

Examen Recuperación

Indice

Ejercicio 3.....	2
Ejercicio4.....	6
Ejercicio 5.....	8

Ejercicio 3

Creamos nuestra carpeta de trabajo llamado Exercici-symfony utilizando mkdir Exercici Symfony

Después crearemos nuestro docker-compose.yml

La versión tiene que ser 3.5

servicio tan solo uno llamado installer

utilizaremos de imagen composer:latest

Creamos un volumen desde nuestra carpeta actual a la carpeta app de contenedor

directorio de trabajo sera el /app

Y nuestro puerto escuchara el 8000 y el del contenedor sera 8000.

```
david@david-VirtualBox:~/Escritorio/Exercici-symfony$ cat docker-compose.yml
version: "3.5"

services:
  installer:
    image: composer:latest
    volumes:
      - ./app
    ports:
      - 8000:8000
    working_dir: /app
```

Después de crear nuestro docker-compose lanzaremos la siguiente linea de comandos.

Docker compose run --rm --service-ports installer bash

```
david@david-VirtualBox:~/Escritorio/Exercici-symfony$ docker compose run --rm --service-ports installer bash
[+] Creating 1/0
✓ Network exercici-symfony_default Created 0.1s
[+] Running 15/15
✓ installer 14 layers [#####] 0B/0B Pulled 13.2s
✓ 4abcf2066143 Pull complete 1.0s
✓ 9b9ef19a461f Pull complete 1.0s
✓ 0f2fc49c6a93 Pull complete 0.4s
✓ 47a4aa92ed63 Pull complete 4.3s
✓ 8f63d27b3e4c Pull complete 2.8s
✓ 6fa0baddbb90 Pull complete 1.5s
✓ 3f19be496d84 Pull complete 4.8s
✓ 7ffb21299d5d Pull complete 3.2s
✓ 3d5aeb239739 Pull complete 3.7s
✓ e0a2f26ff8a6 Pull complete 7.8s
✓ 921e7aa1b80d Pull complete 5.6s
✓ 82033543e129 Pull complete 5.4s
✓ 5539123efeda Pull complete 5.9s
✓ bcf48bff2772 Pull complete 6.1s
p226013f5911:/app#
```

Entraremos en un interfaz de symfony.

Instalaremos el command line interface de symfony para ello utilizaremos el siguiente comando
`curl -sS https://get.symfony.com/cli/installer | bash`

```

b226013f5911:/app# curl -s https://get.symfony.com/cli/installer | bash
Symfony CLI installer

Environment check
[*] cURL is installed
[*] Tar is installed
[*] Git is installed
[*] Your architecture (amd64) is supported

Download
Downloading https://github.com/symfony-cli/symfony-cli/releases/latest/download/symfony-cli_linux_amd64.tar.gz...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
 0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 5379k 100 5379k    0     0  4300k      0  0:00:01 0:00:01 --:--:-- 11.1M
Uncompress binary...
Installing the binary into your home directory...
The binary was saved to: /root/.symfony5/bin/symfony

The Symfony CLI was installed successfully!

Use it as a local file:
/root/.symfony5/bin/symfony

Or add the following line to your shell configuration file:
export PATH="$HOME/.symfony5/bin:$PATH"

Or install it globally on your system:
mv /root/.symfony5/bin/symfony /usr/local/bin/symfony

Then start a new shell and run 'symfony'

```

Después moveremos el binario de symfony tal como nos lo dice el instalador donde pone or install ir globally on your system:

```

b226013f5911:/app# mv /root/.symfony5/bin/symfony /usr/local/bin/symfony

