# Guía de Instalación Moodle Docker

1. Instalar WSL2 y Docker Desktop.

Guía instalación WSL2: <https://learn.microsoft.com/es-es/windows/wsl/install>

Docker Desktop: <https://www.docker.com/products/docker-desktop/>

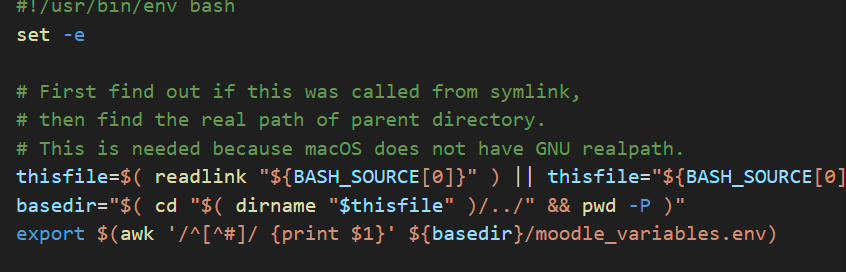
1. Clonar repositorio en Moodle Docker dentro de WSL (path recomendado /home/{$USER}/docker).

Repositorio: <https://github.com/moodlehq/moodle-docker>

1. Modificar el código de la siguiente forma:

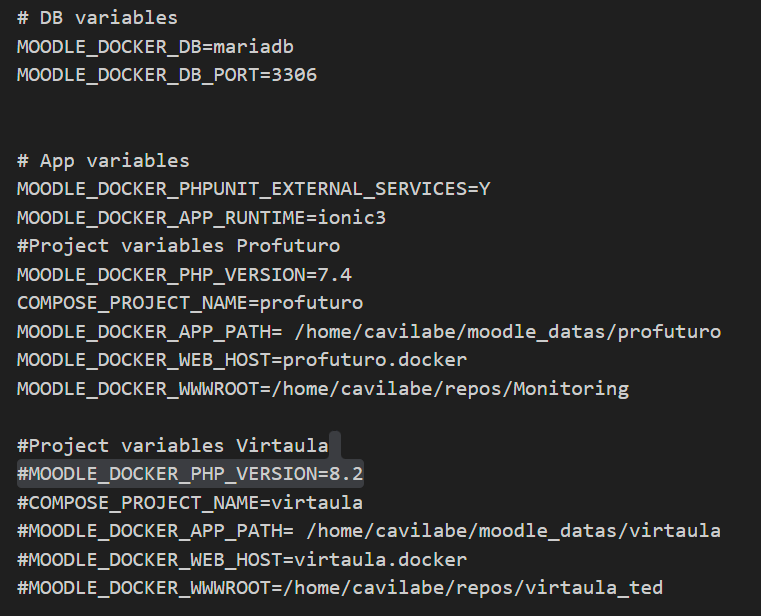
Añadir el comando:

export $(awk '/^[^#]/ {print $1}' ${basedir}/moodle\_variables.env)



En los archivos bin/moodle-docker-compose, bin/moodle-docker-wait-for-app y bin/moodle-docker-wait-for-db

Crear el archivo moodle\_variables.env



Explicación de variables

VARIABLES ESTABLES

# DB variables

MOODLE\_DOCKER\_DB=mariadb -> tipo de BBDD -> pgsql, mariadb, mysql, mssql, oracle

MOODLE\_DOCKER\_DB\_PORT=3306 -> exponer el puerto para acceder fuera Linux (HeidiSQL)

# App variables

MOODLE\_DOCKER\_PHPUNIT\_EXTERNAL\_SERVICES=Y -> Levanta contenedores con las imágenes para probar servicios externos.

MOODLE\_DOCKER\_APP\_RUNTIME=ionic5 Configure esto para indicar el tiempo de ejecución que se utiliza en la aplicación Moodle. Versiones posteriores a versiones 3.9.5, ionic5

VARIABLES POR PROYECTO

#Project variables Profuturo

MOODLE\_DOCKER\_PHP\_VERSION=7.4 -> version de php a utilizar

COMPOSE\_PROJECT\_NAME=profuturo -> Nombre del proyecto

MOODLE\_DOCKER\_APP\_PATH= /home/cavilabe/moodle\_datas/profuturo -> Ruta del moodle data

