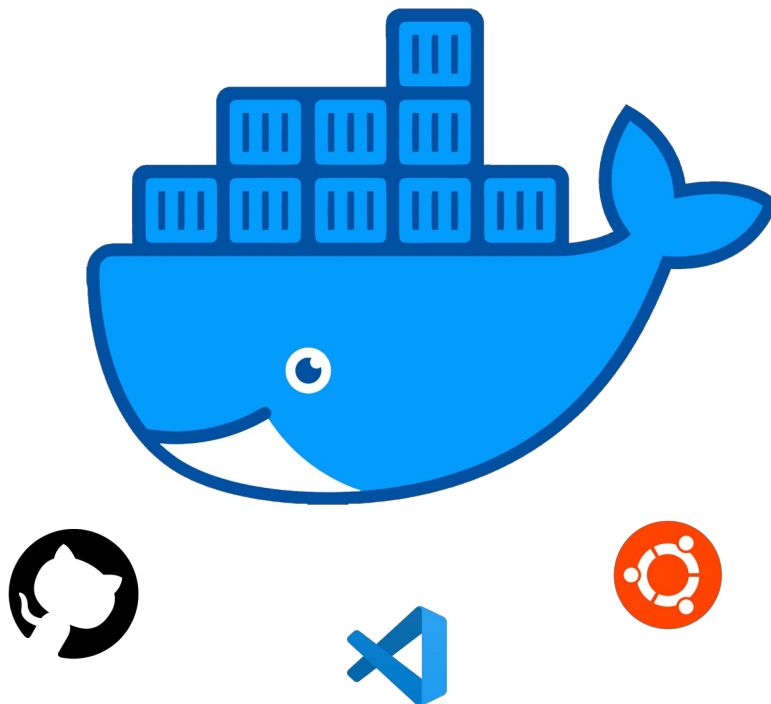


SGE

UD01 – Actividad Evaluable



Marcos Zahonero

Índice

Introducción.....	3
Tiempo utilizado.....	4
Objetivos.....	5
Desarrollo.....	6
Actividad 1.....	6
Actividad 2.....	8
Unidad 3.....	8
Caso práctico 2.....	8
Caso práctico 3.....	10
Unidad 5.....	11
Caso práctico 1.....	11
Unidad 6.....	13
Caso práctico 1.....	13
Caso práctico 4.....	14
Unidad 7.....	16
Caso práctico 1.....	16
Actividad 3.....	20
Actividad 4.....	20
Problemas encontrados, solución y/o sugerencias.....	21
Problema 1# - No tener interfaz gráfica.....	21
Problema 2# - Docker Compose (<i>services.environment must be a mapping</i>).....	22
Problema 3# - Problema con VSC y SSH.....	23
Sugerencia 4# - Docker Desktop.....	24
Sugerencia 5# - Quitar el sudo en comandos.....	24
Conclusión.....	25
Bibliografía/webgrafía.....	26

Introducción

Durante el anterior año vimos una introducción a Docker, la cual veremos un poco más este año en esta actividad de Docker, donde usaremos nuevamente GitHub y Docker para crear contenedores útiles.

Tiempo utilizado

Referente al tiempo utilizado para esta actividad ha constado en 2 veces de hacerlo, el viernes 27 he utilizado 1 hora y media, para empezar con la actividad, tuve problemas ya que cometí el error que no lo estaba haciendo en un contenedor y tuve que empezar de nuevo, en ese momento hice mitad actividad. La segunda vez que continué con la actividad fue el domingo 29 durante 4/5 horas aproximadamente, en la que terminé la actividad 2 e hice el documento, en este tiempo me tomo más ya que tuve que arreglar problemas como he comentado en su apartado.

Total: 6 horas y media aproximadas. (No pude contarlas bien)

Objetivos

- Volver a obtener los conocimientos básicos en Docker
- Poner en práctica GitHub, para ponerle obtener más experiencia
- Conocer el uso del remoto desde VS con Ubuntu Server

Desarrollo

Actividad 1

El repositorio de GitHub: [Maek0s/SGE \(github.com\)](https://github.com/Maek0s/SGE)

The screenshot shows the GitHub repository page for 'Maek0s / SGE'. The repository is private and has 1 watcher, 0 forks, and 0 stars. The page includes a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. Below the navigation bar, there are two main sections: 'Start coding with Codespaces' and 'Add collaborators to this repository'. The 'Quick setup' section provides instructions for cloning the repository using HTTPS or SSH, and for creating a new repository or pushing an existing one from the command line.

