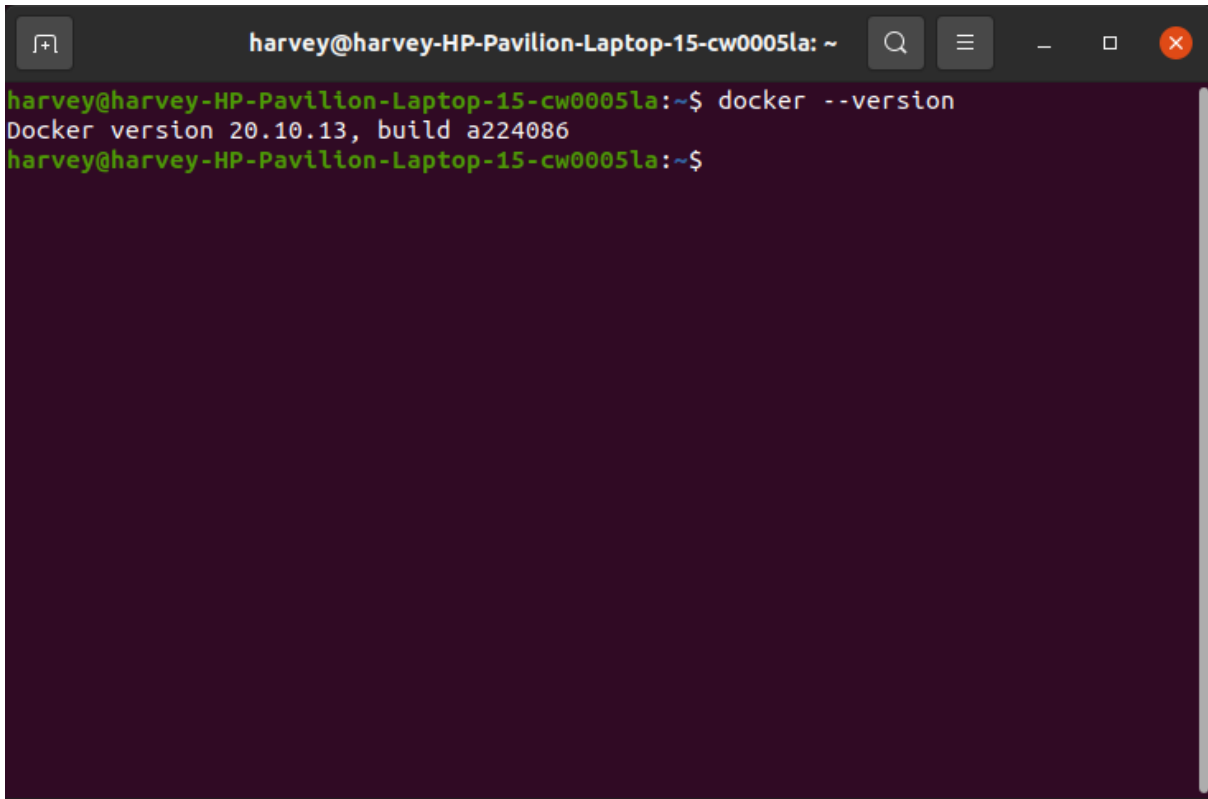


## GUÍA FINAL TERCER CORTE

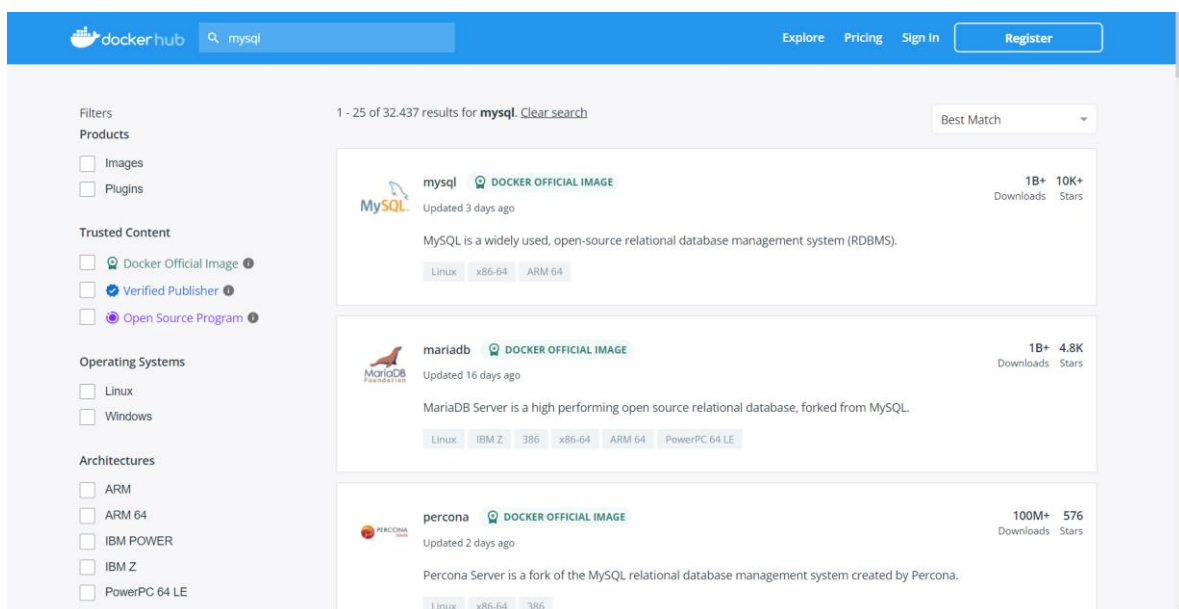
### APARTADO 1: DOCKER

1. Verificar que se tenga instalado docker, para ello ejecutamos el comando `docker --version`



```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ docker --version  
Docker version 20.10.13, build a224086  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$
```

2. Realizar la instalación de las imágenes necesarias, en este caso `mysql` en su versión 5.7 y `php` en su versión 7.0 junto con `apache`, esto por medio del comando `docker pull`. Recuerda que se utiliza el comando `docker pull` y para indicar la versión se utilizan los dos puntos.



```
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$ sudo docker pull mysql:5.7
5.7: Pulling from library/mysql
c32ce6654453: Pull complete
415d08ee031a: Pull complete
7a38fec2542f: Pull complete
352881ee8fe9: Pull complete
b8e20da291b6: Pull complete
66c2a8cc1999: Pull complete
d3a3a8e49878: Pull complete
172aabfba65c: Pull complete
fea17d0b1d1e: Pull complete
fff7f5411ca9: Pull complete
c33d43428e07: Pull complete
Digest: sha256:16e159331007eccc069822f7b731272043ed572a79a196a05ffa2ea127caaf67
Status: Downloaded newer image for mysql:5.7
docker.io/library/mysql:5.7
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$
```

The screenshot shows the Docker Hub search results for the query 'php'. The interface includes a top navigation bar with the Docker Hub logo, a search bar containing 'php', and links for 'Explore', 'Pricing', 'Sign In', and 'Register'. On the left, there are filter sections for 'Products' (Images, Plugins), 'Trusted Content' (Docker Official Image, Verified Publisher, Open Source Program), 'Operating Systems' (Linux, Windows), and 'Architectures' (ARM, ARM 64, IBM POWER, IBM Z, PowerPC 64 LE). The main content area displays '1 - 25 of 71.834 results for php'. Three results are visible: 1. 'php' (DOCKER OFFICIAL IMAGE) with 500M+ downloads and 6.6K stars, updated 2 days ago. 2. 'adminer' (DOCKER OFFICIAL IMAGE) with 100M+ downloads and 711 stars, updated a month ago. 3. 'composer' (DOCKER OFFICIAL IMAGE) with 50M+ downloads and 867 stars, updated 2 days ago. Each result includes a description and a list of supported architectures.

Image Name	Downloads	Stars	Updated	Description	Supported Architectures
php	500M+	6.6K	2 days ago	While designed for web development, the PHP scripting language also provides general-purpose use.	Linux, PowerPC 64 LE, IBM Z, mips64le, x86-64, ARM, ARM 64, 386
adminer	100M+	711	a month ago	Database management in a single PHP file.	Linux, IBM Z, x86-64, ARM, ARM 64, 386, PowerPC 64 LE
composer	50M+	867	2 days ago	Composer is a dependency manager written in and for PHP.	Linux, 386, DockerPPC 64 LE, IBM Z, x86-64, ARM, ARM 64

```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-P... x harvey@harvey-HP-P... x harvey@harvey-HP-P... x  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ sudo docker pull php:7.0-apache  
[sudo] contraseña para harvey:  
7.0-apache: Pulling from library/php  
177e7ef0df69: Pull complete  
9bf89f2eda24: Pull complete  
350207dcf1b7: Pull complete  
a8a33d96b4e7: Pull complete  
c0421d5b63d6: Pull complete  
f76e300fbe72: Pull complete  
af9ff1b9ce5b: Pull complete  
d9f072d61771: Pull complete  
37007e292198: Pull complete  
8ba923990f24: Pull complete  
98af8902979a: Pull complete  
f1548c2cd376: Pull complete  
e1062fd0605a: Pull complete  
Digest: sha256:1d34b2e491a02ba7a8d26478132015e197a5ffea37f0a93b42621d11cfe042cc  
Status: Downloaded newer image for php:7.0-apache  
docker.io/library/php:7.0-apache  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$
```

3. Se verifica que las imágenes se hayan descargado con el comando `docker images`.

```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-P... x harvey@harvey-HP-P... x harvey@harvey-HP-P... x  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$ docker images  
REPOSITORY    TAG          IMAGE ID      CREATED       SIZE  
mysql          5.7          a3d35804fa37 4 days ago   462MB  
php            7.0-apache  aa67a9c9814f 3 years ago   368MB  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la:~$
```

