

Guía de instalación

Esta guía de instalación asume que el usuario hace uso de un dispositivo con una instalación de Docker Engine funcional, en caso de no tener una instalación de Docker funcionando, se remite a <https://docs.docker.com/get-started/get-docker/> para revisar como instalarlo para el sistema operativo deseado. Adicionalmente, se recomienda que el sistema reconozca los comandos docker.

Primeramente, se debe acceder al archivo .env, y modificar el usuario, contraseña, y nombre de la base de datos, variables DB_USER, DB_PASSWORD, y DB_DB, a unos seguros que se consideren apropiados para la instalación, adicionalmente, en casos donde se requiera un puerto diferente al puerto por defecto de la aplicación, el 8000, se debe cambiar también la variable de entorno DJANGO_PORT al valor del puerto deseado, en casos donde se vaya a modificar el proceso de instalación, es posible que se tenga que modificar el host, el puerto, o incluso haya que acceder a settings.py para realizar una configuración más específica. Para instalar la aplicación, se debe acceder a la carpeta de la aplicación, en la que se encuentra el archivo docker-compose.yml, y lanzar por consola el comando:

docker-compose up -d

Este comando sirve para lanzar los contenedores necesarios para el correcto funcionamiento de la aplicación. Si se lanza con éxito, deben generarse dos contenedores bajo el prefijo gestorddbb, llamados gestorddbb-app-1, y gestorddbb-db-1. Una vez hecho esto, se debe lanzar los siguientes comandos:

docker-compose exec app python manage.py makemigrations

docker-compose exec app python manage.py migrate

docker-compose exec app python manage.py seed_database

Esto sirve para integrar la base de datos de la aplicación con los datos por defecto, lo cual incluye un usuario con todos los permisos de nombre admin, y contraseña 1234, con la idea de que el usuario pueda acceder a la gestión de usuarios y cambiar su nombre y contraseña a unos seguros.

Con lo dicho, la aplicación ya debería estar en condiciones de operar, sin embargo, en caso de que el usuario necesite por cualquier motivo realizar cambios sobre archivos estáticos, CSS, o JavaScript de la carpeta static, o generar sus propios archivos, es recomendable que al terminar de generarlos se lance el siguiente comando para que la aplicación reconozca los cambios:

docker-compose exec app python manage.py collectstatic --noinput