

Guía Final de Tercer Corte

Guía elaborada por: **Carlos David Páez Ferreira y Juana Valentina Mendoza Santamaría**

Guía Liderada por: **Harvey Nicolás Echavarria Ortiz**

Fecha: **28 Mayo 2022**

Apartado 1: Docker

Verificar la instalación de Docker

```
docker --version
```

```
C:\Users\carlo>docker --version
Docker version 20.10.14, build a224086
C:\Users\carlo>
```

Instalar imágenes necesarias

Instalar las imágenes de MySQL 5.7 y PHP 7.0 junto Apache.

```
docker pull mysql:5.7 & docker pull php:7.0-apache
```

```
C:\Users\carlo>docker pull mysql:5.7 & docker pull php:7.0-apache
5.7: Pulling from library/mysql
Digest: sha256:16e159331007ecc069822f7b731272043ed572a79a196a05ffa2ea127caaf67
Status: Image is up to date for mysql:5.7
docker.io/library/mysql:5.7
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: sha256:1d34b2e491a02ba7a8d26478132015e197a5ffea37f0a93b42621d11cf042cc
Status: Downloaded newer image for php:7.0-apache
docker.io/library/php:7.0-apache

C:\Users\carlo>
```

Verificar las imágenes instaladas

Images on disk

42 images Total size: 7.41 GB IN USE UNUSED Clean up...

Images Give Feedback

LOCAL REMOTE REPOSITORIES

NAME	TAG	IMAGE ID	CREATED	SIZE
php	7.0-apache	aa67a9c9814f	over 3 years ago	367.57 MB

Connect to Remote Content

Store and backup your images remotely

Unlock vulnerability scanning for greater security

Collaborate with your team

Connect for free

Sign in

Images on disk

42 images Total size: 7.41 GB IN USE UNUSED Clean up...

Images Give Feedback

LOCAL REMOTE REPOSITORIES

NAME	TAG	IMAGE ID	CREATED	SIZE
mysql	IN USE	a3d35804fa37	7 days ago	461.92 MB
mysql/mysql-server	IN USE	434c35b82b08	4 months ago	417.35 MB

Connect to Remote Content

Store and backup your images remotely

Unlock vulnerability scanning for greater security

Collaborate with your team

Connect for free

Sign in

docker images

```
C:\Users\carlo>docker images
REPOSITORY
mysql
hubproxy.docker.internal:5000/docker/desktop-kubernetes
k8s.gcr.io/kube-apiserver
k8s.gcr.io/kube-proxy
k8s.gcr.io/kube-scheduler
k8s.gcr.io/kube-controller-manager
k8s.gcr.io/etcd
p2t1_node_mongo_web
<none>
<none>
<none>
<none>
<none>
hellonode
mongo
hubproxy.docker.internal:5000/docker/desktop-kubernetes
dpage/pgadmin4
k8s.gcr.io/pause
postgres
gcr.io/k8s-minikube/kicbase
B
mysql/mysql-server
hubproxy.docker.internal:5000/docker/desktop-kubernetes-apiserver
k8s.gcr.io/kube-apiserver
hubproxy.docker.internal:5000/docker/desktop-kubernetes-proxy
k8s.gcr.io/kube-proxy
hubproxy.docker.internal:5000/docker/desktop-kubernetes-scheduler
B
k8s.gcr.io/kube-scheduler
B
hubproxy.docker.internal:5000/docker/desktop-kubernetes-controller-manager
k8s.gcr.io/kube-controller-manager
k8s.gcr.io/coredns/coredns
B
hubproxy.docker.internal:5000/docker/desktop-kubernetes-etcd
k8s.gcr.io/etcd
hubproxy.docker.internal:5000/docker/desktop-kubernetes-coredns
B
k8s.gcr.io/coredns/coredns
B
mariadb/server
hubproxy.docker.internal:5000/docker/desktop-vpnkit-controller
docker/desktop-vpnkit-controller
docker/desktop-storage-provisioner
B
hubproxy.docker.internal:5000/docker/desktop-storage-provisioner
B
hubproxy.docker.internal:5000/docker/desktop-kubernetes-pause
k8s.gcr.io/pause
php

C:\Users\carlo>
```

Correr imagen de MySQL

Vamos a ejecutar diferentes banderas al momento de correr la imagen de MySQL:

- **-p:** Indica el puerto por el cual escuchará la aplicación y el puerto del contenedor.

- **--name:** Indica el nombre del proceso o contenedor que se iniciará.
- **-v:** Indica que se creará un volumen, el cual permite almacenar datos de la DB en una carpeta con el fin de salvarlos en caso de que el contenedor se apague o se caiga. La primera rta hace referencia a la carpeta que se creará en nuestro equipo, la segunda hace referencia a la dirección del contenedor por defecto.
- **-e:** Indica la contraseña
- **-d,** para que MySQL corra en segundo plano.

```
docker run -p 3307:3306 --name database -v
C:/Users/carlo/.../P3T1_Guia_Final:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=password
-d mysql:5.7
```

```
C:\Users\carlo>docker run -p 3307:3306 --name database -v C:/Users/carlo/Documents/Noveno_Semestre/DevOps/Tercer_Corte/P3T1_Guia_Final:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=password -d mysql
:5.7
12ce148768567463f54d1af81fdb23d959d5f10f0c886f141a7ecb9288fd8d5a
```

Verificamos que este corriendo nuestro contenedor:

```
docker ps
```

```
C:\Users\carlo>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                         NAMES
428ddaf9b4f0        mysql:5.7          "docker-entrypoint.s..."   43 seconds ago    Up 42 seconds      33060/tcp, 0.0.0.0:3307->3306/tcp   database
cc96093816f3        p2t1_node_mongo_web   "docker-entrypoint.s..."   6 weeks ago       Up 2 hours        0.0.0.0:5000->3000/tcp               exampleapp
21b2aad70c24        postgres           "docker-entrypoint.s..."   2 months ago      Up 2 hours        0.0.0.0:5432->5432/tcp               src_postgres_1
```

Dentro del directorio que hemos escogido, creamos una carpeta para almacenar una página web, y dentro de este directorio, añadimos un nuevo folder para la base de datos.

Correr la imagen de PHP

Vamos a correr la imagen de PHP con las siguientes banderas:

- **-p:** Indicar el puerto
- **-name:** Indicar el nombre del proceso o contenedor
- **-v:** Indicar la creación de un volumen en el que almacenaremos los archivos de nuestro proyecto.
- **-d:** Correr en segundo plano
- **--link:** Comunicar entre el contenedor de PHP y la base de datos.

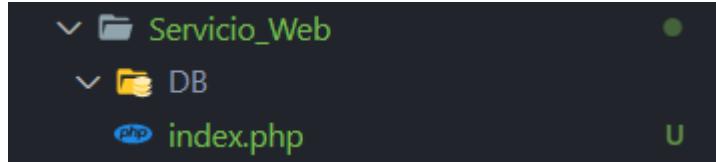
```
docker run -p 1000:80 -v
C:/Users/carlo/.../P3T1_Guia_Final/Servicio_Web:/var/www/html --name servidorphp -
d --link database php:7.0-apache
```

```
C:\Users\carlo>docker run -p 1000:80 -v C:/Users/carlo/
apache
695a031207378fb71baecbdbb39d5c632b205e2e549a93d65a225eb1ee6693cf
/P3T1_Guia_Final/Servicio_Web:/var/www/html --name servidorphp -d --link database php:7.0-
```

```
docker ps
```

C:\Users\carlo>docker ps	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5a67bfeac68c	php:7.0-apache	"docker-php-entrypoi..."	About a minute ago	Up About a minute	0.0.0.0:1000->80/tcp	servidorphp	
428ddaf9b4f0	mysql:5.7	"docker-entrypoint.s..."	4 minutes ago	Up 4 minutes	33060/tcp, 0.0.0.0:3307->3306/tcp	database	
ce96093816f3	p2t1_node_mongo_web	"docker-entrypoint.s..."	6 weeks ago	Up 2 hours	0.0.0.0:5000->3000/tcp	exampleapp	
21b2aad70c24	postgres	"docker-entrypoint.s..."	2 months ago	Up 2 hours	0.0.0.0:5432->5432/tcp	src_postgres_1	

Creamos un archivo llamado **index.php** dentro de la carpeta de **Servicio_Web**:



Dentro de nuestro nuevo archivo, escribimos lo siguiente:

```
<?php
phpinfo();
?>
```

Luego, en el navegador ingresamos la siguiente ruta: <http://localhost:1000> y debemos obtener lo siguiente:

System	Value
Build Date	Dec 29 2018 06:50:15
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-fpm' '--enable-mbstring' '--enable-mysqlind' '--with-curl' '--with-libxml' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-qpc' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	(none)
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, phar, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib*, convert.iconv*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies

zendengine

Configuration	
apache2handler	
Apache Version	Apache/2.4.25 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost

Comprobar que el servidor MySQL funciona

Entrar a la consola interactiva del contenedor de mysql:

```
docker exec -i -t database bash
```

```
C:\Users\carlo>docker exec -i -t database bash  
root@e2749101e305:/#
```

Ejecutar MySQL:

```
mysql -u root -p
```

```
root@e2749101e305:/# 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>
```

Listar las bases de datos:

```
show databases;
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.00 sec)
```

Crear base de datos de usuarios:

```
create database usuarios;
```

```
mysql> create database usuarios;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
| usuarios       |
+-----+
5 rows in set (0.00 sec)
```

Usar la base de datos recién creada:

```
use usuarios;
```

```
mysql> use usuarios;
Database changed
mysql>
```

Crear tabla tabla de clientes:

```
create table clientes(username varchar(20) primary key not null, nombre
varchar(30), correo varchar(50), contra varchar(20));
```

```
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 (0.05 sec)

mysql> show tables;
+-----+
| Tables_in_usuarios |
+-----+
| clientes           |
+-----+
1 row in set (0.00 sec)
```

Insertar datos dentro de la tabla:

```
insert into clientes values ('neo', 'neo', 'neo@gmail.com', 'neo_password');
```

```
mysql> insert into clientes values ('neo', 'neo', 'neo@gmail.com', 'neo_password');
Query OK, 1 row affected (0.01 sec)

mysql> select * from clientes;
+-----+-----+-----+-----+
| username | nombre | correo        | contra      |
+-----+-----+-----+-----+
| neo     | neo   | neo@gmail.com | neo_password |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Conecrtar PHP con la base de datos

Dentro del archivo `index.php` añadimos las siguientes líneas:

