DESPLIEGUE DE TUTRAMITE EN WINDOWS 10

Instalación de Python en Windows

- Dirigirse a https://python.org
- Ir a la sección de descargas
- Descargar cualquier versión igual o superior a 3.6.*
- Debemos tener muy en cuenta check el combo donde nos indica adiciona python al path para poder llamarlo desde el cmd.

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

> python -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 abrimos la cmd y se crea 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 en nuestro caso se creará en el escritorio:

> python -m venv entornoVirtual

- Instalamos el paquete GIT si no lo tenemos instalados nos dirgimos al sitio web https://git-scm.com/downloads y descargamos la versión de windows. para instalarlo ejecutamos el instalador damos click en siguiente, siguiente hasta finalizar.
- Ejecutamos la aplicación **Git Bash** recién instalada, nos posicionamos en el path donde deseamos que se almacene nuestro repositorio clonado de manera local en mi caso en el escritorio.

- Clonamos nuestro proyecto que se encuentra en nuestro repositorio de Github con el siguiente comando.

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:

> "C:\Users\dacostaj\Desktop\entornoVirtual\Scripts\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 porque SQLite usa un archivo autónomo sobre el sistema de archivos para guardar los datos y Django lo crea automáticamente, 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