



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



SERVICIOS SOBRE REDES

ASIGNATURA: Servicios sobre Redes
PROFESOR: Ing. Juan Pable Zaldumbide
PERÍODO ACADÉMICO: 2020A

INFORME

TÍTULO: DOCKER, HAMACHI – Prueba 2

ESTUDIANTES

Erick Bolaños
Estefanía Aguilar

FECHA DE REALIZACIÓN: 2020 / agosto / 20

FECHA DE ENTREGA: 2020 / agosto / 20

CALIFICACIÓN OBTENIDA:

FIRMA DEL PROFESOR:

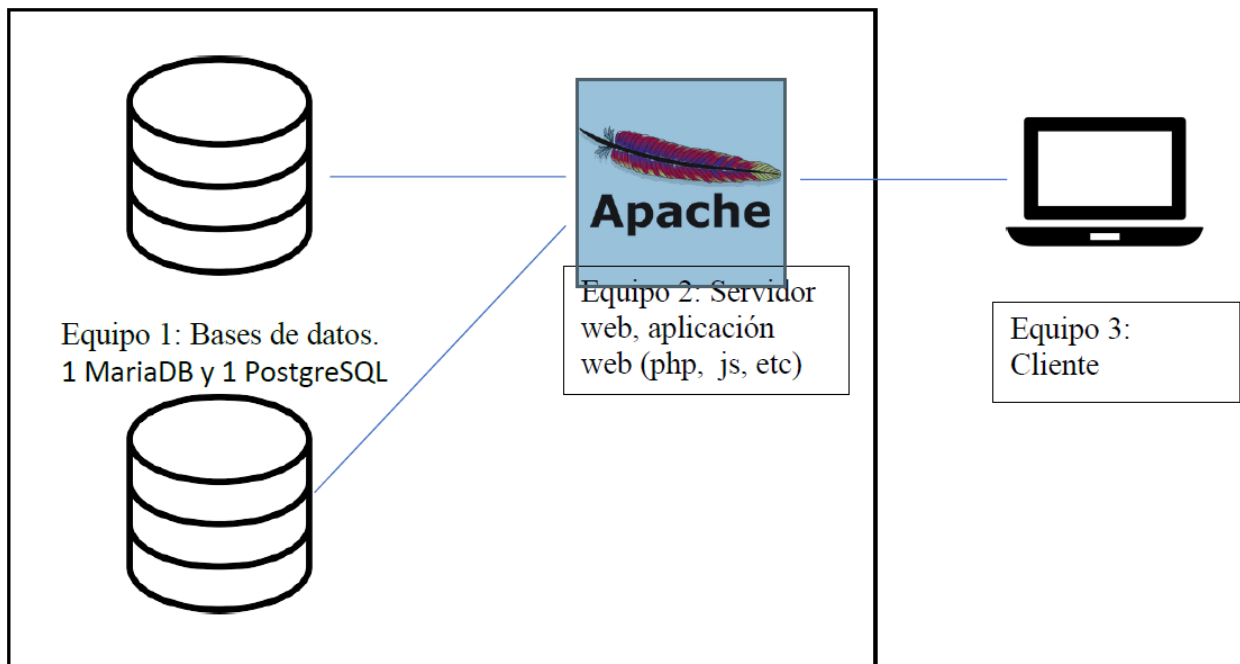
1 PROPÓSITO DE LA PRÁCTICA

Utilizando Docker y Hamachi, implementar la siguiente arquitectura y una aplicación web sencilla que consuma elementos de las bases de datos, documentar los pasos que siguió en un repositorio de github.

2 OBJETIVOS ESPECÍFICOS

Implementar la siguiente arquitectura con dos bases de datos, servidor web y un cliente.

3 DESARROLLO DE LA PRÁCTICA



Equipo 1: PostgreSQL y MariaDB

Utilizar el siguiente comando para encontrar los contenedores de cada una de las bases de datos, sudo Docker search [base de datos]. Seleccionar la versión del contenedor, para luego bajar la imagen con el comando sudo Docker pull [base de datos]

```
erick-linux@ubuntu:~$ sudo docker run --name mariadbtest -e MYSQL_ROOT_PASSWORD=mypass -d mariadb/server:10.3
[sudo] password for erick-linux:
a9f938ba1ae3cfb4c27907d7620d24b5304578839cbaa1e6edb15d9c3f53687f

erick-linux@ubuntu:~$ sudo docker run --name some-postgres -e POSTGRES_PASSWORD=postgres -d postgres
[sudo] password for erick-linux:
6f88c370c0ad31a5df2063a53b37c15912eb4561fd139419a64d578e1fb63bf2
```