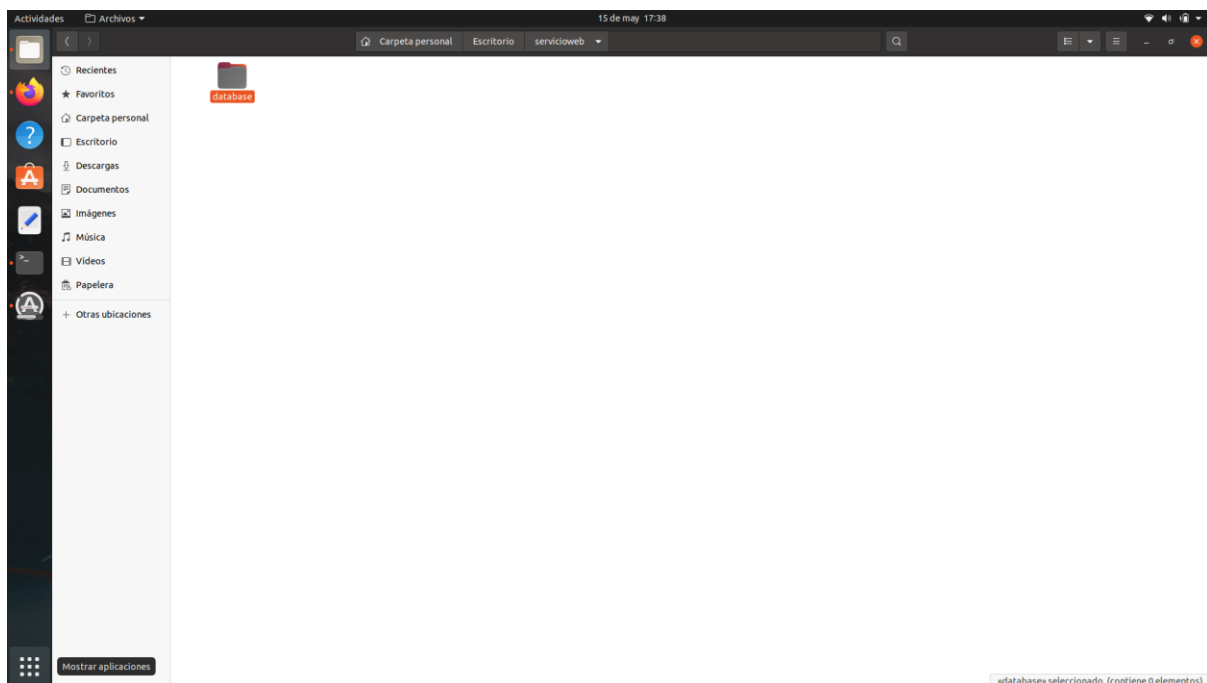
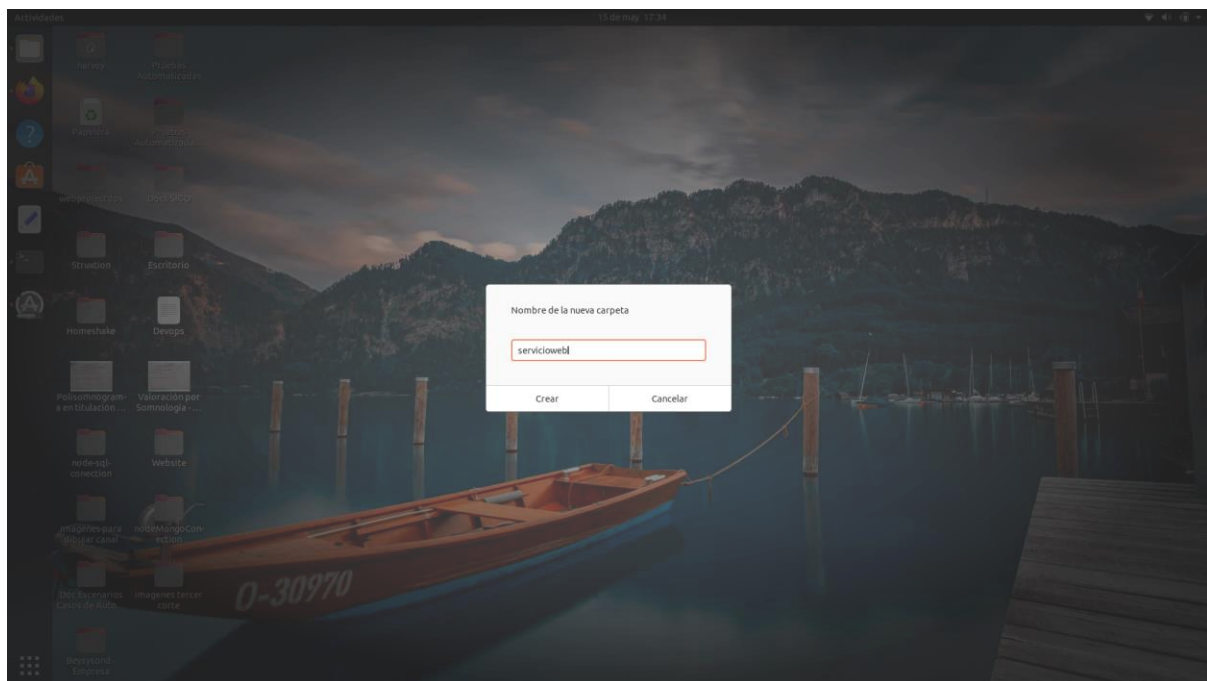
4. Una vez descargadas las imágenes corremos mysql por medio del comando `docker run` con las respectivas banderas o parámetros de nuestro servidor mysql y verificamos que el contenedor inicio por medio del comando `docker ps`. Las banderas son:
  - a) `-p`, es para indicar el puerto por el cual escuchará la aplicación y el puerto del contenedor.
  - b) `-name`, indica el nombre del proceso o contenedor que se iniciará.
  - c) `-v`, indica que se creará un volumen, permite almacenar los datos de la bd en una carpeta con el fin de salvar los datos en caso de que el contenedor se apague o se caiga. La primera ruta hace referencia a la carpeta que se creará en nuestro equipo (se ubicará en el escritorio), la segunda hace referencia a la dirección del contenedor (por defecto).
  - d) `-e`, indica la contraseña, puede ser cualquiera, en este caso `password`.
  - e) `-d`, para que mysql corra en segundo plano.

```

harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$ sudo docker run -p 3307:3306 --name basedatos -v /home/harvey/Escritorio/servicioweb/database:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=password -d mysql:5.7
fd46b0b7d30ef6ac0eb92ee1ac4a6a6ff33b7ea099182f46e38e3eb2d0290066
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS
NAMES
fd46b0b7d30e   mysql:5.7 "docker-entrypoint.s..." 12 seconds ago Up 8 second
s   33060/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3306/tcp   basedatos
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$

```

5. Dentro del escritorio creamos la carpeta llamada servicioweb en la que se alojarán los archivos de nuestra página web y dentro de ella otra carpeta llamada database para almacenar los datos de la base de datos.



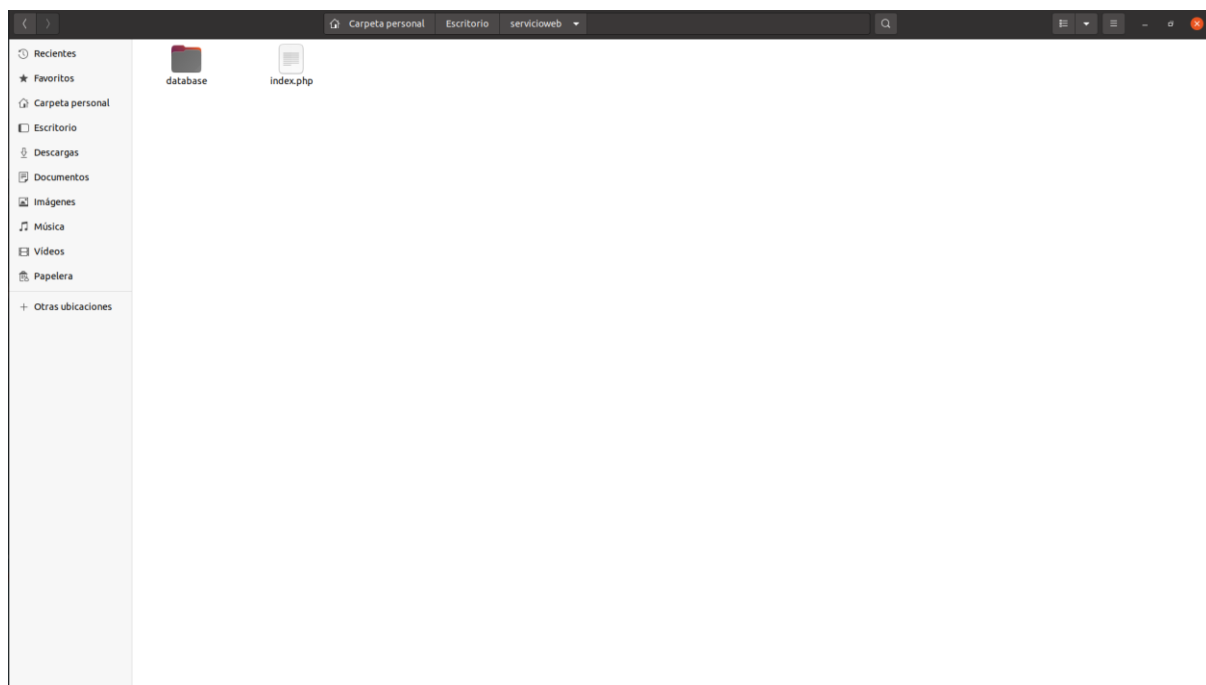
6. Corremos php por medio del comando docker run con las respectivas banderas o parámetros de nuestro servidor php y verificamos que los dos contenedores están corriendo con el comando docker ps. Las banderas son:

- f) -p, es para indicar el puerto por el cual escuchará la aplicación y el puerto del contenedor.
- g) -name, indica el nombre del proceso o contenedor que se iniciará.
- h) -v, indica que se creará un volumen, permite almacenar los archivos de nuestra página web. La primera ruta hace referencia a la carpeta que se creará en nuestro equipo (se ubicará en el escritorio), la segunda hace referencia a la dirección del contenedor (por defecto).
- i) -d, para que php corra en segundo plano.
- j) -link, indica que se pueden comunicar entre el contenedor de php y la base de datos.

