado en eventos donde las solicitudes se manejan en un solo hilo. Con Nginx, un proceso maestro puede controlar múltiples procesos de trabajo. El proceso maestro mantiene los procesos de trabajo, y son estos lo que hacen el procesamiento real.

**Apache**: Apache es un servidor web de código abierto, multiplataforma y gratuito. La función esencial del servidor Apache es servir las webs alojadas en el servidor a los diversos navegadores como Chrome, Firefox, Safari entre otros. Apache consigue que la comunicación entre el servidor web y el cliente web (usuario que solicita la información) sea fluida y constante. Haciendo que cuando un usuario haga una petición HTTP a través de navegador para entrar a una web o URL específica, Apache devuelva la información solicitada a través del protocolo HTTP.

**PHP-FPM:** (FastCGI Process Manager) es una implementación alternativa al PHP FastCGI. FPM se encarga de interpretar código PHP.

## Instalación y configuración de PHP para MariaDB en FreeBSD con un servidor Apache



(1) adicionar información en al menos una tabla de cada una de las bases de datos

(2) Crear/borrar tablas en las bases de datos. Use PHP para la construcción de la página.

Muestre los resultados a su profesor.

Al hacer uso de un servidor Nginx, este no tiene la capacidad de procesar código PHP de forma nativa, es por ello que debemos hacer uso de un intérprete PHP en este caso **php-fpm el cual** se encargará de ensamblar el código html y enviar el código al navegador web correspondiente.

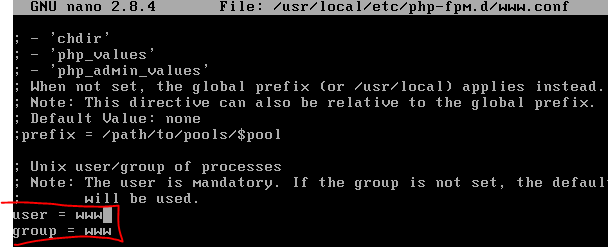
### Instalación PHP

En este caso se hace uso de la última versión de php, es por ello que ejecutamos el siguiente comando para descargar los módulos más importantes de PHP de los cuales haremos uso en la instalación.

**# pkg install php80 php80-mysqli php80-zlib php80-gd mod\_php80 php80-mbstring php80-curl**

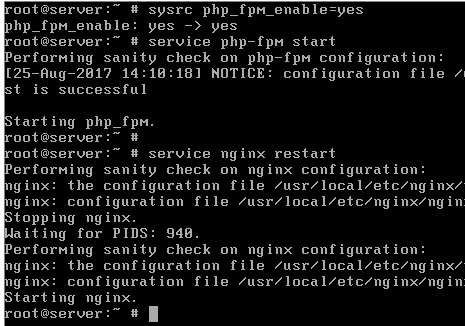
Después de descargar los módulos configuramos PHP-FPM para Nginx y ajustamos los valores de user a ‘www’ y group a ‘www’

### Configuración PHP y php-fpm



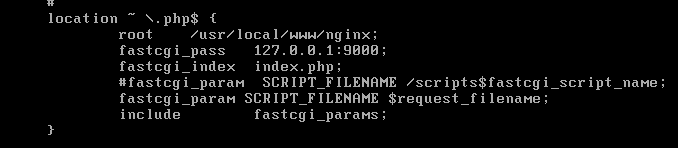
Luego creamos un archivo de configuración PHp.ini en el cual haremos cambios de personalización de php

, en primer lugar, editamos la configuración de tiempo a **date. timezone = America/Bogota,** y luego hacemos un **sysrc php\_fpm\_enable=yes,** para después habilitar el servicio de **php-fpm**

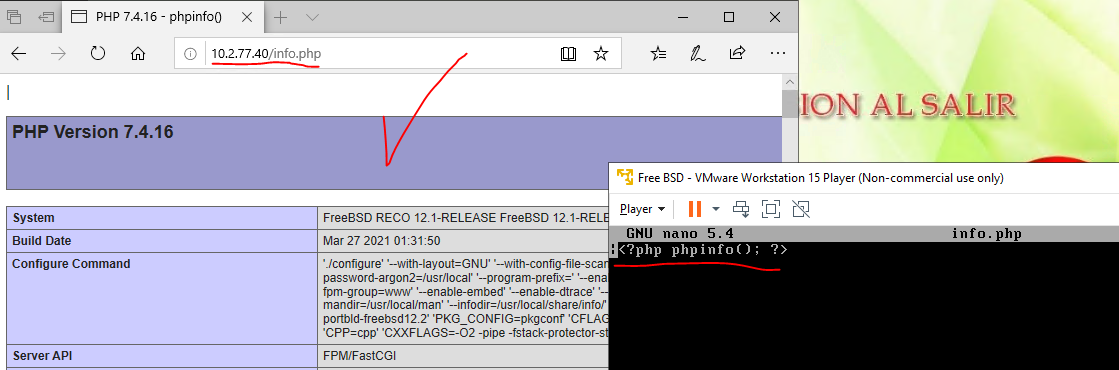


### Configuración Nginx

Una vez allí vamos a **nginx.conf** y des comentamos las líneas de código para la conexión php y añadimos datos del directorio root



Por ultimo probamos la configuración con una prueba llamada ‘**info.php’** el cual nos confirma que la configuración se ha realizado satisfactoriamente.



