

# Informe de Laboratorio 01 **Tema: DOCKER**

Nota				
	1			
	I			
	I			

Estudiante	Escuela	Asignat ura
- Antony Juan Tapia	Escuela Profesional de	Programación Web 2
Huamantuma	Ingeniería de Sistemas	Semestre: I

Laboratorio	Tema	Duración
01	DOCKER	04 horas

Sem est reacadémico	Fecha de inicio	Fecha de entrega
2024 - A	Del 28 Abril 2024	Al 11 Mayo 2024

### ENTREGABLE 1: Capturas de la Instalación y subida del 1. proyecto

### INTEGRANTES DEL GRUPO:

- Antony Juan Tapia Huamantuma
  - CAPTURAS DE LA INSTALACIÓN Y SUBIDA DEL PROYECTO

En primer lugar, lo que hacemos es crear un nuevo contenedor abriendo un total de tres puertos, esto haciendo una copia local de la imagen de Ubuntu, con el comando que se muestra a continuación, el proceso también se encuentra dentro del mismo.

Esto nos permite el utilizar el mismo contenedor, y en este mismo poder instalar todas las dependencias de nuestro proyecto para que tenga un funcionamiento de manera local, lo que nos permitirá el observar nuestro proyecto del semestre pasado.

:\Windows\system32>docker run --name pw2\_lab01 -p 8084:80 -p 8085:3306 -p 8086:22 -it ubuntu:20.04 /bin/bash Unable to find image 'ubuntu:20.04' locally

20.04: Pulling from library/ubuntu d4c3c94e5e10: Pull complete

Digest: sha256:874aca52f79ae5f8258faff03e10ce99ae836f6e7d2df6ecd3da5c1cad3a912b

Status: Downloaded newer image for ubuntu:20.04

oot@0bd18cd27e58:/# apt-get update





Después de eso, lo que hacemos es usar el comando \apt-get update" para poder realizar los comandos que nos serán útiles más adelante:

```
root@0bd18cd27e58:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1203 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
```

Seguidamente usamos el comando que se muestra a continuación para poder instalar el apache dentro de nuestro contenedor:

```
root@0bd18cd27e58:/# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

Seguidamente usamos el comando que se muestra a continuación para poder instalar el vim:

```
root@0bd18cd27e58:/# apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
   alsa-topology-conf alsa-ucm-conf libasound2 libasound2-data libcanberra0 libgpm2 libltdl7 libmpdec2 libogg0
   libpython3.8 libpython3.8-minimal libpython3.8-stdlib libreadline8 libtdb1 libvorbis0a libvorbisfile3
   readline-common sound-theme-freedesktop vim-common vim-runtime xxd
```





Ahora en este caso utilizaremos el mismo comando para MariaDB en este caso, para poder utilizar nuestras bases de datos dentro de nuestro contenedor de Docker, que es donde podremos colocar nuestro proyecto.

```
root@0bd18cd27e58:/# apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
    galera-3 gawk iproute2 libaio1 libatm1 libcap2 libcap2-bin libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
    libdbd-mysql-perl libdbi-perl libelf1 libencode-locale-perl libfgi-perl libhtm1-parser-perl libmn10 libmpfr6
    libmysqlclient21 libpam-cap libpopt0 libreadline5 libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl
    liburi-perl libwrap0 libxtables12 lsof mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common
    mariadb-server-10.3 mariadb-server-core-10.3 mysql-common rsync socat
    root@0bd18cd27e58:/# /etc/init.d/mysql start
    * Starting MariaDB database server mysqld
    root@0bd18cd27e58:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 38
```

Asimismo, configuramos el perl dentro de nuestro servidor local para poder dar funcionalidad a nuestros scripts cgi de nuestro proyecto.

```
root@0bd18cd27e58:/# apt-get install perl apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.41-4ubuntu3.17).
perl is already the newest version (5.30.0-9ubuntu0.5).
perl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@0bd18cd27e58:/# a2enmod cgid
Enabling module cgid.
```

Por último, instalamos git para poder clonar directamente nuestro repositorio del proyecto dentro de los espacios del servidor que estamos utilizando

