DESPLIEGUE DE TUTRAMITE EN UBUNTU LINUX 18.04

Por defecto python viene instalado en ubuntu, así que lo que haremos es actualizar el índice del paquete local para garantizar tener la última versión:

$ sudo apt update

Comprobamos la versión de Python que has instalado (tener la versión 3.6 o superior):

$ python3 -V

Instalar con pip en un entorno virtual

La forma más flexible de instalar Django en su sistema es en un entorno virtual. Le mostraremos cómo instalar Django en un entorno virtual que crearemos con el módulo venv, parte de la biblioteca estándar de Python 3. Esta herramienta le permite crear entornos virtuales de Python e instalar paquetes de Python sin afectar el resto del sistema.

- A continuación, instalemos pip desde los repositorios de Ubuntu:

$ sudo apt install python3-pip

- Una vez que está instalado pip, puedes usarlo para instalar el paquete venv:

$ sudo apt install python3-venv

- A continuación, cree un entorno virtual dentro del directorio del proyecto utilizando el comando python que es compatible con su versión de Python. Llamaremos a nuestro entorno virtual **entornoVirtual**, pero puede nombrarlo como desee esto creará un directorio con el nombre del entorno en el path donde nos encontremos situado:

$ python3 -m venv entornoVirtual

- Instalamos el paquete GIT si no lo tenemos instalados con el siguiente comando

$ sudo apt install git

- Clonamos nuestro proyecto de GIT

git clone https://github.com/dacostaj/Tutramite.git

El comando anterior nos clonará el proyecto que tenemos en nuestro repositorio de github creando un directorio con el nombre **Tutramite** podemos cerrar la consola de Git Bash y regresar a nuestra consola de cmd.

- Paso siguiente ingresamos a nuestro directorio clonado con el siguiente comando

$ cd Tutramite

- Estando posicionado en nuestro directorio clonado, para instalar paquetes en el entorno aislado, debe activarlo escribiendo:

$ source ../entornoVirtual/bin/activate

En su nuevo entorno, puede usar pip para instalar Django. Independientemente de su versión de Python, pip debería llamarse pip cuando esté en su entorno virtual. También tenga en cuenta que no necesita usar sudo ya que está instalando localmente:

INSTALACIÓN DE DEPENDENCIAS.

Dentro del directorio raíz de nuestro proyecto de github tendremos un archivo con el nombre de **requirements.txt**, el cual nos sirve para instalar todo lo necesario para que nuestro proyecto corra perfectamente.

(entornoVirtual)$ pip install -r requirements.txt

Al ejecutar el comando anterior nos debió instalar las aplicaciones tales como DJANGO, CELERY, PILLOW, DJANGORESTFRAMEWORK. entre otros

Un comando útil para conocer el estado real de instalación de los paquetes dentro de los entornos virtuales es el siguiente:

(entornoVirtual)$ pip freeze

CONFIGURACIÓN DE LA BASE DE DATOS Y MIGRACIONES.

- Como primer paso será realizar la configuración para la persistencia de los datos, por defecto este proyecto viene configurado con una base de datos **SQLite**, por lo que si se decide utilizar esta configuración no deberá realizar nada más, pero si su deseo es utilizar otra base de datos soportada por Django como PostgreSQL etc, se sugiere leer el documento **"Configuración de la base de datos persistente en Tutramite"**, que nos brindará los pasos necesarios para configurar una base de datos persistente en Django.

- Nuestro paso siguiente será generar las migraciones correspondientes ejecutando los siguientes comandos:

```
(entornoVirtual)$ python manage.py makemigrations entity

(entornoVirtual)$ python manage.py makemigrations civil_servant

(entornoVirtual)$ python manage.py makemigrations formality

(entornoVirtual)$ python manage.py makemigrations attachment
```

- Paso siguiente corremos el comando migrate para implementar esas migraciones en tablas en nuestra base de datos sqlite corriendo el siguiente comando:

```
(entornoVirtual)$ python manage.py migrate
```

- Crearemos un super usuario con el fin de administrar el portal, al ejecutar el siguiente comando nos pedirá ingresar usuario, email y contraseña, el usuario y contraseña nos servirán como credenciales de acceso.

```
(entornoVirtual)$ python manage.py createsuperuser
```

- Para ejecutar el proyecto sobre el servidor debe usarse el siguiente comando, puede verificar en la URL (regularmente es 127.0.0.1 localhost en el puerto 8000, esto puede variar según su configuración.

```
(entornoVirtual)$ python manage.py runserver
```

- Para abandonar su entorno virtual, debe emitir el comando de desactivación desde cualquier lugar del sistema:

```
(entornoVirtual)$ deactivate
```