

Laboratorio 02

Tema: Git y GitHub

Estudiante: Huamani Condori Jeanpiero Sixto

1. Resolucion del laboratorio

1. Antecedentes

Para este lab es necesario conocer docker compose , como este automatiza la creacion de contenedores y de una red interna donde 2 o mas contenedores podran comunicarse entre sí.

2. Link del repositorio github

<https://github.com/jhuamaniCond/jhuamanicondori/tree/main/pw2-24a/lab02>

3. Link del codigo latex

<https://github.com/jhuamaniCond/jhuamanicondori/blob/main/pw2-24a/lab02/latex/lab02.tex>

4. Principales commits

La mayoría del código fuente es html,css y eso no tiene mucha ciencia .Por eso solo mostraremos los commits, códigos relacionados con docker y con el formulario que guarda consultas en una base de datos docker.

■ Commit 1

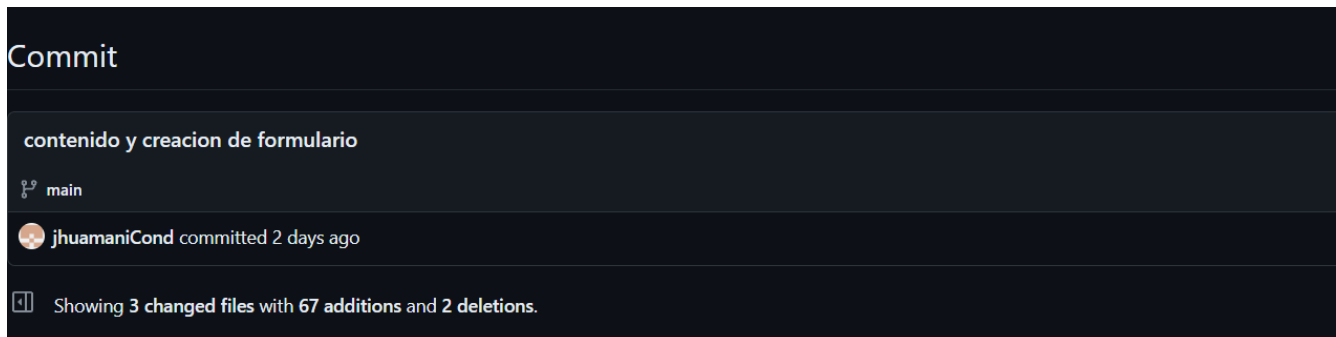


Figura 1: Commit form

En este commit se crea el formulario html para poder recibir consultas y mandarlas a la base de datos a traves de un script perl llamado agregar.pl

Archivo 1: contactarme.html

```
1 <form action="../../../cgi-bin/agregar.pl" method="get">
2   <div>
3     <label for="nombres">Nombres:</label>
4     <input type="text" id="nombres" name="nombres" required>
5   </div>
6   <div>
7     <label for="correo">Correo:</label>
8     <input type="email" id="correo" name="correo" required>
9   </div>
10  <div>
11    <label>Genero:</label>
```

```

12         <label for="masculino"><input type="radio" id="masculino" name="
           genero" value="masculino"> Masculino</label>
13         <label for="femenino"><input type="radio" id="femenino" name="
           genero" value="femenino"> Femenino</label>
14     </div>
15     <div>
16         <label for="nacimiento">Fecha de Nacimiento:</label>
17         <input type="date" id="nacimiento" name="nacimiento" required>
18     </div>
19     <div>
20         <label for="asunto">Asunto:</label>
21         <input type="text" id="asunto" name="asunto" required>
22     </div>
23     <div>
24         <label for="contenido">Contenido:</label>
25         <textarea id="contenido" name="contenido" rows="5" required></
           textarea>
26     </div>
27     <button type="submit">Enviar</button>
28
29 </form>

