

# T3. Despregamento dun Contorno LAMP con Docker Compose

Carlos Martínez López

Deberás entregar un ficheiro pdf que conteña:

O teu ficheiro docker-compose.yml e o Dockerfile.

```
administrador@daw:~/docker_lamps$ cat docker-compose.yml
services:
  db:
    image: mariadb:10.11
    container_name: db_equipo05
    restart: unless-stopped
    environment:
      MARIADB_ROOT_PASSWORD: supersecretpassword
      MARIADB_DATABASE: daw_equipo05
      MARIADB_USER: carlosml
      MARIADB_PASSWORD: password05
    volumes:
      - db_data:/var/lib/mysql
  web:
    build: ./apache-php
    container_name: web_equipo05
    restart: unless-stopped
    volumes:
      - ./src:/var/www/html
    ports:
      - "8005:80"
    depends_on:
      - db
  phpmyadmin:
    image: phpmyadmin:latest
    container_name: phpmyadmin_equipo05
    restart: unless-stopped
    ports:
      - "8105:80"
    environment:
      PMA_HOST: db
      PMA_PORT: 3306
    depends_on:
      - db
  volumes:
    db_data:
administrador@daw:~/docker_lamps$
```

Dockerfile:

```
administrador@daw:~/docker_lamp$ cat apache-php/Dockerfile
# Partimos da imaxe oficial de PHP 8.3 con Apache
FROM php:8.3-apache
# 1. Actualizamos a lista de paquetes e instalamos as dependencias do sistema.
#   - libonig-dev é necesaria para a extensión mbstring.
#   - libxml2-dev para a extensión xml.
#   - libjpeg-dev, libpng-dev, libfreetype-dev para a extensión gd.
#   - libcurl4-openssl-dev para a extensión curl.
#   - --no-install-recommends reduce o tamaño final da imaxe.
RUN apt-get update && apt-get install -y \
    libonig-dev \
    libxml2-dev \
    libjpeg-dev \
    libpng-dev \
    libfreetype-dev \
    libcurl4-openssl-dev \
    && rm -rf /var/lib/apt/lists/*
# 2. Agora instalamos as extensóns de PHP necesarias.
# mysqli é para a conexión coa base de datos.
# mbstring, xml, gd, e curl son comúns en CMS e frameworks modernos.
RUN docker-php-ext-install mysqli mbstring xml gd curl
administrador@daw:~/docker_lamp$
```