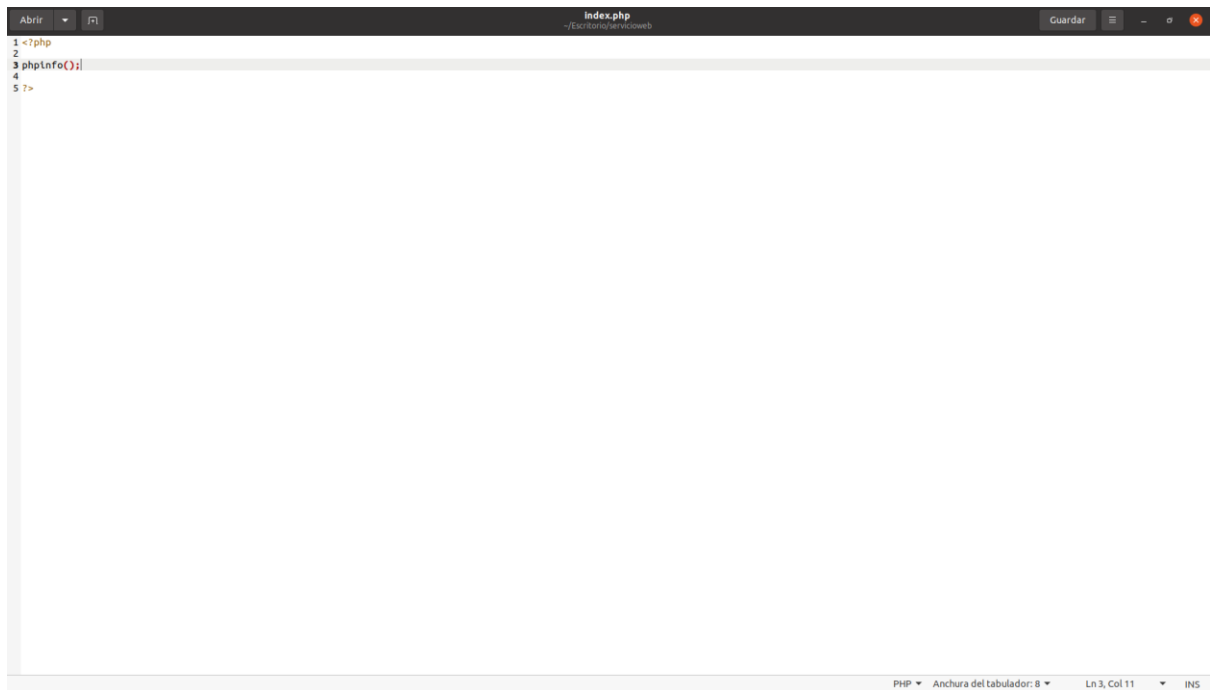
```
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
1c3885da9663   mysql:5.7     "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  33060/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3306/tcp  basedatos

harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$ sudo docker run -p 1000:80 -v /home/harvey/Escritorio/servicioweb:/var/www/html --name miservidorphp
95aa9527915ccc829a3a794e8e933fae404718b6002bd946e1bbf3c7c0f1d782
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
95aa9527915c   php:7.0-apache "docker-php-entrypoi..." 7 seconds ago  Up 6 seconds  0.0.0.0:1000->80/tcp, :::1000->80/tcp  miservidorphp
1c3885da9663   mysql:5.7     "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  33060/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3306/tcp  basedatos
harvey@harvey-HP-Pavilion-Laptop-15-cw00051a:~$
```

7. Dentro de la carpeta servicioweb creamos un nuevo archivo php llamado index.php.



8. Dentro del archivo index.php realizamos el siguiente cambio.



- En nuestro navegador escribimos la siguiente ruta: localhost:puerto, en nuestro caso es el puerto 1000, con esto verificamos que php se encuentra funcionando.



- Ahora comprobaremos que nuestro servidor mysql se encuentra funcionando correctamente, además crearemos la base de datos. Para ello utilizamos el comando `docker exec -i -t basedatos bash`, entrando así al contenedor.

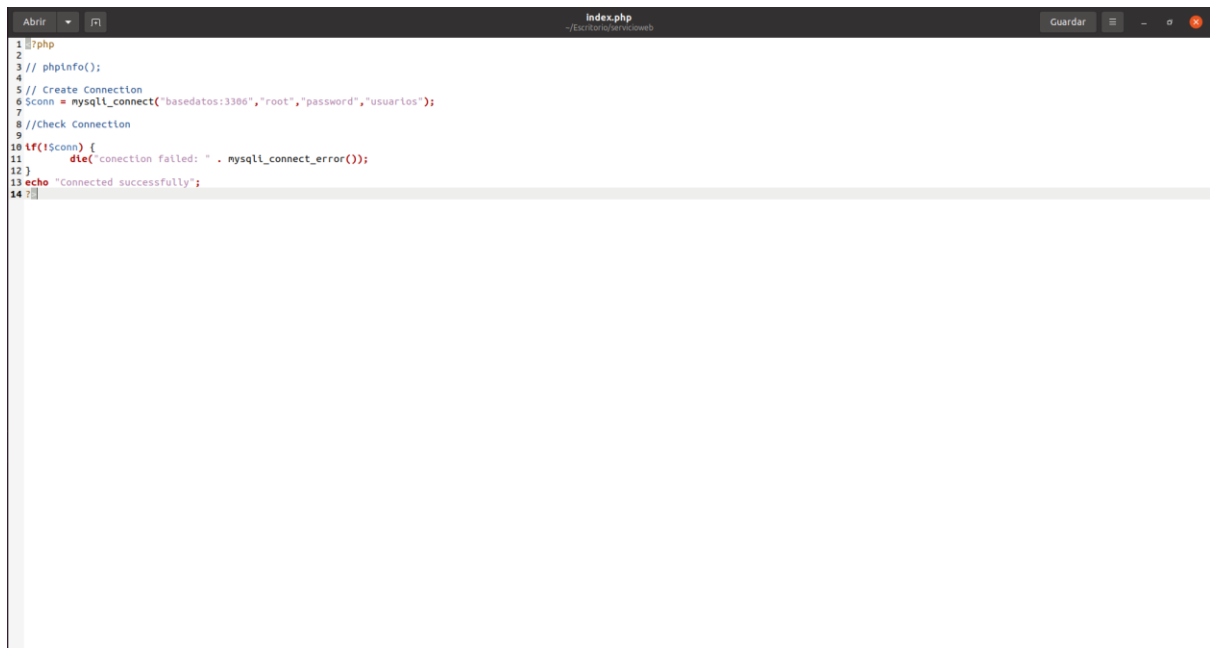
```
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavillon-Laptop-15-cw0005la:~$ sudo docker exec -i -t basedatos /bin/bash  
root@1c3885da9663:/# ls  
bin boot dev docker-entrypoint-initdb.d entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin sr  
v sys tmp usr var  
root@1c3885da9663:/# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2  
Server version: 5.7.38 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.00 sec)
```

11. Accedemos como usuario root a mysql e ingresamos los comandos respectivos para crear la bd y realizar el respectivo ingreso de la bd.

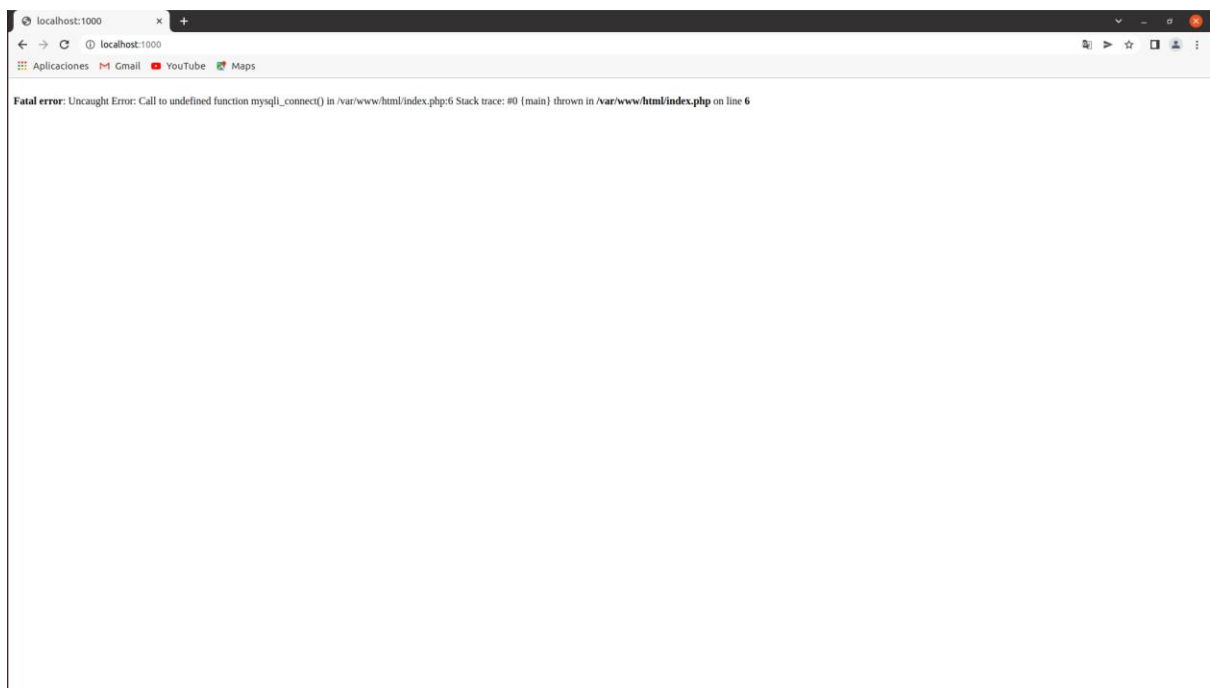
```
mysql> create database usuarios;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| usuarios |  
+-----+  
5 rows in set (0.00 sec)  
  
mysql> use usuarios;  
  
mysql> create table clientes(username varchar(20) primary key not null, nombre varchar(30), correo varchar(50), contra varchar(20));  
Query OK, 0 rows affected (0.28 sec)  
  
mysql> show tables;  
+-----+  
| Tables_in_usuarios |  
+-----+  
| clientes |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> insert into clientes values ('neo', 'nicolas', 'nicolas@gmail.com', '1234');  
Query OK, 1 row affected (0.07 sec)  
  
mysql> select * from clientes;  
+-----+  
| username | nombre | correo | contra |  
+-----+  
| neo | nicolas | nicolas@gmail.com | 1234 |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

12. Dentro de nuestro archivo index.php que se encuentra en la carpeta servicioweb realizamos los siguientes cambios con el fin de comprobar la conexión a la base de datos.

- a. `mysqli_connect("direccionBD:puerto", "usuario", "contraseña", "nombreBD")` ⇒ direccionBD es el nombre de nuestro contenedor para la BD.



13. En nuestro navegador recargamos la siguiente ruta: localhost:puerto, en nuestro caso es el puerto 1000, observamos que se genera un error, es normal ya que la función mysqli\_connect() no se encuentra definida puesto que el servidor que tenemos instalado de php no tiene la extensión para conectarse con mysql.



14. Para corregir esto agregaremos la extensión de mysql para que php pueda reconocer el comando.
- a) Abrir el contenedor php por medio del comando docker exec -i -t miservidorphp /bin/bash
  - b) Ejecutamos el comando cd / y luego ls para listar los archivos.
  - c) Con el comando docker-php-ext-install mysqli instalamos la extensión.



- d) Una vez instalada la extensión tenemos que copiar la dirección en la que se guardó este archivo, el archivo se llama mysqli.os.
- e) Copiamos la ruta que aparece después de: "Installing shared extensions: "
- f) Nos ubicamos en la ruta usr/local/etc/php por medio del comando cd y luego damos ls con el fin listar los archivos que se encuentran.
- g) Debemos abrir el archivo php.ini-development por medio del comando nano, pero para hacer uso de este primero debemos instalarlo por medio de los comando apt-get update y luego apt-get install nano
- h) Ahora si ejecutamos el comando nano php.ini-development que abrirá el archivo.
- i) Buscamos el apartado de extensiones dinámicas y agregamos la ruta que teníamos copiada.
- j) Para salir y guardar los cambios presionamos: control + x, y, Enter.
- k) Realizamos los pasos de los apartados i) y j) después de abrir el archivo php.ini-production con el comando nano.
- l) Reiniciamos el servicio con el comando docker restart miservidorphp
- m) Verificamos que los dos contenedores estén corriendo con el comando docker ps
- n) En el navegador recargamos la ruta localhost:1000 y observamos que aparece el mensaje de conexión exitosa.