```
<?php  
$conn = mysqli_connect("database:3306", "root", "password", "usuarios");  
  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
  
echo "Connected successfully";  
?>
```

Cuando regresamos a la ruta dentro del navegador, vamos a observar el siguiente error:



Esto se debe a que la función `mysqli_connection()` no se encuentra definida, puesto que el servidor que tenemos instalado de php no tiene la extensión para conectarse a mysql. Para solucionar dicho error debemos seguir estos pasos:

- Abrir el contenedor de PHP por medio del siguiente comando:

```
docker exec -i -t servidorphp /bin/bash
```

```
C:\Users\carlo>docker exec -i -t servidorphp /bin/bash  
root@6dbce79d46e1:/var/www/html#
```

- Ir al directorio `/` y listar los archivos y directorios:

```
root@6dbce79d46e1:/var/www/html# cd /  
root@6dbce79d46e1:/# ls  
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var  
root@6dbce79d46e1:/#
```

- Usamos el siguiente comando para instalar la extensión necesaria:

```
docker-php-ext-install mysqli
```

```
root@6dbce79d46e1:/# docker exec -i servidordb/bin/bash
Configuring for:
PHP API Version: 20151012
Zend Engine API No: 20151012
Thread 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 whether the 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 for icc... no
checking for i586xpc-linux-gnu C preprocessor... cc -E
checking for i686xpc-linux-gnu C preprocessor... cc -E
checking for suncc... no
checking whether cc understands -c and -o together... yes
checking for system library directory... lib
checking if compiler supports -R... no
checking if compiler supports -Wl,... -rpath... yes
checking for minimum system version... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for PHP prefix... /usr/local
checking for PHP includes... -I/usr/local/include/php -I/usr/local/include/php/main -I/usr/local/include/php/TSRM -I/usr/local/include/php/zend -I/usr/local/include/php/ext -I/usr/local/include/php/ext/date/lib
checking for PHP extension directory... /usr/local/lib/php/extensions/no-debug-non-zts-20151012
checking for PHP installed headers prefix... /usr/local/include/php
checking if debug is enabled... no
checking if zts is enabled... no
checking for re2c... re2c
checking for re2c version... 0.16 (ok)
checking for gawk... no
checking for awk
checking if awk is broken... no
checking for MySQL support... yes, shared
checking whether to enable embedded MySQL support... no
checking for specified location of the MySQL UNIX socket... no
checking for MySQL UNIX socket location... no
checking for ld used by cc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for BSD-compatible nm... /usr/bin/nm -B
checking whether ln -s works... yes
checking how to recognize dependent libraries... pass_all
checking for ANSI C header files... pass_all
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
```

Debemos copiar la ruta que aparece en la sección de [Installing shared extensions](#) al momento de terminar la instalación de la extensión. ([/usr/local/lib/php/extensions/no-debug-non-zts-20151012/](#))

```
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/*
```

- Nos ubicamos dentro de la ruta [/usr/local/etc/php/](#) y listamos los archivos que se encuentran dentro de dicho directorio

```
root@6dbce79d46e1:/# cd /usr/local/etc/php/
root@6dbce79d46e1:/usr/local/etc/php# ls
conf.d  php.ini-development  php.ini-production
root@6dbce79d46e1:/usr/local/etc/php#
```

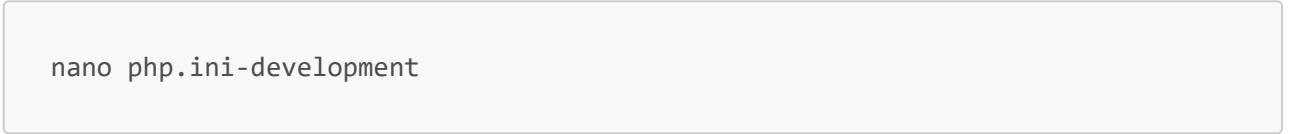
- Instalamos nano con los siguientes comandos:

```
apt-get update
```

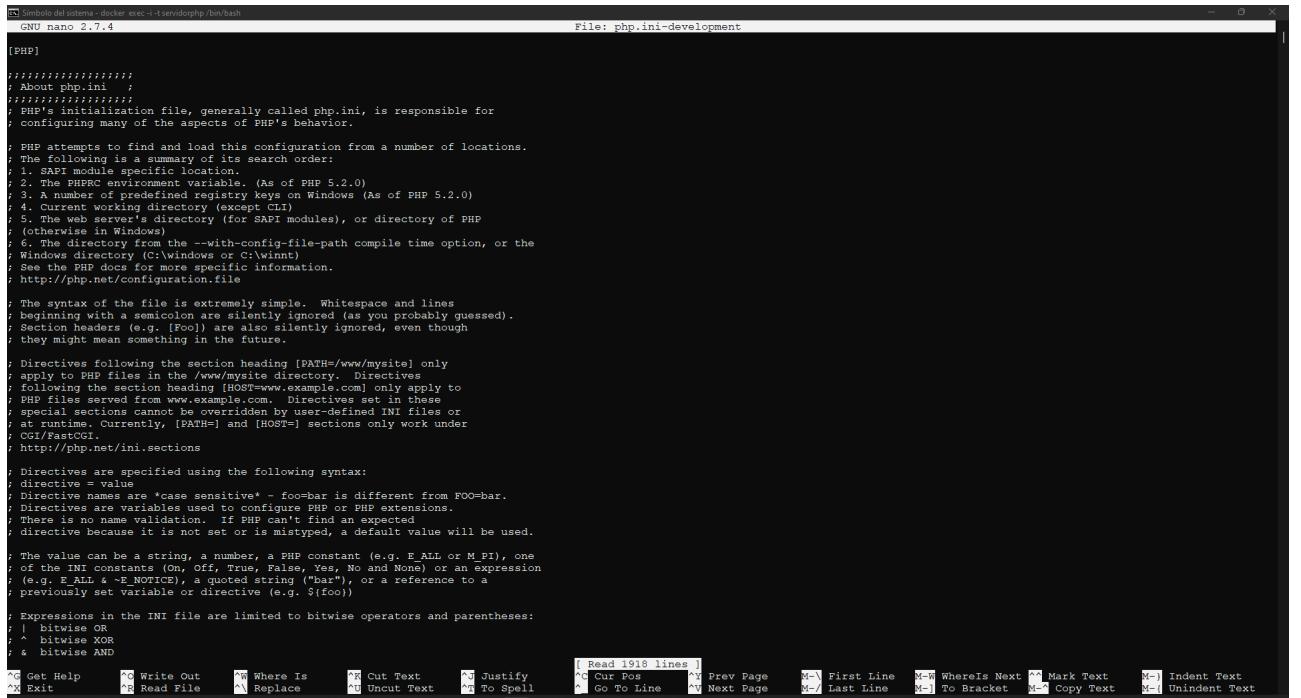
```
apt-get install nano
```

```
root@6dbce79d46e1:/usr/local/lib/php# apt-get update
Hit:1 http://security.debian.org/debian-security stretch/updates InRelease
Ign:2 http://deb.debian.org/debian stretch InRelease
Hit:3 http://deb.debian.org/debian stretch-updates InRelease
Hit:4 http://deb.debian.org/debian stretch Release
Reading package lists... Done
root@6dbce79d46e1:/usr/local/lib/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 64 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 amd64 2.7.4-1 [485 kB]
Fetched 485 kB in 0s (1093 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 76, < line 1.)
debconf: falling back to frontend: Readline
Selecting previously unselected package nano.
(Reading database ... 18637 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/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group pico) doesn't exist
root@6dbce79d46e1:/usr/local/lib/php#
```

- Abrimos el archivo **php.ini-development** mediante el siguiente comando:



```
nano php.ini-development
```



Luego, buscamos el apartado de rutas dinámicas y añadimos la ruta que teníamos copiada, y salimos del editor con **Ctrl + X, y y Enter**:

```
#####
; Dynamic Extensions ;
#####

; If you wish to have an extension loaded automatically, use the following
; syntax:
;
;   extension=modulename.extension
;
; For example, on Windows:
;
;   extension=msql.dll
;
; ... or under UNIX:
;
;   extension=msql.so
;
; ... or with a path:
;
;   extension=/path/to/extension/msql.so

/usr/local/lib/php/extensions/no-debug-non-zts-20151012/
```

- Aplicamos el mismo procedimiento para el archivo de `php.ini-production`:

```
root@6dbce79d46e1:/usr/local/etc/php# nano php.ini-development
root@6dbce79d46e1:/usr/local/etc/php# nano php.ini-production
root@6dbce79d46e1:/usr/local/etc/php#
```

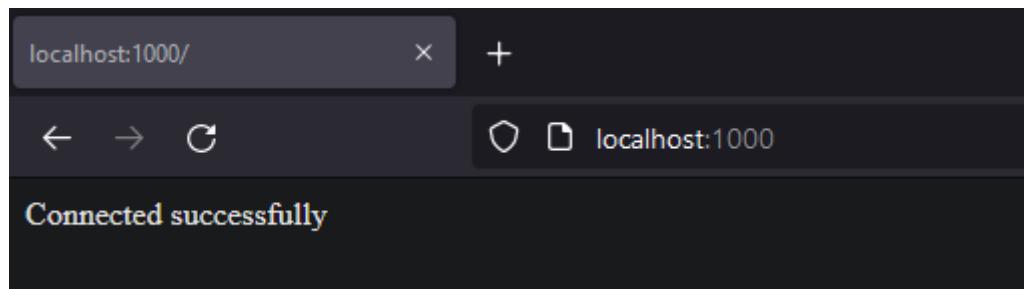
- Reiniciamos el contenedor de PHP con el siguiente comando:

```
docker restart servidorphp
```

```
C:\Users\carlo>docker restart servidorphp
servidorphp
```

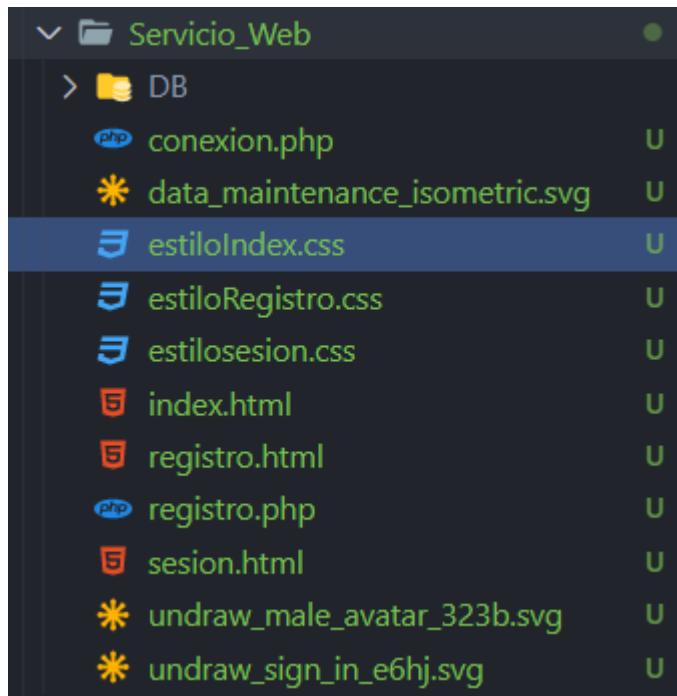
```
C:\Users\carlo>docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED      STATUS      PORTS          NAMES
6dbce79d46e1   php:7.0-apache "docker-php-entrypoi..." 15 hours ago  Up 39 seconds  0.0.0.0:1000->80/tcp, 33060/tcp, 0.0.0.0:3307->3306/tcp   servidorphp
e2749101e305   mysql:5.7    "docker-entrypoint.s..." 15 hours ago  Up 4 hours   3306/tcp, 0.0.0.0:3307->3306/tcp   database
```

- Recargamos el navegador, y esto será lo que debe aparecer:



Página Web

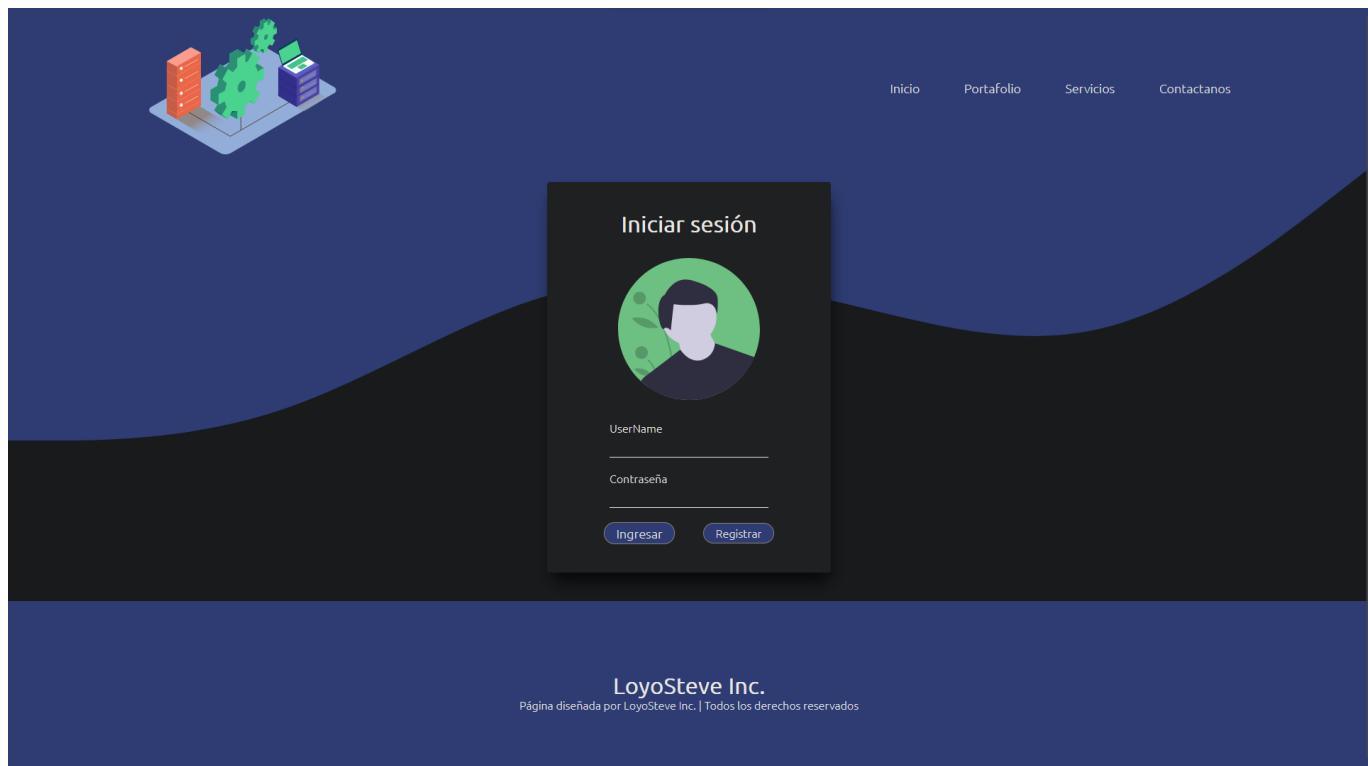
Para efectos del taller vamos a añadir algunos archivos a nuestro directorio `Servicio_Web`:

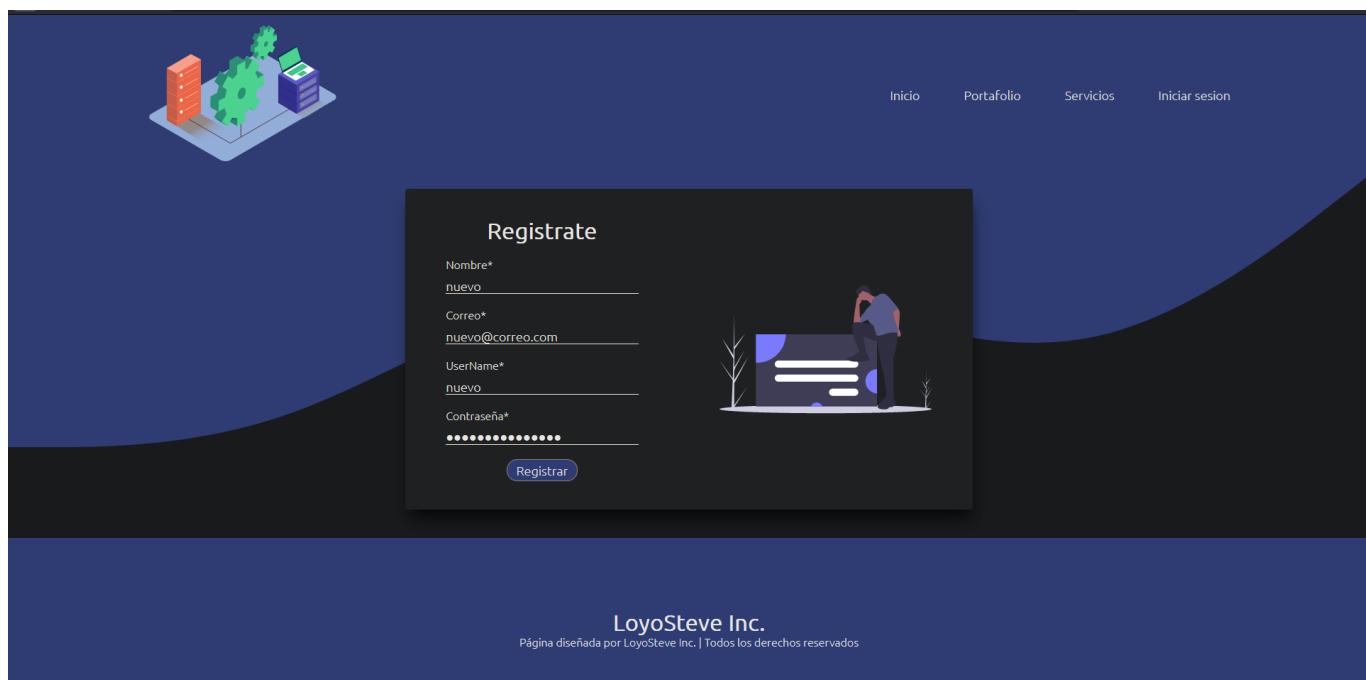
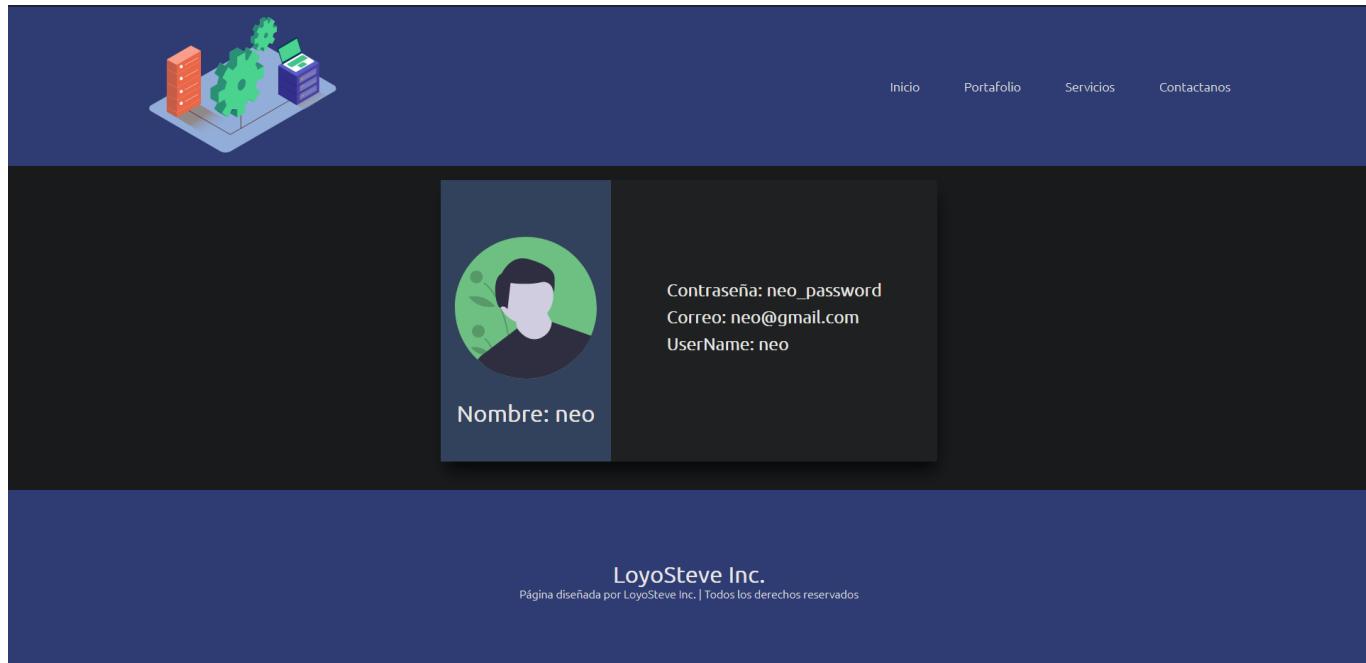


Dentro del archivo `conexion.php` y `registro.php` hacemos los siguientes cambios en la variable de conexión a la base de datos:

```
$con = mysqli_connect("database:3306","root","password","usuarios");
```

Si volvemos al navegador en la dirección `http://localhost:1000`, vamos a observar lo siguiente:





Si vamos al contenedor de la base de datos, podemos verificar que se ha registrado el usuario:

```
docker exec -i -t database bash
```

```
mysql -u root -p
```

```
use usuarios;
```

```
select * from clientes;
```

```
C:\Users\carlo>docker exec -i -t database bash
root@e2749101e305:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
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> use usuarios;
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> select * from clientes;
+-----+-----+-----+-----+
| username | nombre | correo           | contra        |
+-----+-----+-----+-----+
| neo     | neo    | neo@gmail.com   | neo_password |
| nuevo   | nuevo  | nuevo@correo.com | xFEPjtGgs9WKJpq |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Docker-Compose

Dentro del directorio **Servicio_Web** creamos el archivo **docker-compose.yml**, dentro del cual tendremos la siguiente información:

```
version: '3'

services:
  mysql:
    image: mysql:5.7
    container_name: docker-mysql
    environment:
      MYSQL_DATABASE: usuarios
      MYSQL_ROOT_USER: user_docker
      MYSQL_USER: user_docker
```

```

    MYSQL_PASSWORD: password
    MYSQL_ROOT_PASSWORD: password
ports:
  - "3307:3306"
restart: always

```

web:

```

image: php:7.0-apache
container_name: docker-php
ports:
  - "1000:80"
volumes:
  - ./www:/var/www/html
links:
  - mysql

```

Vamos a bajar nuestro contenedores:

```
docker stop database & docker stop servidorphp
```

```
C:\Users\carlo>docker stop database & docker stop servidorphp
database
servidorphp
```

Y ahora usamos el siguiente comando dentro del directorio **Servicio_Web**:

```
docker compose-up
```

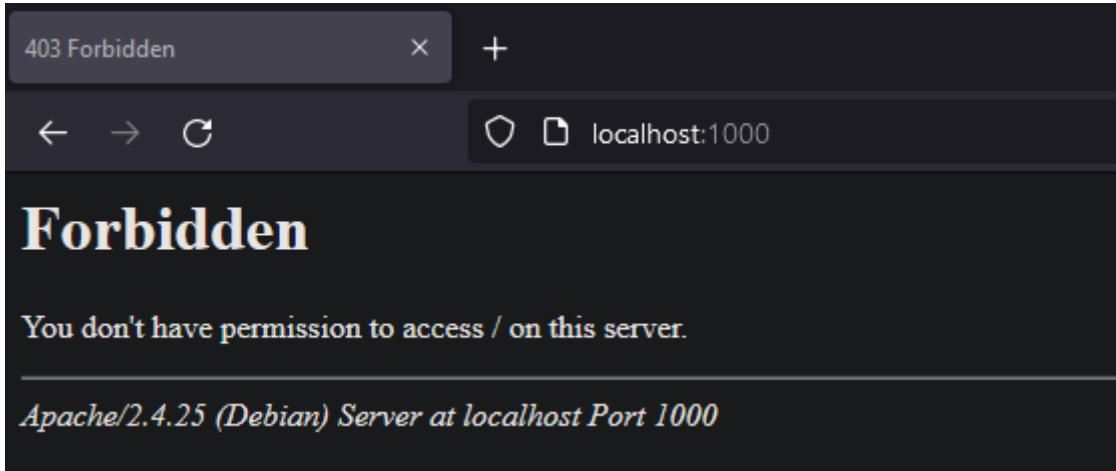
```
C:\Users\carlo\Documents\Noveno_Semestre\DevOps\Tercer_Corte\P3Ti_Guia_Final\Servicio_Web>docker-compose up
Recreating docker-mysql ... done
Recreating docker-php ... done
Recreating docker-database ... done
Recreating docker-servidorphp ... done
docker-mysql | 2022-05-18 18:57:41+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.38-1debian10 started.
docker-mysql | 2022-05-18 18:57:41+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
docker-mysql | 2022-05-18 18:57:41+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.38-1debian10 started.
docker-mysql | 2022-05-18 18:57:41+00:00 [Note] [Entrypoint]: Initializing database files
docker-mysql | 2022-05-18T10:57:41.844473Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
docker-mysql | 2022-05-18T10:57:42.051006Z 0 [Warning] InnoDB: New log files created, LSN=45790
docker-mysql | 2022-05-18T10:57:42.096365Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
docker-mysql | 2022-05-18T10:57:42.109565Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 653
9e74e-d6d0-1ec8-8777-000000000002
docker-mysql | 2022-05-18T10:57:42.110121Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
docker-mysql | 2022-05-18T10:57:42.365370Z 0 [Warning] A deprecated TLS version TlSv1 is enabled. Please use TlSv1.2 or higher.
docker-mysql | 2022-05-18T10:57:42.365421Z 0 [Warning] A deprecated TLS version TlSv1.1 is enabled. Please use TlSv1.2 or higher.
docker-mysql | 2022-05-18T10:57:42.366532Z 0 [Warning] CA certificate ca.pem is self signed.
docker-mysql | 2022-05-18T10:57:42.422885Z 1 [Warning] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
docker-php | AH000558: apache2 Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
docker-php | AH000558: apache2 Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
docker-php | [Wed May 18 18:57:42.561694 2022] [mpm_prefork:notice] [pid 1] AH000163: Apache/2.4.25 (Debian) PHP/7.0.33 configured -- resuming normal operations
docker-mysql | 2022-05-18 18:57:44+00:00 [Note] [Entrypoint]: Database files initialized
docker-mysql | 2022-05-18 18:57:44+00:00 [Note] [Entrypoint]: Starting temporary server
docker-mysql | 2022-05-18 18:57:44+00:00 [Note] [Entrypoint]: Waiting for server startup
docker-mysql | 2022-05-18T10:57:44.940735Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
docker-mysql | 2022-05-18T10:57:44.942082Z 0 [Note] mysqld (mysqld 5.7.38) starting as process 78 ...
docker-mysql | 2022-05-18T10:57:44.944538Z 0 [Note] InnoDB: PUNCH HOLE support available
docker-mysql | 2022-05-18T10:57:44.944584Z 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
docker-mysql | 2022-05-18T10:57:44.944589Z 0 [Note] InnoDB: Uses event mutexes
docker-mysql | 2022-05-18T10:57:44.944592Z 0 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
docker-mysql | 2022-05-18T10:57:44.944595Z 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
docker-mysql | 2022-05-18T10:57:44.944600Z 0 [Note] InnoDB: Using Linux native AIO
docker-mysql | 2022-05-18T10:57:44.944604Z 0 [Note] InnoDB: Using CPU functions
docker-mysql | 2022-05-18T10:57:44.944608Z 0 [Note] InnoDB: Using CPU auto-vectorized instructions
docker-mysql | 2022-05-18T10:57:44.945202Z 0 [Note] InnoDB: Using CPU auto-vectorized instructions
docker-mysql | 2022-05-18T10:57:44.946540Z 0 [Note] InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M
docker-mysql | 2022-05-18T10:57:44.953148Z 0 [Note] InnoDB: Completed initialization of buffer pool
docker-mysql | 2022-05-18T10:57:44.955465Z 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man page of setpriority().
docker-mysql | 2022-05-18T10:57:44.967290Z 0 [Note] InnoDB: Highest supported file format is Barracuda.
docker-mysql | 2022-05-18T10:57:44.975298Z 0 [Note] InnoDB: Creating shared tablespace for temporary tables
docker-mysql | 2022-05-18T10:57:44.975398Z 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
docker-mysql | 2022-05-18T10:57:44.980764Z 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
docker-mysql | 2022-05-18T10:57:44.983813Z 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s) are active.
docker-mysql | 2022-05-18T10:57:44.983820Z 0 [Note] InnoDB: 7.38 started; log sequence number 2749976
docker-mysql | 2022-05-18T10:57:44.990436Z 0 [Note] InnoDB: Error log initialized from ./ib_logfile0
docker-mysql | 2022-05-18T10:57:44.991049Z 0 [Note] Plugin 'FEDEDATED' is loaded.
docker-mysql | 2022-05-18T10:57:44.991752Z 0 [Note] InnoDB: Buffer pool(s) load completed at 220518 18:57:44
docker-mysql | 2022-05-18T10:57:44.996290Z 0 [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to enable SSL support using them.
docker-mysql | 2022-05-18T10:57:44.996331Z 0 [Note] Skipping generation of SSL certificates as certificate files are present in data directory.
docker-mysql | 2022-05-18T10:57:44.996336Z 0 [Warning] A deprecated TLS version TlSv1.1 is enabled. Please use TlSv1.2 or higher.
docker-mysql | 2022-05-18T10:57:44.996338Z 0 [Warning] A deprecated TLS version TlSv1.1 is enabled. Please use TlSv1.2 or higher.
docker-mysql | 2022-05-18T10:57:44.996871Z 0 [Warning] CA certificate ca.pem is self signed.
```

```

docker-mysql | 2022-05-18T18:57:48.954423Z 0 [Note] Shutting down plugin 'mysql_native_password'
docker-mysql | 2022-05-18T18:57:48.954542Z 0 [Note] Shutting down plugin 'binlog'
docker-mysql |
docker-mysql | 2022-05-18 18:57:49+00:00 [Note] [Entrypoint]: Temporary server stopped
docker-mysql |
docker-mysql | 2022-05-18 18:57:49+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.
docker-mysql |
docker-mysql | 2022-05-18T18:57:49.899023Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
docker-mysql | 2022-05-18T18:57:49.900161Z 0 [Note] mysqld (mysqld 5.7.38) starting as process 1 ...
docker-mysql | 2022-05-18T18:57:49.902644Z 0 [Note] InnoDB: PUNCH HOLE support available
docker-mysql | 2022-05-18T18:57:49.902736Z 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
docker-mysql | 2022-05-18T18:57:49.902744Z 0 [Note] InnoDB: Uses event mutexes
docker-mysql | 2022-05-18T18:57:49.902748Z 0 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
docker-mysql | 2022-05-18T18:57:49.902750Z 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
docker-mysql | 2022-05-18T18:57:49.902753Z 0 [Note] InnoDB: Using Linux native AIO
docker-mysql | 2022-05-18T18:57:49.902756Z 0 [Note] InnoDB: Number of pools: 1
docker-mysql | 2022-05-18T18:57:49.903559Z 0 [Note] InnoDB: 1 thread(s) using 16384 pages (65.500 MB) of buffer pool memory
docker-mysql | 2022-05-18T18:57:49.905519Z 0 [Note] InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M
docker-mysql | 2022-05-18T18:57:49.912005Z 0 [Note] InnoDB: completed initialization of buffer pool
docker-mysql | 2022-05-18T18:57:49.913980Z 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man page of setpriority().
docker-mysql | 2022-05-18T18:57:49.925397Z 0 [Note] InnoDB: Highest supported file format is Barracuda.
docker-mysql | 2022-05-18T18:57:49.932393Z 0 [Note] InnoDB: Creating shared tablespace for temporary tables
docker-mysql | 2022-05-18T18:57:49.932500Z 0 [Note] InnoDB: Setting file './ibtmp1' size is now 12 MB.
docker-mysql | 2022-05-18T18:57:49.945966Z 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
docker-mysql | 2022-05-18T18:57:49.946485Z 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s) are active.
docker-mysql | 2022-05-18T18:57:49.946517Z 0 [Note] InnoDB: 32 non-redo rollback segment(s) are active.
docker-mysql | 2022-05-18T18:57:49.947246Z 0 [Note] InnoDB: 5.7.38 started; log sequence number 12660054
docker-mysql | 2022-05-18T18:57:49.947300Z 0 [Note] InnoDB: reading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
docker-mysql | 2022-05-18T18:57:49.947831Z 0 [Note] InnoDB: Buffers initialized from a log file
docker-mysql | 2022-05-18T18:57:49.950253Z 0 [Note] InnoDB: Buffer pool(s) load completed at 220518 18:57:49
docker-mysql | 2022-05-18T18:57:49.952455Z 0 [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to enable SSL support using them.
docker-mysql | 2022-05-18T18:57:49.952500Z 0 [Note] Skipping generation of SSL certificates as certificate files are present in data directory.
docker-mysql | 2022-05-18T18:57:49.952506Z 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
docker-mysql | 2022-05-18T18:57:49.952508Z 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
docker-mysql | 2022-05-18T18:57:49.953008Z 0 [Warning] Ca certificate ca.pem is self signed.
docker-mysql | 2022-05-18T18:57:49.953056Z 0 [Note] Skipping generation of RSA key pair as key files are present in data directory.
docker-mysql | 2022-05-18T18:57:49.953499Z 0 [Note] Server hostname (bind-address): '''; port: 3306
docker-mysql | 2022-05-18T18:57:49.953506Z 0 [Note] IPv6 is available.
docker-mysql | 2022-05-18T18:57:49.953743Z 0 [Note] - '::' resolves to '::'.
docker-mysql | 2022-05-18T18:57:49.953777Z 0 [Note] Server socket created on Ip: '::'.
docker-mysql | 2022-05-18T18:57:49.955691Z 0 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
docker-mysql | 2022-05-18T18:57:49.964403Z 0 [Note] Event Scheduler: Loaded 0 events
docker-mysql | 2022-05-18T18:57:49.964717Z 0 [Note] mysqld: ready for connections
docker-mysql | Version: '5.7.38' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server (GPL)

