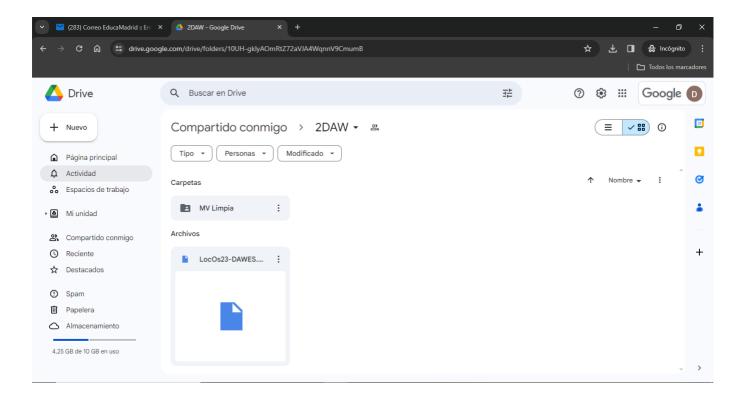
Indice de contenidos:

- 1. Máquina virtual
- 1.1 Adquirir la ova de la máquina virtual
- 1.2 Instalar la máquina virtual
- 1.3 Abrir la máquina virtual
 - 2. Docker
- 2.1 Preparar docker
- 2.2 Hacer el .env del docker
 - 3. Clonar el proyecto
- 3.1 **SSH**
- 3.2 Git clone
- 3.3 Evitar problemas de permisos
- 3.4 Generar el vendor
- 3.5 Crear el .env del proyecto
- 3.6 Compose up
- 3.7 Instalar npm
- 3.8 Instalar Vite
- 3.9 Instalar git flow
- 3.10 Como podrás observar la cosa sigue sin funcionar
- 3.11 Problemas con git

1. Máquina virtual

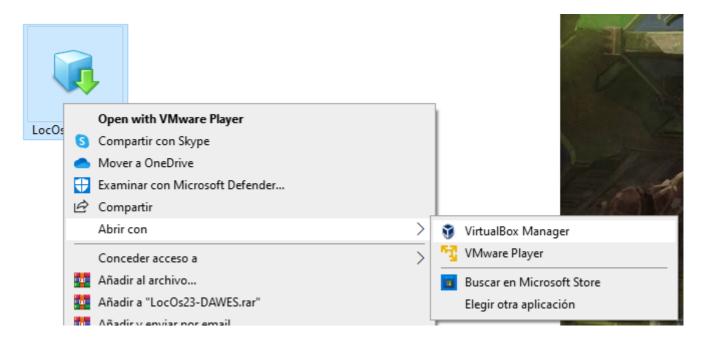
1.1 Adquirir la ova de la máquina virtual

Para conseguir la OVA nos vamos al repositorio que corresponda, como ahora no estoy en el examen toca meterse en Google Drive con la cuenta de educa.madrid.org, una vez en drive seleccionamos a la izquierda "Compartido conmigo" -> "2DAW" y luego descargamos la LocOs23 DAWES OVA, la de 4,2 GB, viene con docker y php instalados, el usuario se llama "usuario" y la contraseña de administrador es "usuario"