```

■ Commit 2

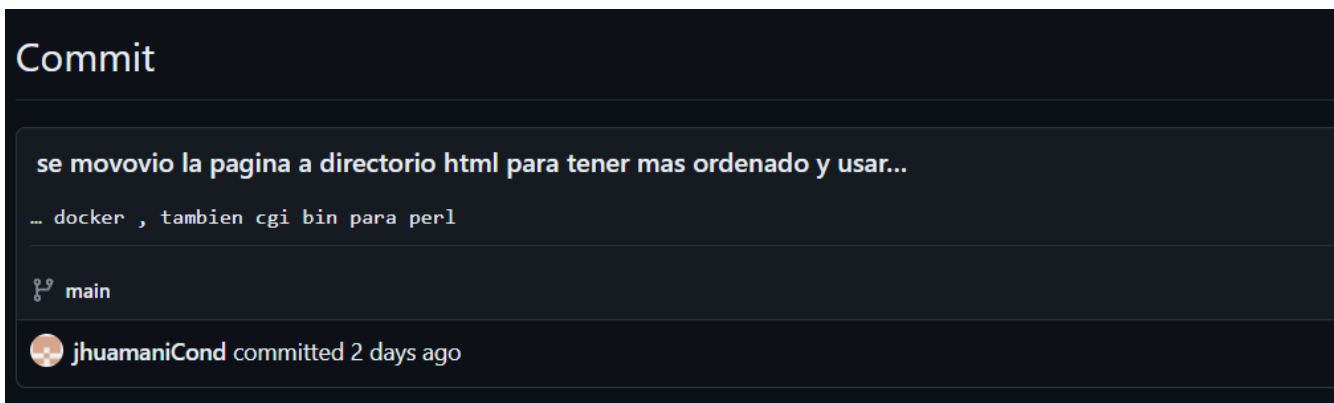


Figura 2: ordenar directorios y crear dockerfiles

Este commit es de los mas importantes , se crea el docker compose, los 2 dockerfiles para crear el servidor apache y mysql , tambien se crea el archivo createData.sql para iniciar las tablas necesarias del proyecto.

Archivo 2: compose.yaml

```

1     services:
2         apache:
3             build:
4                 context: .
5                 dockerfile: apache.dockerfile
6             container_name: apache-inventario
7             ports:
8                 - "80:80"
9             volumes:
10                 - ./html:/usr/local/apache2/htdocs/
11                 - ./cgi-bin:/usr/local/apache2/cgi-bin/
12             depends_on:
13                 - mysql
14
15         mysql:
16             build:

```

```

17         context: .
18         dockerfile: mysql.dockerfile
19         container_name: hc_lab2_sql
20         environment:
21             MYSQL_ROOT_PASSWORD: password
22             MYSQL_DATABASE: mydatabase

```

El archivo compose inicializa la red compuesta por apache y mysql ademas le indica a docker que para crear dichos contenedores se usara apache.dockerfile y mysql.dockerfile

Archivo 3: apache.dockerfile

```

1     # Usa la imagen base de Apache
2     FROM httpd:latest
3
4     # Instala el modulo de Perl para Apache y cpanminus
5     RUN apt-get update && apt-get install -y \
6         libapache2-mod-perl2 \
7         cpanminus \
8         nano \
9         gcc \
10        libdbd-mysql-perl \
11        && rm -rf /var/lib/apt/lists/*
12
13    # Instala el modulo CGI.pm utilizando cpanm
14    RUN cpanm CGI
15
16    # Establece los permisos adecuados para los scripts CGI
17    RUN chmod -R 755 /usr/local/apache2/cgi-bin/*
18    RUN sed -i '1s/^/LoadModule cgi_module modules/mod_cgi.so\n&/' /usr/
19        local/apache2/conf/httpd.conf
20    # Exponer el puerto 80
21    EXPOSE 80
22
23    # Ejecutar Apache en primer plano
24    CMD ["httpd-foreground"]

```

En el archivo apache.dockerfile se definen los comandos que se ejecutaran en el contenedor con RUN , se instalaran todas las librerias necesarias como perl ,cgi y se habilitara la ejecucion de scripts perl con Run sed

Archivo 4: mysql.dockerfile

```

1     FROM mysql:latest
2
3     # Copiar el script SQL a la carpeta /docker-entrypoint-initdb.d/
4     COPY createData.sql /docker-entrypoint-initdb.d/

```

En el archivo mysql.dockerfile se inidica la version de mysql y se copia el archivo .sql para iniciar las tablas necesarias

Archivo 5: createData.sql

```

1     CREATE TABLE IF NOT EXISTS Solicitudes (
2         nombres VARCHAR(255),
3         correo VARCHAR(255),
4         genero VARCHAR(255),
5         nacimiento VARCHAR(255),
6         asunto VARCHAR(255),
7         contenido VARCHAR(255)
8     );

```

creaeteData.sql contiene la ejecucion de comandos para crear las tablas necesarias del proyecto