```

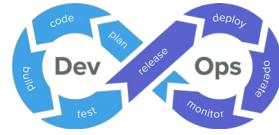
Volvemos a intentar correr nuestra aplicación dentro del navegador, y recibiremos el siguiente mensaje debido a que no se configuro los directorio indices ([index.php](#), [index.html](#)).



```

docker-mysql | 2022-05-18T18:57:49.964403Z 0 [Note] Event Scheduler: Loaded 0 events
docker-mysql | 2022-05-18T18:57:49.964717Z 0 [Note] mysqld: ready for connections
docker-mysql | Version: '5.7.38' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server (GPL)
docker-mysql | 172.19.0.1 - - [18/May/2022:18:59:36 +0000] "GET /estiloIndex.css HTTP/1.1" 404 507 "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0"
docker-php | [Wed May 18 18:59:54.788676 2022] [:error] [pid 18] [client 172.19.0.1:54742] script '/var/www/html/conexion.php' not found or unable to stat, referer: http://localhost:1000/
docker-php | 172.19.0.1 - - [18/May/2022:18:59:54 +0000] "POST /conexion.php HTTP/1.1" 404 504 "http://localhost:1000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0"
docker-php | [Wed May 18 19:00:17:721398 2022] [:error] [pid 19] [client 172.19.0.1:54744] script '/var/www/html/registro.php' not found or unable to stat, referer: http://localhost:1000/registro.php
docker-php | 172.19.0.1 - - [18/May/2022:19:00:17 +0000] "POST /registro.php HTTP/1.1" 404 504 "http://localhost:1000/registro.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0"
docker-php | 172.19.0.1 - - [18/May/2022:19:00:25 +0000] "GET /registro.html HTTP/1.1" 404 505 "http://localhost:1000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0"
docker-php | [Wed May 18 19:01:21.925245 2022] [:autoindex:error] [pid 20] [client 172.19.0.1:54747] AH01276: Cannot serve directory /var/www/html/: No matching DirectoryIndex (index.php,index.html) found, and server-generated directory index forbidden by Options directive
docker-php | [Wed May 18 19:01:21.925245 2022] [:autoindex:error] [pid 21] [client 172.19.0.1:54748] AH01276: Cannot serve directory /var/www/html/: No matching DirectoryIndex (index.php,index.html) found, and server-generated directory index forbidden by Options directive
docker-php | 172.19.0.1 - - [18/May/2022:19:01:21 +0000] "GET / HTTP/1.1" 403 503 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0"

```



Taller Final DevOps - Kubernetes

Juana Valentina Mendoza Santamaría – 2274474

Carlos David Páez Ferreira – 2251184

Parte 1: Tutorial interactivo

Crear un cluster con minikube

1. Ir al link de kubernetes: <https://kubernetes.io/docs/tutorials/hello-minikube/>

Kubernetes Documentation / Tutorials / Hello Minikube

Hello Minikube

This tutorial shows how to run a sample app on Kubernetes using minikube and Katacoda. Katacoda provides a free, in-browser Kubernetes environment.

Note: You can also follow this tutorial if you've installed minikube locally. See [minikube start](#) for installation instructions.

Objectives

- Deploy a sample application to minikube.
- Run the app.
- View application logs.

Before you begin

This tutorial provides a container image that uses NGINX to echo back all the requests.



2. Comenzar minikube con el comando: **minikube start**

```
$ minikube start
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

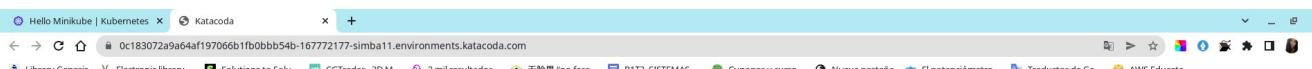
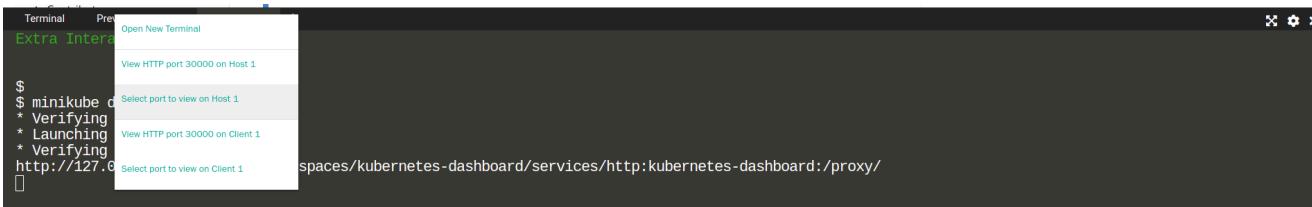
X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
- kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
- Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
```

3. Abrir el panel de control de Kubernets en un navegador, con el comando: **minikube dashboard**

```
$ minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
http://127.0.0.1:41885/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```

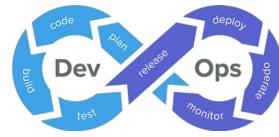
4. Realizar solo en el entorno Katacoda. En la parte superior del panel de la terminal, se da clic en el signo más y luego clic en Seleccionar puerto para ver en el Host 1:



Web Preview Port Selector

To display a HTTP server, please enter the port number below

Display Port



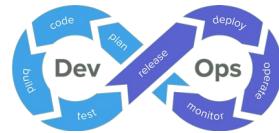
5. Escribir **30000** y realizar clic en “Display Port”:

Screenshot of the Kubernetes Dashboard showing the Overview page. The URL in the address bar is <https://167772177-30000-simba11.environments.katacoda.com/#/overview?namespace=default>.

The sidebar on the left lists various Kubernetes resources under categories like Workloads, Service, Config and Storage, and Cluster.

The main content area displays the following tables:

- Services**: Shows a single entry for the 'kubernetes' service in the default namespace, with Cluster IP 10.96.0.1 and Internal Endpoints kubernetes:443 TCP and kubernetes:0 TCP.
- Config Maps**: Shows a single entry for 'kube-root-ca.crt' in the default namespace.
- Secrets**: Shows a single entry for 'default-token-5vp8s' in the default namespace, which is a service account token.
- Storage Classes**: Shows a single entry for 'k8s-in-minikube-hostpath'.



Abrir Dashboard con URL

- Si no se desea abrir un navegador web, se corre el comando: **minikube dashboard url**

