

Creación de Sistema Web Django con base de datos Postgre con Docker Compose

1. Crear las imagenes

```
$ docker-compose build
```

Configiurar Django

1. Crear el sistema en el contenedor Web

```
docker-compose run web django-admin startproject <nombre_sistema> .
```

Si estamos usando Linux debemos cambiar los usuarios de los archivos creados desde el contenedor

```
sudo chown -R $USER:$USER <nombre_sistema> manage.py
```

2. Conectar la base de datos

En el archivo /settings.py cambiar el siguiente código:

Agregar:

```
import os
```

Arriba de:

```
from pathlib import Path
```

Y modificar:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

Por lo siguiente:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': os.environ.get('POSTGRES_NAME'),  
        'USER': os.environ.get('POSTGRES_USER'),  
        'PASSWORD': os.environ.get('POSTGRES_PASSWORD'),  
        'HOST': 'db',  
        'PORT': 5432,  
    }  
}
```

3. Correr los contenedores

```
$ docker-compose up
```

Si en la consola nos dice algo como:

```
web_1 | You have 18 unapplied migration(s). Your project may not work properly until
web_1 | you apply the migrations for app(s): admin, auth, contenttypes, sessions.
web_1 | Run 'python manage.py migrate' to apply them.
web_1 | November 24, 2022 - 00:07:42
web_1 | Django version 4.1.2, using settings 'sistema_web.settings'
web_1 | Starting development server at http://0.0.0.0:8000/
web_1 | Quit the server with CONTROL-C.
```

Entonces está funcionando

4. Correr las migraciones de Django

```
docker exec -it <id contenedor web> python manage.py migrate
```

5. Crear un superusuario (admin) Django

```
docker exec -it <id contenedor web> python manage.py createsuperuser
```

Y completas los campos que nos pide adecuadamente

6. Navegar al admin de Django

<http://localhost:8000/admin/> Ingresar con las credenciales del superusuario creado

Probar el sistema

1. Crear un usuario nuevo en el panel de Admin de Django
2. Detener los contenedores
3. Volver a correr los contenedores
4. Verificar que el usuario creado siga existiendo (esté persistido en la base de datos)