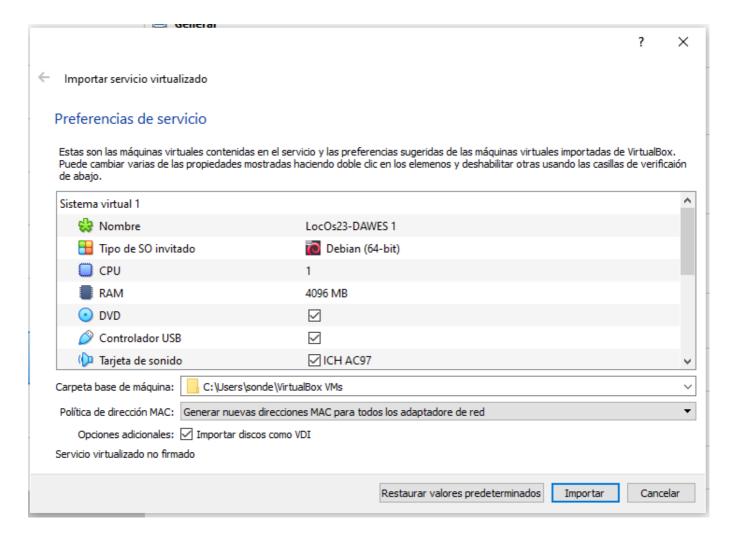


1.2 Instalar la máquina virtual

Cuando tengais la OVA teneis que hacer click derecho -> Abrir con -> virtualbox

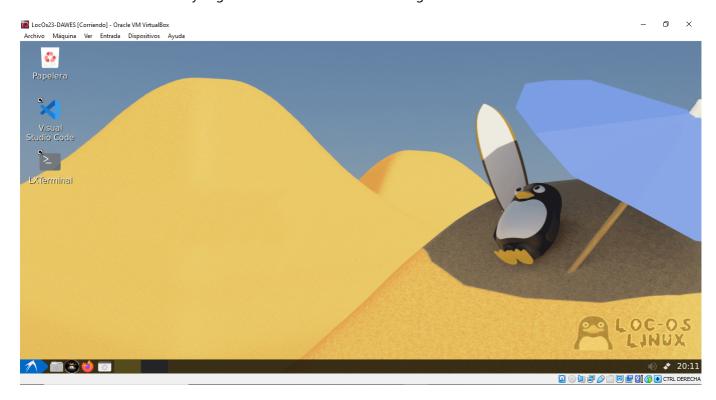


Os aparecera el menú de VirtualBox, dejais todo como está excepto que en "Politica de direcciones MAC" abrís el desplegable y elegís generar nuevas MAC para todos los adaptadores. Le dais a importar.



1.3 Abrir la máquina virtual

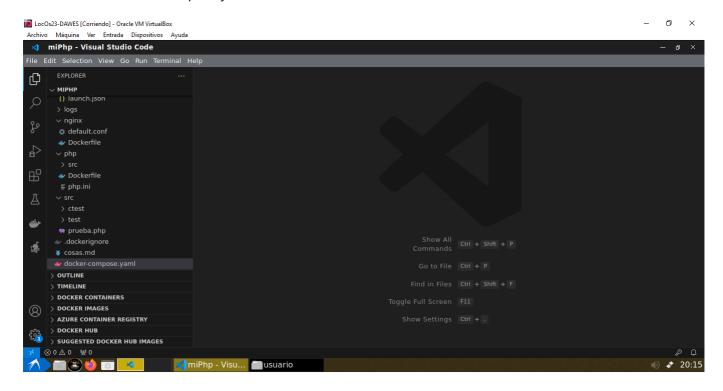
La abrimos con normalidad y según nos metemos ha de verse algo así:

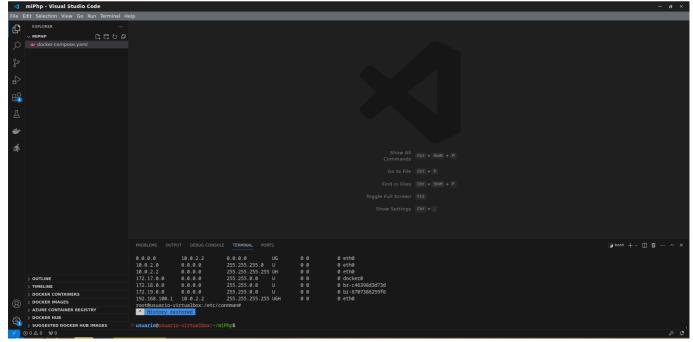


2.Docker

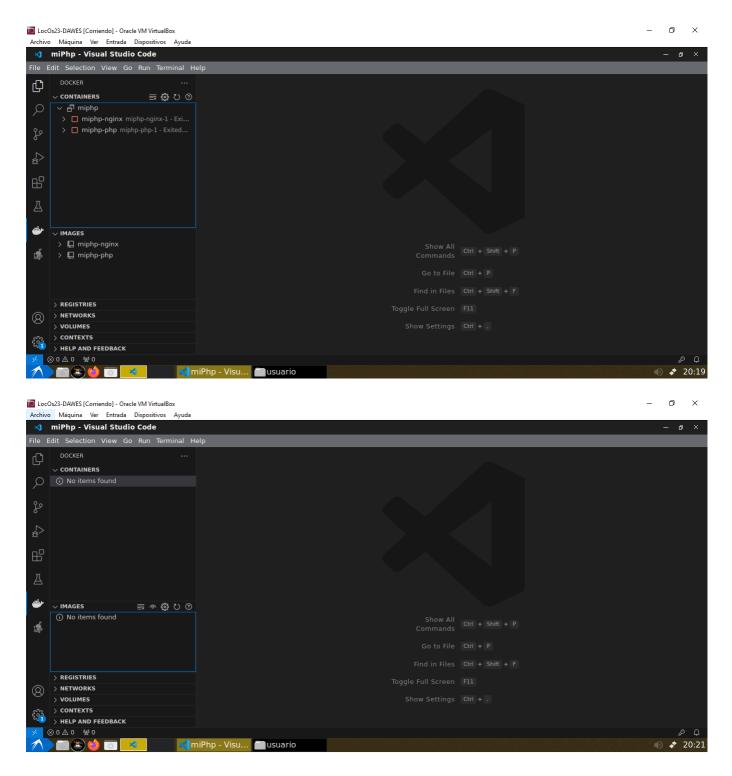
2.1 Preparar docker

Abrimos el visual y nos deberiamos encontrar en una carpeta llamada "miPHP", borramos todo lo que halla en ella menos el "docker-compose.yaml"





Luego entramos en el apartado de Docker y borramos todo, los contenedores y las imágenes:



Ahora volvemos y en la carpeta miPHP el "docker compose.yaml" que teniamos vamos a sustituir su contenido por el siguiente:

```
version: '2'

services:
    mariadb:
    image: docker.io/bitnami/mariadb:11.2
    ports:
        - '3306:3306'
    expose:
        - '3306'
    env_file:
```

```
- ".env"
  environment:
    # ALLOW_EMPTY_PASSWORD is recommended only for development.
    - ALLOW_EMPTY_PASSWORD=yes
    - MARIADB USER=${MARIADB USER}
    - MARIADB_DATABASE=${MARIADB_DATABASE}
    - MARIADB_PASSWORD=${MARIADB_PASSWORD}
myapp:
  tty: true
  image: docker.io/bitnami/laravel:10
  labels:
   kompose.service.type: nodeport
  ports:
    - '8000:8000'
  env_file:
    - ".env"
  environment:
    - DB HOST=mariadb
    - DB PORT=3306
    - DB_USERNAME=${MARIADB_USER}
    - DB_DATABASE=${MARIADB_DATABASE}
    - DB_PASSWORD=${MARIADB_PASSWORD}
  volumes:
    - './Proyecto-Julian:/app'
  depends_on:
    - mariadb
```

2.2 Hacer el .env del docker

Crearemos en miPHP un archivo llamado ".env" con el siguiente contenido:

```
# MaribaDB - Desarrollo
MARIADB_VERSION=8.0.21
MARIADB_HOST=mariadb
MARIADB_DATABASE=bitnami_myapp
MARIADB_USER=bn_myapp
MARIADB_PASSWORD=secret
```

3. Clonar el proyecto

3.1 SSH

Para poder clonar nuestro proyecto hay que conectarse por SSH con nuestra cuenta de GitHub, esto lo lograremos con los siguientes pasos:

```
ssh-keygen -t rsa -b 4096 -C tucorreo@mail
```

Ejecutamos el comando "ssh-keygen -t rsa -b 4096 -C tucorreo@mail" para generar la clave pública y la privada. Cuando nos pida donde guardarlo ponemos /home/usuario/.ssh/id_rsa . La passphrase la dejamos vacia dandole a enter.

```
TERMINAL
                                                                                                  🍞 bash +∨ 🏻 🛍 … ∧ 🗙
• usuario@usuario-virtualbox:~/miPhp$ ssh-keygen -t rsa -b 4096 -C davidortegahoyas@gmail.com
 Generating public/private rsa key pair.
 Enter file in which to save the key (/home/usuario/.ssh/id_rsa): /home/usuario/.ssh/id_rsa
 Created directory '/home/usuario/.ssh'
 Enter passphrase (empty for no passphrase):
 Enter same passphrase again:
 Your identification has been saved in /home/usuario/.ssh/id_rsa
 Your public key has been saved in /home/usuario/.ssh/id_rsa.pub
 The key fingerprint is:
 SHA256:zZqJjKLn1s6Y3l3KqEyT/Z2g2INw/q/D6SuiSRsX5dg davidortegahoyas@gmail.com
 The key's randomart image is:
  +---[RSA 4096]----
     o E S o
    .0.0 . +
    *==0 + =
  =B0*%** o
   ---[SHA256]----+
 usuario@usuario-virtualbox:~/miPhp$
```

