

“IMPLEMENTACIÓN DE UN SERVIDOR L.A.M.P. PARA VISUALIZACIÓN DE DATOS - RASPBERRY S.O.”

Informe N°12

Laboratorio de Sistemas Embebidos

Melanny Dávila

*Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec*

2nd Jonathan Álvarez

*Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
jonathan.alvarez@epn.edu.ec*

Abstract—En el presente informe se analizarán los beneficios y aplicaciones de un servidor L.A.M.P, luego se describirán cuales fueron las complicaciones mas significativas que se tuvo para la ejecución del servidor y como se solucionaron dichos inconvenientes, finalmente se hablará de como es posible hacer una instancia en la nube AWS para guardar la base de datos en la nube

Index Terms—Raspberry, PHP, Mariadb, Apache, L.A.M.P.

I. INTRODUCCIÓN

LAMP es un acónimo de “Linux, Apache, MySQL y PHP”, es decir, las cuatro tecnologías que conforman esta plataforma que corre desde el lado del servidor.

Gracias a LAMP se puede crear sitios web, aplicaciones, realizar testing de páginas dinámicas y estáticas, entre muchas otras cosas más. En inglés se lo conoce como LAMP stack, es decir, una ampliación de servicios y tecnologías que nos permiten una gracias a la otra, conformar la plataforma que necesitamos.

II. OBJETIVOS

- Relacionar al estudiante con el uso de las aplicaciones de un servidor L.A.M.P.
- Realizar un conjunto de configuraciones e instalaciones de servicios, que permitan desarrollar una página web simple, para mostrar al usuario datos ambientales [3].

III. CUESTIONARIO

A. *Analizar los beneficios y aplicaciones de un servidor L.A.M.P.*

En sus beneficios se encuentra:

- **Desarrollo y sencillez de implementación:** Las aplicaciones web potentes se pueden construir mediante un servidor L.A.M.P. con un código simple y fácil de modificar o ampliar la aplicación según los requisitos del negocio. La mayoría de los servicios de alojamiento proporcionan entornos estándar basados en LAMP y se

pueden implementar fácilmente sin tarifas de licencia y a través de diversas distribuciones de Linux como Debian, Fedora, etc.

- **Flexibilidad:** A diferencia de otros proveedores de tecnología, L.A.M.P. no limita sus opciones de desarrollo. Ofrece una flexibilidad completa para construir e implementar aplicaciones teniendo en cuenta tus necesidades comerciales únicas.
- **Personalización:** Como los componentes de L.A.M.P. son de código abierto, proporcionan excelentes características de personalización con una amplia gama de módulos y funciones adicionales.
- **Seguridad:** Su implementación es segura y estable. Tiene una poderosa característica de seguridad para mitigar los ataques a vulnerabilidades y, si se produce algún error, se puede solucionar rápidamente en un sentido económico viable.
- **Comunidad de apoyo:** Hay un gran número de personas experimentadas en su uso, las mismas que pueden ofrecer apoyo rápido durante la fase de desarrollo, implementación, etc.
- **Económicamente inteligente:** En comparación con otros paquetes de software, L.A.M.P. es económicamente inteligente, ya que se puede adquirir a un precio relativamente bajo.

Como se menciono anteriormente, este servidor ofrece una gran cantidad de opciones de implementación debido a su flexibilidad, por lo que en sí, las aplicaciones de este servidor están limitadas a las capacidades de quien este a cargo de manejarlo, pero por hacer mención a algunas aplicaciones se puede decir que este servidor puede funcionar como una plataforma pequeña para una empresa, guardaría la información de sus clientes y presentaría sus servicios por medio de una página web. A su vez se puede usar este servidor para el monitoreo de variables ambientales en varios puntos y almacenar estos datos en una página web designada.

B. Describir cuáles fueron las complicaciones más significativas que se tuvo para ejecutar el servidor L.A.M.P., y describir las líneas de código utilizadas para solventar los problemas presentados.

Durante la implementación del servidor, uno de los principales problemas que se tuvo fue el instalar phpmyadmin. Normalmente, esta herramienta resultaría fácil de implementar en diversas distribuciones de Linux, con la aplicación del siguiente comando:

```
1 sudo apt-get install phpmyadmin
```

Pero, para la distribución de Raspbian Buster, ya que todavía es nueva, no existen los repositorios pertinentes dentro del sistema operativo, por lo que se procedió a instalar una versión que se obtuvo mediante la descarga de una página web. Posterior a esto se crearon algunos archivos de configuración para implementar esta herramienta, cada uno de sus campos se llenó de acuerdo a lo especificado en una guía. A continuación se indican los cambios mencionados:

- Descarga de la herramienta:

```
1 wget https://files.phpmyadmin.net/phpMyAdmin
  /4.9.0.1/phpMyAdmin-4.9.0.1-all-languages.
  zip
```

