

MANUAL DE INSTALACION BACKEND LOCALMENTE

Herramientas utilizadas:

1. Python (<https://es.wikipedia.org/wiki/Python>)

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.



2. PIP ([https://es.wikipedia.org/wiki/Pip_\(administrador_de_paquetes\)](https://es.wikipedia.org/wiki/Pip_(administrador_de_paquetes)))

Pip es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python. Es un acrónimo recursivo que se puede interpretar como Pip Instalador de Paquetes o Pip Instalador de Python.

3. PostgreSQL (<https://openwebinars.net/blog/que-es-postgresql/>)

PostgreSQL es un sistema gestor de bases de datos relacionales, está orientado a objetos, es decir, todos los elementos de nuestra base de datos van a poder tratarse como objetos, algo parecido a un lenguaje de programación, además es multiplataforma y open source.



El código fuente del Backend que se encuentra dentro del CD, se encuentra igual a como está subido en el servidor www.heroku.com

Para instalarlo localmente, se requiere:

1. Descargar e instalar Python: <https://www.python.org/>
2. Descargar e instalar PIP: <https://tecnonucleous.com/2018/01/28/como-instalar-pip-para-python-en-windows-mac-y-linux/>
3. Descargar e instalar PostgreSQL: <https://www.postgresql.org/>

Base de Datos:

Es necesario crear una Base de Datos, para ello se ejecuta PostgreSQL:

1. Al ejecutar por primera vez PostgreSQL, le solicitara una contraseña
2. Crear un usuario de tipo superuser y dar permisos para crear bases de datos
3. Crear una base de datos, con propietario el usuario creado anteriormente

Backend:

Es necesario ingresar por consola a la carpeta donde se encuentra el Backend, después:

1. Crear un entorno virtual: `python -m venv myenv`
2. Activar el entorno virtual:
 - a. Ingresar a la carpeta Scripts del entorno virtual: `cd myenv/Scripts/`
 - b. Escribir `actívale` para activar el entorno virtual
 - c. Después de activarlo, regresar a la carpeta raíz del Backend
3. Instalar las dependencias necesarias: `pip install -r requirements.txt`

Después de tener creada la Base de Datos, será necesario modificar en el archivo settings dentro de la carpeta BackendTG, el valor DEBUG a TRUE, y la sección DATABASES:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'mydatabase',  
        'USER': 'mydatabaseuser',  
        'PASSWORD': 'mypassword',  
        'HOST': '127.0.0.1',  
        'PORT': '5432',  
    }  
}
```

Siguiente, se ejecutarán las migraciones, para ello el comando:

1. `python manage.py makemigrations`
2. `python manage.py migrate`

Crear superusuario que tendrá acceso a todo:

```
python manage.py createsuperuser --username usuario --email usuario@usuario.com
```

Luego ejecutar: `python manage.py runserver`

Es posible que, por cuestiones de seguridad, ya que el Backend se encuentra alojado en un servidor, le solicite autenticación para poder ver la información localmente, para lo cual será necesario ingresar a la carpeta de cada modulo (categorías, comentarios, eventos, lugares, municipios, opiniones, preferencias, productos, solicitudes, suscripciones, tags, usuarios, visitas) `/api/views.py` y comentar los valores `render_classes` y `permission_classes`.