```

harvey@harvey-HP-Pavillon-Lap... x harvey@harvey-HP-Pavillon-Lap... x harvey@harvey-HP-Pavillon-Lap... x harvey@harvey-HP-Pavillon-Lap... x
harvey@harvey-HP-Pavillon-Laptop-15-cw00051a:~$ docker exec -i -t miservidorphp /bin/bash
root@95aa9527915c:/var/www/html# cd /
root@95aa9527915c:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@95aa9527915c:/# docker-php-install mysql
bash: docker-php-install: command not found
root@95aa9527915c:/# docker-php-ext-install mysqli
Configuring for:
PHP Api Version:      20151012
Zend Module Api No:   20151012
Zend Extension Api No: 320151012
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed
checking for cc... cc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether cc accepts -g... yes
checking for cc option to accept ISO C89... none needed
checking how to run the C preprocessor... cc -E
checking for icc... no
checking for suncc... no
checking whether cc understands -c and -o together... yes

```

```

harvey@harvey-HP-Pavillon-Laptop-15-cw00051a:~$
If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
Build complete.
Don't forget to run 'make test'.

Installing shared extensions: /usr/local/lib/php/extensions/no-debug-non-zts-20151012/
Installing header files: /usr/local/include/php/
find . -name *.gcno -o -name *.gcda | xargs rm -f
find . -name *.lo -o -name *.o | xargs rm -f
find . -name *.la -o -name *.a | xargs rm -f
find . -name *.so | xargs rm -f
find . -name *.libs -a -type d|xargs rm -rf
rm -f libphp.la modules/* libs/*
root@95aa9527915c:/#

```

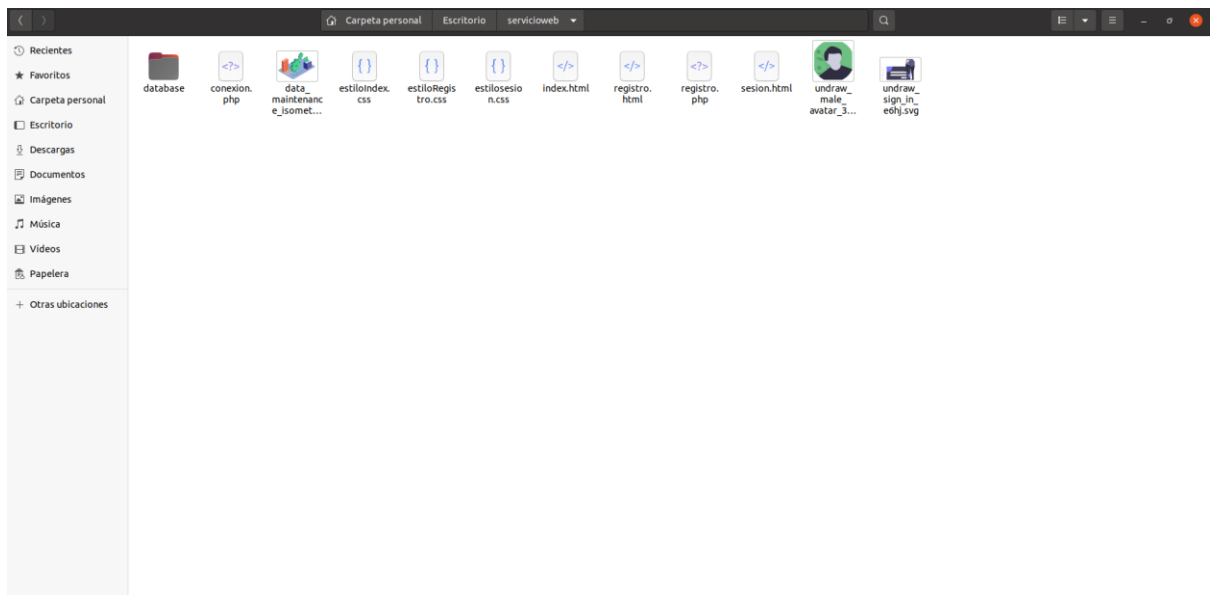
```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
find -name '*.so' | xargs rm -f  
find -name '*.lib' -a -type d | xargs rm -rf  
rm -f libphp.la  
modules/* lib/*  
root@95aa9527915c:/# cd /usr/local/lib/php/extensions/no-debug-non-zts-20151012/  
root@95aa9527915c:/usr/local/lib/php/extensions/no-debug-non-zts-20151012# ls  
mysql.so opcache.so  
root@95aa9527915c:/usr/local/lib/php/extensions/no-debug-non-zts-20151012# cd /  
root@95aa9527915c:/# cd usr/local  
bash: cd: usr/local: No such file or directory  
root@95aa9527915c:/# cd /usr/local/etc/php/  
root@95aa9527915c:/usr/local/etc/php# ls  
conf.d php.ini-development php.ini-production  
root@95aa9527915c:/usr/local/etc/php# nano php.ini-development  
bash: nano: command not found  
root@95aa9527915c:/usr/local/etc/php# apt-get install nano  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
E: Unable to locate package nano  
root@95aa9527915c:/usr/local/etc/php# apt-get update  
Ign:1 http://deb.debian.org/debian stretch InRelease  
Get:2 http://deb.debian.org/debian stretch-updates InRelease [93.6 kB]  
Get:3 http://security.debian.org/debian-security stretch/updates InRelease [53.0 kB]  
Get:4 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [765 kB]  
Get:5 http://deb.debian.org/debian stretch Release [118 kB]  
Get:6 http://deb.debian.org/debian stretch Release.gpg [3177 B]  
Get:7 http://deb.debian.org/debian stretch/main amd64 Packages [7080 kB]  
Fetched 8113 kB in 4s (1665 kB/s)  
Reading package lists... Done  
root@95aa9527915c:/usr/local/etc/php# apt-get install nano  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Suggested packages:  
  spell  
The following NEW packages will be installed:  
  nano  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 485 kB of archives.  
After this operation, 2092 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian stretch/main amd64 nano 2.7.4-1 [485 kB]  
Fetched 485 kB in 0s (928 kB/s)  
debconf: delaying package configuration, since apt-utils is not installed  
Selecting previously unselected package nano.  
(Reading database ... 13099 files and directories currently installed.)  
Preparing to unpack ../nano.2.7.4-1_amd64.deb ...  
Unpacking nano (2.7.4-1) ...  
Setting up nano (2.7.4-1) ...  
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode  
update-alternatives: warning: skip creation of /usr/share/man/man1/nano.1.gz (of link group editor) doesn't exist  
update-alternatives: using /bin/nano to provide /usr/bin/plco (plco) in auto mode  
update-alternatives: warning: skip creation of /usr/share/man/man1/plco.1.gz (of link group pico) doesn't exist  
root@95aa9527915c:/usr/local/etc/php# nano php.ini-development
```