Script 1. Descarga del archivo

- Descompresión del archivo y envío de lo obtenido al directorio /usr/share/ creando a la vez un directorio que contenga a la aplicación.

```
1 unzip phpMyAdmin-4.9.0.1-all-languages.zip
2 sudo mv phpMyAdmin-4.9.0.1-all-languages /usr/
  share/phpmyadmin
```

Script 2. Creación del directorio

- Creación de un usuario propietario del servidor web:

```
1 sudo chown -R www-data:www-data /usr/share/
  phpmyadmin
```

Script 3. Creación del usuario propietario

- Instalación de los módulos de PHP requeridos y recomendados:

```
1 sudo apt install php-imagick php-phpseclib php-
  php-gettext php7.3-common php7.3-mysql php7
  .3-gd php7.3-imap php7.3-json php7.3-curl
  php7.3-zip php7.3-xml php7.3-mbstring php7
  .3-bz2 php7.3-intl php7.3-gmp
```

Script 4. Instalación de los módulos

- Edición del archivo de configuración:

```
1 # phpMyAdmin default Apache configuration
2
3 Alias /phpmyadmin /usr/share/phpmyadmin
4
5 <Directory /usr/share/phpmyadmin>
6     Options SymLinksIfOwnerMatch
7     DirectoryIndex index.php
8
9     <IfModule mod_php5.c>
10         <IfModule mod_mime.c>
11             AddType application/x-httpd-php .php
12         </IfModule>
13         <FilesMatch ".+\.php$">
```

```
14         SetHandler application/x-httpd-php
15     </FilesMatch>
16
17     php_value include_path .
18     php_admin_value upload_tmp_dir /var/lib /
  phpmyadmin/tmp
19     php_admin_value open_basedir /usr/share /
  phpmyadmin:/etc/phpmyadmin:/var/lib /
  phpmyadmin:/usr/share/php/php-gettext:/usr
  /share/php/php-gettext:/usr/share /
  javascript:/usr/share/php/tcpdf:/usr/share
  /doc/phpmyadmin:/usr/share/php/phpseclib /
  php_admin_value mbstring.func_overload 0
20 </IfModule>
21 <IfModule mod_php.c>
22     <IfModule mod_mime.c>
23         AddType application/x-httpd-php .php
24     </IfModule>
25     <FilesMatch ".+\.php$">
26         SetHandler application/x-httpd-php
27     </FilesMatch>
28
29     php_value include_path .
30     php_admin_value upload_tmp_dir /var/lib /
  phpmyadmin/tmp
31     php_admin_value open_basedir /usr/share /
  phpmyadmin:/etc/phpmyadmin:/var/lib /
  phpmyadmin:/usr/share/php/php-gettext:/usr
  /share/php/php-gettext:/usr/share /
  javascript:/usr/share/php/tcpdf:/usr/share
  /doc/phpmyadmin:/usr/share/php/phpseclib /
  php_admin_value mbstring.func_overload 0
32 </IfModule>
33
34 </Directory>
35
36 # Disallow web access to directories that don't
  need it
37
38 <Directory /usr/share/phpmyadmin/templates>
39     Require all denied
40 </Directory>
41 <Directory /usr/share/phpmyadmin/libraries>
42     Require all denied
43 </Directory>
44 <Directory /usr/share/phpmyadmin/setup/lib>
45     Require all denied
46 </Directory>
```

Script 5. Modificación del contenido del archivo

Posterior a esto, los problemas a solventar fueron fácilmente superados, entre ellos está la creación de la base de datos y la definición de variables dentro del mismo, ya que en el caso de la implementación de la lectura del código de barras, se planteó en un inicio leer las variables desde una base de datos, pero esta no podía almacenar espacios en blanco, por lo que se cambió la lectura de una base de datos a un archivo .txt ya que este almacena todo, una vez hecho esto, el código se simplificó y se pudieron leer las variables desde el mismo hacia la página web requerida.

Otro problema que surgió fue la utilización de MariaDB y Python debido a que no existía una librería en python en las librerías por defecto fue necesario utilizar un módulo para poder instalar la librería apropiada para el manejo de bases de datos mediante el script. para solucionar esto se utilizaron los siguientes códigos:

- Actualizar el modulo pip usando el siguiente comando

```
1 pip install -U pip
```

Script 6. Actualización de pip

- Una vez ingresado al modulo se instalan los paquetes que se desean instalar

```
1 pip install mysqlclient
```

Script 7. Instalación de los paquetes

- Para instalar el paquete de conector para python se ingresa el siguiente comando

```
1 pip install mysql-connector-python
```

Script 8. Instalación conector de Python

- Se instala el comando pymysql

```
1 pip install pymysql
```

Script 9. Instalación de pymysql

C. Describir detalladamente los procedimientos necesarios para crear una instancia en una nube de AWS y describir los cambios necesarios a realizar en los scripts de la parte práctica para guardar las dos bases de datos en la nube.