```

Creamos un proyecto en symfony utilizando el siguiente comandos: symfony new --webapp symfony-dockeritzat.

```

b226013f5911:/app# symfony new --webapp symfony-dockeritzat
* Creating a new Symfony project with Composer
(running /usr/bin/composer create-project symfony/skeleton /app/symfony-dockeritzat --no-interaction)

* Setting up the project under Git version control
(running git init /app/symfony-dockeritzat)

Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

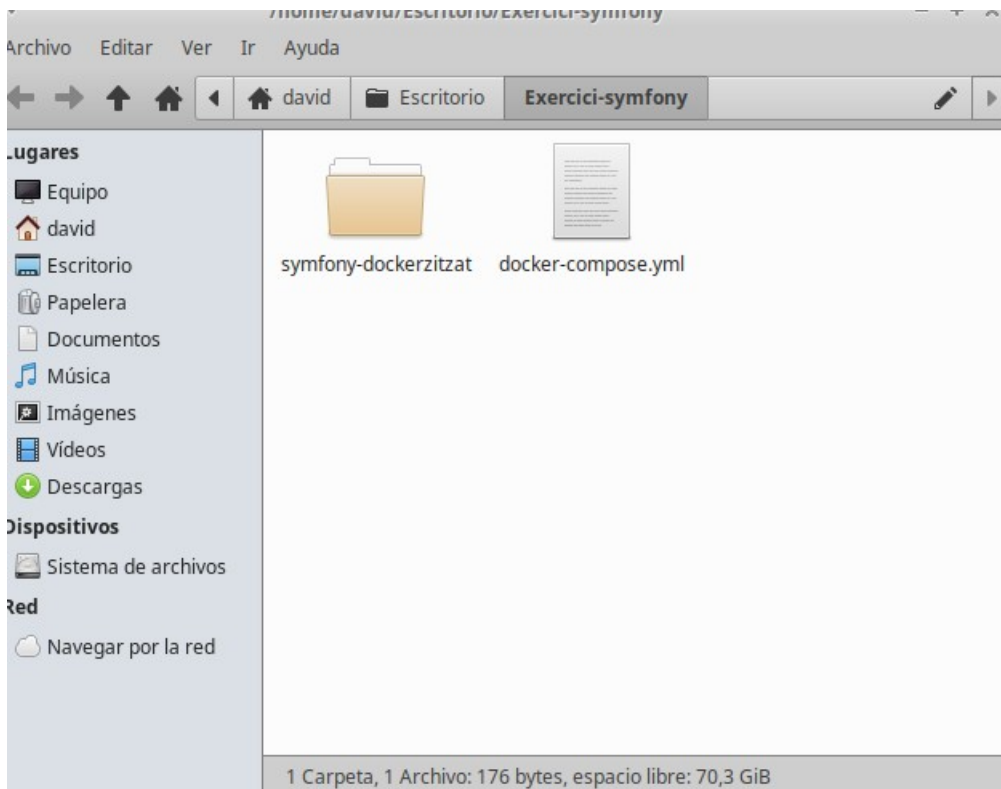
to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'root@b226013f5911.(none)')

exit status 128

```

con estos pasos se nos creara una carpeta en nuestra maquina llamada symfony-dockeritzat.



