

Ejercicios sección 6

— Caso práctico 01 – Wordpress con Docker Compose

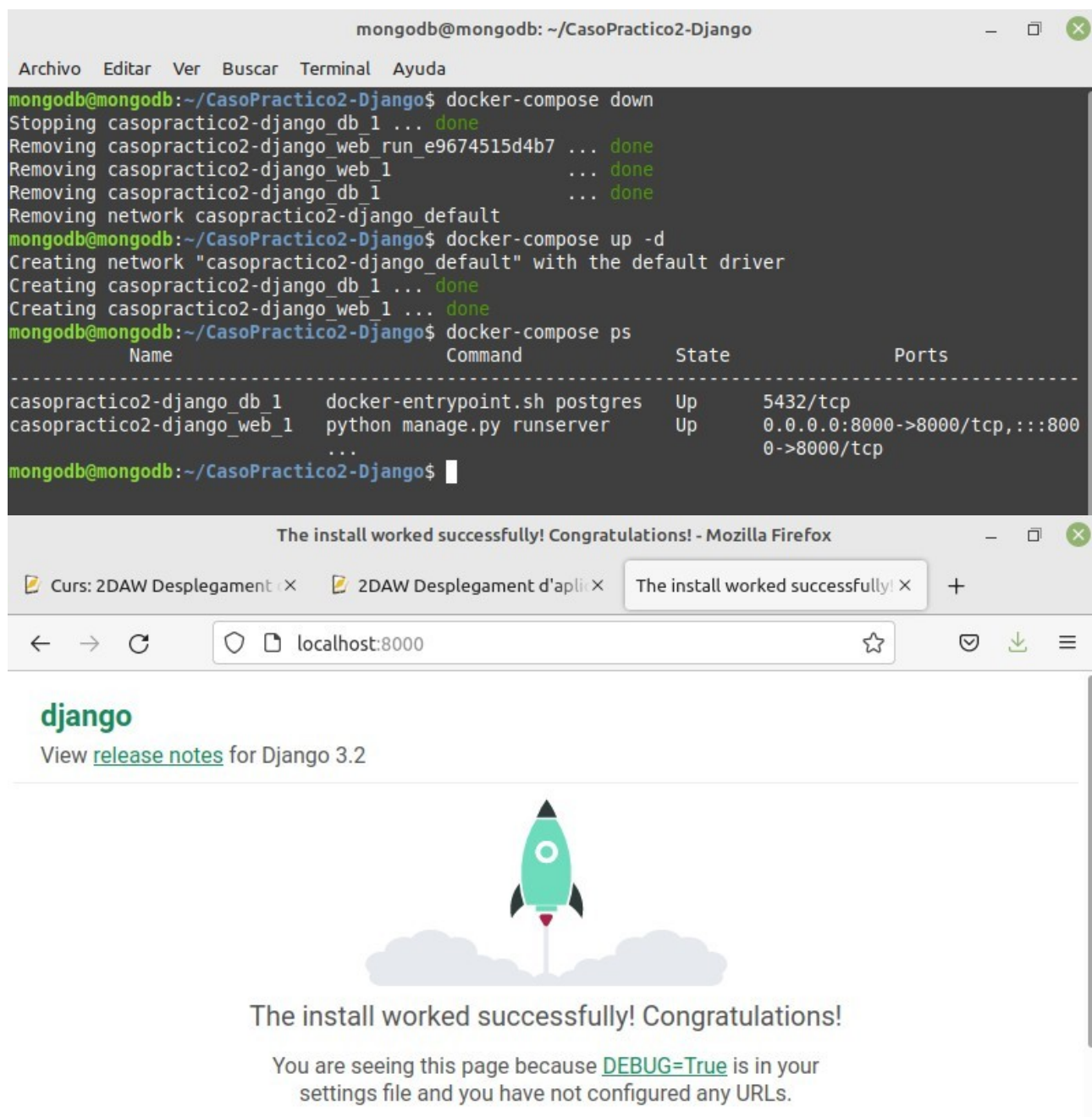


```
mongodb@mongodb: ~/CasoPractico1-Wordpress
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
mongodb@mongodb:~/CasoPractico1-Wordpress$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
AMES          35aaf9b0c114        wordpress:latest        "docker-entrypoint.s..." 2 minutes ago Up 2 minutes 0.0.0.0:8000->80/tcp, :::8000->80/tcp
asopractico1-wordpress_wordpress_1  0bebc28b5d12        mysql:5.7               "docker-entrypoint.s..." 2 minutes ago Up 2 minutes 3306/tcp, 33060/tcp
asopractico1-wordpress_db_1
mongodb@mongodb:~/CasoPractico1-Wordpress$
```

```
mongodb@mongodb: ~/CasoPractico1-Wordpress
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
mongodb@mongodb:~/CasoPractico1-Wordpress$ docker-compose down
Stopping casopractico1-wordpress_wordpress_1 ... done
Stopping casopractico1-wordpress_db_1 ... done
Removing casopractico1-wordpress_wordpress_1 ... done
Removing casopractico1-wordpress_db_1 ... done
Removing network casopractico1-wordpress_default
mongodb@mongodb:~/CasoPractico1-Wordpress$
```

```
mongodb@mongodb: ~/CasoPractico1-Wordpress
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
mongodb@mongodb:~/CasoPractico1-Wordpress$ docker-compose up -d
Creating network "casopractico1-wordpress_default" with the default driver
Creating casopractico1-wordpress_db_1 ... done
Creating casopractico1-wordpress_wordpress_1 ... done
mongodb@mongodb:~/CasoPractico1-Wordpress$
```