```
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
harvey@harvey-HP-Pavilion-Laptop-15-cw0005la: ~  
GNU nano 2.7.4 File: php.ini-development  
[PHP]  
;;;;;;;;;;;;;;;;;;  
; About php.ini  
;;;;;;;;;;;;;;;;;;  
; PHP's initialization file, generally called php.ini, is responsible for  
; configuring many of the aspects of PHP's behavior.  
;  
; PHP attempts to find and load this configuration from a number of locations.  
; The following is a summary of its search order:  
; 1. SAPI module specific location.  
; 2. The PHPRC environment variable. (As of PHP 5.2.0)  
; 3. A number of predefined registry keys on Windows (As of PHP 5.2.0)  
; 4. Current working directory (except CLI)  
; 5. The web server's directory (for SAPI modules), or directory of PHP  
; (otherwise in Windows)  
; 6. The directory from the --with-config-file-path compile time option, or the  
; Windows directory (C:\Windows or C:\Winnt)  
; see the PHP docs for more specific information.  
; http://php.net/configuration-file  
;  
; The syntax of the file is extremely simple. Whitespaces and lines  
; beginning with a semicolon are silently ignored (as you probably guessed).  
; Section headers (e.g. [foo]) are also silently ignored, even though  
; they might mean something in the future.  
;  
; Directives following the section heading [PATH=/www/mysite] only  
; apply to PHP files in the /www/mysite directory. Directives  
; following the section heading [HOST=www.example.com] only apply to  
; PHP files served from www.example.com. Directives set in these  
; special sections cannot be overridden by user-defined INI files or  
; at runtime. Currently, [PATH=] and [HOST=] sections only work under  
; CGI/FastCGI.  
; http://php.net/ini.sections  
;  
; Directives are specified using the following syntax:  
; directive = value  
; Directive names are "case sensitive" - foo=bar is different from FOO=bar.  
; Directives are variables used to configure PHP or PHP extensions.  
; There is no name validation. If PHP can't find an expected  
; directive because it is not set or is mistyped, a default value will be used.  
;  
; The value can be a string, a number, a PHP constant (e.g. E_ALL or M_PI), one  
; of the INI constants (On, Off, True, False, Yes, No and None) or an expression  
; (e.g. E_ALL & E_NOTICE), a quoted string ("bar"), or a reference to a  
; previously set variable or directive (e.g. ${foo})  
;  
; Expressions in the INI file are limited to bitwise operators and parentheses:  
Get Help Write Out Where Is Cut Text Justify Cur Pos Prev Page First Line WhereIs Next Mark Text Indent Text Undo  
Exit Read File Replace Uncut Text To Spell Go To Line Next Page Last Line To Bracket Copy Text Unindent Text Redo
```





15. Para efectos de este taller eliminamos el archivo `index.php` que teníamos en nuestra carpeta `servicioweb`, descargamos el archivo rar llamado “`paginaweb.rar`” que se encuentra en plataforma, lo descomprimos y copiamos su contenido dentro de la carpeta `servicioweb`.



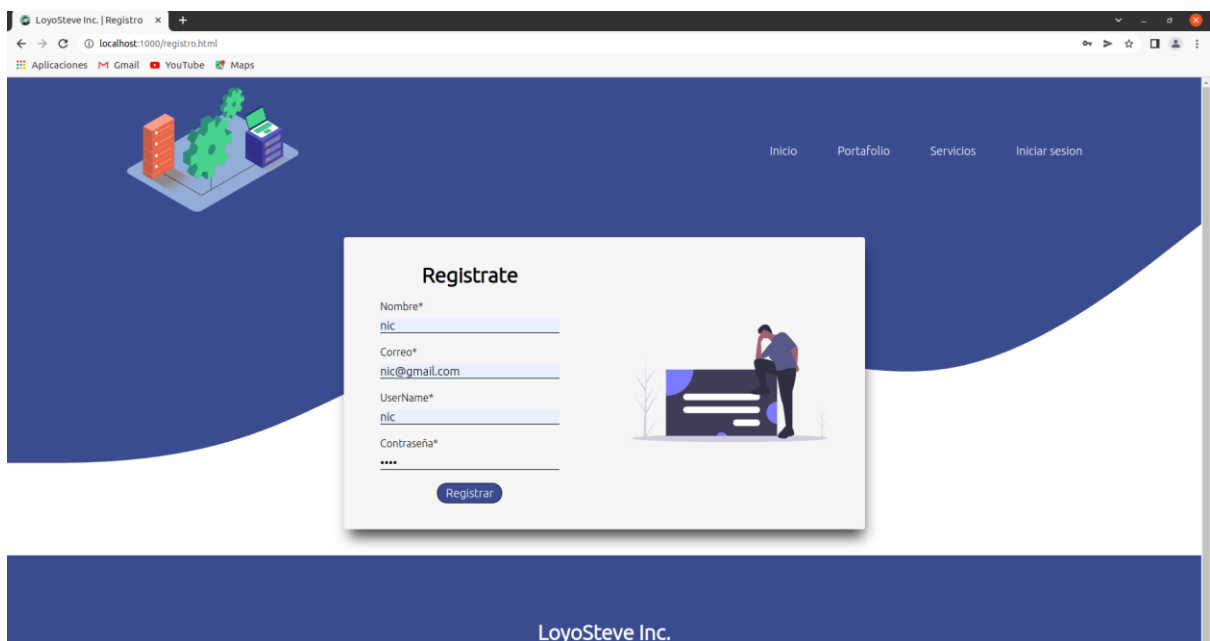
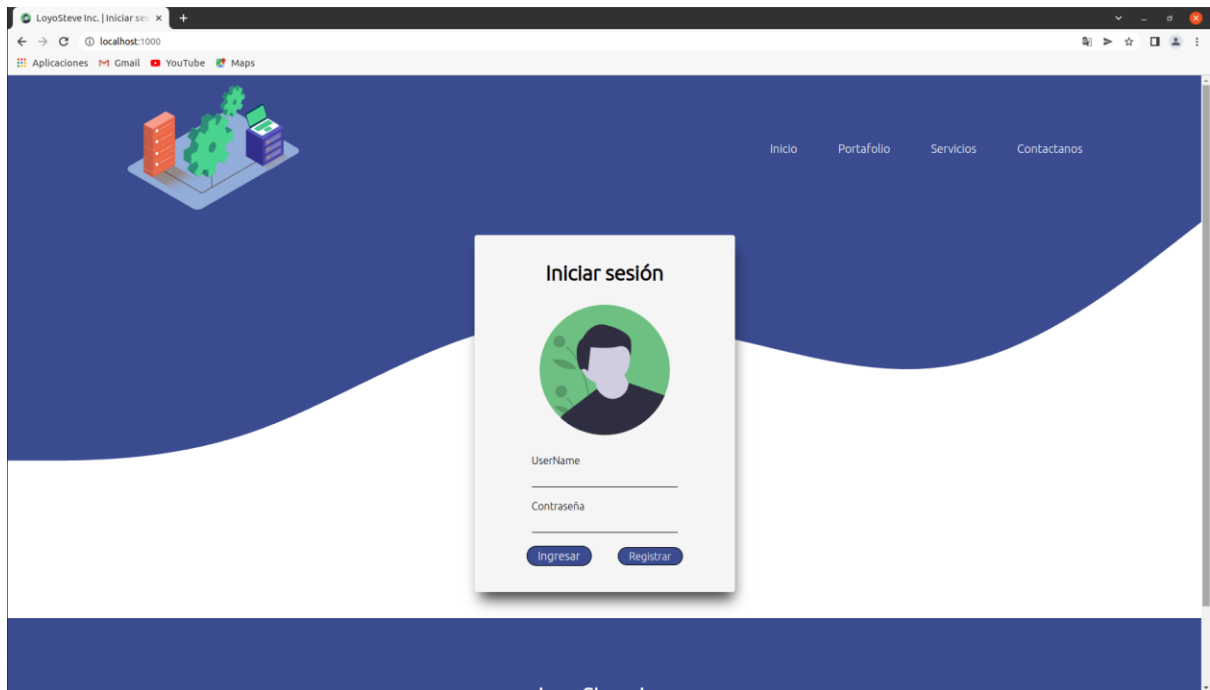
16. Abrimos el archivo llamado `conexion.php` y modificamos los parámetros de conexión de la función `mysqli_connect`.

```
Abrir  conexion.php  Guardar
1 <?php
2 $nombre = $_POST['nombre'];
3 $pass = $_POST['password'];
4
5 $conn = mysqli_connect("basedatos:3306","root","password","usuarios");
6 $query = "select * from clientes where username='".$nombre."' and contrasena='".$pass."'";
7
8 if($conn){
9     $result = mysqli_query($conn, $query);
10
11     if ($result->num_rows > 0) {
12         // output data of each row
13         while($row = mysqli_fetch_assoc()) {
14             $nombre = $row['nombre'];
15             $username = $row['username'];
16             $password = $row['contrasena'];
17             $correo = $row['correo'];
18
19             echo "<doctype html>";
20             echo "<html lang='es'>";
21
22             <head>
23             <meta charset='UTF-8'>
24             <meta name='viewport' content='width=device-width, initial-scale=1.0'>
25             <title>LoyosSteve Inc. | Sesión</title>
26             <link rel='stylesheet' href='estilos.css'>
27             <link rel='shortcut icon' href='undraw_male_avatar_323b.svg' type='image/x-icon'>
28             </head>
29
30             <body>
31                 <header>
32                     <div class='contentheader'>
33                         <img class='imagenh' src='data_maintenance_isometric.svg'>
34                         <div></div>
35                         <nav>
36                             <ul class='menu'>
37                                 <li><a href='index.html'>Inicio</a></li>
38                                 <li><a href='index.html'>Portafolio</a></li>
39                                 <li><a href='index.html'>Servicios</a></li>
40                                 <li><a href='index.html'>Contactanos</a></li>
41                             </ul>
42                         </nav>
43                     </header>
44
45                     <div class='formulario'>
46                         <form action=''>
47                             <div class='user'>
48                                 <img src='undraw_male_avatar_323b.svg'>
49                                 <input type='text' value='".$nombre."'>
50                             </div>
51                             <div class='password'>
52                                 <input type='password' value='".$password."'>
53                                 <input type='text' value='".$correo."'>
```

17. Abrimos el archivo llamado registro.php y modificamos los parámetros de conexión de la función mysqli\_connect.