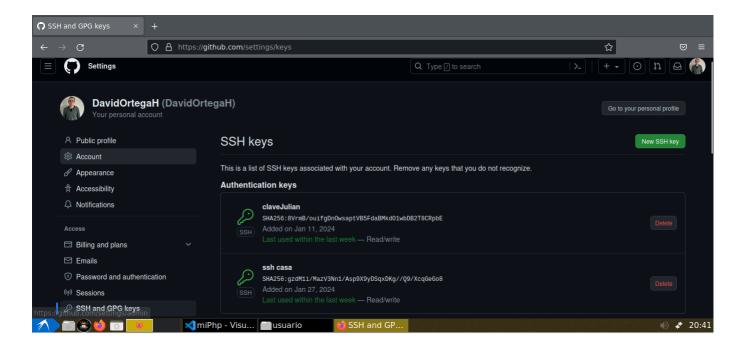
```
cd /home/usuario/.ssh/
ssh-add id_rsa
```

Ahora nos movemos al directorio en cuestión con el comando "cd /home/usuario/.ssh/", despues ejecutamos "ssh-add id_rsa" para añadir la clave privada.

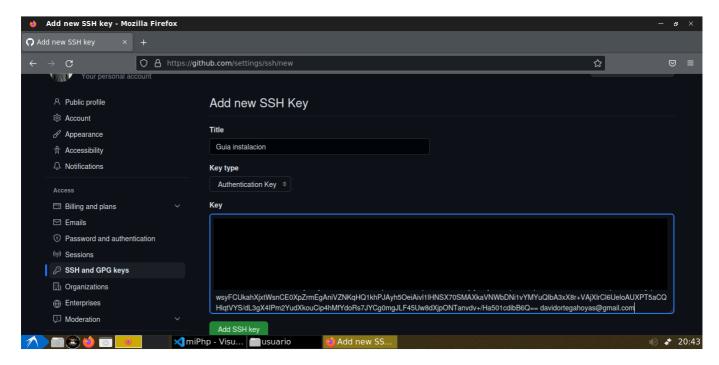
```
usuario@usuario-virtualbox:~/miPhp$ cd /home/usuario/.ssh/
usuario@usuario-virtualbox:~/.ssh$ ls
id_rsa id_rsa.pub
usuario@usuario-virtualbox:~/.ssh$ ssh-add id_rsa
Identity added: id_rsa (davidortegahoyas@gmail.com)
usuario@usuario-virtualbox:~/.ssh$
```

Luego hacemos "cat id_rsa.pub" y copiamos todo lo que nos devuelve.

Ahora nos logeamos en Github y vamos a https://github.com/settings/keys . Entonces pulsamos en New SSH Key.

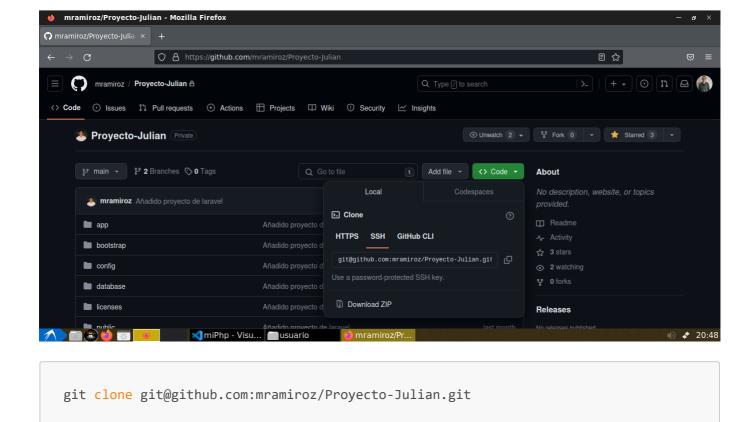


Una vez dentro le ponemos un título y en Key pegamos el tocho de la clave que copiamos antes. Entonces pulsamos en Add Key



3.2 Git clone

Nos vamos al repositorio de github del proyecto https://github.com/mramiroz/Proyecto-Julian y copiamos la opción SSH para clonarlo.