Para poder implementar este paso se seguirá los pasos mostrados en la documentación de AWS

Se debe ya tener una instancia usando Amazon Linux, con un nombre DNS público que se pueda acceder a través de internet. Se debe configurar un grupo de seguridad para permitir las conexiones SSH, HTTP y HTTPS. Se debe seguir los siguientes pasos

- 1) Conectarse con la instancia
- 2) Asegurarse que todos los paquetes estén actualizados, para esto se ejecuta los comandos de actualización

```
1 $ sudo yum update -y
```

Script 10. Comando de actualización

- 3) Con la instancia actualizada se debe instalar Apache web server, MySQL, y PHP

```
1 $ sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd
```

Script 11. Instalación LAMP en AWS

- 4) Iniciar el web server Apache

```
1 $ sudo service httpd start Starting httpd:
```

Script 12. Iniciar el servidor

- 5) Use el comando chkconfig para configurar el servidor Apache e inicie al iniciar el sistema

```
1 $ sudo chkconfig httpd on
```

Script 13. Comando para iniciar apache al inicio del sistema

- 6) Agregar una regla de seguridad para permitir conexiones en el puerto 80 de HTTP hacia la instancia.
- 7) Probar su servidor web en un buscador la dirección publica DNS de la instancia, se puede obtener al consultar la consola de Amazon EC2. Probar que el servidor

funcione adecuadamente al verificar en el directorio /var/www/html que exista la pagina de prueba de Apache

Para permitir permisos en los archivos,

- 1) Agregar el usuario al grupo este caso de ejemplo se usa el usuario ec2-user al grupo apache

```
1 [ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

Script 14. Agregar usuario al grupo apache

- 2) Salir y volver a ingresar de la consola comprobar que el usuario se encuentre en el grupo

```
1 [ec2-user ~]$ groups
2 ec2-user wheel apache
```

Script 15. Comprobar que el usuario se encuentre en el grupo

- 3) Cambiar la propiedad del directorio /var/www y su contenido al grupo apache

```
1 [ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

Script 16. Cambiar la posesión del directorio para el grupo apache

- 4) Para agregar permisos de escritura al grupo para directorios futuros se debe cambiar los permisos de directorio para /var/www y sus subdirectorios con el siguiente comando

```
1 [ec2-user ~]$ sudo chmod 2775 /var/www
2 [ec2-user ~]$ find /var/www -type d -exec sudo
   chmod 2775 {} \;
```

Script 17. Cambio de posesión para el grupo apache del directorio y sus subdirectorios

- 5) Para agregar permisos de grupo de manera recursiva cambie los permisos de /var/www y sus subdirectorios

```
1 [ec2-user ~]$ find /var/www -type f -exec sudo
   chmod 0664 {} \;
```

Script 18. Cambio de recursivo de permisos de escritura para el grupo apache del directorio y sus subdirectorios

- 6) El usuario y los futuros miembros del grupo apache pueden agregar, borrar o editar archivos en el directorio Apache

Para probar el servidor se siguen los siguientes pasos

- 1) Crear un archivo PHP en la carpeta Apache

```
1 [ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Script 19. Creación de documento PHP

- 2) En un navegador escriba la URL del archivo creado

```
1 http://my.public.dns.amazonaws.com/phpinfo.php
```

Script 20. URL del archivo creado

- 3) Se debería ver la pagina de información de PHP
- 4) Borrar el archivo de la pagina web, debido a que exponerlo sería inseguro

```
1 [ec2-user ~]$ rm /var/www/html/phpinfo.php
```

Script 21. Borrado del archivo de prueba

Otra preparación del servidor consiste en asegurar la instalación de MySQL como se lo hizo con el servidor LAMP local

1) Inicie el servidor web MySQL

```
1 [ec2-user ~]$ sudo service mysqld start
2 Initializing MySQL database:
3 ...
4
5 PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL
   root USER !
6 ...
7
8 Starting mysqld:
```

Script 22. Inicio de MySQL

2) Corra el script mysql_secure_installation

```
1 [ec2-user ~]$ sudo mysql_secure_installation
```

Script 23. Inicio del Script de instalación segura

3) Siga las instrucciones del Script para definir herramientas básicas de seguridad como contraseñas, usuarios, remover la base de datos de prueba y recargar los privilegios y tablas

Instalación de phpMyAdmin, como ya se vio es una herramienta de manejo de bases de datos, para poder instalar y configurar se siguen los siguientes pasos. Debemos tener en cuenta que se debe contar con SSL/TLS para poder transmitir las claves de manera segura

- 1) Ingresar en la instancia en AWS usando SSH
- 2) Instalar las dependencias necesarias