```
Abrir  registro.php  Guardar
1 <?php
2
3 $nombre = $_POST['nombre'];
4 $correo = $_POST['correo'];
5 $username = $_POST['username'];
6 $contra = $_POST['contra'];
7
8 $conn = mysqli_connect("basedatos:3306","root","password","usuarios");
9 $query = "insert into clientes values('".$username."', '".$nombre."', '".$correo."', '".$contra."')";
10
11 if($conn){
12     $result = mysqli_query($conn, $query);
13     if($result){
14         echo "<script type='text/javascript'>";
15         alert("Registro Correcto");
16         window.location.href="index.html";
17         </script>";
18     }else{
19         echo "<script type='text/javascript'>";
20         alert("Algo salio mal. :/");
21         window.location.href="registro.html";
22         </script>";
23     }
24 }else{
25     echo "no conectado bro..";
26 }
27
28 >
```

18. Abrimos nuestro navegador con la ruta localhost:1000, observando que carga la página principal y realizamos la prueba de registro de usuarios e inicio de sesión.

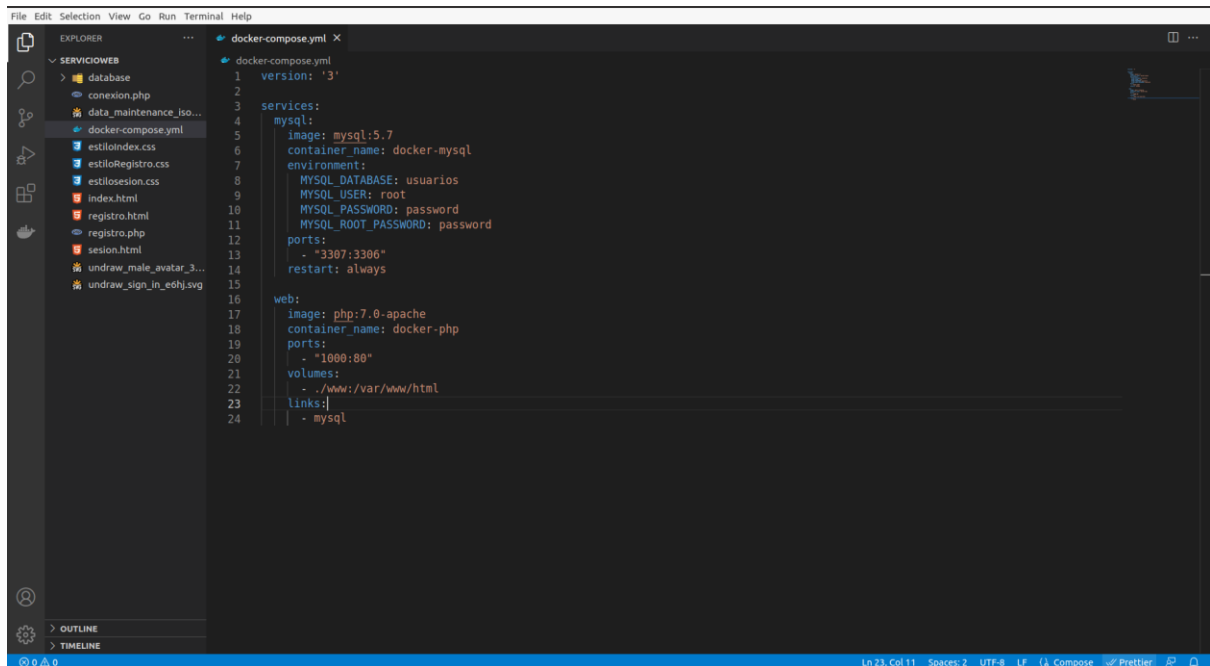
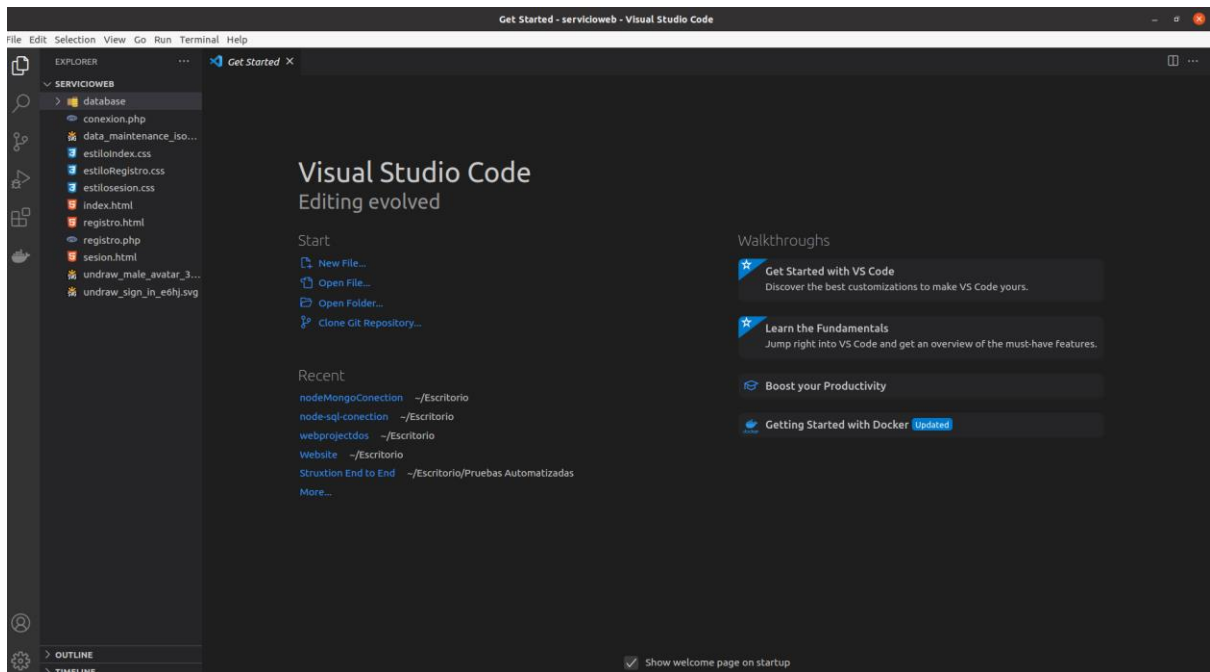


19. Podemos validar los datos guardados en la BD

```
mysql> select * from clientes;
+-----+-----+-----+-----+
| username | nombre | correo | contra |
+-----+-----+-----+-----+
| neo | nicolas | nicolas@gmail.com | 1234 |
| nic | nic | nic@gmail.com | 0000 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

20. Por último, abrimos nuestra carpeta en Visual Studio Code y creamos nuestro archivo docker-compose.yml



## APARTADO 2: KUBERNETES

### Parte 1

Ingresa a la siguiente página y desarrolla el tutorial interactivo:

<https://kubernetes.io/docs/tutorials/hello-minikube/>

### Parte 2

Prerrequisito: Tener instalado Docker, sino es así acceder a este enlace para descargar el instalador y posteriormente realizar la instalación.

[https://drive.google.com/file/d/1mbykZVf\\_4ZGKwwVAoBq4ugzmJUEy0WdS/view?usp=sharing](https://drive.google.com/file/d/1mbykZVf_4ZGKwwVAoBq4ugzmJUEy0WdS/view?usp=sharing)

- Prerrequisito: Tener instalado Minikube, sino es así acceder a este enlace para descargar el

instalador y posteriormente realizar la instalación.

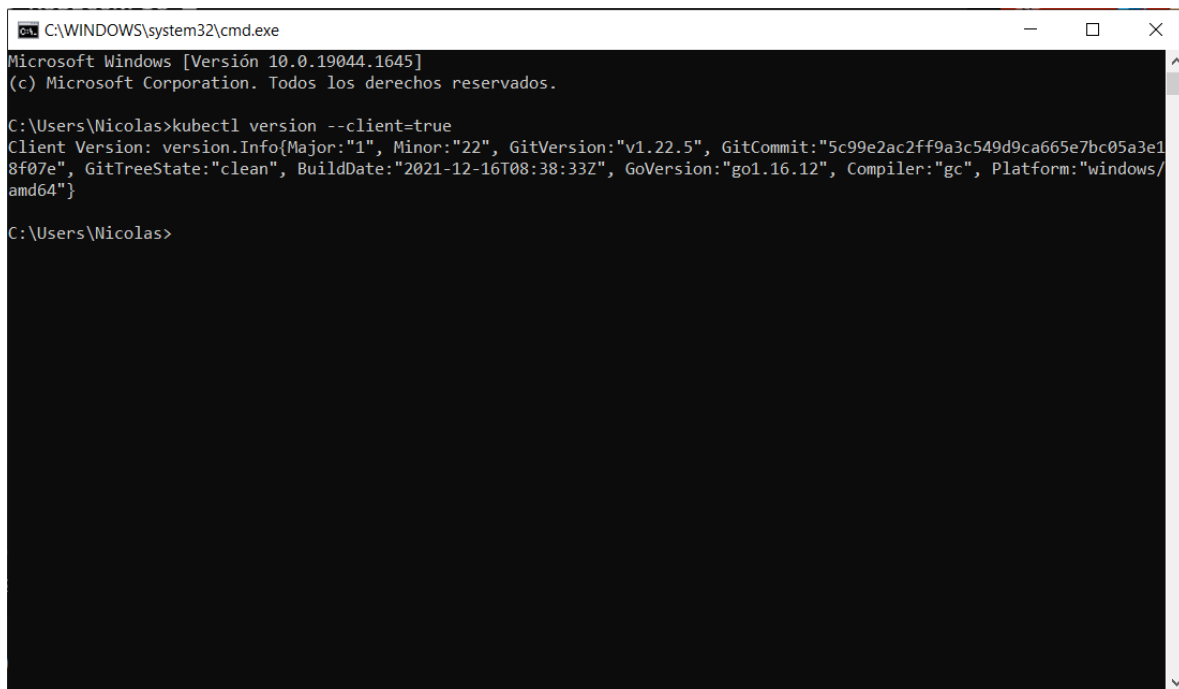
<https://minikube.sigs.k8s.io/docs/start/>

- Prerrequisito: Tener instalado Kubectl, sino es así acceder a este enlace para descargar el

instalador y posteriormente realizar la instalación.

<https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>

1. Ejecutamos este comando para conocer la versión de kubectl que se tiene instalado, recordando que esta herramienta nos permite interactuar con el cluster.