```
Terminal Preview Port 30000 Terminal 2 Terminal 3 +
Extra Interactive Bash Terminal

$ minikube dashboard --url
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
http://127.0.0.1:34047/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```

Crear una implementación

- El Pod corre un Container basado en imagen provista de Docker. Para crear un despliegue que maneje un Pod, se utiliza el comando:

kubectl create deployment hello-node –image=k8s.gcr.io/echoserver:1.4

```
Terminal Preview Port 30000 Terminal 2 Terminal 3 Terminal 4 +
Extra Interactive Bash Terminal

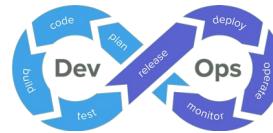
$ kubectl create deployment hello-node --image=k8s.gcr.io/echoserver:1.4
deployment.apps/hello-node created
$
```

- Para visualizar el despliegue, se digita el comando: **kubectl get deployments**

```
Terminal Preview Port 30000 Terminal 2 Terminal 3 Terminal 4 +
$ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
hello-node 1/1 1 1 2m36s
$
```

- Para visualizar el Pod: **kubectl get pods**

```
Terminal Preview Port 30000 Terminal 2 Terminal 3 Terminal 4 +
$ kubectl get pods
NAME READY STATUS RESTARTS AGE
hello-node-7567d9fdc9-66m5f 1/1 Running 0 4m18s
$
```



4. A continuación, para visualizar los eventos del cluster: **kubectl get events**

```

Terminal  Preview Port 30000 Terminal 2 Terminal 3 Terminal 4 +
$ kubectl get events
LAST SEEN   TYPE    REASON          OBJECT
8m45s       Normal   Scheduled      pod/hello-node-7567d9fdc9-66m5f
8m44s       Normal   Pulling        pod/hello-node-7567d9fdc9-66m5f
8m43s       Normal   Pulled         pod/hello-node-7567d9fdc9-66m5f
8m43s       Normal   Created        pod/hello-node-7567d9fdc9-66m5f
8m43s       Normal   Started        pod/hello-node-7567d9fdc9-66m5f
8m45s       Normal   SuccessfulCreate replicaset/hello-node-7567d9fdc9
8m45s       Normal   ScalingReplicaSet deployment/hello-node
32m         Normal   Starting        node/minikube
32m         Normal   NodeHasSufficientMemory node/minikube
32m         Normal   NodeHasNoDiskPressure node/minikube
32m         Normal   NodeHasSufficientPID node/minikube
32m         Normal   NodeAllocatableEnforced node/minikube
31m         Normal   RegisteredNode  node/minikube
31m         Normal   NodeReady       node/minikube
31m         Normal   Starting        node/minikube
                                                 MESSAGE
                                                 Successfully assigned default/hello-node-7567d9fdc9-66m5f to minikube
                                                 Pulling image "k8s.gcr.io/echoserver:1.4"
                                                 Successfully pulled image "k8s.gcr.io/echoserver:1.4" in 510.281479ms
                                                 Created container echoserver
                                                 Started container echoserver
                                                 Created pod: hello-node-7567d9fdc9-66m5f
                                                 Scaled up replica set hello-node-7567d9fdc9 to 1
                                                 Starting kubelet.
                                                 Node minikube status is now: NodeHasSufficientMemory
                                                 Node minikube status is now: NodeHasNoDiskPressure
                                                 Node minikube status is now: NodeHasSufficientPID
                                                 Updated Node Allocatable limit across pods
                                                 Node minikube event: Registered Node minikube in Controller
                                                 Node minikube status is now: NodeReady
                                                 Starting kube-proxy.
$ 

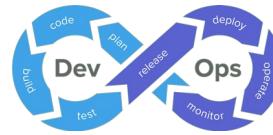
```

5. Para visualizar la configuración, se utiliza el comando: **kubectl config view**

```

Terminal  Preview Port 30000 Terminal 2 Terminal 3 Terminal 4 +
$ kubectl config view
apiVersion: v1
clusters:
- cluster:
  certificate-authority: /root/.minikube/ca.crt
  extensions:
  - extension:
    last-update: Sat, 28 May 2022 20:40:21 UTC
    provider: minikube.sigs.k8s.io
    version: v1.18.0
    name: cluster_info
    server: https://10.0.0.17:8443
    name: minikube
contexts:
- context:
  cluster: minikube
  extensions:
  - extension:
    last-update: Sat, 28 May 2022 20:40:21 UTC
    provider: minikube.sigs.k8s.io
    version: v1.18.0
    name: context_info
    namespace: default
    user: minikube
    name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /root/.minikube/profiles/minikube/client.crt
    client-key: /root/.minikube/profiles/minikube/client.key
$ 

```



Crear un servicio

1. Exponer el Pod al internet público usando el comando:

```
kubectl expose deployment hello-node --type=LoadBalancer --port=8080
```

```
Terminal Preview Port 30000 +
$ kubectl expose deployment hello-node --type=LoadBalancer --port=8080
service/hello-node exposed
$
```

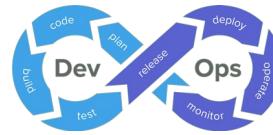
nota: la bandera --type=LoadBalancer indica que tu quieres exponer tu servicio fuera del cluster.

El código de la aplicación dentro de la imagen k8s.gcr.io/echoserver solo escucha el puerto TCP 8080. Si quiere usar kubectl expose para exponer en un puerto diferente, los clientes no podrán conectarse al otro puerto.

2. Para visualizar el servicio creado, digitar el comando: **kubectl get services**

```
Terminal Preview Port 30000 +
$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
hello-node  LoadBalancer  10.103.121.27  <pending>      8080:31712/TCP  2m30s
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      4m43s
$
```

notas: En los proveedores de cloud que soportan balanceo de carga, una IP externa debe ser provista para acceder al servicio. En minikube, el tipo LoadBalancer hace accesible el servicio mediante el comando minikube service



3. Correr el siguiente comando: **minikube service hello-node**

```
Terminal Preview Port 30000 +
$ minikube service hello-node
|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|
| default   | hello-node | 8080 | http://10.0.0.9:31712 |
|-----|
* Opening service default/hello-node in default browser...
Minikube Dashboard is not supported via the interactive terminal experience.

Please click the 'Preview Port 30000' link above to access the dashboard.
This will now exit. Please continue with the rest of the tutorial.

X Exiting due to HOST_BROWSER: exit status 1
*
* If the above advice does not help, please let us know:
- https://github.com/kubernetes/minikube/issues/new/choose
```

4. Abrir el entorno de Katacoda:



5. El puerto tiene 5 dígitos que se muestra en oposición al puerto 8080 en la salida del punto 3. El número se genera aleatoriamente, corresponde al 31712.

Hello Minikube | Kubernetes X Katacoda x +

← → × ⌂ 167772169-30369-olllie09.environments.katacoda.com

Library Genesis V Electronic library... S Solutions to Solu... CGTrader - 3D M... 2 mil resultados... 无脸男 "no face..." P1T3_SISTEMAS_... Cupones y curso... Nueva pestaña El potenciómetro... Traductor d...

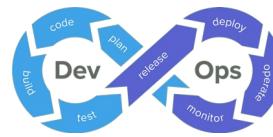
Connecting to Port 30369

We're currently trying to connect to a HTTP service running on 30369. Services can sometimes take a few moments to start, even up to five minutes.

Display a different port

If the service is running on a different port then please enter it below

Display Port



```

Hello Minikube | Kubernetes x https://167772169-31712-ollie09.environments.katacoda.com +
← → C ⌂ 167772169-31712-ollie09.environments.katacoda.com
Library Genesis Electronic library... Solutions to Solu... CGTrader - 3D M... 2 mil resultados... 无脸男 "no face..." P1T3_SISTEMAS_... Cupones y curso... Nueva pestaña El potenciōr

CLIENT VALUES:
client_address=172.18.0.1
command=GET
real_path/
query=nil
request_version=1.1
request_uri=http://167772169-31712-ollie09.environments.katacoda.com:8080/

SERVER VALUES:
server_version=nginx: 1.10.0 - lua: 10001

HEADERS RECEIVED:
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding=gzip
accept-language=es-419,es;q=0.9
host=167772169-31712-ollie09.environments.katacoda.com
referer=https://167772169-30369-ollie09.environments.katacoda.com/
sec-ch-ua="Not A;Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"
sec-ch-ua-mobile=?0
sec-ch-ua-platform="Linux"
sec-fetch-dest=document
sec-fetch-mode=navigate
sec-fetch-site=same-site
sec-fetch-user=?1
upgrade-insecure-requests=1
user-agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36
via=1.1 google
x-cloud-trace-context=936d89abedc46670efcee8a71c690f8f/16198241373820607742
x-forwarded-for=181.60.111.199, 35.201.124.219, 35.191.10.97, 35.226.247.243, 172.18.0.14
x-forwarded-proto=https
x-katacoda-host=ollie09
x-real-ip=35.226.247.243
BODY:
-no body in request-

```

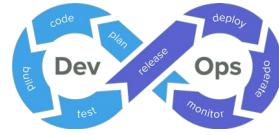
Habilitar los complementos

1. Se listan los complementos actualmente soportados: **minikube addons list**

```

Terminal Preview Port 30000 +
$ minikube addons list
-----|-----|-----|
ADDON NAME | PROFILE | STATUS |
-----|-----|-----|
ambassador | minikube | disabled |
auto-pause | minikube | disabled |
csi-hostpath-driver | minikube | disabled |
dashboard | minikube | enabled | ✓ |
default-storageclass | minikube | enabled | ✓ |
efk | minikube | disabled |
freshpod | minikube | disabled |
gcp-auth | minikube | disabled |
gvisor | minikube | disabled |
helm-tiller | minikube | disabled |
ingress | minikube | disabled |
ingress-dns | minikube | disabled |
istio | minikube | disabled |
istio-provisioner | minikube | disabled |
kubevirt | minikube | disabled |
logviewer | minikube | disabled |
metallb | minikube | disabled |
metrics-server | minikube | enabled | ✓ |
nvidia-driver-installer | minikube | disabled |
nvidia-gpu-device-plugin | minikube | disabled |
olm | minikube | disabled |
pod-security-policy | minikube | disabled |
registry | minikube | disabled |
registry-aliases | minikube | disabled |
registry-creds | minikube | disabled |
storage-provisioner | minikube | enabled | ✓ |
storage-provisioner-gluster | minikube | disabled |
volumesnapshots | minikube | disabled |
-----|-----|-----|
$ 