— Caso práctico 02 – Django con Docker Compose



The screenshot shows a terminal window titled 'mongodb@mongodb: ~/CasoPractico2-Django' with the following commands and output:

```
mongodb@mongodb:~/CasoPractico2-Django$ docker-compose down
Stopping casopractico2-django_db_1 ... done
Removing casopractico2-django_web_run_e9674515d4b7 ... done
Removing casopractico2-django_web_1 ... done
Removing casopractico2-django_db_1 ... done
Removing network casopractico2-django_default
mongodb@mongodb:~/CasoPractico2-Django$ docker-compose up -d
Creating network "casopractico2-django_default" with the default driver
Creating casopractico2-django_db_1 ... done
Creating casopractico2-django_web_1 ... done
mongodb@mongodb:~/CasoPractico2-Django$ docker-compose ps
```

Name	Command	State	Ports
casopractico2-django_db_1	docker-entrypoint.sh postgres	Up	5432/tcp
casopractico2-django_web_1	python manage.py runserver	Up	0.0.0.0:8000->8000/tcp, :::8000->8000/tcp

The terminal output is followed by a web browser window (Mozilla Firefox) showing a message: 'The install worked successfully! Congratulations!'. The browser address bar shows 'localhost:8000'.

Below the browser window, the Django logo is displayed, followed by the text: 'View [release notes](#) for Django 3.2'.

The browser window also shows a message: 'The install worked successfully! Congratulations!'.

Below the browser window, the Django logo is displayed, followed by the text: 'View [release notes](#) for Django 3.2'.

The browser window also shows a message: 'The install worked successfully! Congratulations!'.

Below the browser window, the Django logo is displayed, followed by the text: 'View [release notes](#) for Django 3.2'.

The browser window also shows a message: 'The install worked successfully! Congratulations!'.

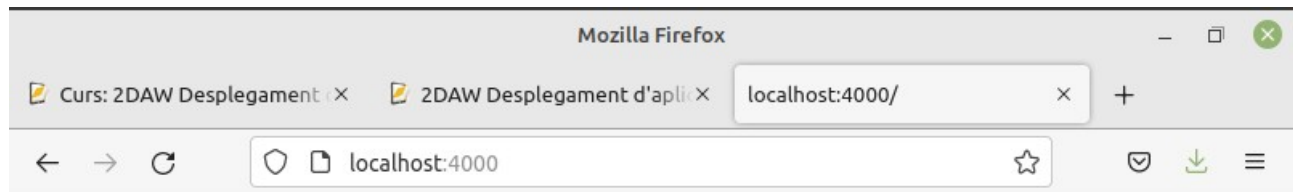
Con los pasos dados no funcionaba, ya que al crear el proyecto ejemplodjango se creaba con la estructura de carpetas:

```
codigo
├── ejemplodjango
│   ├── manage.py
│   └── ejemplodjango
│       ├── settings.py
│       └── ...
```

Para que funcione, he tenido que mover la carpeta ejemplodjango y el manage un nivel arriba:

```
codigo
├── manage.py
└── ejemplodjango
    ├── settings.py
    └── ...
```

— Caso práctico 03 – Proxy Nginx y balanceo escalado con Docker Compose



Servido por: Servidor con IP 172.20.0.2 y hostname f3bf8ceaab00