■ Commit 3

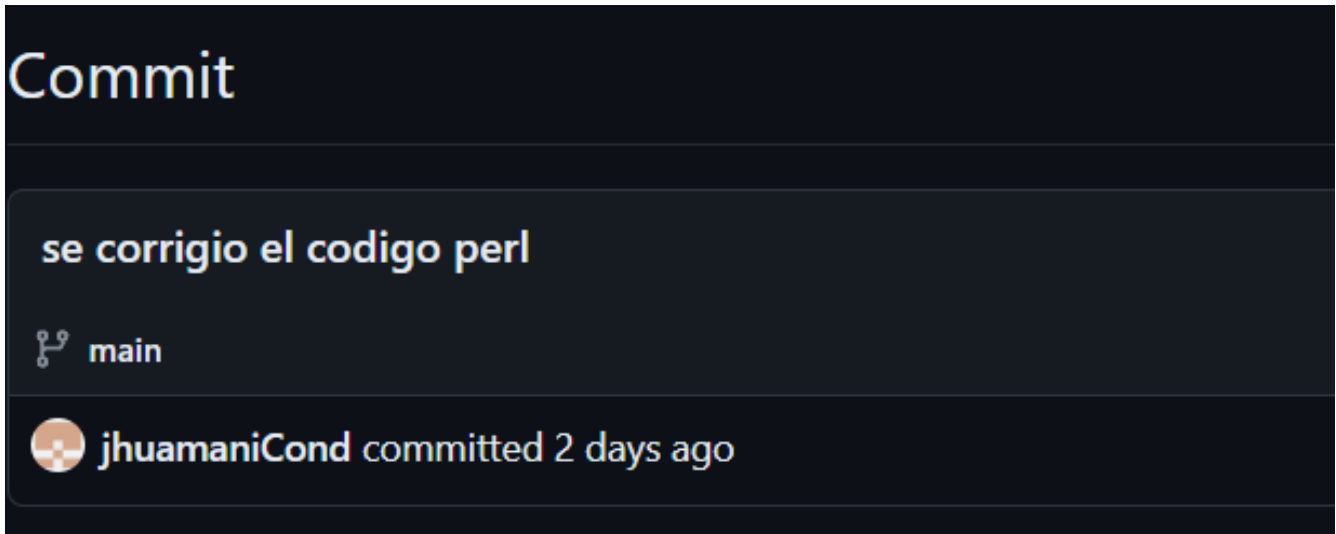


Figura 3: modificar archivo perl

En este commit se modifica el archivo perl para poder recibir la informacion enviada del form , conectarse a la base dedatos , y guardar esa informacion

Archivo 6: agregar.pl

```
1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4  use CGI;
5  use DBI;
6
7
8  my $q=CGI->new;
9  my $nombres=$q->param("nombres");
10 my $correo=$q->param("correo");
11 my $genero=$q->param("genero");
12 my $nacimiento=$q->param("nacimiento");
13 my $asunto=$q->param("asunto");
14 my $contenido=$q->param("contenido");
15
16 print "Content-type: text/html\n\n";
17 print <<HTML;
18 <!DOCTYPE html>
19 <html>
20 <head>
21
22     <link rel="stylesheet" href="/css/header.css">
23     <link rel="stylesheet" href="/css/resto.css">
24     <title></title>
25 </head>
26 <body>
27     <video autoplay loop class="back-video" >
28         <source src="/background.mp4" type="video/mp4">
29         <source src="/background.ogg" type="video/ogg">
30     </video>
31     <!-- Site navigation menu -->
32     <ul class="navbar">
33         <li><a href="/index.html">Home page</a>
34         <li><a href="/autor.html">Autor</a>
```

```

35         <li><a href="/estandaresWeb.html">Estandares Web</a>
36         <li><a href="/contactame.html">Contactame</a>
37     </ul>
38     <div class="content-container">
39         <div class="content">
40 HTML
41
42
43
44
45     if(!($nombres eq "") and !($correo eq "") and !($genero eq "") and !($nacimiento eq "") and !($asunto eq "") and !($contenido eq "")){
46
47         crearArticulo($nombres,$correo,$genero,$nacimiento,$asunto,$contenido);
48         print "<h2>el articulo se agrego correctamente</h2>";
49
50     }
51     else{
52         print "<h2>Nose pudo agregar el articulo</h2>";
53     }
54
55     sub crearArticulo{
56
57         my $nombres=$_[0];
58         my $correo=$_[1];
59         my $genero=$_[2];
60         my $nacimiento=$_[3];
61         my $asunto=$_[4];
62         my $contenido=$_[5];
63
64
65         my $user = 'root';
66         my $password1= 'password';
67         my $dsn = 'DBI:mysql:database=mydatabase;host=hc_lab2_sql';
68         my $dbh = DBI ->connect($dsn,$user,$password1) or die ("No se pudo conectar");
69
70
71         my $sth=$dbh->prepare("INSERT INTO Solicitudes(nombres,correo,genero,nacimiento,asunto,contenido) VALUES (?,?,?,?,?,?,?)");
72
73         $sth->execute($nombres,$correo,$genero,$nacimiento,$asunto,$contenido);
74         $sth ->finish;
75         $dbh->disconnect;
76     }