---

1. Un ficheiro historial.txt co resultado completo do comando history da túa terminal.

```
administrador@dw:~/docker_lamp$ cat historial.txt
 1 sudo adduser administrador vboxsf
 2 cd /media
 3 sudo mkdir Compartida
 4 sudo chown administrador:administrador Compartida/
 5 exit
 6 node -v
 7 node version
 8 ng version
 9 sudo apt update; sudo apt install nodejs -y
10 node -v
11 sudo apt install npm -y
12 npm -v
13 sudo npm cache verify
14 sudo npm install -g @angular/cli@18.2.12
15 ng version
16 clear
17 poweroff
18 cd /var/www/html
19 ls
20 cd ..
21 sudo chmod -r 777 html
22 sudo chmod 777 -r html
23 sudo chmod 777 -R html
24 sudo apt-get install mysql-server mysql-common mysql-client
25 sudo apt-get install phpmyadmin php-mbstring
26 sudo service apache2 restart
27 sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
28 cd html
29 ls
30 mkdir laravel
31 sudo apt-get update
32 ls
33 cd laravel/
34 sudo apt-get install curl composer
35 sudo apt-get install php-cli php-json php-dom xdebug
36 sudo a2enmod rewrite
37 sudo a2enmod ssl
38 sudo service apache2 restart
39 sudo apt-get install php-xdebug
40 sudo service apache2 restart
41 sudo apt-get install php-dev
42 composer global require "laravel/installer"
43 export PATH="~/.config/composer/vendor/bin:PATH"
44 source ~/.bashrc
45 nano ~/.bashrc
46 sudo nano ~/.bashrc
47 laravel --version
48 sudo apt-get update
49 sudo apt install postman
50 sudo apt install snapd
51 sudo apt autoremove
52 sudo add-apt-repository --remove ppa:linuxuprising/apps
53 sudo snap install postman
54 clear
55 sudo su
56 exit
57 sudo su
```

```
78 poweroff
79 groups
80 sudo usermod -l administrador eu
81 sudo usermod -l eu administrador
82 whoami
83 sudo apt update
84 sudo apt upgrade -y
85 sudo apt install apache2 -y
86 sudo ufw allow 'Apache Full'
87 sudo ufw allow 'Apache Full'
88 sudo ufw enable
89 sudo systemctl status apache2
90 ip
91 ip a
92 sudo apt install mariadb-server mariadb-client -y
93 sudo mysql_secure_installation
94 mysql -u root -p
95 abc123..
96 mysql -u root -p
97 sudo mysql_secure_installation
98 mysql -u root -p
99 sudo apt install php libapache2-mod-php php-mysql -y
100 sudo nano /var/www/html/info.php
101 sudo rm /var/www/html/info.php
102 sudo apt install phpmyadmin -y
103 sudo apt purge phpmyadmin
104 sudo apt install phpmyadmin -y
105 sudo nano /etc/apache2/sites-available/proxecto.daw.conf
106 sudo mkdir -p var/www/proxecto.daw
107 sudo chown -R administrador:administrador /var/www/proxecto.daw
108 sudo chown -R administrador:administrador /var/www/proxecto.daw
109 sudo chown -R administrador:administrador /var/www/proxecto.daw
110 sudo mkdir -p /var/www/proxecto.daw
111 sudo chown -R administrador:administrador /var/www/proxecto.daw
112 nano /var/www/proxecto.daw/index.html
113 sudo nano /etc/apache2/sites-available/proxecto.daw.conf
114 sudo nano a2ensite proxecto.daw.conf
115 sudo a2ensite proxecto.daw.conf
116 sudo a2ensite 000-default.conf
117 sudo a2dissite 000-default.conf
118 sudo apache2ctl configtest
119 sudo systemctl restart apache2
120 127.0.0.1 proxecto.daw
121 sudo nano etc/hosts
122 sudo nano /etc/hosts
123 sudo nano /var/www/html/info.php
124 ip a
125 sudo a2dissite proxecto.daw.conf
126 sudo a2ensite 000-default.conf
127 sudo apache2ctl configtest
128 sudo systemctl restart apache2
129 cd docker_lamp/
130 apache compose ps
131 docker compose ps
132 cat docker-compose.yml
133 cat apache-php/Dockerfile
134 history > historial.txt
administrador@daw:~/docker_lamp$ █
```