Maek0s / SGE Private

Unwatch 1 Fork 0 Star 0

Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.
[Create a codespace](#)

Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

HTTPS SSH <https://github.com/Maek0s/SGE.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

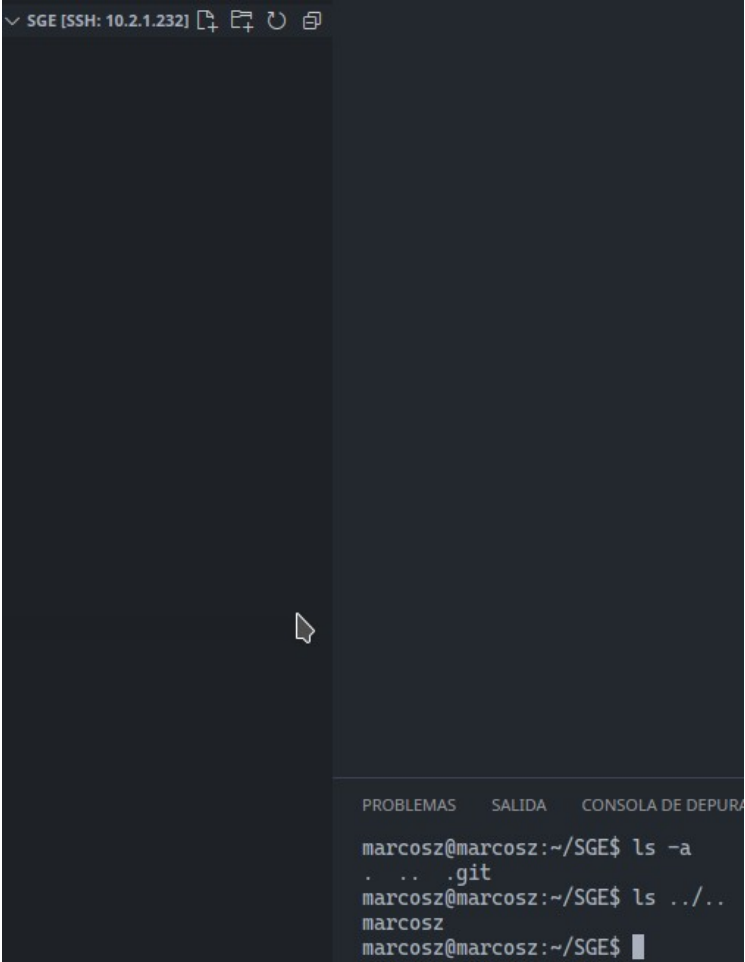
...or create a new repository on the command line

```
echo "# SGE" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Maek0s/SGE.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Maek0s/SGE.git
git branch -M main
git push -u origin main
```

Conectamos VS con nuestra máquina virtual con Ubuntu server, esto se ha conseguido con SSH conectando con una extensión de VS que permite usar SSH y entrar utilizando el comando “ssh (nombreServer)@(ip)” (sin comillas) y luego introducir la contraseña del usuario.



```
SGE [SSH: 10.2.1.232]
marcosz@marcosz:~/SGE$ ls -a
.  ..  .git
marcosz@marcosz:~/SGE$ ls ../../
marcosz
marcosz@marcosz:~/SGE$
```

Actividad 2

Unidad 3

Caso práctico 2

Preparamos el Apache para utilizar Wordpress haciendo un fichero de configuración en la siguiente ruta **“/etc/apache2/sites-available/wordpress.conf”** que configurará el sitio Wordpress, copiamos el siguiente texto en el archivo creado:

```
GNU nano 6.2 /etc/apache2/sites-available/wordpress.conf
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>

[ Read 13 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Recargamos Apache con los siguientes comandos:

```
root@bdec4148bb73:/# a2ensite wordpress
Enabling site wordpress.
To activate the new configuration, you need to run:
  service apache2 reload
root@bdec4148bb73:/# service apache2 reload
Usage: apache2 {start|stop|graceful-stop|restart|reload|force-reload}
root@bdec4148bb73:/# service apache2 reload
* Reloading Apache httpd web server apache2
*
root@bdec4148bb73:/# a2ensite wordpress
Site wordpress already enabled
root@bdec4148bb73:/# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 restart
root@bdec4148bb73:/# service apache2 restart
* Restarting Apache httpd web server apache2
```


Instalamos MariaDB y creamos una base de datos, un usuario y entregamos los permisos al usuario creado, después actualizamos los privilegios para guardar cambios.

```
installation should now be secure.

Thanks for using MariaDB!
root@bdec4148bb73:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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

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

MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wordpress'@'%' IDENTIFIED BY 'MiPass-2023';
Query OK, 0 rows affected (0.032 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.045 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

Salimos de la configuración de MariaDB y reiniciamos todo para verificar, en este momento deberíamos comprobar entrando al localhost para ver que tal ha quedado pero nos fiamos de los comandos. -> *

```
MariaDB [(none)]> exit
Bye
root@bdec4148bb73:/# nano /etc/wordpress/config-localhost.php
root@bdec4148bb73:/# service apache2 start
 * Starting Apache httpd web server apache2
 *
root@bdec4148bb73:/# service mariadb start
 * Starting MariaDB database server mariadbd
root@bdec4148bb73:/# [ OK ]
```

