

# Guía de Configuración y Solución de Problemas en Laravel con Sail y MySQL

## Eliminación y Creación del Contenedor MySQL

1. Para eliminar todos los contenedores existentes, se utilizó el siguiente comando:

```
docker rm -f $(docker ps -aq)
```

2. Luego, el contenedor de MySQL fue recreado usando Docker Compose con Sail, asegurando que se estuviera ejecutando la última configuración del archivo `docker-compose.yml`.

```
./vendor/bin/sail up -d
```

## Configuraciones del archivo docker-compose.yml

services:

laravel.test:

build:

context: './vendor/laravel/sail/runtimes/8.3'

dockerfile: Dockerfile

ports:

- '\${APP\_PORT:-80}:80'

environment:

WWWUSER: '\${WWWUSER}'

volumes:

- './var/www/html'

networks:

- sail

depends\_on:

- mysql

mysql:

image: 'mysql/mysql-server:8.0'

ports:

- '3306:3306'

environment:

MYSQL\_ROOT\_PASSWORD: '\${DB\_PASSWORD}'

MYSQL\_DATABASE: '\${DB\_DATABASE}'

MYSQL\_USER: '\${DB\_USERNAME}'

MYSQL\_PASSWORD: '\${DB\_PASSWORD}'

volumes:

- 'sail-mysql:/var/lib/mysql'

networks:

- sail

phpmyadmin:

image: 'phpmyadmin:latest'

ports:

- '8088:80'

networks:

sail:

driver: bridge

## Conexión a MySQL y Configuración de Permisos

3. Para acceder a MySQL usando el usuario 'ximo' con la contraseña 'admin', se ejecutó:

```
mysql -uximo -padmin
```

4. Se crearon usuarios y permisos para la base de datos de la siguiente manera:

```
CREATE USER 'ximo'@'%' IDENTIFIED BY 'admin';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'ximo'@'%' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

5. Para verificar la existencia de datos, se usó Tinker en Laravel con los siguientes comandos:

```
./vendor/bin/sail php artisan tinker
```

```
App\Models\Noticia::count();
```

6. También se realizó la eliminación de permisos utilizando:

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'ximo'@'%';
```