

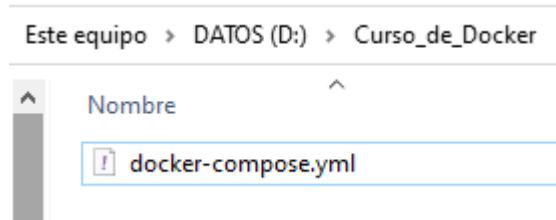
Para armar el docker-compose.yml seguí lo indicado en Docker hub para Symfony (<https://hub.docker.com/r/bitnami/symfony>) dónde nos indica un link para ver el código para dicho archivo (<https://raw.githubusercontent.com/bitnami/containers/main/bitnami/symfony/docker-compose.yml>).

- 1 Dicho código es el siguiente

```
version: '2'

services:
  mariadb:
    image: docker.io/bitnami/mariadb:10.6
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes
      - MARIADB_USER=bn_myapp
      - MARIADB_DATABASE=bitnami_myapp
  myapp:
    image: docker.io/bitnami/symfony:6.2
    ports:
      - '8000:8000'
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes
      - SYMFONY_DATABASE_HOST=mariadb
      - SYMFONY_DATABASE_PORT_NUMBER=3306
      - SYMFONY_DATABASE_USER=bn_myapp
      - SYMFONY_DATABASE_NAME=bitnami_myapp
volumes:
  - './my-project:/app'
depends_on:
  - mariadb
```

- 2 Creo el archivo "docker-compose.yml" en un directorio (en este caso en "Curso_de_Docker")



Luego copio lo indicado el punto 1, pero realizando las siguientes modificaciones

- a- El nombre de la app:
myapp => appcompras
- b- Personalizé el nombre de los contenedores agregando lo siguiente
El de mariadb: container_name: c_1_mariadb
El de appcompras: container_name: c_app_compras
- c- Personalizé el mapeo de puertos:
8000:8000 => 8080:8000
- d- El nombre de la bd y del usuario
En bn_myapp => user_db
En bitnami_myapp => compras_db

e- Agrego variable de entorno para trabajar con symfony/skeleton

En appcompras:

...

enviromente:

...

- SYMFONY_PROJECT_SKELETON=symfony/skeleton

f- El nombre con el que quiero que se cree mi volumen/app

volumes:

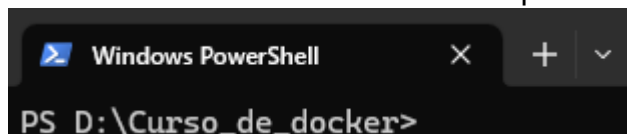
- './my-project:/app' => - './docker_tp_integrador:/app'

-3 El archivo “docker-compose.yml” me queda así:

```
version: '2'

services:
  mariadb:
    image: docker.io/bitnami/mariadb:10.6
    container_name: c_1_mariadb
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes
      - MARIADB_USER=user_db
      - MARIADB_DATABASE=compras_db
  appcompras:
    image: docker.io/bitnami/symfony:6.2
    container_name: c_app_compras
    ports:
      - '8080:8000'
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes
      - SYMFONY_DATABASE_HOST=mariadb
      - SYMFONY_DATABASE_PORT_NUMBER=3306
      - SYMFONY_DATABASE_USER=user_db
      - SYMFONY_DATABASE_NAME=compras_db
      - SYMFONY_PROJECT_SKELETON=symfony/skeleton
    volumes:
      - './docker_tp_integrador:/app'
    depends_on:
      - mariadb
```

-4 Por consola accedo al directorio que contiene el archivo “docker-compose.yml” creado



-5 Ejecuto “docker-compose up -d” (o simplemente “docker-compose up” para ver los detalles y que quede escuchando).

Con esto descarga las imágenes (si no están descargadas), crea los contenedores (si no están creados) y los pone a correr.

La primera vez va a descargar lo necesario

```
Windows PowerShell
PS D:\Curso_de_Docker> docker-compose up -d
[+] Running 0/4
- mariadb Pulling
- 78fbd6370c08 Downloading [=====> ]
- appcompras Pulling
- 10947a65b32d Downloading [=====> ]
```

Al finalizar lo anterior, nos indicará que se completó y que la red está creada y los contenedores están corriendo.

```
Windows PowerShell
PS D:\Curso_de_Docker> docker-compose up -d
[+] Running 4/4
- mariadb Pulled
- 78fbd6370c08 Pull complete
- appcompras Pulled
- 10947a65b32d Pull complete
[+] Running 3/3
- Network curso_de_docker_default Created
- Container c_1_mariadb Started
- Container c_app_compras Started
PS D:\Curso_de_Docker> |
```

