



IES Enric Valor  
C/ Duanes, 17 - Pego  
96 640 99 60  
03007613@edu.gva.es



## **ACTIVDAD 0301:**

# **Despliegue de una aplicación web en un entorno LAMP dockerizado**

Andrea Berenice Mendez  
Despliegue de Aplicaciones Web  
2 DAW

## Sumario

I. Objectivos.....	3
II. Enunciado.....	3
III. Producto final.....	3
IV. Desarrollo.....	4
1. Análisis de requisitos.....	4
2. Creación del fichero Dockerfile.....	5
3. Creación y ejecución del fichero docker-compose.yml.....	5
4. Instalación y configuración de Joomla.....	8
5. Despliegue de la página web.....	10
6. Creación de usuarios y definición de roles.....	12
7. Preparación de la aplicación web.....	13



## I. Objectivos

- Desplegar una aplicació web totalment funcional en un entorn *dockeritzado*.
- Ajustar la configuració del servidor web per a fer funcionar la aplicació.
- Depurar els errors i avisos.
- Documentar el procés de desplegament d'una aplicació web.
- Personalitzar la aplicació.

## II. Enunciado

- Haurà de desplegar un CMS, sistema gestor de continguts en un entorn LAMP basant-se en contenidors. Si és possible es pot aprofitar l'entorn desenvolupat en sessions anteriors.
- Haurà de analitzar correctament els requisits per poder adaptar l'entorn
- El sistema gestor de continguts a desplegar serà indicat pel professor de entre els següents: Drupal, Wordpress, Joomla o Typo3.
- Se tindrà de documentar tot el procés.
- Una vegada creat haurà de personalitzar-lo de forma que:
  - La pàgina principal s'asemeje a la del projecte.
  - Aparezca el logotip.
  - Haja 3 usuaris definits amb 3 rols diferents.

## III. Producto final

- Fitxer zip amb la aplicació web preparada per a ser desplegada en el seu contenidor. És a dir, una vegada desplegat el contenidor es podrà accedir a la aplicació web amb els canvis realitzats.
- Document tècnic on es documenta tot el procés, seguint les indicacions de la *Guía adaptada de presentació de treballs amb ordinador*.

## IV. Desarrollo

### 1. Análisis de requisitos

Uno de los pasos importantes previos a desarrollar el archivo *docker-compose* para crear el entorno LAMPP que permitiera ejecutar Joomla es identificar los requisitos técnicos necesarios para garantizar un despliegue exitoso. Esta información se puede consultar en manual de la página oficial de Joomla, donde se especifican las versiones de software requeridas.

Requirements for Joomla! 5.x			
Software	Recommended <sup>1</sup>	Minimum <sup>2</sup>	More Information
PHP	8.3	8.1.0	Modules: json, simplexml, dom, zlib, gd, mysqlnd or pdo_mysql or pdo_pgsql
<b>Databases</b>			
MySQL	8.1	8.0.13	
MariaDB	11.1.0	10.4.0	
PostgreSQL	16.0	12.0	
<b>Web Servers</b>			
Apache <sup>3</sup>	2.4	2.4	
Nginx	1.25	1.21	
Microsoft IIS <sup>4</sup>	10	10	

*Figura 1: Requisitos técnicos de Joomla*

En este caso se utilizará PHP 8.3 y MySQL 8.1 como se recomienda.



## 2. Creación del fichero Dockerfile

Otro de los pasos antes de comenzar a desarrollar el fichero *docker-compose* es crear un archivo *Dockerfile*, en el mismo directorio que el *docker-compose*, donde se incluyen las extensiones necesarias para personalizar los contenedores. En este caso, es imprescindible añadir la extensión PDO que permitirá a las aplicaciones PHP conectarse a la base de datos.

### Dockerfile

```
#Usa la imagen base de PHP con Apache
FROM php:8.3-apache
#Instala la extensión PDO en PHP
RUN docker-php-ext-install mysqli pdo_mysql
#Configura Apache
RUN a2enmod rewrite
```

## 3. Creación y ejecución del fichero docker-compose.yml

- En el directorio donde se encontrará el docker-compose, se creará un nuevo directorio denominado **/www** donde se almacenarán los archivos de Joomla, previamente descargados de la página oficial <https://downloads.joomla.org/>
- Una vez reconocidos los requisitos técnicos y creado el Dockerfile se procede a desarrollar el fichero docker-compose que creará el entorno LAMPP donde se desplegará el CMS de Joomla.