```



2. Para habilitar un complemento, por ejemplo: metrics-server, se utiliza el comando:

minikube addons enable metrics-server

```
Terminal Preview Port 30000 +
$ minikube addons enable metrics-server
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
* The 'metrics-server' addon is enabled
$
```

3. Para visualizar el Pod y el Servicio que se creo, se digita el comando:

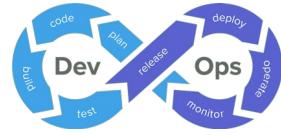
kubectl get pod,svc -n kube-system

```
Terminal Preview Port 30000 +
$ kubectl get pod,svc -n kube-system
NAME                                         READY   STATUS    RESTARTS   AGE
pod/coredns-74ff55c5b-fwcqx                1/1     Running   0          25m
pod/etccd-minikube                          1/1     Running   0          25m
pod/kube-apiserver-minikube                 1/1     Running   0          25m
pod/kube-controller-manager-minikube        1/1     Running   0          25m
pod/kube-proxy-qnvnm                        1/1     Running   0          25m
pod/kube-scheduler-minikube                 1/1     Running   0          25m
pod/metrics-server-56c4f8c9d6-gq575         1/1     Running   0          25m
pod/storage-provisioner                      1/1     Running   0          25m

NAME                  TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
service/kube-dns     ClusterIP  10.96.0.10    <none>       53/UDP, 53/TCP, 9153/TCP   25m
service/metrics-server ClusterIP  10.106.114.33  <none>       443/TCP     25m
$
```

4. Para desactivar el metrics-server, digitar el comando: **minikube addons disable metrics-server**

```
Terminal Preview Port 30000 +
$ minikube addons disable metrics-server
* "The 'metrics-server' addon is disabled
$
```



Limpieza

1. Para limpiar los recursos que se crearon en el cluster, digitar:

kubectl delete service hello-node & kubectl delete deployment hello-node

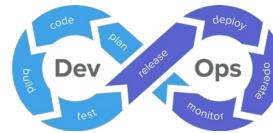
```
Terminal Preview Port 30000 +
$ kubectl delete service hello-node & kubectl delete deployment hello-node
[1] 12568
deployment.apps "hello-node" deleted
$ service "hello-node" deleted
$
```

2. Opcionalmente, se detiene la máquina virtual de Minikube: **minikube stop**

```
Terminal Preview Port 30000 +
$ minikube stop
* Stopping node "minikube" ...
* 1 nodes stopped.
$
```

3. Asimismo, se puede eliminar la máquina virtual de Minikube: **minikube delete**

```
Terminal Preview Port 30000 +
$ minikube delete
* Uninstalling Kubernetes v1.20.2 using kubeadm ...
* Deleting "minikube" in none ...
* Removed all traces of the "minikube" cluster.
$
```



Parte 2: Kubectl

1. Se verifica la versión del kubectl con el comando: **kubectl version --client=true**

```
juana@fedora:~
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora ~]$ kubectl version --client=true
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.7", GitCommit:"b56e432f2191419647a6a13b9f5867801850f969", GitTreeState:"archive", BuildDate:"2022-03-01T00:00:00Z", GoVersion:"go1.16.15", Compiler:"gc", Platform:"linux/amd64"}
(base) [juana@fedora ~]$
```

2. Se inicia minikube con el comando: **minikube start**

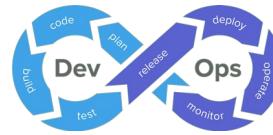
```
juana@fedora:~
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora ~]$ minikube start
🕒 minikube v1.25.2 en Fedora 35
💡 Using the docker driver based on existing profile
docker is currently using the btrfs storage driver, consider switching to overlay2 for better performance
👉 Starting control plane node minikube in cluster minikube
_PULLING_ base image...
Updating the running docker "minikube" container ...
🚀 Preparando Kubernetes v1.23.3 en Docker 20.10.12...
  kubelet.housekeeping-interval=5s
💡 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
(base) [juana@fedora ~]$
```

3. Se verifica que esté corriendo, con el comando: **minikube status**

```
juana@fedora:~
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora ~]$ minikube status
minikube
  type: Control Plane
  host: Running
  kubelet: Running
  apiserver: Running
  kubeconfig: Configured

(base) [juana@fedora ~]$
```

nota: en caso de la distribución de Fedora no es posible instalar Docker-Desktop.



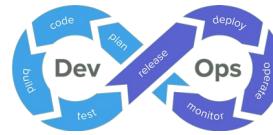
4. Para observar los nodos del cluster de kubernetes, se utiliza el comando: **kubectl get nodes**

```
juana@fedora:~ [juana@fedora ~]$ kubectl get nodes
NAME      STATUS    ROLES      AGE   VERSION
minikube  Ready     control-plane,master  26d   v1.23.3
[juana@fedora ~]$
```

Parte 3: Implementación de la aplicación PHP Guestbook con Redis

1. Ir a la página <https://kubernetes.io/docs/tutorials/stateless-application/guestbook/>

The screenshot shows a browser window with the URL <https://kubernetes.io/docs/tutorials/stateless-application/guestbook/>. The page is titled "Example: Deploying PHP Guestbook application with Redis". The left sidebar contains a navigation menu with links like Home, Getting started, Concepts, Tasks, and various Tutorial sections. The main content area describes the tutorial and lists objectives. The right sidebar provides options to edit the page, create child pages, and print the section.



Creación del despliegue de Redis

El archivo de manifiesto, que se incluye a continuación, especifica un controlador de implementación que ejecuta una sola réplica de Redis Pod.

1. Se aplica la implementación de Redis desde el archivo redis-leader-deployment.yaml, por medio del comando:

```
kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml
deployment.apps/redis-leader created
(base) [juana@fedora Descargas]$
```

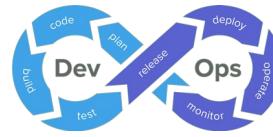
2. Posteriormente, se consulta la lista de Pods, para verificar que el Pod de Redis esté corriendo:

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-66c9c7669-hnqvk	1/1	Running	1 (108m ago)	18d
nginx-deployment-66c9c7669-q4qf9	1/1	Running	1 (108m ago)	18d
nginx-deployment-v85nz	1/1	Running	1 (108m ago)	18d
redis-leader-766465cd9c-b6gr4	1/1	Running	0	2m48s

3. Para ver los logs del Pod de Redis líder, correr el comando:

```
kubectl logs -f deployment/redis-leader
```

```
juana@fedora:~/Descargas -- kubectl logs -f deployment/redis-leader
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl logs -f deployment/redis-leader
1:C 28 May 2022 22:19:44.089 # 000000000000 Redis is starting 000000000000
1:C 28 May 2022 22:19:44.090 # Redis version=6.0.5, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 28 May 2022 22:19:44.090 # 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 28 May 2022 22:19:44.093 * Running mode=standalone, port=6379.
1:M 28 May 2022 22:19:44.093 # Server initialized
1:M 28 May 2022 22:19:44.094 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with
Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain
the setting after a reboot. Redis must be restarted after THP is disabled.
1:M 28 May 2022 22:19:44.094 * Ready to accept connections
```



Creación del servicio líder de Redis

La aplicación del libro de visitas debe comunicarse con Redis para escribir sus datos. Debe aplicar un Servicio para enviar el tráfico al Redis Pod. Un Servicio define una política para acceder a los Pods.

1. Se aplica un servicio de Redis desde el archivo redis-leader-server.yaml, con el comando:

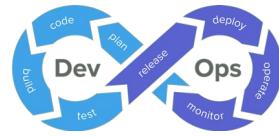
```
kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-service.yaml
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juana@fedora:~/Descargas — kubectl logs -f deployment/redis-leader
juana@fedora:~/Descargas
(base) [juana@fedora Descargas]$ kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-service.yaml
service/redis-leader created
(base) [juana@fedora Descargas]$
```

2. Ahora, se listan los servicios para verificar que el servicio de Redis esté corriendo, con el comando:

```
kubectl get service
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juana@fedora:~/Descargas — kubectl logs -f deployment/r...
juana@fedora:~/Descargas
(base) [juana@fedora Descargas]$ kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP      26d
redis-leader   ClusterIP   10.99.66.196  <none>        6379/TCP     109s
(base) [juana@fedora Descargas]$
```



Configurar los seguidores de Redis

Si bien el líder de Redis es un pod único, puede hacerlo altamente disponible y cumplir con las demandas de tráfico agregando algunos seguidores de Redis o réplicas.

1. Se aplica el despliegue de Redis desde el archivo redis-follower-deployment.yaml, por medio del comando:

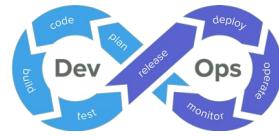
```
kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-deployment.yaml
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-deployment.yaml
deployment.apps/redis-follower created
(base) [juana@fedora Descargas]$
```

2. Se verifica que las 2 replicas de Redis estén corriendo:

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-66c9c7669-hnqvk	1/1	Running	1 (122m ago)	18d
nginx-deployment-66c9c7669-q4qf9	1/1	Running	1 (122m ago)	18d
nginx-deployment-v85nz	1/1	Running	1 (122m ago)	18d
redis-follower-84fcc94dfc-g7z9q	1/1	Running	0	109s
redis-follower-84fcc94dfc-p245s	1/1	Running	0	109s
redis-leader-766465cd9c-b6gr4	1/1	Running	0	17m



Creación del Servicio de seguidores de Redis

La aplicación guestbook necesita comunicarse con los seguidores de Redis para leer la data. Para hacer que los seguidores de Redis sean detectables, debemos configurar otro servicio.