2. Un documento PDF que inclúa:

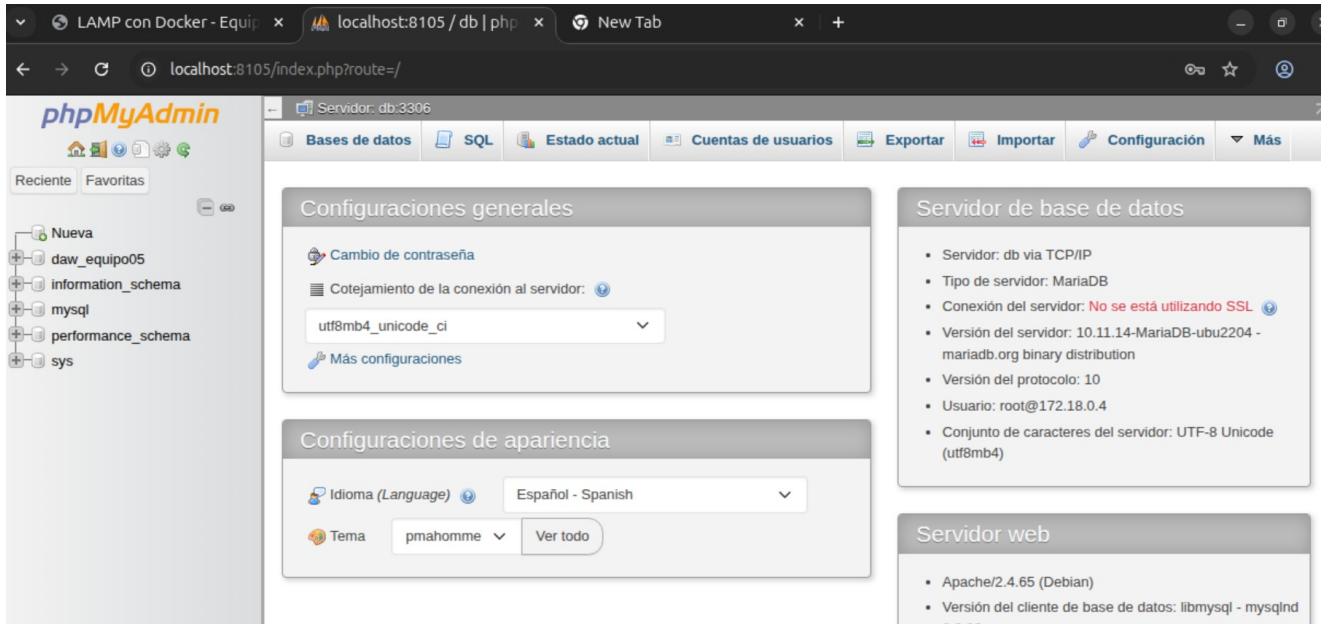
- **Capturas de pantalla** que demostren o correcto funcionamento:
  - A saída do comando docker compose ps amosando os teus tres contedores.

```
administrador@daw:~/docker_lamp$ docker compose ps
NAME           IMAGE          COMMAND                  SERVICE    CREATED        STATUS        PORTS
db_equipo05    mariadb:10.11  "docker-entrypoint.s..." db         10 minutes ago Up 8 minutes  3306/tcp
phpmyadmin_equipo05  phpmyadmin:latest  "/docker-entrypoint..." phpmyadmin  10 minutes ago Up 8 minutes  0.0.0.0:81
05->80/tcp, [::]:8105->80/tcp
web_equipo05   docker_lamp-web  "docker-php-entrypoi..." web        10 minutes ago Up 8 minutes  0.0.0.0:80
05->80/tcp, [::]:8005->80/tcp
administrador@daw:~/docker_lamp$
```

- O navegador amosando a páxina index.php no porto 80XX.



- O navegador amosando a páxina de phpMyAdmin no porto 81XX.



- **Análise Comparativa:** Unha sección escrita onde compares as dúas metodoloxías (instalación nativa vs. Docker Compose). Debes explicar coas túas propias palabras:

- Cal foi a principal diferenza na preparación do sistema anfitrión (a túa VM de Ubuntu)?

Na preparación do docker para a contedorización e bastante distinta e ademais non fixo falta preparar todas as instalacións do LAMP solo o docker.

- **Vantaxes de usar Docker Compose:** Menciona e explica polo menos tres vantaxes (p. ex., portabilidade, illamento, consistencia, facilidade de despregamento...).

Actualiza so os novos paquetes que manden, e mais portable porque ao usar contenedores permite una instalación igual e non problemas de “na miña maquina funciona”, e mais facil de despregar pois faise so con un comando.

- **Desvantaxes de usar Docker Compose:** Menciona e explica polo menos dúas desvantaxes (p. ex., curva de aprendizaxe, consumo de recursos, complexidade inicial...).

Pareceume mais dificil de aprender que una instalacion clasica (curva de aprendizaxe), e a primeira instalación e bastante complexa.

- En que tipo de proxecto ou situación elixirías cada un dos enfoques? Xustifica a túa resposta.

Para una instalación que so necesite usalo no meu equipo, usaria a instalación clasica por ser mas simple pero nunha empresa o similar, utilizaria Docker por a sua portabilidade e por tanto funcionamento correcto compatible cos meus compañeiros.