

Informe de Laboratorio 5

Tema: Python

Nota

Estudiante	Escuela	Asignatura
Chambilla Perca Ricardo Mauricio rchambillap@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1701213

Laboratorio	Tema	Duración
5	Python	08

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 29 de Mayo 2024	Al 31 de Mayo 2024

1 Tarea

- URL GitHub de Tarea del Ajedrez: <https://github.com/rescobedoq/pw2/tree/main/labs/lab04/Tarea-del-Ajedrez>
- En esta tarea usted pondrá en práctica sus conocimientos de programación en Python para dibujar un tablero de Ajedrez.
- La parte gráfica ya está programada, usted sólo tendrá que concentrarse en las estructuras de datos subyacentes.
- Con el código proporcionado usted dispondrá de varios objetos de tipo Picture para poder realizar su tarea:

- 1.1 Una vez instalado python y creado un virtual environment para la resolución de este laboratorio nos fijamos en el repositorio de github compartido por el profesor y nos fijamos que una de las dependencias es el modulo de python pyGame, lo instalamos con pip y descargamos el código fuente.

```
pip3 install pygame
curl -LJO
  https://github.com/rescobedoq/pw2/blob/main/labs/lab04/Tarea-del-Ajedrez/chessPictures.py
curl -LJO
  https://github.com/rescobedoq/pw2/blob/main/labs/lab04/Tarea-del-Ajedrez/colors.py
```

```
curl -LJO
  https://github.com/rescobedoq/pw2/blob/main/labs/lab04/Tarea-del-Ajedrez/interpreter.py
curl -LJO
  https://github.com/rescobedoq/pw2/blob/main/labs/lab04/Tarea-del-Ajedrez/picture.py
curl -LJO
  https://github.com/rescobedoq/pw2/blob/main/labs/lab04/Tarea-del-Ajedrez/pieces.py
```

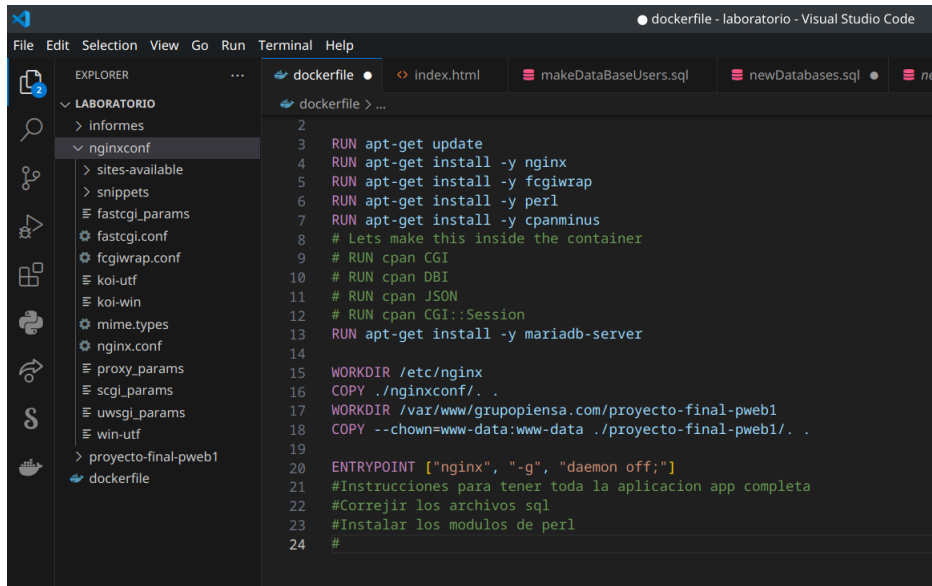
- Estos objetos estarán disponibles importando la biblioteca: `chessPictures` y estarán internamente representados con arreglos de strings que podrá revisar en el archivo `pieces.py`.
- La clase `Picture` tiene un sólo atributo: el arreglo de strings `img`, el cual contendrá la representación en caracteres de la figura que se desea dibujar.
- La clase `Picture` ya cuenta con una función implementada, no debe modificarla, pero sí puede usarla para implementar sus otras funciones:
- La clase `Picture` contará además con varios métodos que usted deberá implementar
 - `verticalMirror`: Devuelve el espejo vertical de la imagen
 - `horizontalMirror`: Devuelve el espejo horizontal de la imagen
 - `negative`: Devuelve un negativo de la imagen
 - `join`: Devuelve una nueva figura poniendo la figura del argumento al lado derecho de la figura actual
 - `up`: Devuelve una nueva figura poniendo la figura recibida como argumento, encima de la figura actual
 - `under`: Devuelve una nueva figura poniendo la figura recibida como argumento, sobre la figura actual
 - `horizontalRepeat`: Devuelve una nueva figura repitiendo la figura actual al costado la cantidad de veces que indique el valor de `n`
 - `verticalRepeat`: Devuelve una nueva figura repitiendo la figura actual debajo, la cantidad de veces que indique el valor de `n`