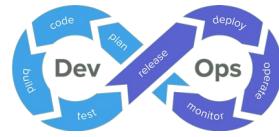
1. Se aplica el servicio de Redis desde el archivo redis-follower-service.yaml. Con el comando:

```
kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-service.yaml
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-service.yaml
service/redis-follower created
(base) [juana@fedora Descargas]$
```

2. Para consultar que los servicios de Redis estén corriendo, se digita el comando: **kubectl get service**

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	26d
redis-follower	ClusterIP	10.107.83.34	<none>	6379/TCP	110s
redis-leader	ClusterIP	10.99.66.196	<none>	6379/TCP	11m



Configurar y exponer el Frontend de Guestbook: Crear un despliegue del Frontend

Ahora que tiene el almacenamiento Redis de su libro de visitas en funcionamiento, inicie los servidores web del libro de visitas. Al igual que los seguidores de Redis, la interfaz se implementa mediante una implementación de Kubernetes.

La aplicación de libro de visitas utiliza una interfaz PHP. Está configurado para comunicarse con los servicios de líder o seguidor de Redis, dependiendo de si la solicitud es de lectura o escritura. La interfaz expone una interfaz JSON y sirve una experiencia de usuario basada en jQuery-Ajax.

1. Se aplica el despliegue del frontend desde frontend-deployment.yaml, con el comando:

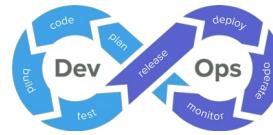
kubectl apply -f <https://k8s.io/examples/application/guestbook/frontend-deployment.yaml>

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml
deployment.apps/frontend created
(base) [juana@fedora Descargas]$
```

2. Se consultan los pods para verificar que las tres replicas del frontend estén corriendo:

kubectl get pods -l app=guestbook -l tier=frontend

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl get pods -l app=guestbook -l tier=frontend
NAME          READY   STATUS    RESTARTS   AGE
frontend-57df59b89c-4fcc4   1/1     Running   0          2m2s
frontend-57df59b89c-htz94   1/1     Running   0          2m2s
frontend-57df59b89c-q5wdk   1/1     Running   0          2m2s
(base) [juana@fedora Descargas]$
```



Creando el servicio del Frontend

Solo se puede acceder a los servicios de Redis que se aplicó dentro del clúster de Kubernetes porque el tipo predeterminado para un servicio es ClusterIP. ClusterIP proporciona una única dirección IP para el conjunto de pods a los que apunta el servicio. Solo se puede acceder a esta dirección IP dentro del clúster.

Si desea que los invitados puedan acceder a su libro de visitas, debe configurar el Servicio de interfaz para que sea visible externamente, de modo que un cliente pueda solicitar el Servicio desde fuera del clúster de Kubernetes. Sin embargo, un usuario de Kubernetes puede usar el reenvío de puerto de kubectl para acceder al servicio aunque use un ClusterIP.

1. Se aplica el servicio del frontend desde el archivo frontend-service.yaml, con el comando:

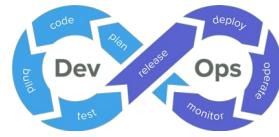
```
kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
service/frontend created
(base) [juana@fedora Descargas]$
```

2. Se consulta la lista de servicios para verificar que esté corriendo el servicio del Frontend:

```
kubectl get services
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
frontend   ClusterIP  10.107.78.85  <none>        80/TCP      25s
kubernetes ClusterIP  10.96.0.1    <none>        443/TCP     26d
redis-follower   ClusterIP  10.107.83.34  <none>        6379/TCP    16m
redis-leader    ClusterIP  10.99.66.196  <none>        6379/TCP    26m
(base) [juana@fedora Descargas]$
```



Observar el servicio del frontend via kubectl port-forward

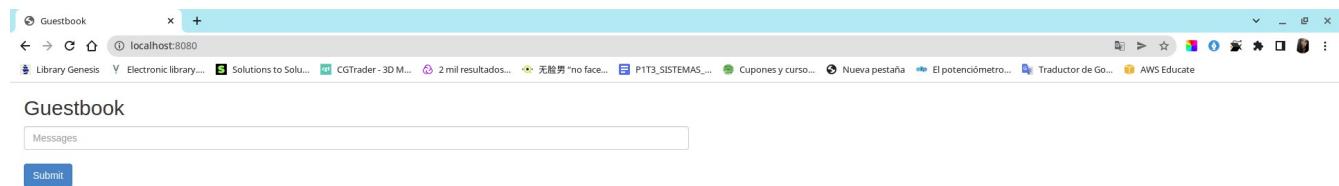
1. Se corre el siguiente comando por el puerto delantero 8080 en la maquina local, al puerto 80 en el servicio: **kubectl port-forward svc/frontend 8080:80**

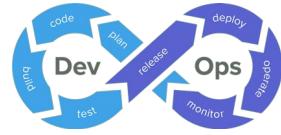
```

juana@fedora:~/Descargas — kubectl port-forward svc/frontend 8080:80
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl port-forward svc/frontend 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [:1]:8080 -> 80

```

2. Se abre la siguiente dirección en el navegador: <http://localhost:8080>





Observar el servicio del frontend via LoadBalancer

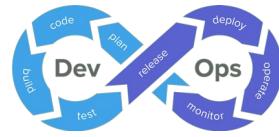
Si se implementa el manifiesto frontend-service.yaml con el tipo: LoadBalancer, se necesita encontrar la dirección IP para ver su libro de visitas.

1. Se ejecuta el siguiente comando para obtener la dirección IP del servicio del frontend:

kubectl get service frontend

A screenshot of a terminal window titled 'juana@fedora:~/Descargas'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', 'Pestañas', and 'Ayuda'. Below the menu is a toolbar with three colored dots (yellow, green, red). The terminal content shows the command 'kubectl get service frontend' being run, followed by its output:

```
(base) [juana@fedora Descargas]$ kubectl get service frontend
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
frontend   ClusterIP  10.107.78.85  <none>          80/TCP      7m3s
(base) [juana@fedora Descargas]$
```



Escalar el Frontend Web

Puede escalar hacia arriba o hacia abajo según sea necesario porque sus servidores se definen como un servicio que utiliza un controlador de implementación.

1. Se ejecuta el siguiente comando para aumentar el número de pods del frontend:

```
kubectl scale deployment frontend --replicas=5
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juana@fedora:~/Descargas — kubectl port-forward svc/fronten...
juana@fedora:~/Descargas
(base) [juana@fedora Descargas]$ kubectl scale deployment frontend --replicas=5
deployment.apps/frontend scaled
(base) [juana@fedora Descargas]$
```

2. Se consultan los pods para verificar el número de pods del frontend corriendo el comando:

```
kubectl get pods
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juana@fedora:~/Descargas — kubectl port-forward svc/fronten...
juana@fedora:~/Descargas
(base) [juana@fedora Descargas]$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
frontend-57df59b89c-4fcc4   1/1     Running   0          24m
frontend-57df59b89c-8tx86   1/1     Running   0          95s
frontend-57df59b89c-d7cbg   1/1     Running   0          95s
frontend-57df59b89c-htz94   1/1     Running   0          24m
frontend-57df59b89c-q5wdk   1/1     Running   0          24m
nginx-deployment-66c9c7669-hnqvk 1/1     Running   1 (155m ago) 18d
nginx-deployment-66c9c7669-q4qf9   1/1     Running   1 (155m ago) 18d
nginx-deployment-v85nz        1/1     Running   1 (155m ago) 18d
redis-follower-84fcc94dfc-g7z9q  1/1     Running   0          34m
redis-follower-84fcc94dfc-p245s  1/1     Running   0          34m
redis-leader-766465cd9c-b6gr4   1/1     Running   0          50m
(base) [juana@fedora Descargas]$
```



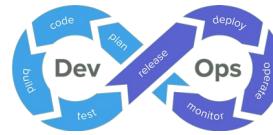
3. Para disminuir el número de pods del frontend, correr el siguiente comando:

kubectl scale deployment frontend --replicas=2

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juana@fedora:~/Descargas — kubectl port-forward svc/fro... juana@fedora:~/Descargas
(base) [juana@fedora Descargas]$ kubectl scale deployment frontend --replicas=2
deployment.apps/frontend scaled
(base) [juana@fedora Descargas]$
```

4. Para verificar el número de pods del frontend correr el comando: **kubectl get pods**

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juana@fedora:~/Descargas — kubectl port-forward svc/fro... juana@fedora:~/Descargas
(base) [juana@fedora Descargas]$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
frontend-57df59b89c-4fcc4   1/1     Running   0          27m
frontend-57df59b89c-htz94   1/1     Running   0          27m
nginx-deployment-66c9c7669-hnqvk  1/1     Running   1 (158m ago) 18d
nginx-deployment-66c9c7669-q4qf9  1/1     Running   1 (158m ago) 18d
nginx-deployment-v85nz       1/1     Running   1 (158m ago) 18d
redis-follower-84fcc94dfc-g7z9q  1/1     Running   0          38m
redis-follower-84fcc94dfc-p245s  1/1     Running   0          38m
redis-leader-766465cd9c-b6gr4   1/1     Running   0          53m
(base) [juana@fedora Descargas]$
```



Limpieza

1. Eliminar los despliegues y servicios, así como cualquier Pod corriendo. Se usa labels para eliminar múltiples recursos con un comando.

```
kubectl delete deployment -l app=redis
```

```
kubectl delete service -l app=redis
```

```
kubectl delete deployment frontend
```

```
kubectl delete service frontend
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl delete deployment -l app=redis
deployment.apps "redis-follower" deleted
deployment.apps "redis-leader" deleted
(base) [juana@fedora Descargas]$ kubectl delete service -l app=redis
service "redis-follower" deleted
service "redis-leader" deleted
(base) [juana@fedora Descargas]$ kubectl delete deployment frontend
deployment.apps "frontend" deleted
(base) [juana@fedora Descargas]$ kubectl delete service frontend
service "frontend" deleted
(base) [juana@fedora Descargas]$
```

2. Se consulta la lista de pods que están corriendo:

```
kubectl get pods
```

```
juana@fedora:~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
(base) [juana@fedora Descargas]$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-66c9c7669-hnqvk  1/1     Running   1 (165m ago)   18d
nginx-deployment-66c9c7669-q4qf9  1/1     Running   1 (165m ago)   18d
nginx-deployment-v85nz          1/1     Running   1 (165m ago)   18d
(base) [juana@fedora Descargas]$
```