Si los detenemos y volvemos a ejecutar “docker-compose up -d”, al finalizar nos indicará que la red está creada y los contenedores están corriendo.

```
Windows PowerShell
PS D:\Curso_de_Docker> docker-compose up -d
[+] Running 3/3
- Network curso_de_docker_default Created
- Container c_1_mariadb Started
- Container c_app_compras Started
PS D:\Curso_de_Docker>
```

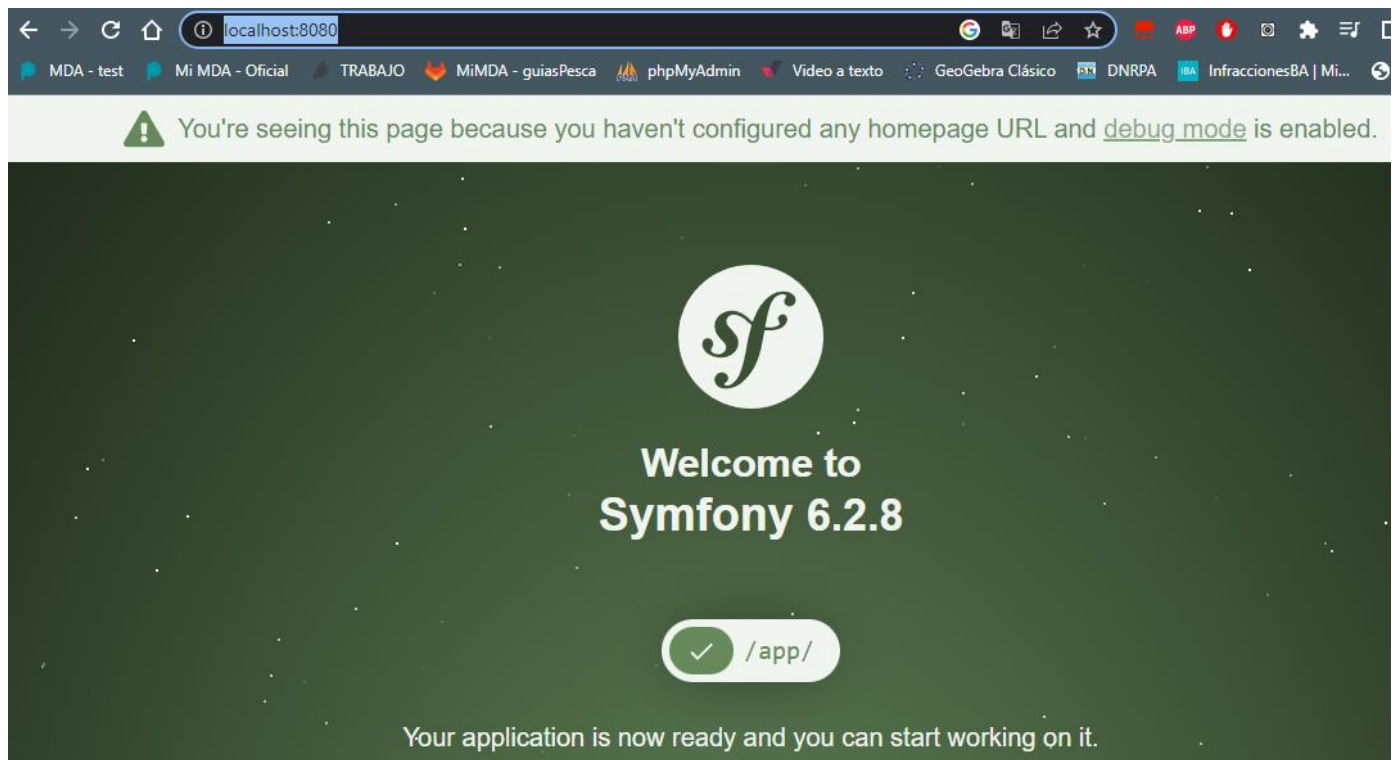
-6 Ejecuto “docker ps” para verificar que los contenedores están ejecutándose

```
Windows PowerShell
PS D:\Curso_de_Docker> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f1370370c6dd	bitnami/symfony:6.2	"/opt/bitnami/script..."	29 minutes ago	Up 29 minutes	0.0.0.0:8080->8080/tcp	c_app_compras
fc0f782d116d	bitnami/mariadb:10.6	"/opt/bitnami/script..."	29 minutes ago	Up 29 minutes	3306/tcp	c_1_mariadb

```
PS D:\Curso_de_Docker> |
```

Ya puedo acceder a mi app a través de “localhost:8080”



-7 Con los contenedores corriendo ejecuto “docker exec -it c_app_compras /bin/bash”

(dónde “c_app_compras” es el nombre del contenedor de mi app).

Con esto puedo trabajar con symfony como lo hago desde una consola fuera de un contenedor