En la carpeta miPHP hacemos un "git clone git@github.com:mramiroz/Proyecto-Julian.git", en la opción de que si queremos seguir conectando le decimos que yes.

```
• usuario@usuario-virtualbox:~/miPhp$ ls
• usuario@usuario-virtualbox:~/miPhp$ git clone git@github.com:mramiroz/Proyecto-Julian.git
Clonando en 'Proyecto-Julian'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 1340, done.
remote: Counting objects: 100% (74/74), done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 1340 (delta 28), reused 52 (delta 21), pack-reused 1266
Recibiendo objetos: 100% (1340/1340), 5.61 MiB | 6.46 MiB/s, listo.
Resolviendo deltas: 100% (733/733), listo.

usuario@usuario-virtualbox:~/miPhp$ □
```

3.3 Evitar problemas de permisos

```
sudo chown -R usuario Proyecto-Julian/
```

Para hacerlo ejecutamos el comando "sudo chown -R usuario Proyecto-Julian/"

```
• usuario@usuario-virtualbox:~/miPhp$ sudo chown -R usuario Proyecto-Julian/
```

3.4 Generar el vendor

```
docker run --rm -it --volume $(pwd):/app prooph/composer:8.2 install --ignore-
platform-reqs
```

Para conseguir esta hazaña solo hay usar el comando "docker run --rm -it --volume \$(pwd):/app prooph/composer:8.2 install --ignore-platform-reqs" en la carpeta del proyecto.

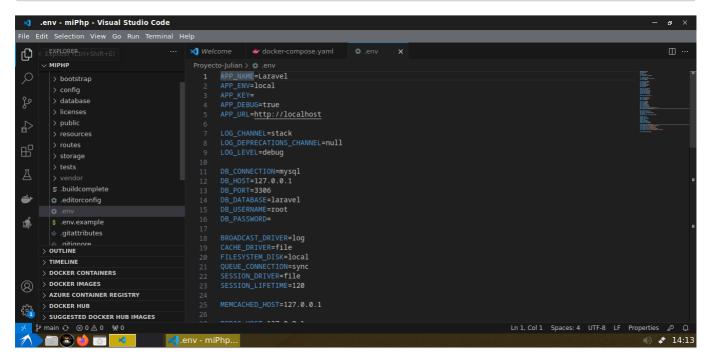
o usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian\$ docker run --rm -it --volume \$(pwd):/app prooph/composer:8.2 install --ignore-platform-reqs

3.5 Crear el .env del proyecto

Este archivo es importante tenerlo en la carpeta del proyecto, debemos de crearlo, lo llamamos .env y dentro ponemos lo siguiente (si no lo copiamos del .env.example):

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP DEBUG=true
APP_URL=http://localhost
LOG CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
MEMCACHED_HOST=127.0.0.1
REDIS HOST=127.0.0.1
REDIS PASSWORD=null
REDIS_PORT=6379
MAIL MAILER=smtp
MAIL_HOST=mailpit
MAIL PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
```

```
MAIL_FROM_NAME="${APP_NAME}"
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false
PUSHER APP ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER=mt1
VITE_APP_NAME="${APP_NAME}"
VITE PUSHER APP KEY="${PUSHER APP KEY}"
VITE_PUSHER_HOST="${PUSHER_HOST}"
VITE_PUSHER_PORT="${PUSHER_PORT}"
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
SCOUT_DRIVER=collection
```



3.6 Compose up

```
docker compose up
```

Una vez hecho esto tiramos el comando "docker compose up" desde la carpeta miphp.

```
.env
                                                                                                                                                                                                                      1 APP_NAME=Laravel
                           > Proyecto-Julian
                               docker-compose.yaml
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      🍞 docker 🕂 ∨ 🔲 🛍 🙃
                                                                                                                                                                                                           Failed to load /home/usuario/miPhp/.env: open /home/usuario/miPhp/.env: no such file or directory
usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian$ cd ..
                                                                                                                                                                                                          usuario@usuario-virtualbox:-/miPhp/Proyecto-Julian$ cd ..

usuario@usuario-virtualbox:-/miPhp$ docker compose up

WARN[0000] The "MARIADB_USER" variable is not set. Defaulting to a blank string.

WARN[0000] The "MARIADB_DATABASE" variable is not set. Defaulting to a blank string.

WARN[0000] The "MARIADB_DATABASE" variable is not set. Defaulting to a blank string.