Caso práctico 3

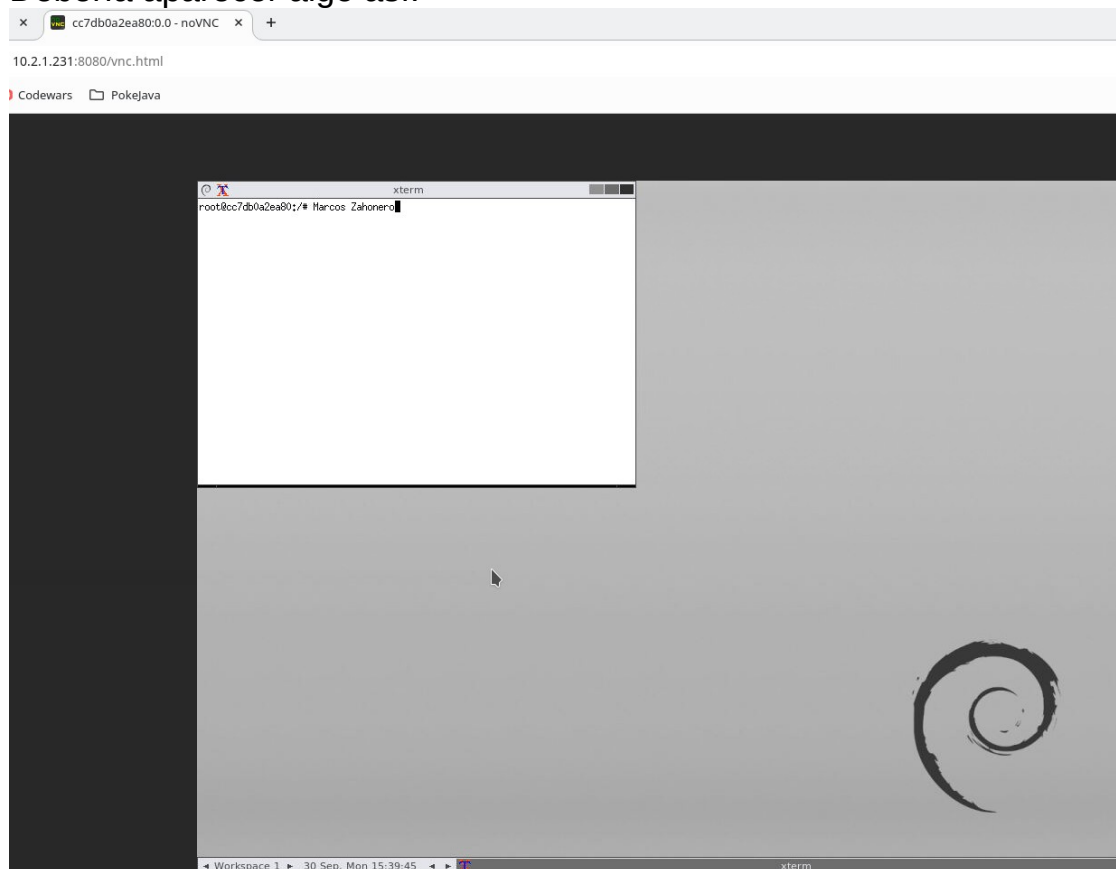
Iniciamos el contenedor con el siguiente comando **`docker run --rm -it -p 8080:8080 theasp/novnc`**

```
docker run --rm -it -p 8080:8080 theasp/novnc
```

⚠ **Atención:** como es una prueba de concepto, hemos creado la imagen con el parámetro `--rm`, por lo cual al cerrar la terminal el contenedor se borrará.

```
root@marcosz:/etc/apache2/sites-available# docker run --rm -it -p 8080:8080 theasp/novnc
Unable to find image 'theasp/novnc:latest' locally
latest: Pulling from theasp/novnc
ba83bbfca944: Pull complete
92e0276c4450: Pull complete
ed1f829c06de: Pull complete
Digest: sha256:ee874a13eeab9a8d8d990832039dac151fe2c7496a29010453d3aae4bf26e456
Status: Downloaded newer image for theasp/novnc:latest
```

Debería aparecer algo así:



Unidad 5

Caso práctico 1

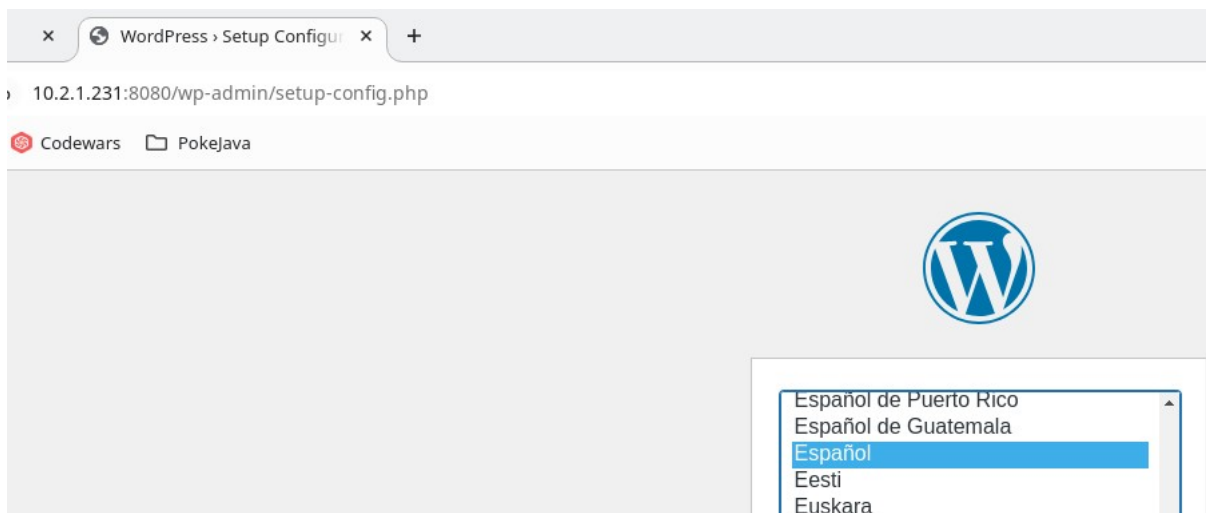
Lo primero que haremos será utilizar la base de datos de MariaDB, de la siguiente forma:

```
docker run --name nuestromariadb --network redwp -v /home/(tuUsuario)/mariadbdata:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=cefireroot -e MARIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass -e MARIADB_DATABASE=cefiredb -d mariadb:10.6
```

```
root@marcosz:/etc/apache2/sites-available# docker run --name nuestromariadb --network redwp -v /home/marcosz/mariadbdata:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=cefireroot -e MARIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass -e MARIADB_DATABASE=cefiredb -d mariadb:10.6
Unable to find image 'mariadb:10.6' locally
10.6: Pulling from library/mariadb
602d8ad51b81: Pull complete
906bf2208a98: Pull complete
06102c8e12da: Pull complete
79322a8d69e6: Pull complete
9cd48d1e81c5: Pull complete
eb82348957f6: Pull complete
b1dd5e075a7e: Pull complete
712710a7d704: Pull complete
```

Creamos el contenedor de Wordpress y lo podemos hacer así: **`docker run --name nuestrowp --network redwp -p 8080:80 -d wordpress`**

En este momento deberíamos entrar al localhost y saldría lo siguiente, en donde tendríamos que configurar la web, poniendo el idioma y poniendo los parámetros de la base de datos creados anteriormente:



Configuramos los parámetros como he dicho anteriormente (en clase):

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

Nombre de la base de datos
El nombre de la base de datos que quieres usar con WordPress.

Nombre de usuario
El nombre de usuario de tu base de datos.

Contraseña [Mostrar](#)
La contraseña de tu base de datos.

Servidor de la base de datos
Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

Prefijo de tabla
Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Enviar](#)

Después vamos a ver como podríamos migrar de versión de nuestra base de datos, para que en un futuro podamos migrar sin ningún problema.

```
docker run --name nuestromariadb --network redwp -v /home/(tuUsuario)/mariadbdata:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=cefireroot -e MARIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass -e MARIADB_DATABASE=cefiredb -d mariadb:10.7
```

```
root@marcosz:/etc/apache2/sites-available# docker stop nuestromariadb
nuestrumariadb
root@marcosz:/etc/apache2/sites-available# docker rm nuestromariadb
nuestrumariadb
root@marcosz:/etc/apache2/sites-available# docker run --name nuestromariadb --network redwp -v /home/marcosz/mariadbdata:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=cefireroot -e MARIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass -e MARIADB_DATABASE=cefiredb -d mariadb:10.7
Unable to find image 'mariadb:10.7' locally
10.7: Pulling from library/mariadb
99803d4b97f3: Pull complete
b8bc823a83fd: Pull complete
16685f710f5d: Pull complete
b5660ff63058: Pull complete
8b13582f0741: Pull complete
7d98acc143e6: Pull complete
6e4fbb6a63e0: Pull complete
d3da17671559: Pull complete
```

Si todo ha funcionado podríamos entrar al localhost de nuevo y funcionaría como antes.

Unidad 6

Caso práctico 1

Tenemos que realizar un archivo Compose, en mi caso por problemas desde casa con la conectividad entre VS y Ubuntu Server tuve que hacerlo a mano, aunque ahora veremos un pequeño problema con este código que en mi caso me dio error, después lo veremos.

```
GNU nano 6.2 docker-compose.yml *
version: "3.9"

services:
  db:
    image: mariadb:10.11.2
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: cefireroot
      MARIADB_DATABASE: cefiredb
      MARIADB_USER: cefireuser
      MARIADB_PASSWORD: cefirepass_

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress

volumes:
  db_data:
```

Ahora hacemos “**docker compose up -d**” con sudo en caso de que no estemos con el “**sudo su**” y saltará un error, que será normal, ya que a mi también me ocurrió en caso de que no sea así puede ser algo bueno o malo pero si puedes seguir con la actividad genial, en caso de que nos hayamos topado con un error vamos a hacer clic aquí y seguir las instrucciones: [Click!](#)

Si todo ha funcionado al hacerlo de nuevo deberíamos recibir esto, sino es el caso me asegurate de haber tabulado bien y que no hayan fallos en el archivo.