## Instalación y configuración de PHP para PostgresSQL en Slackware con un servidor Apache

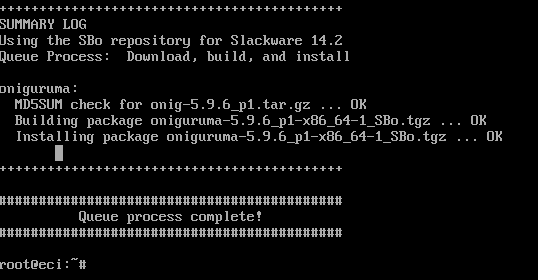
Como deseamos instalar la versión más reciente ,eliminamos el paquete de PHP5 que ya viene instalado en Slackware .

Descargamos las versiones 2.4.46 para apache y la versión de php 8.0.3 desde los sitios web oficiales y procedemos a descomprimir

### Instalación PHP

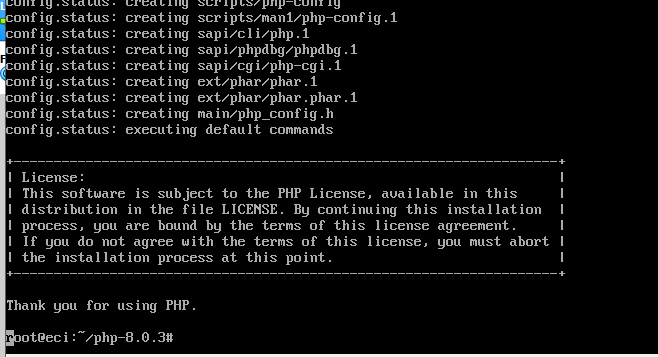
Nos dirigimos a la carpeta de php una vez ya descomprimida y procedemos a actualizamos primero el paquete de expresiones regurales del sistema para no tener complicaciones con la instalación





Luego configuramos los complementos a necesitar no olvídelos añadir la etiqueta **–with-pdo-pgsql** ya que vamos a necesitar los complementos en este caso para PostgresSQL.

./configure --with-mysqli --with-apxs2=/usr/local/httpd/bin/apxs --enable-gd --enable-pdo --with-zlib --with-pdo-sqlite --with-pdo-mysql **–with-pdo-pgsql** --enable-opcache --enable-mbstring --enable-ftp --enable-sockets --with-openssl --with-curl



Hacemos make y make install y tendremos PostgresSQL instalado

### Configuracion PHP

En /usr/local/lib/php.ini des comentamos los complementos que instalamos y por ultimo indicamos al servidor Apache que todo archivo .php o con contenido php será procesado por php con los siguientes comandos:

AddType application/x-httpd-php .php AddType application/x-httpd-php .html

Grabamos e iniciamos el php

/et/rc.d/rc.httpd start

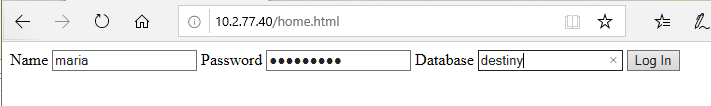
Para verificar instalacion probamos con un archivo <?phpinfo()?> en htdocs y verificamos su funcionamiento en el navegador

## Mostrando información de la base de datos

Una vez instalado php procedemos a crear un archivo html y php para mostrar los datos de una tabla en especifico para ello primero establecemos la conexión con la base de datos a través de mysqli() y pg\_connect() para postgrestql

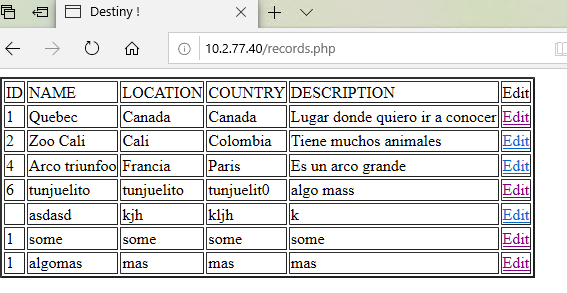
En este caso realizamos un script con la configuración inicial de conexión el cual vamos a llamar cada vez que hagamos un cambio de página de la página web