WARN[0000] The "MARIADB_DATABASE" variable is not set. Defaulting to a blank string.

WARN[0000] The "MARIADB_DATABASE" variable is not set. Defaulting to a blank string.

WARN[0000] The "MARIADB_PASSWORD" variable is not set. Defaulting to a blank string.

WARN[0000] The "MARIADB_PASSWORD" variable is not set. Defaulting to a blank string.

                                                                                                                                                                                                                    1
                                                                                                                                                                                                                    ✓ mariadb 1 layers [∷] 0B.
✓ aa14c7233fe9 Pull complete
                                                                                                                                                                                                            | Additional 2/1 | Created | Created | Created | Container miphp-mariadb-1 | Created | Container miphp-myapp-1 | Created | Cre
                          > TIMELINE
                           > DOCKER CONTAINERS
                        > DOCKER IMAGES
                           > AZURE CONTAINER REGISTRY
                                                                                                                                                                                                            miphp-mariadb-1 | mariadb 13:17:07.71 INFO ==> welcome to the Bitnami mariadb container
miphp-mariadb-1 | mariadb 13:17:07.71 INFO ==> belcome to the Bitnami mariadb container
miphp-mariadb-1 | mariadb 13:17:07.71 INFO ==> Subscribe to project updates by watching https://github.com/bitnami/contain
                           > DOCKER HUB
                          > SUGGESTED DOCKER HUB IMAGES
                % main → ⊗ 0 <u>M</u> 0
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ln 1, Col 1 Spaces: 4 UTF-8 LF Properties 🔑 🚨
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        14:18
```

3.7 Instalar npm

```
sudo apt-get install npm
```

Como nuestro proyecto utiliza Vite, hemos de instalar npm, hay que ejecutar "sudo apt-get install npm"

```
■ usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian$ sudo apt-get install npm
```

```
sudo apt-get update
```

Además de "sudo apt-get update"

ousuario@usuario-virtualbox:~/miPhp/Proyecto-Julian\$ sudo apt-get update

3.8 Instalar Vite

```
npm install vite
```

Hacemos "npm install vite"

usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian\$ npm install vite

```
echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo
sysctl -p
```

Ahora nos daría un error por eso tenemos que ejecutar el comando "echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p"

```
• usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian$ echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p
```

```
npm run dev
```

Con el error subsanado podemos ejecutar "npm run dev", de esta manera veremos nuestra aplicación funcionando en localhots:8000

3.9 Instalar git flow

```
sudo apt-get install git-flow
```

Ya que usamos git flow toca instalarlo con "sudo apt-get install git-flow".

usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian\$ sudo apt-get install git-flow

```
git flow init
```

Luego hacemos "git flow init", respondemos como se ve en la imagen.

```
• usuario@usuario-virtualbox:~/miPhp/Proyecto-Julian$ git flow init

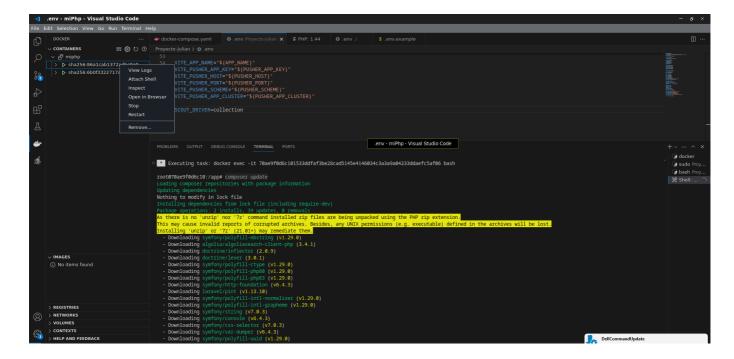
Which branch should be used for bringing forth production releases?
    - main
Branch name for production releases: [main] main
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] feature/
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [notfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/home/usuario/miPhp/Proyecto-Julian/.git/hooks]
```

3.10 Como podrás observar la cosa sigue sin funcionar

```
composer update
```

Para arreglarlo nos metemos en la terminal del contenedor de docker y ejecutamos "composer update"



3.11 Problemas con git

```
git config --global user.email "you@example.com"

git config --global user.name "Tu Nombre"
```

Tirar estos comandos si hace falta con vuestro datos.

"git config --global user.email "you@example.com"" "git config --global user.name "Tu Nombre""

FIN