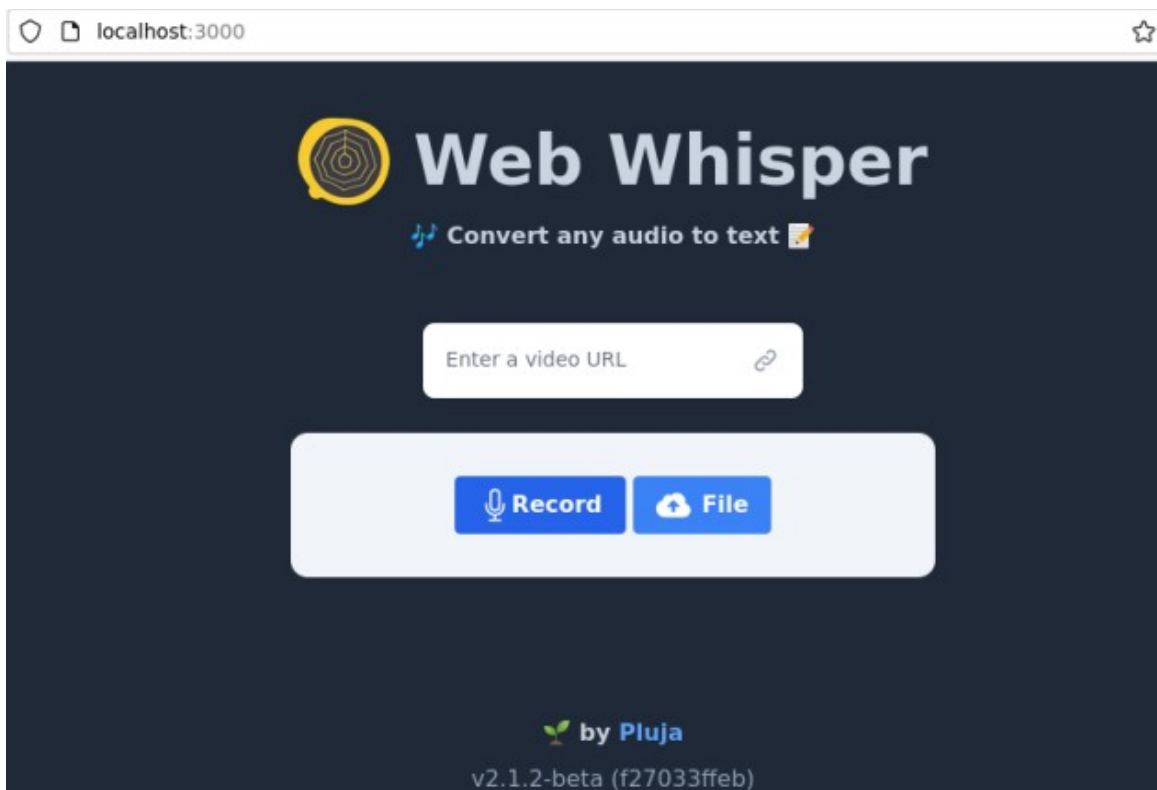
```
marcosz@marcosz:~/Desktop/miwordpress$ sudo docker compose up -d
[sudo] password for marcosz:
[+] Running 8/9
 * db 8 layers [+++++ ] 0B/0B Pulling 10.9s
 * 1bc677758ad7 Download complete 2.3s
 * 196e1740aea4 Download complete 0.5s
 * 3d4df0997938 Download complete 1.0s
 * 9fa4e4184824 Download complete 1.2s
 * 22001c5fbbcd Download complete 1.7s
 * 49ed08e2853d Download complete 6.2s
 * ca39212f2f3c Download complete 2.4s
 * 439f9b904603 Download complete 2.9s
```

```
marcosz@marcosz:~/Desktop/miwordpress$ sudo docker compose down
[+] Running 3/3
  ✖ Container miwordpress-wordpress-1   Removed          4.0s
  ✖ Container miwordpress-db-1          Removed          0.7s
  ✖ Network miwordpress_default         Removed          0.2s
```

```
marcosz@marcosz:~/Desktop/miwordpress$ sudo docker compose up -d
[+] Running 3/3
  * Network miwordpress_default          Created                                0.2s
  * Container miwordpress-db-1          Started                                0.5s
  * Container miwordpress-wordpress-1   Started                                0.1s
```

[illegible]

Entramos en <https://localhost:3000> y veremos lo siguiente:



Lo he intentado unas 5 veces, probando todo con las IPs y no se abre en el anfitrión, no puedo más.

Unidad 7

Caso práctico 1

¡Aviso! En mi caso lo realizaré con mi ordenador anfitrión por comodidad, pero debería usarse en todo caso una máquina virtual de Windows/Ubuntu con interfaz gráfica.