Luego paso a realizar las implementaciones, esto se realiza de manera rápida, ya que estos ejercicios son similares a los que hemos visto en FP1 en java

```
RUN apt-get install -y mariadb-server
```

1.2 Cree un usuario `pweb2` con contraseña: `12345678`. Otorgue permisos al usuario para acceder a la aplicación web. (Read-Write)

Esta operación se realizó a través del terminal del contenedor una vez creado la imagen y corriendo. En resumen el `Dockerfile` que utilicé para crear la imagen se ve tal que así:



```

2
3 RUN apt-get update
4 RUN apt-get install -y nginx
5 RUN apt-get install -y fcgiwrap
6 RUN apt-get install -y perl
7 RUN apt-get install -y cpanminus
8 # Lets make this inside the container
9 # RUN cpan CGI
10 # RUN cpan DBI
11 # RUN cpan JSON
12 # RUN cpan CGI::Session
13 RUN apt-get install -y mariadb-server
14
15 WORKDIR /etc/nginx
16 COPY ./nginxconf/. .
17 WORKDIR /var/www/grupopiensa.com/proyecto-final-pweb1
18 COPY --chown=www-data:www-data ./proyecto-final-pweb1/. .
19
20 ENTRYPOINT ["nginx", "-g", "daemon off;"]
21 #Instrucciones para tener toda la aplicacion app completa
22 #Corregir los archivos sql
23 #Instalar los modulos de perl
24 #
  
```

Una vez adentro del terminal del contenedor puedo comenzar los procesos de MariaDB así como el servicio de fcgiwrap el cual me permite ejecutar cgi en servidores estáticos administrados por nginx.

```

docker run -dp 8000:80 nginxserver
docker exec -it mystifying_ellis /bin/bash
root@1a30749290f4:/var/www/grupopiensa.com/proyecto-final-pweb1# /etc/init.d/mariadb
start
* Starting MariaDB database server mariabdb

[ OK ]
root@1a30749290f4:/var/www/grupopiensa.com/proyecto-final-pweb1# /etc/init.d/fcgiwrap
start
* Starting FastCGI wrapper fcgiwrap

[ OK ]
root@1a30749290f4:/var/www/grupopiensa.com/proyecto-final-pweb1# mysql
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.6.16-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database pweb1
-> ;
Query OK, 1 row affected (0.001 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON pweb1.* TO 'alumno'@'%' IDENTIFIED BY 'pweb1';
Query OK, 0 rows affected (0.015 sec)

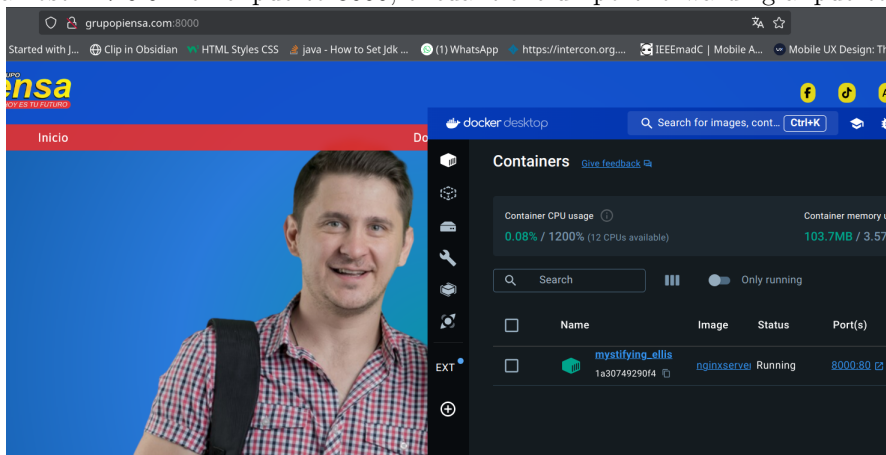
MariaDB [(none)]> exit
Bye
  
```

1.3 Finalmente implemente el trabajo final del curso de pw1 en ese contenedor.

Elabore un informe paso a paso para donde explique funcionalmente el proyecto demostrando que se trata de un contenedor docker.

Adjunte la URL de un video donde muestre que se trata de un contenedor Docker.

Entonces para probar mi contenedor de docker que este correctamente sirviendo los servidores de nginx voy a entrar al buscador de firefox y buscar la pagina web grupopiensa, el cual esta mapeado al localhost 127.0.0.1 en el puerto 8000, el cual tiene un port forwarding al puerto 80 del contenedor



Aqui dejo el enlace al video de YT en el que se muestra la veracidad del uso de Docker para hostear paginas web: <https://youtu.be/ERklT6IMV7E>

2 Rúbricas

2.1 Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

2.2 Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25%	En Proceso 50%	Satisfactorio 75%	Sobresaliente 100%
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	0	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas.	4	X	4	
Total		20		16	

3 Referencias

- https://drive.google.com/file/d/19hjCsMtViypEj3c0PRF_cflH1r8r9vBn/view?usp=sharing