-



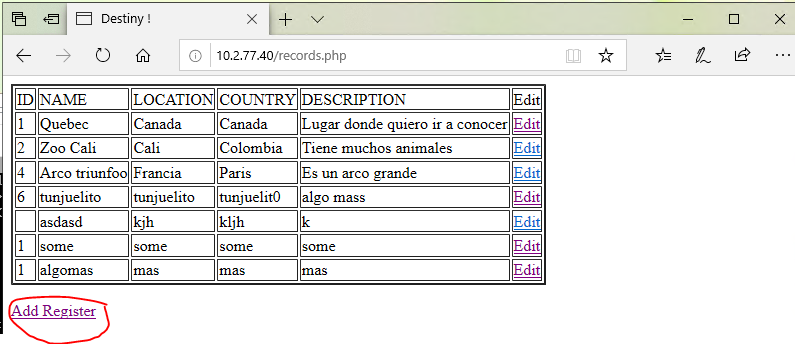
Una vez establecida creamos un archivo en este caso llamado records.php con el cual buscamos en los datos de la base de datos en la etiqueta <?php ?> y se mostraran en pantalla conforme vaya encontrando registros en la tabla.

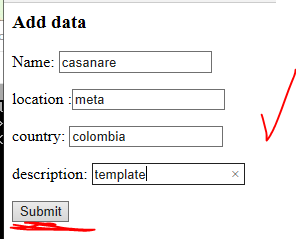
En este caso mostraremos la información relacionada a la tabla **info**.



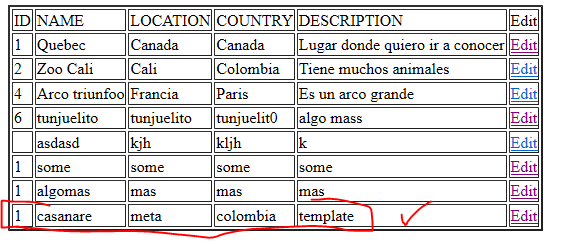
## Añadiendo información a la base de datos

Para añadir un registro lo que hacemos es crear un acceso desde la página principal a otra en donde haremos el resto de los datos en la tabla **info**  .



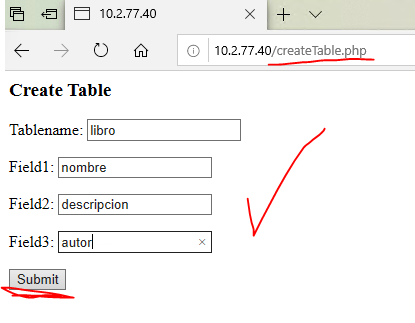


Una vez ingresado los datos ésta pagina nos redireccionará a la pagina principal donde podemos observar el registro hecho en la base de datos

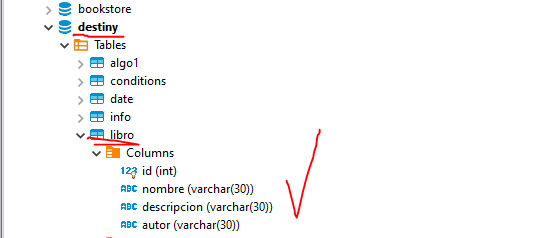


## Creando Tablas en la base de datos

Para crear una nueva tabla, se creó un link en la página principal en la cual nos re direccionará a una página de configuración en donde nos dejará crear una tabla nueva



Una vez ingresados los datos podemos corroborar esta información. En este caso vamos a acceder al dbeaver y ver la tabla creada.



## Eliminando Tablas de la base de datos

Para eliminar una tabla de la base de datos creamos una nueva hoja de configuración en donde proporcionamos el nombre de la tabla a eliminar y una vez eliminada nos redirigirá a la página principal

Esta vez procederemos a eliminar la tabla **libro P**reviamente creada

# 

# 

# 

# Conclusiones

En conclusión, se pudo identificar como se puede acceder a la informacion de la base de datos desde un una pagina web usando los servidores de MariaDB y PostgresSQL, permitiendonos conocer cómo se puede acceder desde cualquier dispositivo a las aplicaciones y datos de una organización, modificar la información almacenada en las bases de datos, eliminarla o editarla por medio de Scripts simples en PHP.

# Bibliografía

* Josedomingo.org(3 Ago. 2019 ) Introducción a PHP-FPM .Recuperado de:<https://www.josedomingo.org/pledin/2017/11/introduccion-a-php-fpm/>
* Kinsta.(12 abr.2021) ¿Qué Es Nginx y Cómo Funciona?. Recuperado de:<https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>
* CarlLinuxConsole09323(12 Feb 2019) Cómo instalar Nginx, MariaDB y PHP Stack (FEMP) en FreeBSD. Recuperado de:<https://es.linux-console.net/?p=569>
* Aaditya . (2017/02/18) howtos:databases:postgresql .Recuperado de:<https://docs.slackware.com/howtos:databases:postgresql>
* Jelastic(2021). Connection to MySQL/MariaDB for PHPRecuperado de:<https://docs.jelastic.com/connection-to-mysql-php/#connect-db>
* Oscar Meza(13 May, 2020) Como instalar PHP 7 en Linux Slackware.Recuperado de:<https://www.vidagnu.com/como-instalar-php-7-en-linux-slackware/>