## docker-compose.yml

```
version: '3.2'

services:
  web:
    image: php:8.3-apache
    build:
      context: .
      dockerfile: Dockerfile
    container_name: php-apache_joomla
    ports:
      - "8080:80"
    volumes:
      - ./www:/var/www/html
    depends_on:
      - db
    environment:
      - ./www:/var/www/html

  db:
    image: mysql:8.1
    container_name: mysql_joomla
    environment:
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_DATABASE=joomla_db
      - MYSQL_USER=andrea
      - MYSQL_PASSWORD=andrea
    volumes:
      - ./mysql:/docker-entrypoint-initdb.d/
      - db_data:/var/lib/mysql

  phpmyadmin:
    image: phpmyadmin:5.2.1
    container_name: phpmyadmin_joomla
    ports:
      - "8088:80"
    environment:
      - PMA_HOST=db
      - PMA_PORT=3306
      - PMA_USER=root
      - PMA_PASSWORD=root
    depends_on:
      - db

volumes:
  db_data:
  www:
```

#Se monta el volumen ./www, donde se almacenará y persistirán los archivos de Joomla.

#Se definen variables de entorno para la configuración de la base de datos.

\*\*\*\*\*

#Se incluye el contenedor phpMyAdmin que permitirá gestionar la base de datos

#Se define la variable PMA\_HOST para que phpMyAdmin se conecte al servicio de BBDD 'db'.

#Volumenes persistentes

**\*\*ES MUY IMPORTANTE NO DEFINIR ROOT COMO MYSQL\_USER EN EL CONTENEDOR DE MYSQL YA QUE DARÁ UN ERROR, INTERRUPIENDO LA EJECUCIÓN DEL MISMO\*\***

```
mysql_joomla | 2024-11-24 14:34:00+00:00 [ERROR] [Entrypoint]: MYSQL_USER="root", MYSQL_
USER and MYSQL_PASSWORD are for configuring a regular user and cannot be used for the root use
r
```

Figura 2: Error de ejecución estableciendo root como usuario

Esto se debe a que se estaría configurando al superusuario para tareas específicas de la aplicación, lo que rompería el principio de menor privilegio.

**Principio del menor privilegio:** otorgar a los usuarios solo los privilegios que necesitan para realizar sus tareas.

- c) Luego se ejecuta uno de los siguientes comandos para crear todos los contenedores definidos en el archivo:

**docker-compose up** (los contenedores se inician en primer plano y se pueden ver los logs desde la terminal, lo cual es útil para depurar)

**docker-compose up -d** (los contenedores se inician en segundo plano, liberando la terminal inmediatamente)

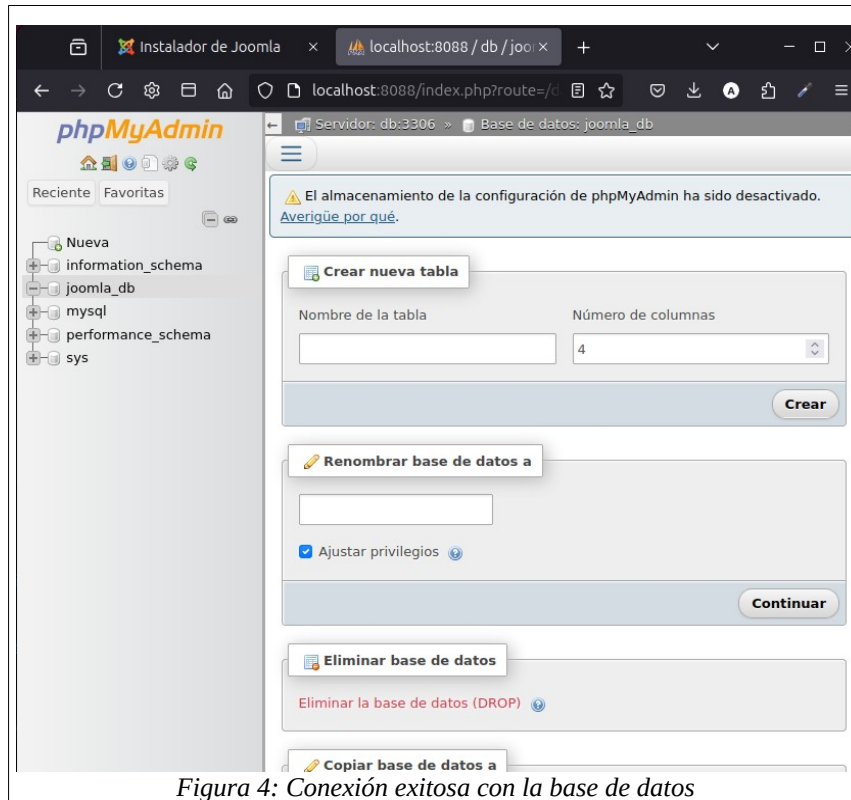
- d) Una vez creados los contenedores, se puede verificar su correcto funcionamiento accediendo a:

- <http://localhost:8080> Donde se podrá comenzar a instalar Joomla



Figura 3: Instalación exitosa de joomla

- <http://localhost:8088> Donde se podrá acceder a la base de datos



*Figura 4: Conexión exitosa con la base de datos*

## 4. Instalación y configuración de Joomla


- a) Se puede realizar la instalación y configuración desde el terminal o desde la interfaz gráfica. En este caso se hará desde la última mencionada.
- b) Luego de seguir los pasos dándole el nombre al sitio y establecer los datos de acceso. El paso más importante es la configuración de la bases de datos, en el que se debe prestar especial atención a los siguientes puntos:





### Configuración de la base de datos

Seleccione el tipo de base de datos. \*

MySQL 


Introduzca el nombre del hospedaje. Normalmente es "localhost" o el nombre proporcionado por su hospedaje. \*

db

Introduzca el nombre de usuario de la base de datos que haya elegido o el facilitado por quien le sirva el hospedaje. \*

andrea

Introduzca la contraseña de la base de datos que haya elegido o la facilitada por quien le sirva el hospedaje.

..... 

Introduzca el nombre de la base de datos. \*


joomla\_db

Introduzca un prefijo para la base de datos o use uno generado aleatoriamente. \*

ng69q\_

Si está utilizando una base de datos existente con tablas con el mismo prefijo, Joomla renombrará las tablas existentes añadiendo el prefijo "bak\_".

Cifrado de la conexión \*

Predeterminado (controlado por el servidor) 

**Instalar Joomla >**

Joomla! is Free Software released under the GNU General Public License.

Figura 5: Configuración de la base de datos

**A.** El nombre del hospedaje que se debe usar para conectarse a la base de datos de MySQL es el nombre del servicio establecido en el archivo docker-compose (**db**).  
## Esto se debe a que todos los contenedores definidos en el docker-compose están conectados a una red virtual interna y dentro de esta se pueden acceder a ellos por el nombre del servicio asigando, en lugar de una dirección de IP. De esta manera, Docker se encarga de dirigir las peticiones de red al contenedor correcto utilizando este nombre. ##



**B.** Los datos de usuario y contraseña para acceder en la bases de datos son los establecidos en el docker-compose.

- c) Luego de terminar la configuración de la base de datos, saldrá una advertencia por utilizar un host que no se encuentra en el servidor local. Por lo que habrá que realizar un paso extra de eliminar un archivo que se crea en la ruta **/www/installation/nombreDelArchivo.txt**



*Figura 6: Advertencia de hospedaje que no se encuentra local*

- d) Una vez terminada la instalación, en el directorio **/www** se crea automáticamente un archivo de **configuration.php** que contiene todas las configuraciones necesarias para que el sitio web funcione correctamente. En este se definen la conexión a la base de datos (tipo, nombre del host, usuario, contraseña, etc) , configuración del sitio web (nombre, etc), opciones de depuración y errores, entre otros datos importantes.

**\*\* Cada vez que se detengan y eliminen todos los contenedores, redes y volúmenes asociados y se quieran volver a crear estos contenedores, es necesario eliminar el archivo **configuration.php**, ya que contiene configuraciones específicas generadas para cada instalación. Tener este archivo significaría un conflicto en la nueva instalación ya que los nuevos parámetros no coincidirán con este. Además una vez configurado se eliminará el directorio **/installation** automáticamente, por lo que habrá que volver a cargarlo cuando se quiera volver a instalar.\*\***

## 5. Despliegue de la página web

Para el desarrollo y despliegue de la página de Stockflow en Joomla se utilizaron dos herramientas que facilitaron la edición del sitio, estas son el template Helix Ultimate y el plugin SP Page Builder.



Helix Ultimate es un template para Joomla que ofrece una estructura flexible y personalizable. A través de este se configuraron aspectos como el menú, el footer y los colores principales de la página web.