```

5. Imagenes de la pagina web y ejecucion

- Una vez creado los archivos dockerfile debemos ejecutar los siguientes comandos para que se armen los contenedores y nuestra pagina web funcione

```

PS C:\users\user\Downloads\pweb\jhuamanicondori\pw2-24a\lab02> docker compose up -d --build
time="2024-05-11T14:44:36-05:00" level=warning msg="C:\\users\\user\\Downloads\\pweb\\jhuamanicondori\\pw2-24a\\lab02\\compose.yaml: `vers
ion` is obsolete"
[+] Building 0.0s (0/0) docker:default
[+] Building 6.2s (7/7)
[+] Building 17.5s (17/17) FINISHED
=> [mysql internal] load build definition from mysql.dockerfile
=> => transferring dockerfile: 288B
=> [mysql internal] load metadata for docker.io/library/mysql:latest
=> [mysql internal] load .dockerignore
=> => transferring context: 2B
=> [mysql internal] load build context
=> => transferring context: 36B
=> [mysql 1/2] FROM docker.io/library/mysql:latest
=> CACHED [mysql 2/2] COPY createData.sql /docker-entrypoint-initdb.d/
=> [mysql] exporting to image
=> => exporting layers
=> => writing image sha256:0cd6f880b0923a47c90dd57c6182a2582930f1411e26ffd615e34a05bac7fe11
=> => naming to docker.io/library/lab02-mysql
=> [apache internal] load build definition from apache.dockerfile
=> => transferring dockerfile: 691B
=> [apache internal] load metadata for docker.io/library/httpd:latest
=> [apache auth] library/httpd:pull token for registry-1.docker.io
=> [apache internal] load .dockerignore
=> => transferring context: 2B
=> [apache 1/5] FROM docker.io/library/httpd:latest@sha256:36c8c79f900108f0f09fd4148ad35ade57cba0dc19d13f3d15be24ce94e6a639
=> CACHED [apache 2/5] RUN apt-get update && apt-get install -y libapache2-mod-perl2 cpanminus nano gcc libdbd
=> CACHED [apache 3/5] RUN cpanm CGI
=> CACHED [apache 4/5] RUN chmod -R 755 /usr/local/apache2/cgi-bin/*
=> CACHED [apache 5/5] RUN sed -i '1s/^/LoadModule cgi_module modules/mod_cgi.so\n&/' /usr/local/apache2/conf/httpd.conf
=> [apache] exporting to image
=> => exporting layers
=> => writing image sha256:a3f2b667759c320c1536d75f3a84c9c74f0ff4f95f489c3ad7f8bedd7455421f
=> => naming to docker.io/library/lab02-apache
[+] Running 3/3
  Network lab02_default      Created
  Container hc_lab2_sql      Started
  Container apache-inventario Started

```

Figura 4: Comando docker compose

Despues de dicho comando el servidor se iniciara automaticamente y estara listo para mostrar la pagina web ademas de para usar la base de datos

- Verificando si la base de datos esta funcionando Vamos a enviar datos a traves de la seccion **contactame**

```

mysql> use mydatabase;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_mydatabase |
+-----+
| Solicitudes           |
+-----+
1 row in set (0.00 sec)

mysql> select * from Solicitudes;
Empty set (0.00 sec)

```

Figura 5: mysql inicial

Home page Autor Estandares Web Contactame

Contactame

Nombres:

Correo:

Género: ☒ Masculino ☐ Femenino

Fecha de Nacimiento:

Asunto:

Contenido:

Figura 6: llenando el form

```
mysql> select * from Solicitudes;
Empty set (0.00 sec)

mysql> select * from Solicitudes;
+-----+-----+-----+-----+-----+-----+
| nombres | correo          | genero  | nacimiento | asunto | contenido          |
+-----+-----+-----+-----+-----+-----+
| carlos  | cherrera@gmail.com | masculino | 2002-01-11 | hola   | me gusto tu pagina web |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figura 7: mysql final

6. Estructura de directorios