MOODLE\_DOCKER\_WEB\_HOST=profuturo.docker -> dominio del proyecto

MOODLE\_DOCKER\_WWWROOT=/home/cavilabe/repos/Monitoring -> Ruta al códio

\*Todas las rutas (moodledatas and code) deben estar en WSL2

#Project variables Virtaula -> configuraciones para otro proyecto

#MOODLE\_DOCKER\_PHP\_VERSION=8.2

#COMPOSE\_PROJECT\_NAME=virtaula

#MOODLE\_DOCKER\_APP\_PATH= /home/cavilabe/moodle\_datas/virtaula

#MOODLE\_DOCKER\_WEB\_HOST=virtaula.docker

#MOODLE\_DOCKER\_WWWROOT=/home/cavilabe/repos/virtaula\_ted

\*\*IMPORTANTE\*\*: Tener bien configurado este punto al lanzar comandos bin/moodle-docker-compose y archivo con salto de línea LF no CRLF

1. Clonar el repositorio y la carpeta moodle data en las rutas de arriba.
2. Iniciar el contenedor
   1. Copiar el archivo cofing.docker-template.php al repositorio, borrar config.php y renombrar cofing.docker-template.php -> config.php
   2. Desde ubuntu y posicionandonos en la carpeta de moodle-docker lanzamos los siguientes comandos:

bin/moodle-docker-compose up –d

bin/moodle-docker-wait-for-db

Si todo ha ido bien se prodría acceder a moodle a través de

{$MOODLE\_DOCKER\_WEB\_HOST}:8000

Y procede a proceso de instalación de moodle

## PHP UNIT

Para ejecutar PHPUNIT es necesario tener instalada la version de PHP utilizada en el proyecto y seteada como version por defecto. Posteriomente ejecutar los siguientes comandos desde moodle-docker.

Crear el entorno PHPUNIT:

bin/moodle-docker-compose exec webserver php admin/tool/phpunit/cli/init.php

Instalar decoradores:

bin/moodle-docker-compose exec webserver php admin/tool/phpunit/cli/util.php --buildcomponentconfigs

Prueba para un testsuite:

bin/moodle-docker-compose exec webserver php vendor/bin/phpunit --verbose --testdox --testsuite=core\_auth\_testsuite --colors --configuration auth

Prueba sobre un archivo en específico:

bin/moodle-docker-compose exec webserver php vendor/bin/phpunit auth/manual/tests/manual\_test.php

CAMBIAR PHP UBUNTU -> sudo update-alternatives --set php /usr/bin/php8.2

## BEHAT

Para ejecutar BEHAT es necesario tener instalada la version de PHP utilizada en el proyecto y seteada como version por defecto. Posteriomente ejecutar los siguientes comandos desde moodle-docker.

Crear el entorno BEHAT:

bin/moodle-docker-compose exec webserver php admin/tool/behat/cli/init.php

Prueba para un testsuite:

bin/moodle-docker-compose exec -u www-data webserver php admin/tool/behat/cli/run.php --tags=@auth\_manual

\*\*ERRORES POSIBLES\*\*: Problemas de permisos en los siguientes directorios:

/vendor/bin/behat -> chmod +x /vendor/bin/behat

Carpetas /tmp. Comando -> chmod 777 –R /tmp -> es dentro del contenedor

XDEBUG

Para ejecutar BEHAT es necesario tener instalada la version de PHP utilizada en el proyecto y seteada como version por defecto. Posteriomente ejecutar los siguientes comandos desde moodle-docker.

Instalar XDebug:

PHP 7.4 -> moodle-docker-compose exec webserver pecl install xdebug-3.1.6

PHP 8.2 -> moodle-docker-compose exec webserver pecl install xdebug

Add config file

read -r -d '' conf <<'EOF'

; Settings for Xdebug Docker configuration

xdebug.mode=debug

xdebug.client\_host=host.docker.internal

; Some IDEs (eg PHPSTORM, VSCODE) may require configuring an IDE key, uncomment if needed

xdebug.idekey=VSCODE

zend\_extension=xdebug

xdebug.start\_with\_request=yes

xdebug.client\_port=9003

output\_buffering=off

xdebug.log=/var/log/xdebug.log

EOF   
bin/moodle-docker-compose exec webserver bash -c "echo '$conf' >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini"

Activar XDEBUG

bin/moodle-docker-compose exec webserver docker-php-ext-enable xdebug   
bin/ moodle-docker-compose restart webserver

Abrir puerto 9003 de ubuntu

sudo ufw allow 9003

Desactivar XDEBUG

# Disable XDebug extension in Apache and restart the webserver container  
bin/ moodle-docker-compose exec webserver sed -i 's/^zend\_extension=/; zend\_extension=/' /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini  
bin/ moodle-docker-compose restart webserver

Activar XDEBUG  
# Enable XDebug extension in Apache and restart the webserver container  
bin/ moodle-docker-compose exec webserver sed -i 's/^; zend\_extension=/zend\_extension=/' /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini  
bin/ moodle-docker-compose restart webserver

NODE – GRUNT

Compilar JS