Vamos a Visual Studio Code

En caso de no tenerlo instalado → <https://code.visualstudio.com/>

En caso de que quieras aprender de VSC → <https://code.visualstudio.com/learn>

Vamos a las extensiones y buscamos las siguientes



Figura 1: Visual Studio Code Marketplace -> Docker

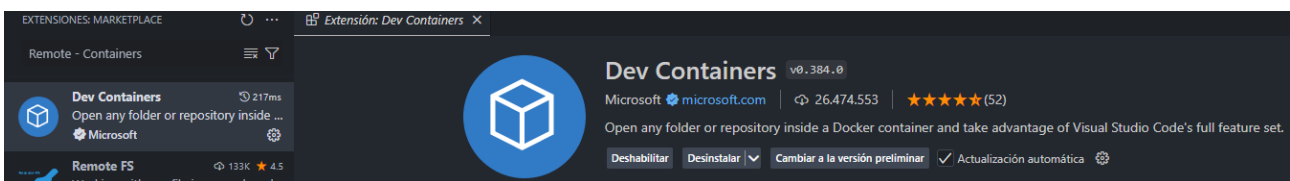


Figura 2: Visual Studio Code Marketplace -> Dev Containers

Para más información sobre contenedores en Visual Studio Code

<https://code.visualstudio.com/docs/containers/overview>

En caso de querer hacer esto en Windows si no te funciona tal vez deberías probar a descargar esto: <https://docs.docker.com/desktop/install/windows-install/>

Creamos un contenedor “**docker run -d --name servidordesarrollo -p 8080:80 php:7.2-apache**” que contendrá Apache y PHP

```
Símbolo del sistema X + v
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Maek0s>docker run -d --name servidordesarrollo -p 8080:80 php:7.2-apache
Unable to find image 'php:7.2-apache' locally
7.2-apache: Pulling from library/php
6ec7b7d162b2: Pull complete
db606474d60c: Pull complete
afb30f0cd8e0: Pull complete
3bb2e8051594: Pull complete
4c761b44e2cc: Pull complete
c2199db96575: Pull complete
1b9a9381eea8: Pull complete
fd07bbc59d34: Pull complete
72b73ab27698: Pull complete
983308f4f0d6: Pull complete
6c13f026e6da: Pull complete
e5e6cd163689: Pull complete
5c5516e56582: Pull complete
154729f6ba86: Pull complete
Digest: sha256:4dc0f0115acf8c2f0df69295ae822e49f5ad5fe849725847f15aa0e5802b55f8
Status: Downloaded newer image for php:7.2-apache
ad12bec4436f4ea2330a52bd3881d5d2cbf43c57ca12c0d06edc504212dd89bd

C:\Users\Maek0s>
```

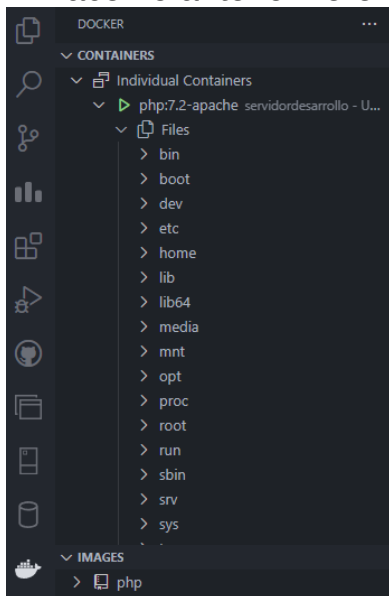
Entramos a <http://localhost:8080/> y nos saldrá lo siguiente:

Forbidden

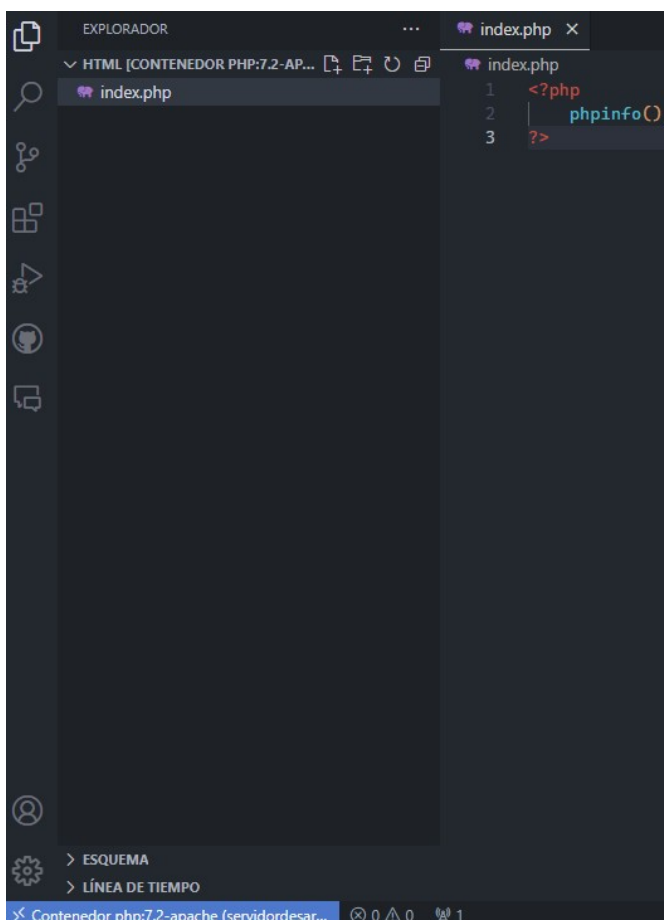
You don't have permission to access this resource.