Ahora podremos probar que funciona lo que hemos realizado utilizando `symfony server:start` y se lanzara el servicio

```
fatal: unable to auto-detect email address (got 'root@b226013f5911.(none)')
exit status 128
b226013f5911:/app# symfony server:start

[WARNING] run "symfony server:ca:install" first if you want to run the web server with TLS support, or use "--p12" or
"--no-tls" to avoid this warning

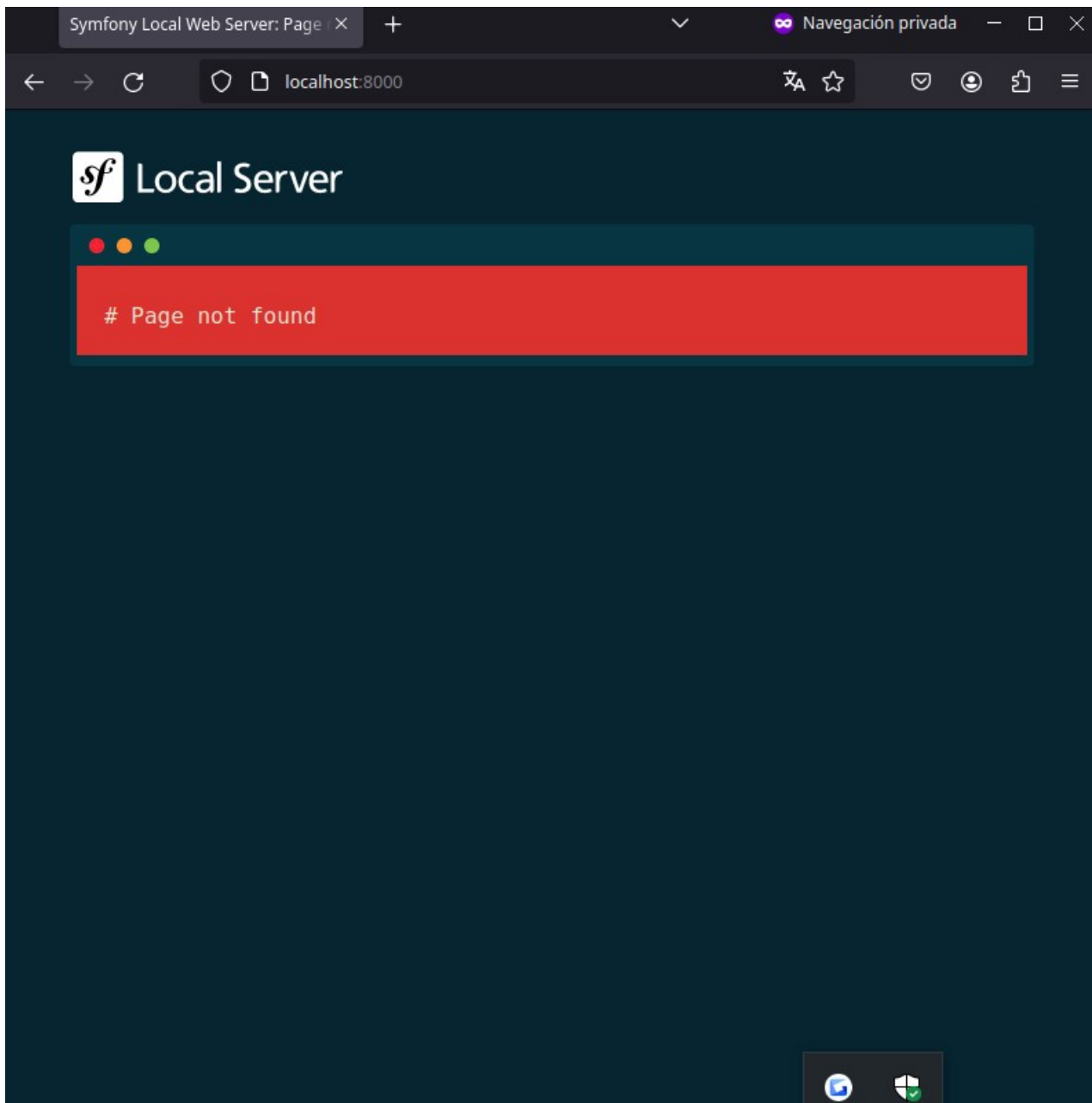
Following Web Server log file (/root/.symfony5/log/0c35eebf403cf91fe77a64921d76aa1ca6411d20.log)

[WARNING] The local web server is optimized for local development and MUST never be used in a production setup.

[OK] Web server listening
http://127.0.0.1:8000

[Web Server ] Feb 21 14:55:46 [WARN] | HTTP  unable to detect the front controller, disabling the PHP server error="Pa
ss thru script \"/index.php\" does not exist under /app/: stat /app/index.php: no such file or directory"
[Web Server ] Feb 21 14:56:06 [WARN] | SERVER GET  (404) / ip="192.168.0.1"
[Web Server ] Feb 21 14:56:06 [WARN] | SERVER GET  (404) /favicon.ico
```

Si entramos desde nuestra maquina a localhost:8000 debería de verse algo creado por symfony.



No funciona porque no encuentra el index.php según el error y además está utilizando una versión de symfony muy baja al instalarlo (5.3) y en el siguiente ejercicio sí que funciona bien con una versión más alta.