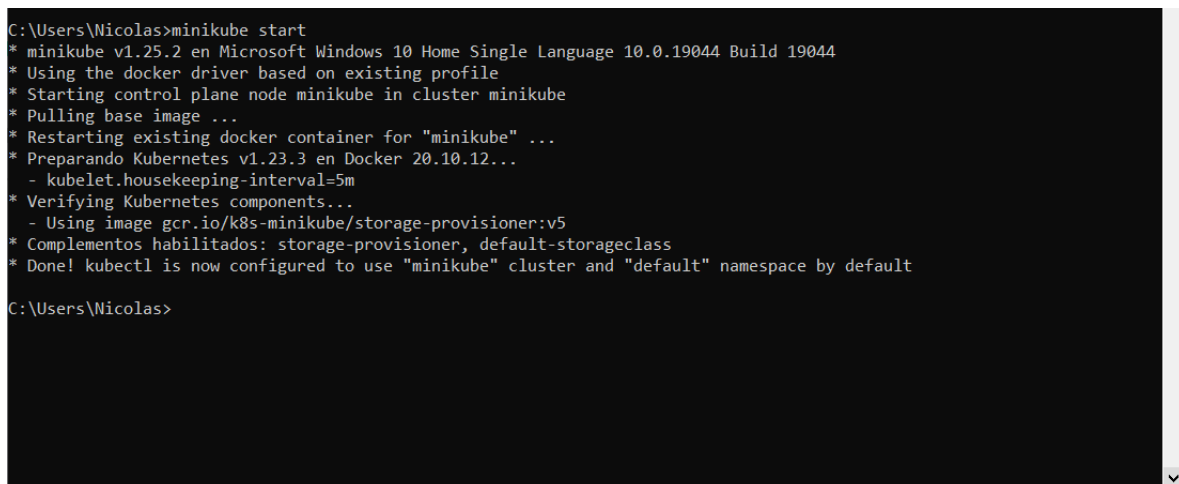


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.1645]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Nicolas>kubectl version --client=true
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.5", GitCommit:"5c99e2ac2ff9a3c549d9ca665e7bc05a3e18f07e", GitTreeState:"clean", BuildDate:"2021-12-16T08:38:33Z", GoVersion:"go1.16.12", Compiler:"gc", Platform:"windows/amd64"}

C:\Users\Nicolas>
```

2. Iniciamos minikube.

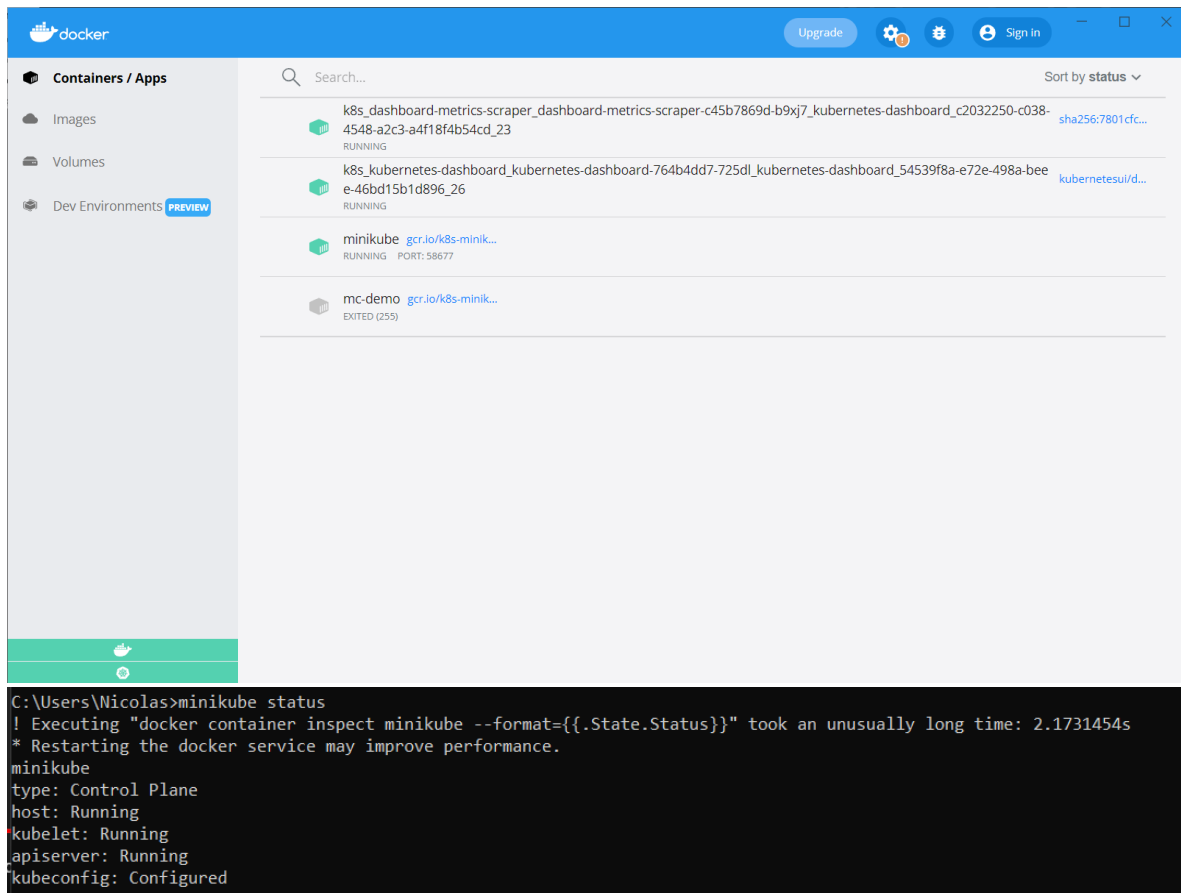


```
C:\Users\Nicolas>minikube start
* minikube v1.25.2 en Microsoft Windows 10 Home Single Language 10.0.19044 Build 19044
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Restarting existing docker container for "minikube" ...
* Preparando Kubernetes v1.23.3 en Docker 20.10.12...
  - kubelet.housekeeping-interval=5m
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Complementos habilitados: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

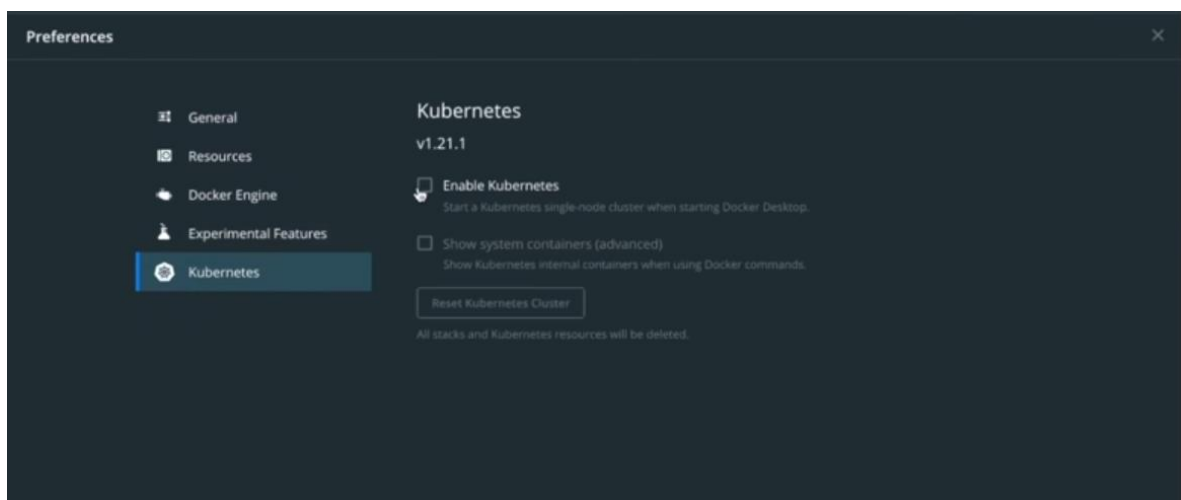
C:\Users\Nicolas>
```

3. Validamos que minikube este corriendo con Docker.

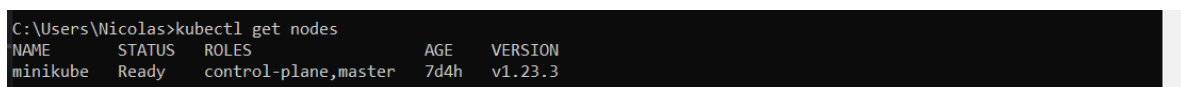




4. Verificamos que kubernetes está habilitado con docker.



5. Con este comando podremos observar los nodos de nuestro cluster de kubernetes.



Nos dirigimos a la siguiente página: <https://kubernetes.io/docs/tutorials/stateless-application/guestbook/>, con el fin de realizar la práctica, además de tener los resultados de la página encontrarán las capturas de pantalla del proceso.

```
C:\WINDOWS\system32\cmd.exe - kubectl logs -f deployment/redis-leader

Microsoft Windows [Versión 10.0.19044.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Nicolas>kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml
deployment.apps/redis-leader unchanged

C:\Users\Nicolas>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
my-csi-app-set-0                    0/1     Pending   0           8d
nginx-deployment-66c9c7669-qzh4s    1/1     Running   1 (8d ago)  8d
nginx-deployment-66c9c7669-tqftp    1/1     Running   1 (8d ago)  8d
nginx-deployment-vlhlr              1/1     Running   1 (8d ago)  8d
redis-leader-766465cd9c-td5lf       1/1     Running   0           2m56s

C:\Users\Nicolas>kubectl logs -f deployment/redis-leader
1:C 17 May 2022 03:54:43.829 # o000o000o000o Redis is starting o000o000o000o
1:C 17 May 2022 03:54:43.829 # Redis version=6.0.5, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 17 May 2022 03:54:43.829 # Warning: no config file specified, using the default config. In order to specify a config
file use redis-server /path/to/redis.conf
1:M 17 May 2022 03:54:43.829 * Running mode=standalone, port=6379.
1:M 17 May 2022 03:54:43.830 # Server initialized
1:M 17 May 2022 03:54:43.830 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will c
reate latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transpa
rent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis mu
st be restarted after THP is disabled.
1:M 17 May 2022 03:54:43.830 * Ready to accept connections
```

```
C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Versión 10.0.19044.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Nicolas>kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-service.yaml
service/redis-leader created

C:\Users\Nicolas>kubectl get service
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes ClusterIP   10.96.0.1     <none>        443/TCP    22d
redis-leader ClusterIP   10.101.58.22 <none>        6379/TCP   13s

C:\Users\Nicolas>kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-deployment.yaml
deployment.apps/redis-follower created

C:\Users\Nicolas>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
my-csi-app-set-0                    0/1     Pending   0           8d
nginx-deployment-66c9c7669-qzh4s    1/1     Running   1 (8d ago)  8d
nginx-deployment-66c9c7669-tqftp    1/1     Running   1 (8d ago)  8d
nginx-deployment-vlhlr              1/1     Running   1 (8d ago)  8d
redis-follower-84fcc94dfc-s5v8j     1/1     Running   0           9s
redis-follower-84fcc94dfc-ss2hg     1/1     Running   0           9s
redis-leader-766465cd9c-td5lf       1/1     Running   0           5m27s

C:\Users\Nicolas>
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Nicolas>kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-service.yaml
service/redis-follower created

C:\Users\Nicolas>kubectl get service
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)    AGE
kubernetes      ClusterIP   10.96.0.1        <none>       443/TCP    22d
redis-follower   ClusterIP   10.102.132.195   <none>       6379/TCP    7s
redis-leader     ClusterIP   10.101.58.22     <none>       6379/TCP    115s