```

jhuamanicond/
|-- pw2-24a
|   |-- lab01
|   |   |-- README.md
|   |   |-- createData.sql
|   |   |-- inventario
|   |   |   |-- cgi-bin
|   |   |   |   |-- agregar.pl
|   |   |   |   |-- prueba.pl
|   |   |   |   |-- retirar.pl
|   |   |   |   '-- verLista.pl
|   |   '-- html
|   |       |-- estilos
|   |       |   '-- style.css
|   |       |-- formulario.html
|   |       |-- img
|   |       |   |-- alpaca
|   |       |   |   |-- Nuevo\ Archivo\ WinRAR\ ZIP.zip
|   |       |   |   |-- chalina.jpg
|   |       |   |   |-- chompa.jpg
|   |       |   |   |-- img1.jpg
|   |       |   |   '-- medias.jpg
|   |       |   |-- index
|   |       |   |   |-- banner.jpg
|   |       |   |   |-- banner2.png
|   |       |   |   |-- banner3.jpg
|   |       |   |   |-- banner4.webp
|   |       |   |   |-- banner5.png
|   |       |   |   |-- banner6.png
|   |       |   |   |-- banner7.png
|   |       |   |   '-- fondo.gif
|   |       |   |-- inventario
|   |       |   |   |-- img1.jpg
|   |       |   |   |-- img1.png
|   |       |   |   '-- img2.png
|   |       |   |-- lana
|   |       |   |   |-- Nuevo\ Archivo\ WinRAR\ ZIP.zip
|   |       |   |   |-- chalina.jpg
|   |       |   |   |-- chompa.webp
|   |       |   |   |-- chullo.webp
|   |       |   |   '-- img1.jpg
|   |       |   '-- vicu\303\261a
|   |       |       |-- chalina.jpg
|   |       |       |-- chaqueta.png
|   |       |       |-- chullo.jpg
|   |       |       '-- img1.jpg
|   |       |-- index.html
|   |       |-- javascript
|   |       |   '-- verlista.js
|   |       '-- retirar.html
|   '-- ubuntu.dockerfile
'-- lab02
    |-- apache.dockerfile
    |-- cgi-bin
    '-- agregar.pl

```



```

|         |-- compose.yaml
|         |-- createData.sql
|         |-- html
|         |   |-- README.md
|         |   |-- autor
|         |   |   |-- galeria.html
|         |   |   |-- hobbies.html
|         |   |   '-- ingSoftware.html
|         |   |-- autor.html
|         |   |-- background.mp4
|         |   |-- contactame.html
|         |   |-- css
|         |   |   |-- autor.css
|         |   |   |-- header.css
|         |   |   '-- resto.css
|         |   |-- estandaresWeb.html
|         |   |-- imgs
|         |   |   |-- anime.png
|         |   |   |-- cat.jpeg
|         |   |   |-- codigo.jpg
|         |   |   |-- dog.jpeg
|         |   |   |-- mando.png
|         |   |   |-- playa.jpeg
|         |   |   '-- system.jpeg
|         |   '-- index.html
|         |-- latex
|         |   |-- imgs
|         |   |   |-- comit2.png
|         |   |   |-- comit3.png
|         |   |   |-- commit_form.png
|         |   |   |-- compose.png
|         |   |   |-- contactame.png
|         |   |   |-- database1.png
|         |   |   '-- database2.png
|         |   |-- lab02.tex
|         |   '-- lab02.pdf
|         '-- mysql.dockerfile

```

2. Pregunta

- Mencione tres aportes a su adquisición de conocimientos que este laboratorio le proporcionó.
 1. En este laboratorio aprendí a usar dockerfiles para automatizar los comandos al crear un contenedor docker además de copiar el proyecto directamente con el dockerfile sin necesidad de usar sftp.
 2. Aprendí a usar docker compose, que básicamente es un agrupador de contenedores para juntarlos en una red interna, además que también puedes indicar dockerfiles para que dichos contenedores se inicialicen.
 3. Aprendí que se puede inicializar una base de datos con un archivo .sql solo necesitas copiar dicho archivo a /docker-entrypoint-initdb.d/ y se inicializarán tus tablas automáticamente.

3. Autoevaluación

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o omisión)	4	✓	4	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	✓	3	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4	✓	3	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	✓	2	
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	✓	1	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	✓	3	
	Total			16	