Ejercicio4

Vamos dentro del directorio que hemos creado antes al hacer un nuevo docker-compose.yml

```
version: "3.5"

services:
  app:
    image: php:8.2-fpm-alpine
    volumes:
      - ./app
    working_dir: /app

  webserver:
    image: nginx:latest
    ports:
      - 8000:80
    volumes:
      - ./nginx/conf.d:/etc/nginx/conf.d
```

Como vemos hemos creado 2 servicios nuevos App y webserver.

En app utilizamos la imagen php:8.2 fpm alpine y en webserver sera el que utilice nginx como servidor

los volúmenes serán volumes es desde app desde el mismo directorio:. Al directorio del contenedor utilizando /app

y en el webserver utilizara ./nginx/conf desde nuestro repositorio local que el contenedor utilizara etc/nginx/conf.d

Vamos a Aules y nos descargamos un archivo llamado app.conf con ello vamos a la carpeta nginx creamos el directorio conf.d y lo copiamos en ese directorio.

```
david@david-VirtualBox:~/Escritorio/Exercici-symfony/symfony-dockeritzat/nginx$ tree .
.
├── conf.d
│   └── app.conf
└── 1 directory, 1 file
```

Ahora utilizamos docker-compose up -d y si todo va correctamente debiera de estar 2 contenedores lanzados.


CONTAINER ID	IMAGE NAMES	COMMAND	CREATED	STATUS	PORTS
f04997b9b961	php:8.2-fpm-alpine	"docker-php-entrypoi..."	9 minutes ago	Up 9 minutes	9000/tcp
b63868aa545d	nginx:latest	"/docker-entrypoint..."	9 minutes ago	Up 9 minutes	0.0.0.0:8000->80/tcp, :::8000->80/tcp
symfony-dockeritzat_app_1					
symfony-dockeritzat_webserver_1					

Ahora si accedemos a localhost:8000 deberiamos ver la pantalla de symfony.


Welcome to Symfony!


Navegación privada


localhost:8000





Welcome to Symfony 7

 You are using Symfony **7.0.3** version

 Your application is ready at:
/app/


 You are seeing this page because the homepage URL is not configured and debug mode is enabled.

NEXT STEP  [Create your first page](#) to replace this placeholder page.




Learn

- [Read Symfony Docs](#)
- [Watch Symfony Screencast](#)
- [Read Symfony Book](#)



Community & Support

- [Symfony Support](#)
- [Join the Symfony Community](#)
- [Contribute to Symfony](#)



Stay Updated

- [Symfony Blog](#)
- [Follow Symfony](#)
- [Attend Symfony Events](#)

Ejercicio 5

Vamos a crear un nuevo servicio llamado db en el docker compose.

Utilizaremos una imagen mysql:latest, las variables de entorno serán.

MYSQL_ROOT_PASSWORD = admin

MYSQL_DATABASE= db

volumen definido con el nombre symfony-data contra la carpeta /var/lib/mysql

fuera de servicio creamos un apartado llamado volumes y dentro definiremos el volumen llamado symfony-data.

```
version: "3.5"

services:
  app:
    image: php:8.2-fpm-alpine
    volumes:
      - ./app
    working_dir: /app

  webserver:
    image: nginx:latest
    ports:
      - 8000:80
    volumes:
      - ./nginx/conf.d:/etc/nginx/conf.d

  db:
    image: mysql:latest
    environment:
      - MYSQL_ROOT_PASSWORD= admin
      - MYSQL_DATABASE= db
    volumes:
      - symfony-data:/var/lib/mysql

volumes:
  symfony-data:
```

Vamos a configurar nuestra base de datos y instalar una extensión en nuestra app.