C:\Users\Nicolas>kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml
deployment.apps/frontend created

C:\Users\Nicolas>kubectl get pods -l app=guestbook -l tier=frontend
NAME                                READY   STATUS             RESTARTS   AGE
frontend-57df59b89c-5msvx           0/1     ContainerCreating   0          7s
frontend-57df59b89c-chbxd           0/1     ContainerCreating   0          7s
frontend-57df59b89c-wg64r           0/1     ContainerCreating   0          7s
```

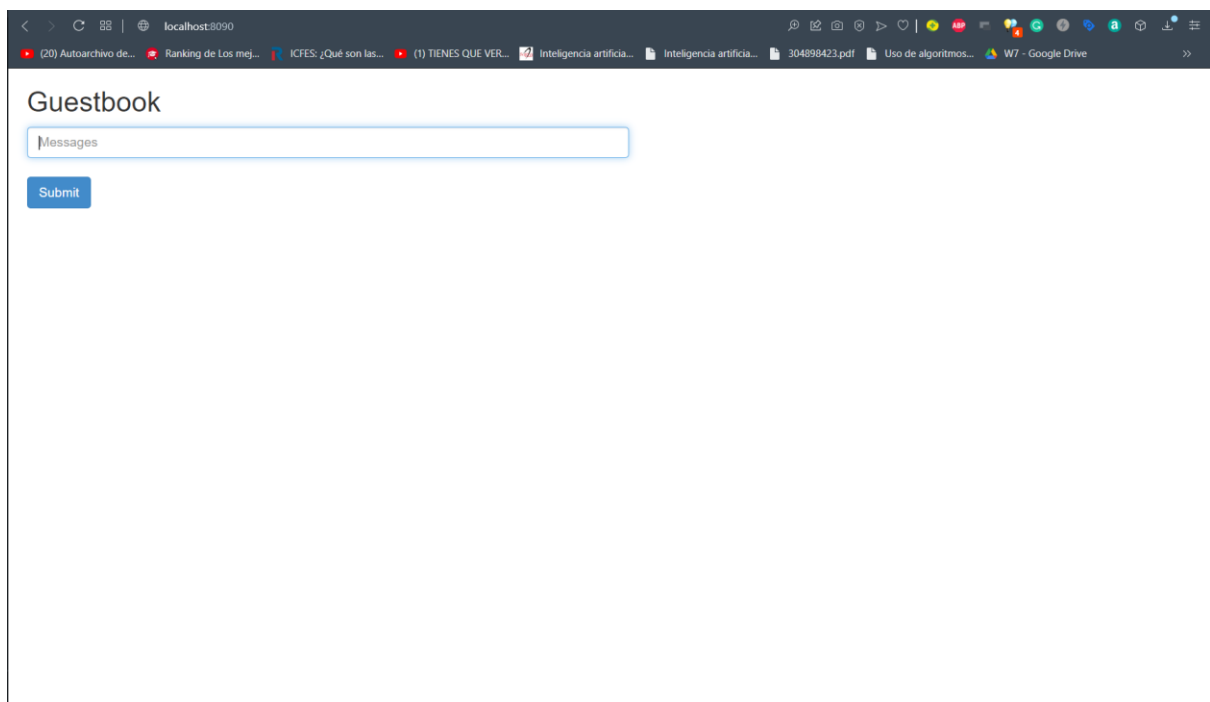
```
C:\Users\Nicolas>kubectl get pods -l app=guestbook -l tier=frontend
NAME                                READY   STATUS    RESTARTS   AGE
frontend-57df59b89c-5msvx           1/1     Running   0          3m49s
frontend-57df59b89c-chbxd           1/1     Running   0          3m49s
frontend-57df59b89c-wg64r           1/1     Running   0          3m49s

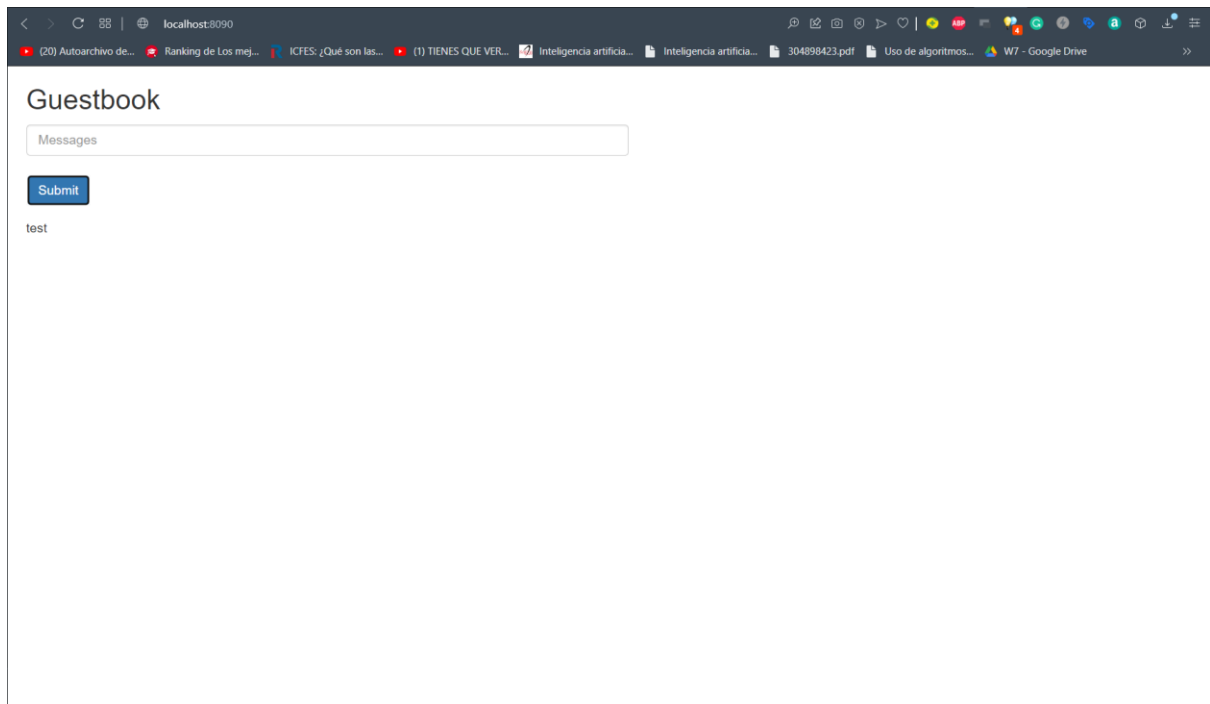
C:\Users\Nicolas>
```

```
C:\WINDOWS\system32\cmd.exe - kubectl port-forward svc/frontend 8090:80
C:\Users\Nicolas>kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
service/frontend created

C:\Users\Nicolas>kubectl get services
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)    AGE
frontend        ClusterIP   10.107.19.46     <none>       80/TCP     9s
kubernetes      ClusterIP   10.96.0.1        <none>       443/TCP    22d
redis-follower   ClusterIP   10.102.132.195   <none>       6379/TCP    85s
redis-leader     ClusterIP   10.101.58.22     <none>       6379/TCP    3m13s
```

```
C:\Users\Nicolas>kubectl port-forward svc/frontend 8090:80
Forwarding from 127.0.0.1:8090 -> 80
Forwarding from [::1]:8090 -> 80
Handling connection for 8090
Handling connection for 8090
```





```
C:\Users\Nicolas>kubectl get service frontend
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
frontend  ClusterIP   10.107.19.46  <none>         80/TCP     8m35s
```

```
C:\Users\Nicolas>kubectl scale deployment frontend --replicas=3
deployment.apps/frontend scaled
```

```
C:\Users\Nicolas>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
frontend-57df59b89c-5msvx           1/1     Running   0           11m
frontend-57df59b89c-chbxh           1/1     Running   0           11m
frontend-57df59b89c-wg64r           1/1     Running   0           11m
my-csi-app-set-0                     0/1     Pending   0           8d
nginx-deployment-66c9c7669-qzh4s     1/1     Running   1 (8d ago)  8d
nginx-deployment-66c9c7669-tqftp     1/1     Running   1 (8d ago)  8d
nginx-deployment-vlh1r               1/1     Running   1 (8d ago)  8d
redis-follower-84fcc94dfc-s5v8j      1/1     Running   0           13m
redis-follower-84fcc94dfc-ss2hg      1/1     Running   0           13m
redis-leader-766465cd9c-td51f       1/1     Running   0           18m
```

```
C:\Users\Nicolas>kubectl scale deployment frontend --replicas=2
deployment.apps/frontend scaled
```

```
C:\Users\Nicolas>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
frontend-57df59b89c-5msvx           1/1     Running   0           12m
frontend-57df59b89c-chbxh           1/1     Running   0           12m
my-csi-app-set-0                     0/1     Pending   0           8d
nginx-deployment-66c9c7669-qzh4s     1/1     Running   1 (8d ago)  8d
nginx-deployment-66c9c7669-tqftp     1/1     Running   1 (8d ago)  8d
nginx-deployment-vlh1r               1/1     Running   1 (8d ago)  8d
redis-follower-84fcc94dfc-s5v8j      1/1     Running   0           13m
redis-follower-84fcc94dfc-ss2hg      1/1     Running   0           13m
redis-leader-766465cd9c-td51f       1/1     Running   0           19m
```

```
C:\Users\Nicolas>
```

```
C:\Users\Nicolas>kubectl delete deployment -l app=redis
deployment.apps "redis-follower" deleted
deployment.apps "redis-leader" deleted
```

```
C:\Users\Nicolas>kubectl delete service -l app=redis
service "redis-follower" deleted
service "redis-leader" deleted
```

```
C:\Users\Nicolas>kubectl delete deployment frontend
deployment.apps "frontend" deleted
```

```
C:\Users\Nicolas>kubectl delete service frontend
service "frontend" deleted
```

```
C:\Users\Nicolas>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
my-csi-app-set-0                   0/1     Pending   0           8d
nginx-deployment-66c9c7669-qzh4s   1/1     Running   1 (8d ago)  8d
nginx-deployment-66c9c7669-tqftp   1/1     Running   1 (8d ago)  8d
nginx-deployment-vlhlr             1/1     Running   1 (8d ago)  8d
redis-follower-84fcc94dfc-s5v8j    1/1     Terminating 0           16m
```