Apache/2.4.38 (Debian) Server at localhost Port 8080


Al hacer lo anterior veremos un contenedor con lo siguiente:



Hacemos “Attach Visual Studio Code” y se nos abre un nuevo VSC, abrimos una carpeta en “/var/www/html” y creamos un archivo index.php y ponemos lo siguiente:




Accedemos al <http://localhost:8080/> de nuevo y veremos esto:

PHP Version 7.2.34


System	Linux ad12bec4436f 5.15.153.1-microsoft-standard-WSL2 #1 SMP Fri Mar 29 23:14:13 UTC 2024 x86_64
Build Date	Dec 11 2020 10:50:00
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' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' '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	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, 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 v3.2.0, Copyright (c) 1998-2018 Zend Technologies



Configuration

apache2handler

Actividad 3

Los kubernetes es una forma de llevar los contenedores, es agil y tiene mucha compatibilidad entre diferentes nubes o distribuciones, como por ejemplo Ubuntu, Google Kubernetes Engine, etc. GitHub por ejemplo utiliza Kubernetes, utiliza formas de seguridad para mantener los datos seguros, de forma que si se llega a detener un contenedor se hará una copia casi exacta para mantener los datos los máximos posibles.

Actividad 4

El repositorio de GitHub: [Maek0s/SGE \(github.com\)](https://github.com/Maek0s/SGE)

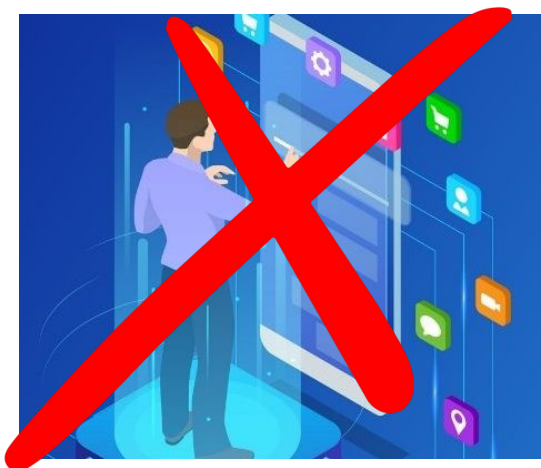


Problemas encontrados, solución y/o sugerencias

Problema 1# - No tener interfaz gráfica

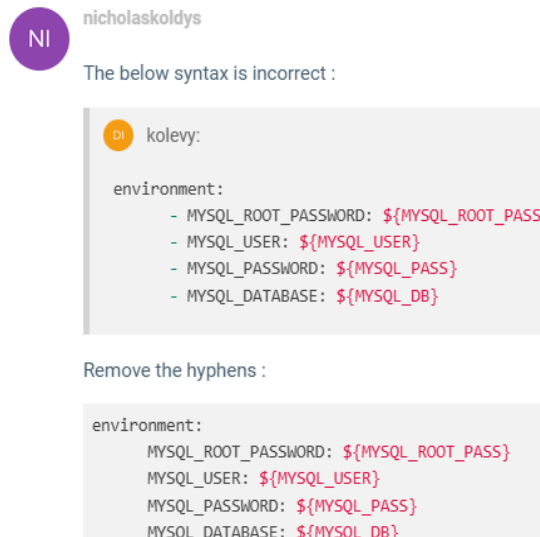
Al hacer la actividad en un Ubuntu Server sin interfaz gráfica nuestra capacidad para enviar capturas de las pruebas o revisar si ha funcionado correctamente era imposible, teníamos que fiarnos un poco de los comandos y era algo incomodo para la actividad y en mi opinión, un poco desastre.

La solución es desde el anfitrión acceder en vez de localhost poner la IP y asegurarte que este en Adaptador puente, y funcionará, en mi caso al final pude hacerlo en todas menos una que era problema al parecer de la propia actividad.



Problema 2# - Docker Compose (*services.environment must be a mapping*)

Realizando la actividad normal y siguiendo los pasos me encontré con el error de “*services.db.environment must be a mapping*” al hacer el **docker compose up -d** y lo que hice fue ir a Google y preguntar “¿Por qué?” a lo que me llevo a esta web de Docker: [Enlace](#) pero resumidamente pone lo siguiente:



En su caso era eliminar los “hyphens” que es este carácter → - ← pero vi que estaba tabulado así que probé a hacer lo mismo y funcionó.

¿Arreglado? Vuelve con la actividad haciendo [clic aquí](#).