```
Windows PowerShell
PS D:\Curso_de_Docker> docker exec -it c_app_compras /bin/bash
root@f1370370c6dd:/app#
```

Por ejemplo: creación de clase Cliente (php bin/console make:entity Cliente)

```
Windows PowerShell
PS D:\Curso_de_Docker> docker exec -it c_app_compras /bin/bash
root@f1370370c6dd:/app# php bin/console make:entity Cliente

created: src/Entity/Cliente.php
created: src/Repository/ClienteRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
>
```

Hago completa la clase Cliente, la clase Compra y actualizo la bd (no creo la bd, dado que se crea junto con el contenedor).

NOTA: No agrego todos los pasos para la creación de las clases y actualización de la base de datos, porque literalmente es como se realizan habitualmente con symfony.

Para salir del bash ejecuto “exit”

```
Windows PowerShell
PS D:\Curso_de_Docker> docker exec -it c_app_compras /bin/bash
root@f1370370c6dd:/app# exit
exit
PS D:\Curso_de_Docker> |
```

-8 Acedo a la base de datos por consola (siempre con los contenedores corriendo) para verificar que se crea la bd y las tablas, para acceder ejecuto:

“docker exec -it c_1_mariadb mysql -u user_db -p”
(dónde “c_1_mariadb” es el nombre del contenedor de mi bd y user_db es el usuario).

Cuando pide la contraseña, simplemente doy enter, ya que especifiqué que para el user user_db no hay contraseña y luego puedo realizar las consultas como en mysql común.

```
Windows PowerShell
PS D:\Curso_de_Docker> docker exec -it c_1_mariadb mysql -u user_db -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.6.12-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

Muestro las bases de datos

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| compras_db |
| information_schema |
| test |
+-----+
3 rows in set (0.000 sec)
```

Muestro que las tablas Cliente y Compra están vacías

```
MariaDB [(none)]> SELECT * FROM compras_db.cliente;
Empty set (0.000 sec)

MariaDB [(none)]> SELECT * FROM compras_db.compra;
Empty set (0.000 sec)
```

Muestro en la app que no hay registros



Voy al formulario para dar de alta a un Cliente, completo los dato y doy clic en el botón para guardar



Muestro en la app que el Cliente se agregó



Muestro en la consola del contenedor de la bd que el Cliente se agregó

```
MariaDB [(none)]> SELECT * FROM compras_db.cliente;
+----+-----+-----+-----+-----+
| id | nombre | apellido | email | saldo |
+----+-----+-----+-----+-----+
| 1 | Sergio | Avalos | sergio-hdd@hotmail.com | 20000 |
+----+-----+-----+-----+-----+
```

Voy al formulario para dar de alta a una Compra, completo los dato y doy clic en el botón para guardar

← → ↻ 🏠 ⓘ localhost:8080/compra/new

MDA - test Mi MDA - Oficial TRABAJO 🔥

Create new Compra

Producto

Precio unitario

Cantidad

Cliente

[back to list](#)

Muestro en la app que el Cliente se agregó

← → ↻ 🏠 ⓘ localhost:8080/compra/

MDA - test Mi MDA - Oficial TRABAJO 🔥 MiMDA - guiasPesca 🐟

Compra index

Id	Producto	PrecioUnitario	Cantidad	ImporteCompra	actions
1	Jabón blanco	125	20	2500	show edit

[Create new](#)

Muestro en la consola del contenedor de la bd que la Compra se agregó

```
MariaDB [(none)]> SELECT * FROM compras_db.compra;
+----+-----+-----+-----+-----+-----+
| id | cliente_id | producto | precio_unitario | cantidad | importe_compra |
+----+-----+-----+-----+-----+-----+
| 1 | 1 | Jabón blanco | 125 | 20 | 2500 |
+----+-----+-----+-----+-----+-----+
```

Muestro en la y en en la consola del contenedor de la bd que al Cliente se le descontó, de su saldo, el importe de la compra

← → ↻ 🏠 ⓘ localhost:8080/cliente/

MDA - test Mi MDA - Oficial TRABAJO MiMDA - guiasPe

Cliente index

Id	Nombre	Apellido	Email	Saldo	actions
1	Sergio	Avalos	sergio-hdd@hotmail.com	17500	show edit

[Create new](#)

```
MariaDB [(none)]> SELECT * FROM compras_db.cliente;
+----+-----+-----+-----+-----+
| id | nombre | apellido | email | saldo |
+----+-----+-----+-----+-----+
| 1 | Sergio | Avalos | sergio-hdd@hotmail.com | 17500 |
+----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```