```
root@0bd18cd27e58:/# apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
    git-man less libcbor0.6 libcurl3-gnutls liberror-perl libfido2-1 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6
    libxext6 libxmuu1 openssh-client patch xauth
```





Con ello resuelto, lo que hacemos es buscar la dirección del directorio HTML, y posicionarnos en el directorio que lo contiene, porque en ese será el lugar en el que clonaremos nuestro repositorio para poder observar nuestro proyecto pasado. En ese mismo lugar usamos el comando \git clone https://github.com/RyanValdivia/pweb1-trabajoFinal" para poder clonar el proyecto de nuestro proyecto

root@0bd18cd27e58:/# cd ~/../var/www/html/
root@0bd18cd27e58:/var/www/html# git clone https://github.com/ynoacamino/bancaFinal.git
Cloning into 'bancaFinal'...
remote: Enumerating objects: 485, done.
remote: Counting objects: 100% (485/485), done.
remote: Compressing objects: 100% (310/310), done.





Ahora lo que procede es el poder configurar la base de datos con la información que teníamos en nuestro directorio: Esto lo usamos utilizando el comando \mysql-u root -p" después de haber inicializado el sql dentro de nuestro contenedor, esto nos permitirá ingresar al root de la base de datos, y para poder importar nuestras tablas, primero creamos una base de datos general, en este caso con el nombre de \biblioteca" y después importamos con el comando \source" para poder importar nuestra información:

```
root@0bd18cd27e58:/# /etc/init.d/mysql start

* Starting MariaDB database server mysqld

* Starting MariaDB database server mysqld

Foot@0bd18cd27e58:/# mysql - u root -p

Enter password:

Welcome to the MariaDB monitor. Commands end with; or \g.

Your MariaDB connection id is 38

Server version: 10.3.39-MariaDB-0ubuntu0.20.04.2 Ubuntu 20.04

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)]> -- phpMyAdmin SQL Dump

MariaDB [(none)]> -- version 5.2.1

MariaDB [(none)]> -- servidor: 127.0.0.1

MariaDB [(none)]> -- Servidor: 127.0.0.1

MariaDB [(none)]> -- Servidor: 127.0.0.1

MariaDB [(none)]> -- Versinnn del servidor: 10.4.28-MariaDB

MariaDB [(none)]> -- Versinnn de PHP: 8.2.4

MariaDB [(none)]> CREATE DATABASE IF NOT EXISTS `bancafinal` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)] USE `bancafinal`;
la tabla `operarioDatabase changed

MariaDB [bancafinal]> START TRANSACTION;