```
1 [ec2-user ~]$ sudo yum install php72-mbstring.
   x86_64 -y
```

Script 24. Instalar dependencias

3) Reiniciar Apache

```
1 [ec2-user ~]$ sudo service httpd restart
2 Stopping httpd: [ OK ]
3 Starting httpd:
```

Script 25. Reiniciar apache

4) Navegar al root de apache

```
1 [ec2-user ~]$ cd /var/www/html
2 [ec2-user html]$
```

Script 26. Navegar al root de Apache

5) Descargue directamente desde la página de phpMyAdmin a la instancia usando el comando wget

```
1 [ec2-user html]$ wget https://www.phpmyadmin.
   net/downloads/phpMyAdmin-latest-all-
   languages.tar.gz
```

Script 27. Descargando phpMyAdmin

6) Cree la carpeta phpMyAdmin y extraiga los contenidos del paquete con el siguiente comando

```
1 [ec2-user html]$ mkdir phpMyAdmin && tar -xvzf
   phpMyAdmin-latest-all-languages.tar.gz -C
   phpMyAdmin --strip-components 1
```

Script 28. Extracción de los archivos

7) Inicie el servidor MySQL usando:

```
1 [ec2-user ~]$ sudo service mysqld start
2 Starting mysqld:
```

Script 29. Inicio servicio MySQL

8) En un navegador ingrese el URL publica seguido de phpMyAdmin

```
1 http://my.public.dns.amazonaws.com/phpMyAdmin
```

Script 30. Ingreso a phpMyAdmin

Finalmente sería necesario copiar los archivos y subirlos a las carpetas correspondientes del servidor AWS

D. Conclusiones:

Jonathan Álvarez

- Se pudo comprobar que la implementación de un servidor L.A.M.P. se encuentra dentro del alcance de un estudiante, convirtiéndolo en una alternativa viable para sistemas de monitoreo y sistemas de registro para empresas pequeñas que requieran de estos servicios y no deseen contratar compañías de terceros.
- Los servidores L.A.M.P. permiten una integración completa entre diferentes servicios que permiten guardar en bases de datos mediante MySQL, procesar los datos o realizar acciones mediante scripts en Python en la plataforma Linux, utilizar servidores para guardar los archivos web mediante Apache y mostrar los datos en páginas web usando PHP
- Se puede utilizar servicios en la nube para poder acceder a los datos de nuestro servidor o manejar el servidor sin la necesidad de guardarlo localmente.

Melanny Dávila

- Un servidor L.A.M.P. es un sustituto apropiado de los paquetes comerciales y funciona como programas de software en capas que ofrecen una plataforma indispensable para desarrollar e implementar aplicaciones. Hay disponible una amplia gama de alternativas que son: L.N.M.P. o L.E.M.P., W.A.M.P., W.I.M.P., etc. Todos estos poseen principios similares para usar como una solución de código abierto completo, y el soporte para versiones fácilmente instalables con distribuciones de Linux.
- La implementación de una base de datos para el almacenamiento y manejo de la información recolectada, permite que se implemente aplicaciones aún más eficientes; debido a que el usuario puede presentar gran cantidad de información sin mayor problema y a su vez garantizado que dicha recolección es la correcta.
- A partir de esta clase de servidores se puede crear una interfaz fácil de manejar y leer para poder visualizar la información obtenida mediante sensores conectados a la placa Raspberry Pi en el internet.

E. Recomendaciones:

Jonathan Álvarez

- Consultar la documentación de los servicios que conforman parte del servidor cuando se cometan errores.

- Copiar y pegar los errores que salgan en la consola en algún buscador para verificar que se pueda hacer en caso de errores.

Melanny Dávila

- Así como esta clase de servidores existen varios que pueden ser instalados en Linux, por lo que pueden ser explorados para enfocarlos en una aplicación específica.
- Un lenguaje orientado a objetos que puede ser usado como última capa en esta clase de servidores es Python y es recomendable su uso debido a su gran capacidad.

REFERENCES

- [1] “Servidor LAMP: una solución económica para webs dinámica”, IONOS Digitalguide. <https://www.ionos.es/digitalguide/servidores/know-how/servidor-lamp-la-solucion-para-webs-dinamicas/> (accedido mar. 08, 2021).
- [2] E. Espinosa, E. Tatayo, “IMPLEMENTACIÓN DE UN SERVIDOR L.A.M.P. PARA VISUALIZACIÓN DE DATOS - RASPBERRY S.O.”. C.P. SISTEMAS EMBEBIDOS, Accedido: mar. 06, 2021. [En línea].
- [3] Tutorial: Install a LAMP web server on the Amazon Linux AMI - Amazon Elastic Compute Cloud. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html> (accedido mar. 11, 2021).