
	Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 1

INFORME DE LABORATORIO

INFORMACION BASICA					
ASIGNATURA:	Programación Web 2				
TITULO DE LA PRACTICA:	Docker				
NUMERO DE LA PRACTICA:	01	AÑO LECTIVO:	2024 - A	N° SEMESTRE:	III
FECHA DE PRESENTACION:	05 / 05 / 2024	HORA DE PRESENTACION:	23:59 PM		
INTEGRANTE (s): ▪ Huayhua Hillpa Yourdyy Yossimar				NOTA:	
DOCENTE (s): ▪ Dr(a). Richart Smith Escobedo Quispe					



1. Proyecto Final de Programación web I

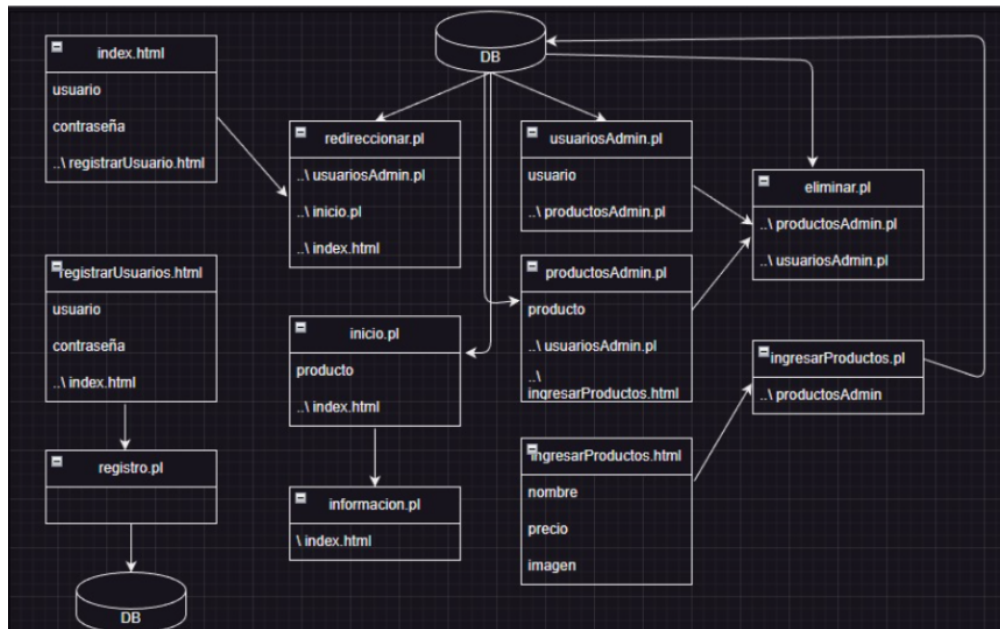
El proyecto de programación web I consiste en la creación de una página web que permita la gestión de una tienda de productos. En este caso la aplicación web estaba orientada a una tienda de lentes, en la cual se podía realizar las siguientes acciones:

- Registro de usuarios.
- Inicio de sesión.
- Visualización de productos.
- Administrar los pagina web (solo para el administrador).
- Comprar productos.

1.1. Estructura del proyecto

El proyecto se encuentra estructurado de la siguiente manera:

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 2



Para la elaboración del proyecto se utilizaron archivos Perl, para el envío de formularios, la escritura de documentos html y la conexión con la base de datos. Para el servidor local se utilizó una herramienta llamada Xampp.

XAMPP: es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser fácil de instalar y usar.

2. El uso de Docker

Para probar el funcionamiento del proyecto en docker, se instaló Docker Desktop. Luego se instaló la imagen de ubuntu para crear un contenedor para el proyecto.

Listing 1: Descargando la imagen de ubuntu

```
1 > docker pull ubuntu
```

Listing 2: Creando el contenedor ubu

```
2 > docker create --name ubu ubuntu
```

Listing 3: Creando el contenedor ubu



```
3 > docker create -p8080:80 --name ubu ubuntu
```

Listing 4: Iniciando el servidor

```
4 > docker start ubu
```

Listing 5: Ingresamos al contenedor

```
5 > docker exec -it ubu bash
```

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 3</p>

Listing 6: Instalando apache2 dentro del contenedor

```
6 $ apt-get update
7 $ apt-get install apache2
8 $ a2enmod cgi
```

Listing 7: Prendemos apache

```
9 $ service apache2 start
```

Listing 8: Se instala MariaDB y seguidamente lo activamos

```
10 $ apt-get install mariadb-server
11 $ service mariadb start
```

Listing 9: Instalamos neovim para la edicion de archivos

```
12 $ apt-get install neovim
```

Una vez que comprobamos que podemos usar la base de datos y el servidor apache, procedemos a copiar los archivos del proyecto al contenedor. Para ello se utilizo el comando **docker cp**.

Listing 10: Copiando los archivos al contenedor

```
13 > docker cp TrabajoFinalPweb1 ubu:/var/www/html
```

Listing 11: Ingresamos al contenedor

```
14 > docker exec -it ubu bash
```

Para que funcionen los archivos perl y cgi, tuve que mover los archivos a la carpeta /usr/lib/cgi-bin.

Listing 12: Moviendo los archivos

```
15 $ mv acciones/ comprar.cgi inicio.cgi registro.pl usuariosAdmin.pl catalogoAdmin.pl
    ingresarArticulos.html redireccionar.pl registroAdmin.pl /usr/lib/cgi-bin/
```

Luego en un archivo *.sql se insertaron los datos de la base de datos que tenia en XAMPP, para que luego los exporte a MariaDB en el contenedor.

Listing 13: Exportando la base de datos

```
16 $ mysql -u root -p < login.sql
```



Para configurar el el usuario y contraseña de la base de datos, se utilizo los siguientes comandos:

Listing 14: Abriendo la configuración para establecer el usuario y contraseña

```
17 $ service mariadb stop
18 $ mysqld_safe --skip-grant-tables --skip-networking &
19 $ mysql -u root
20 $ FLUSH PRIVILEGES;
```

Listing 15: Estableciendo el usuario y contraseña

```
21 $ ALTER USER 'root'@'localhost' IDENTIFIED BY 'fangtastic';
```

	<p>Universidad Nacional de San Agustín de Arequipa Facultad de Ingeniería de Producción y Servicios Escuela Profesional de Ingeniería de Sistemas</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 4</p>

Listing 16: Reiniciando el servicio

```

22 $ quit
23 $ service mariadb restart

```

Al momento de poner a flote la aplicación web las paginas con terminación .cgi no funcionaban, por que en cada una se utilizaba una libreria que se instala con CPAN la cual es CGI::Ajax. Para instalarla trate de utilizar el siguiente comando: instal CGI::Ajax, pero no funciona. Por lo que tuve que quitar parte de la funcionalidad de la pagina web.

Sin embargo aun queda una gran parte de ella. Esta se mostrara en el video de Youtube.

3. URL's

- Video de Youtube: https://youtu.be/qBstTw_4QxM

REFERENCIAS

- XAMPP: <https://www.apachefriends.org/es/index.html>
- Docker Desktop: <https://www.docker.com/products/docker-desktop/>