Nota: ingresar el puerto correspondiente para cada contenedor, con `-p` o `-P`.

```
erick-linux@ubuntu:~$ sudo docker run -p 5432:5432 --name some-postgres -e POSTGRES_PASSWORD=postgres -d postgres
b0ba2ab917d185f98830936b8ef18f2d92b32a4d7c8835a56c3bfe3c4a5e28ec
docker: Error response from daemon: driver failed programming external connectivity on endpoint some-postgres (f452ee4d9face3974181d999497020234cabcf1045533e79163cfed04a89ceb3b9): (iptables failed: iptables --wait -t nat -A DOCKER -p tcp -d 0/0 --dport 5432 -j DNAT --to-destination 172.17.0.2:5432 ! -i docker0: iptables: No chain/target/match by that name.
(exit status 1)).
erick-linux@ubuntu:~$ docker run -p 3306:3306 --name mariadbtest -e MYSQL_ROOT_PASSWORD=mypass -d mariadb/server:10.3
d1081db42984188479afa53eb3f9ed20b9681dbe5094081e773eddf789ad52bb
docker: Error response from daemon: driver failed programming external connectivity on endpoint mariadbtest (f45085d97eef22b93837cba611c18a565ef64a0ffc5e1d312a53ed931eb08c6): (iptables failed: iptables --wait -t nat -A DOCKER -p tcp -d 0/0 --dport 3306 -j DNAT --to-destination 172.17.0.2:3306 ! -i docker0: iptables: No chain/target/match by that name.
(exit status 1)).
```

```
erick-linux@ubuntu:~$ docker inspect 3b32ff9c97b2 | grep -i tcp
"CPUsetCpus": "",
"KernelMemoryTCP": 0,
"5432/tcp": {}
"5432/tcp": null
erick-linux@ubuntu:~$ docker inspect 575814f5d8ab | grep -i tcp
"CPUsetCpus": "",
"KernelMemoryTCP": 0,
"3306/tcp": {}
"3306/tcp": null
```

```
erick-linux@ubuntu:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	latest	62473370e7ee	5 days ago	314MB
mariadb/server	10.3	16783b3dc9fc	9 days ago	346MB
hello-world	latest	bf756fb1ae65	7 months ago	13.3kB

Creamos el servidor para levantar el servicio.

```
1 <?php
2
3 $mysql = new mysqli("25.2.91.197", "root", "", "base1");
4
5 $resultados = $mysql->query("select * from datos");
6
7 echo "
8 <table>
9 <tr>
10 <th>Nombre</th>
11 <th>Apellido</th>
12 <th>Edad</th>
13 <th>Sexo</th>
14 </tr>";
15
16 foreach($resultados as $r){
17
18     echo "<tr> <td>";
19     echo $r["nombre"]."</td><td>";
20     echo $r["apellido"]."</td><td>";
21     echo $r["edad"]."</td><td>";
22     echo $r["sexo"]."</td><td>";
23
24 }
25
26 echo "</table>";
```

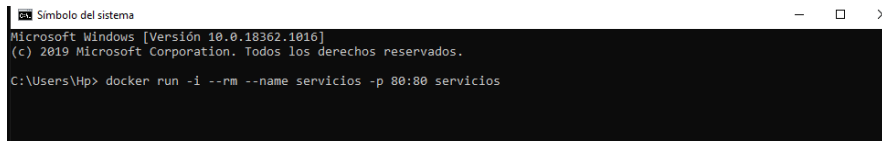
Este archivo se guardará dentro de un directorio, y dentro de este dentro de otro directorio llamado sites.

Luego abrimos la línea de comandos, y ejecutamos el siguiente comando para poder construir la imagen:

```
Microsoft Windows [Versión 10.0.18362.1016]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

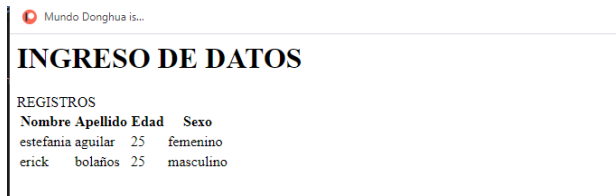
C:\Users\Hp> docker build -t servicios .
```

Ahora ejecutamos el siguiente comando para realizar la última configuración y levantar el servicio y pasar a la comprobación.

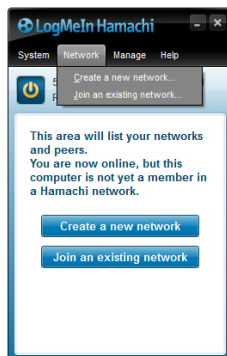


```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.1016]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Hp> docker run -i --rm --name servicios -p 80:80 servicios
```

Y el resultado es el siguiente.



El uso de Hamachi es necesario para que los ordenadores se encuentren en red:



4 CONCLUSIONES

El uso de los contenedores de Docker permite reducir el uso de memoria de tu disco físico aprovechando mejor sus capacidades.

5 BIBLIOGRAFÍA

- [1] IONOS, «Digital Guide,» 9 Julio 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/configuracion/tutorial-docker-instalacion-y-primeros-pasos/>. [Último acceso: 17 Agosto 2020].
- [2] B. Hogan, «Digital Ocean Community,» Digital Ocean, 26 Abril 2019. [En línea]. Available: <https://www.digitalocean.com/community/tutorials/como-instalar-y-usar-docker-en-ubuntu-18-04-1-es>. [Último acceso: 17 Agosto 2020].