Para facilitar la creación y edición del contenido visual, se utilizó el plugin SP Page Builder. Este plugin de Joomla permite crear páginas web mediante un sistema drag and drop, lo que hace que la construcción de páginas web sea mucho más intuitiva y accesible.

SP Page Builder permitió estructurar todo el contenido del principal de la página Home, basándose en un grid de filas y columnas.

Para hacer uso de ambos fue necesario modificar el archivo php.ini dentro del contenedor para que ampliar el tamaño máximo de subida. Esto se logró ejecutando los siguientes comandos en el terminal:

- a) Se accede al contenedor php-apache\_joomla

***docker exec -it php-apache\_joomla bash***

- b) Se busca y edita el archivo php.ini

***cd /usr/local/etc/php/php.ini***

- c) Luego se accede al archivo

***nano php.ini***

**\*\* Si nano no está instalado, habrá que instalarlo \*\***

***apt-get update***

***→ apt-get install nano***

- d) Una vez dentro se buscan y modifican los siguientes datos

***upload\_max\_filesize = 100M***

***post\_max\_size = 100M***

- e) Una forma de persistir esta información, a la hora de ejecutar el docker-compose en otra máquina, es copiando este archivo fuera del contenedor y ejecutarlo en el Dockerfile:

***mkdir php.ini*** (se crea un directorio para guardar el archivo)

***nombre del contenedor ruta dentro del contenedor***  
***docker cp php-apache\_joomla:/usr/local/etc/php.ini php.ini/***  
***ruta fuera del contenedor***

Luego se modifica el Dockerfile añadiendo la siguiente línea:

### Dockerfile

```
#Usa la imagen base de PHP con Apache
FROM php:8.3-apache
#Instala la extensión PDO en PHP
RUN docker-php-ext-install mysqli pdo_mysql
#Copia el archivo php.in dentro del contenedor
COPY ./php.ini /usr/local/etc/php/php.ini
#Configura Apache
RUN a2enmod rewrite
```

## 6. Creación de usuarios y definición de roles


Desde Joomla se accede desde el panel derecho en la opción **Usuarios > Gestionar > + Nuevo** y se crean 3 usuarios definiendo sus roles desde la pestaña de "**Grupos de usuarios asignados**"



The screenshot shows the Joomla user creation interface. It includes fields for Name, Username, Password, Confirm Password, and Email. The password field has a strength indicator showing 'Contraseña aceptada'.

Detalles de la cuenta	
Nombre *	Vicent Jorda
Nombre de usuario (Username) *	vicent_jorda
Contraseña	VicentJorda1
Confirmar contraseña	VicentJorda1
Correo electrónico *	v.jordaescriva@edu.gva.es

Figura 7: Creación de usuario



The screenshot shows the Joomla user role assignment interface. It includes tabs for 'Detalles de la cuenta', 'Grupos de usuario asignados', 'Configuración básica', and 'Configuración de accesibilidad'. The 'Grupos de usuario asignados' tab is active, showing a list of user groups with checkboxes for selection.

Detalles del nuevo usuario	
API Token de Joomla	
Grupos de usuario asignados	
<input type="checkbox"/>	Public
<input type="checkbox"/>	- Guest
<input type="checkbox"/>	- Manager
<input type="checkbox"/>	- Administrator
<input checked="" type="checkbox"/>	- Registered
<input checked="" type="checkbox"/>	- Author
<input type="checkbox"/>	- Editor
<input type="checkbox"/>	- Publisher
<input type="checkbox"/>	- Super Users

Figura 8: Asignación de rol

Los usuarios creados son los siguientes:

Nombre de usuario	Contraseña	Rol
vicent_jorda	VicentJorda1	Author
j.signes	JavierSignes1!	Editor
d.crespo	DavidCrespo1!	Publisher
Andrea	Andreagulener	Administrador

## 7. Preparación de la aplicación web

A la hora de compartir los contenedores con la página web desplegada es necesario tener en consideración lo siguiente:

- Se debe exportar la base de datos des de phpmyadmin y cargarla en el volumen /mysql para poder acceder a los archivos de la página desplegada una vez se ejecute el docker-compose.
- Configurar los permisos correctos antes de ejecutar el docker-compose:

***chmod -R 777 /www***

- Luego se comprimen los archivos y ya está lista para ser desplegada en cualquier ordenador con el siguiente comando:

***docker-compose up -d --build***