Problema 3# - Problema con VSC y SSH

Haciendo la práctica probé a conectarme desde casa desde un Windows y tenía muchos problemas, no era por la IP sino porque me daba errores de conectividad, hacia pings entre los maquina virtual y anfrition y no funcionaban, desactive el firewall y eso empezó a funcionar pero no se conectaba aún, fui a la terminal a hacer el comando desde ahí y me dio más información:

```
C:\Users\Maek0s>ssh marcosz@192.168.1.182
Bad permissions. Try removing permissions for user: DESKTOP-77B3NCS\Usuario (S-1-5-21-240272102-1221773416-3054530593-1002) on file C:/Users/Maek0s/.ssh/config.
Bad owner or permissions on C:\\Users\\Maek0s/.ssh/config

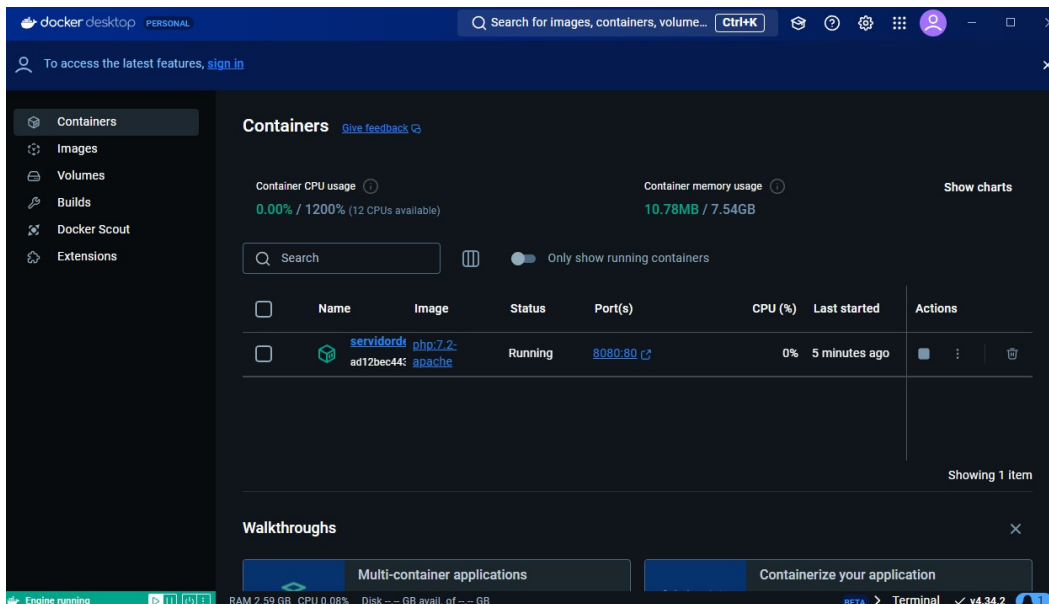
C:\Users\Maek0s>ssh marcosz@192.168.1.182
C:\\Users\\Maek0s/.ssh/config line 11: Missing argument.
C:\\Users\\Maek0s/.ssh/config: terminating, 1 bad configuration options

C:\Users\Maek0s>ssh marcosz@192.168.1.182
The authenticity of host '192.168.1.182 (192.168.1.182)' can't be established.
ED25519 key fingerprint is SHA256:qabHNVfEQ9QEEkc9K9HpYyJDruLXBEVktRLXAPJpQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.182' (ED25519) to the list of known hosts.
marcosz@192.168.1.182's password:
```

Me decia cosas de permisos con otro usuario que tengo creado, lo que hice fue ir al Archivo > Seguridad > Opciones avanzadas > Deshabilitar herencia y luego podemos cerrar eso y Editar permisos y quitar los permisos del Usuario que en mi caso me daba problemas (tengo 2 usuarios en mi ordenador) y funciona por arte de magia. No tiene sentido.

Sugerencia 4# - Docker Desktop

Al estar con Docker en Windows instale algo sinceramente sin querer que no me arrepiento y es Docker Desktop que se puede descargar aquí [Docker Desktop: The #1 Containerization Tool for Developers](#) aquí un ejemplo de como se ve:



En mi opinión, es muy bonita de ver y sencilla y aún tampoco la entiendo al 100% pero seguro que puede ser útil para el futuro y tal vez no la conocías aún y esta bien conocer cosas como estas.

Sugerencia 5# - Quitar el sudo en comandos

Debemos usar el siguiente comando y no necesitaremos el sudo en los comandos, yo hice la práctica sin esto puesto y es por eso que se ve como lo hago con sudo.

```
marcosz@marcosz:~$ sudo usermod -aG docker ${USER}
marcosz@marcosz:~$
```


Conclusión

He concluido con la actividad con dificultades, con **muchas** dificultades, por errores de conexión y por nuestra falta de experiencia en esta forma de realizar la actividad (utilizar el anfitrión para probar las cosas), al final he conseguido hacerlo lo mejor posible tras dedicar mucho tiempo e investigación con búsquedas.

Bibliografía/webgrafía

- Para solucionar el error del Docker Compose: [Docker Community Forums](#)
- Guía para realizar los cursos: [Introducción a Docker \(sergarb1.github.io\)](#)
- Docker Desktop: <https://www.docker.com/products/docker-desktop/>
- Kubernetes: <https://kubernetes.io/docs/concepts/overview>