Primero lo lanzamos en segundo plano utilizando docker-compose up -d

```
david@david-VirtualBox:~/Escritorio/Exercici-symfony/symfony-dockeritzat$ docker-compose up -d
Creating volume "symfony-dockeritzat_symfony-data" with default driver
Pulling db (mysql:latest)...
latest: Pulling from library/mysql
81badc5f380f: Pull complete
c490e5dd1a9d: Pull complete
87aeb61f1478: Pull complete
1cacbea6ceda: Pull complete
1e72891ace67: Pull complete
42b720363d36: Pull complete
6b3b50f9990a: Pull complete
3811d52cfa61: Pull complete
05bc7a0277d8: Pull complete
cc0abd25a274: Pull complete
Digest: sha256:ff5ab9cdce0b4c59704b4e2a09deed5ab8467be795e0ea20228b8528f53fcf82
Status: Downloaded newer image for mysql:latest
symfony-dockeritzat_webserver_1 is up-to-date
symfony-dockeritzat_app_1 is up-to-date
Creating symfony-dockeritzat_db_1 ... done
```

Vamos a entrar en una sesión interactiva en nuestro servicio db.

```
david@david-VirtualBox:~/Escritorio/Exercici-symfony/symfony-dockeritzat$ docker compose exec db mysql -uroot -padmin
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
```


A mi no me va por algun motivo de configuración interna y me sale siempre yes pero el resto de la practica me funciona y me sale el resultado final.

```

      NAMES
07d2f214f5f6  mysql:latest      "docker-entrypoint.s..." About a minute ago  Up About a minute  3306/tcp, 33060/tcp
p             symfony-dockeritzat_db_1
f04997b9b961  php:8.2-fpm-alpine "docker-php-entrypoi..." 31 minutes ago     Up 31 minutes      9000/tcp
             symfony-dockeritzat_app_1
b63868aa545d  nginx:latest      "/docker-entrypoint.s..." 31 minutes ago     Up 31 minutes      0.0.0.0:8000->80/tcp, :::8000->80/tcp
             symfony-dockeritzat_nginx_1
```

Se ven los servidores alzados.

Vamos a crear el Dockerfile ahora en la raiz de nuestro proyecto symfony-dockeritzat.

```

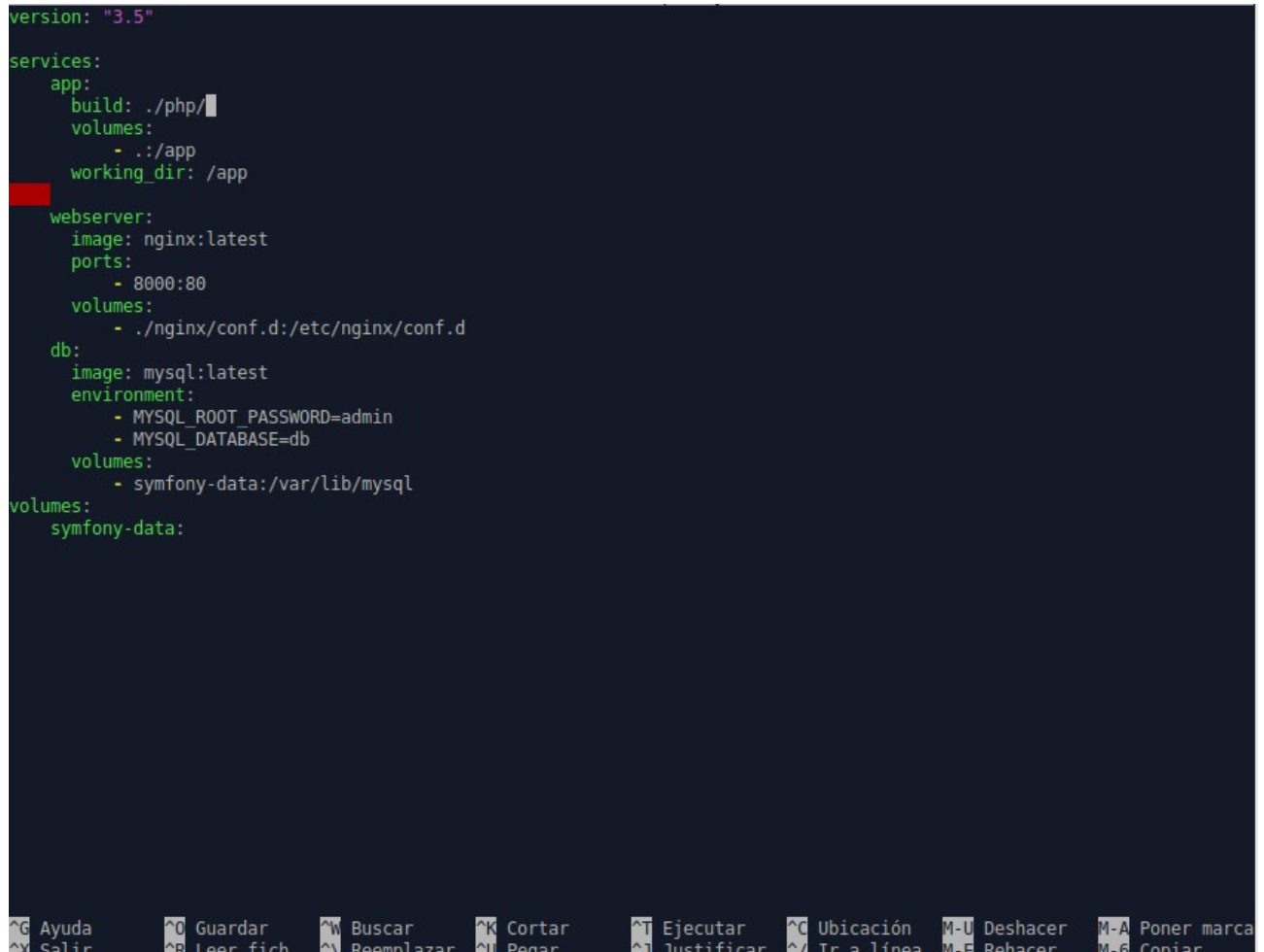
FROM php:8.2-fpm-alpine
RUN docker-php-ext-install pdo_mysql
```

