Django

Es importante trabajar en un virtual environment para tener control de paquetes

# Virtual environments

## usando pipenv

-------

pip install pipenv

mkdir del proyecto

cd alli

pipenv install django==3.2.12 --python 3.10.4

-------

-------------------------

## usando python crudo

-------------------------

nos paramos en el proyecto

python -m venv venv //el segundo venv es la carpeta que se genera

luego hacemos: venv/Scripts/activate

y estamos en el (venv)

podemos instalar paquetes con pip install y quedan solo en ese venv

para salir hacemos deactivate

para ver los paquetes instalados

pip list

### generar el requirements.txt

hacemos

pip freeze > requirements.txt

### crear el venv con los paquetes del sistema

agregamos --system-site-packages

#### borrar el venv

rmdir venv /s

//eliminamos el directorio basicamente.

# DJANGO

Estando en el venv

instalar django

pip install django==3.2.12

## Comandos:

django-admin //Para ver los comandos de django

django-admin startproject djangoApp //creamos el proyectou

python manage.py runserver //con esto levantamos el server

una vez tenemos el proyecto creado, creamos una app dentro del proyecto

python manage.py startapp testApp

por cada app creada la debemos agregar en INSTALLED APPS (el nombre de la clase en apps.py dentro de mi app)

ORM : Object Relational Model

en models.py ponemos la clase de nuestra entidad

python manage.py makemigrations //para actualizar los archivos de las entidades en migrations

nos genera el código de la entidad

python manage.py sqlmigrate testApp 001

python manage.py migrate //para actualizar las entidades en base

python manage.py createsuperuser (necesita haber corrido antes el python manage.py migrate al menos una vez)

## Dockerizando

estando dentro del virtual environment

pip freeze > requirements.txt