MariaDB [bancafinal]> START TRANSACTION;
```





```
NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
MariaDB [bancafinal]>
               [bancafinal]>
 MariaDB
 MariaDB
               [bancafinal]> -- Estructura de tabla para la tabla `clientes`
               [bancafinal]> --
 MariaDB
MariaDB [bancafinal]> --
MariaDB [bancafinal]>
MariaDB [bancafinal]> CREATE TABLE `clientes` (
    -> `id` int(11) NOT NULL,
    -> `nombres` varchar(8) NOT NULL,
    -> `paterno` varchar(30) NOT NULL,
    -> `materno` varchar(30) NOT NULL,
    -> `nacimiento` date NOT NULL,
    -> `creado` timestamp NOT NULL DEFAULT current_timestamp(),
    -> `estado` tinyint(1) NOT NULL DEFAULT 1
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
ietOuery OK, 0 rows affected (0.287 sec)
 jetQuery OK, 0 rows affected (0.287 sec)
MariaDB [bancafinal]>
MariaDB [bancafinal]> --
MariaDB [bancafinal]> -- Volcado de datos para la tabla `clientes`
MariaDB [bancafinal]> --
MariaDB [bancafinal]> --
MariaDB [bancafinal]>
NMariaBB [bancafinal]> INSERT INTO `clientes` (`id`, `dni`, `nombres`, `paterno`, `materno`, `nacimiento`, `creado`, `esado`) VALUES
-> (1, '54587912', 'Christian Raul', 'Mestas', 'Zegarra', '2006-04-17', '2024-01-17 02:19:50', 1);
te_id`)Query OK, 1 row affected (0.055 sec)
MariaDB [bancafinal]>
MariaDB [bancafinal]> --
MariaDB [bancafinal]>
MariaDB [bancafinal]> --
 ADEMariaDB [bancafinal]> -- Estructura de tabla para la tabla `cuentas`
MariaDB [bancafinal]>
EMariaDB [bancafinaĺ]> --
AMariaDB [bancafinal]> -- Volcado de datos para la tabla `cuentas`
MariaDB [bancafinal]> --
MariaDB [bancafinal]>
MariaDB [bancafinal]> INSERT INTO `cuentas` (`id`, `usuario`, `clave`, `creado`, `estado`, `cliente_id`) VALUES
-> (1, 'cmestasz', '123456789', '2024-01-17 02:20:14', 1, 1);
Query OK, 1 row affected (0.055 sec)
```





Esto es todo lo que se necesita para poder correr nuestro proyecto dentro de docker, y para poder visualizarlo simplmente obserbamos los puertos que se utilizan dentro de nuestro desktop, en este caso usamos el puerto 8084 para poder visualizar nuestro proyecto, teniendo lo siguiente:

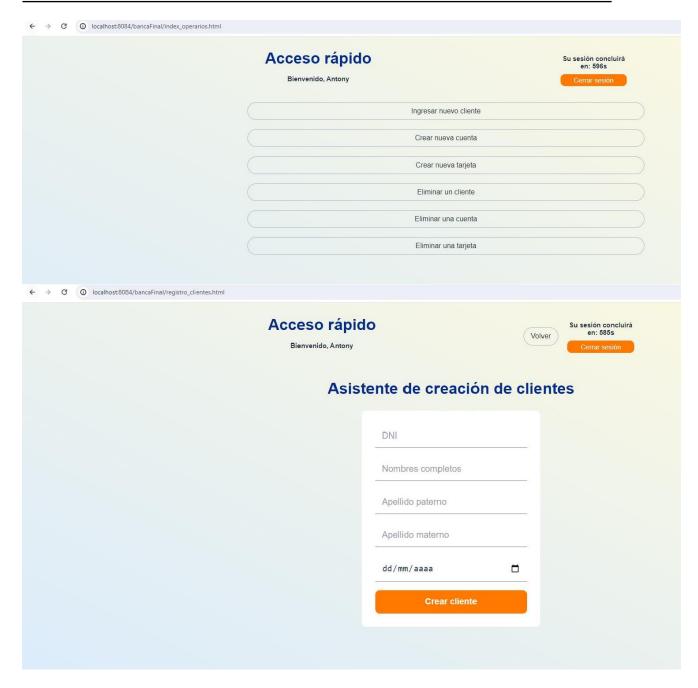


- CAPTURAS DE LA INICIALIZACIÓN DE AMBOS PROYECTOS:
- PROYECTO:



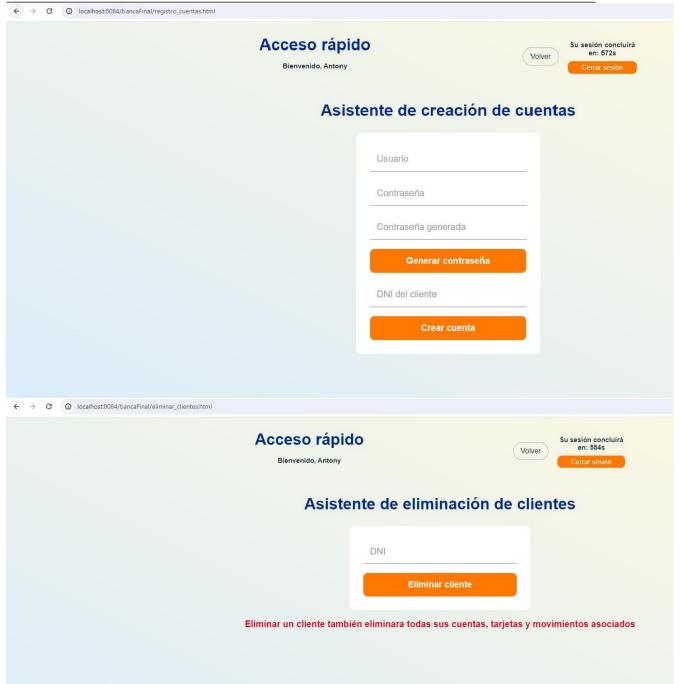






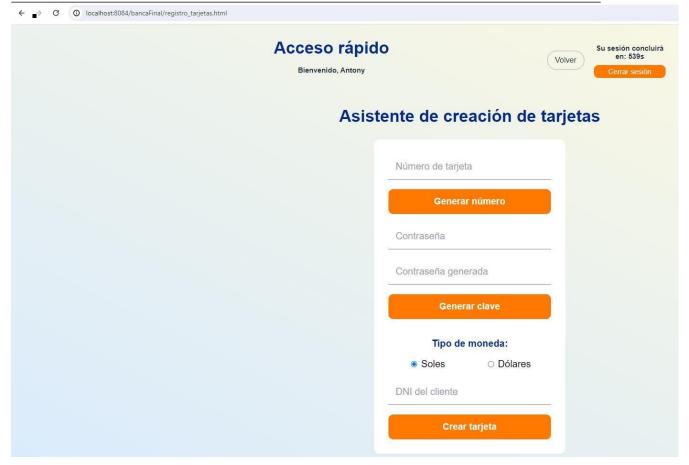
















## 2. ENTREGABLE 2: links de los materiales

Link del video(youtube) PROYECTO 1: https://www.youtube.com/watch?v= mfNkSx YW LI

- Link de Dockerhub PROYECTO 1: https://hub.docker.com/repository/doc ker/anthonyajra/proyectoweb/ge neral
- Comando de descarga de Dockerhub: https://hub.doc ker.c o m/r/anton ytapia/p we b2

# 3. Referencias

hhttps://www.docker.com/