[3 líneas escritas]

Ahora tenemos que sustituir la imagen que utiliza el servicio app por lo escrito en Dockerfile.

```
version: "3.5"

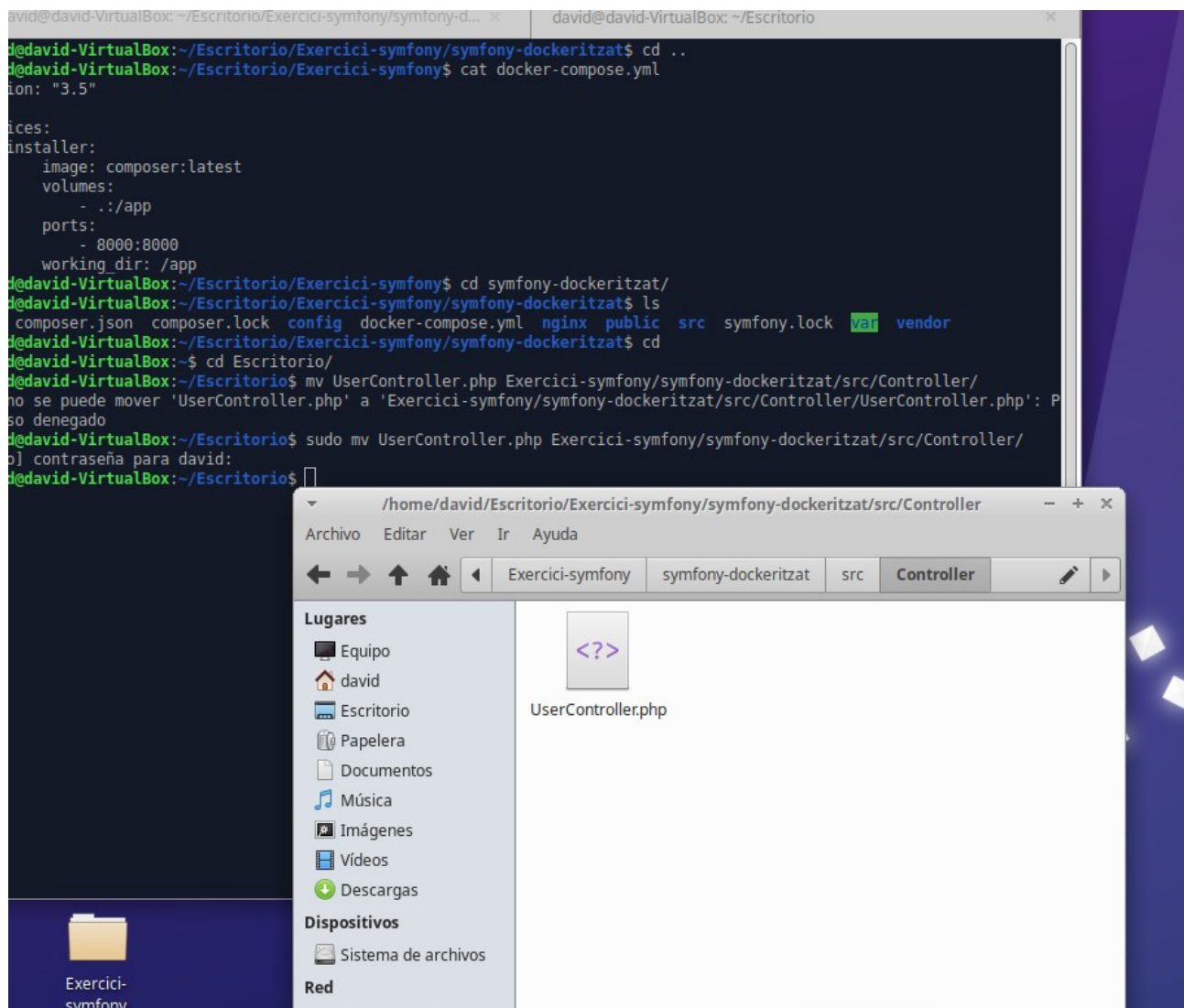
services:
  app:
    build: ./php/
    volumes:
      - ./app
    working_dir: /app
  webserver:
    image: nginx:latest
    ports:
      - 8000:80
    volumes:
      - ./nginx/conf.d:/etc/nginx/conf.d
  db:
    image: mysql:latest
    environment:
      - MYSQL_ROOT_PASSWORD=admin
      - MYSQL_DATABASE=db
    volumes:
      - symfony-data:/var/lib/mysql
volumes:
  symfony-data:
```

A terminal window with a dark blue background and green text. It displays a Docker Compose configuration file. The configuration defines three services: 'app' (built from a local Dockerfile), 'webserver' (nginx:latest), and 'db' (mysql:latest). It also defines a volume named 'symfony-data'. At the bottom of the terminal, there is a menu bar with various keyboard shortcuts and their corresponding actions in Spanish, such as '^G Ayuda', '^O Guardar', '^W Buscar', etc.

Si no lo tenemos en marcha hay que ponerlo en marcha.(en mi caso ya están alzados).

Ahora movemos los siguientes archivos:

UserController.php a → src/Controller/



routes.yaml a → config/



Y modificamos .env que esta en la raiz del archivo y ponemos lo siguiente:

DATABASE_URL=mysql://root:[admin@db](#):3306/db

```
GNU nano 6.2 .env
# In all environments, the following files are loaded if they exist,
# the latter taking precedence over the former:
#
# * .env                contains default values for the environment variables needed by the app
# * .env.local          uncommitted file with local overrides
# * .env.$APP_ENV       committed environment-specific defaults
# * .env.$APP_ENV.local uncommitted environment-specific overrides
#
# Real environment variables win over .env files.
#
# DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.
# https://symfony.com/doc/current/configuration/secrets.html
#
# Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).
# https://symfony.com/doc/current/best_practices.html#use-environment-variables-for-infrastructure-configuration
###> symfony/framework-bundle ###
APP_ENV=dev
APP_SECRET=2ca2010d40dd390825570fb98a5b1f60
DATABASE_URL=mysql://root:admin@db:3306/db
###< symfony/framework-bundle ###
```

[21 líneas escritas]

Si todo esta correcto deberíamos entrar en <http://localhost/user/test> y ver lo siguiente.

