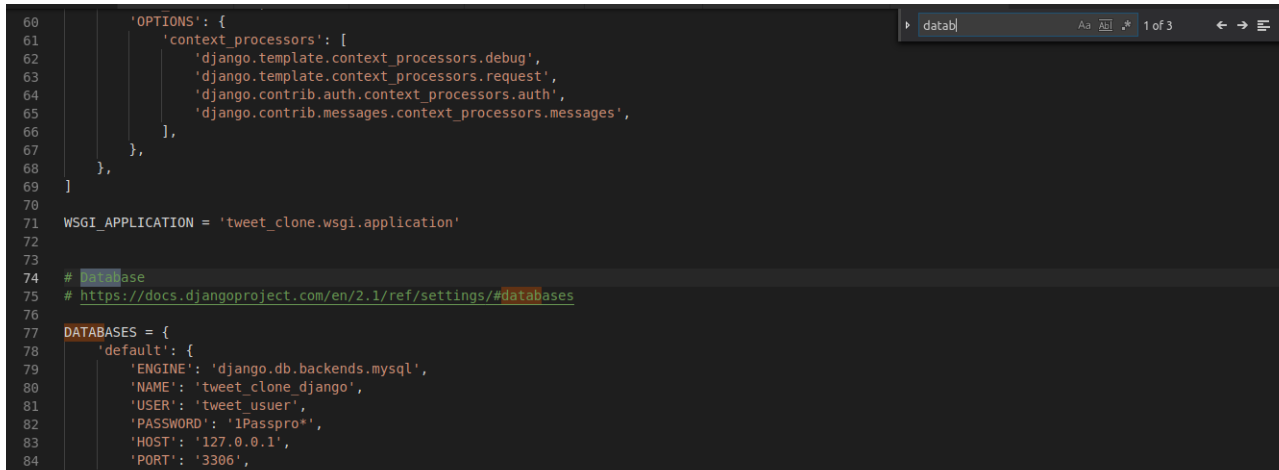


Configurar la base de datos mysql

antes de configurar la base de datos debemos tener instalado el cliente servidor de mysql en nuestro sistema

iremos a configurar el backend que esta en la parte de DATABASES en el settings.py

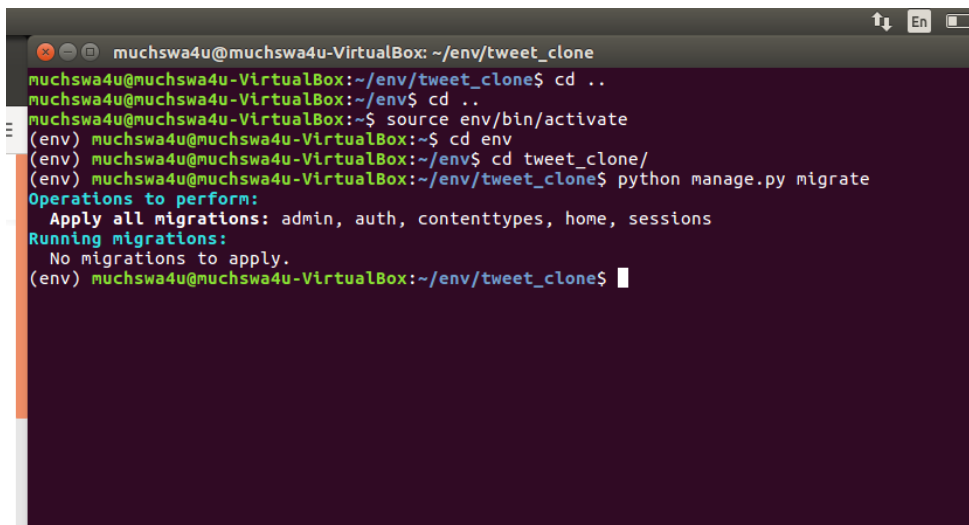


```
60     'OPTIONS': {
61         'context_processors': [
62             'django.template.context_processors.debug',
63             'django.template.context_processors.request',
64             'django.contrib.auth.context_processors.auth',
65             'django.contrib.messages.context_processors.messages',
66         ],
67     },
68 },
69 ]
70
71 WSGI_APPLICATION = 'tweet_clone.wsgi.application'
72
73
74 # Database
75 # https://docs.djangoproject.com/en/2.1/ref/settings/#databases
76
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.mysql',
80         'NAME': 'tweet_clone_django',
81         'USER': 'tweet_usuer',
82         'PASSWORD': '1Passpro*',
83         'HOST': '127.0.0.1',
84         'PORT': '3306',
```

cambiaremos la base por default y le pondemos mysql en la parte de ENGINE en la linea 79 de la patalla anterior posteriormente agregaremos los datos siguientes de la base de dato tales como el
NAME hace referencia al nombre de la base de datos creada
USER hace referencia el usuario administrador creado
PASSWORD la contrasenia asginada a ese usuario
HOST la ip en la que va acorrer la base de datos
PORT : el puerto donde se asignara la base de datos local.

Guardamos y despues iremos a nuestra consola y realizaremos las migraciones

ingresando en nuestra consola el siguiente comando
python manage.py migrate



```
muchswa4u@muchswa4u-VirtualBox: ~/env/tweet_clone
muchswa4u@muchswa4u-VirtualBox:~/env/tweet_clone$ cd ..
muchswa4u@muchswa4u-VirtualBox:~/env$ cd ..
muchswa4u@muchswa4u-VirtualBox:~$ source env/bin/activate
(env) muchswa4u@muchswa4u-VirtualBox:~$ cd env
(env) muchswa4u@muchswa4u-VirtualBox:~/env$ cd tweet_clone/
(env) muchswa4u@muchswa4u-VirtualBox:~/env/tweet_clone$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, home, sessions
Running migrations:
  No migrations to apply.
(env) muchswa4u@muchswa4u-VirtualBox:~/env/tweet_clone$
```

despues crearemos un modelo , nos iremos a models.py y escribiremos la siguiente linea de codigo

```
1 from django.db import models
2
3 # Create your models here.
4 class Tweet(models.Model):
5     content = models.CharField(max_length=256)
6     timestamp = models.DateTimeField(auto_now_add=True)
```

despues crearemos migraciones el comando para realizar migraciones

el comando sera el siguiente
python manage.py makemigrations

```
No migrations to apply.
(env) muchswa4u@muchswa4u-VirtualBox:~/env/tweet_clone$ python manage.py makemigrations
No changes detected
(env) muchswa4u@muchswa4u-VirtualBox:~/env/tweet_clone$
```

despues tendremos que registrar en el archivo admin.py

ingresando el siguiente modelo
admin.site.register(Tweet)

asi tambien importando del modelo

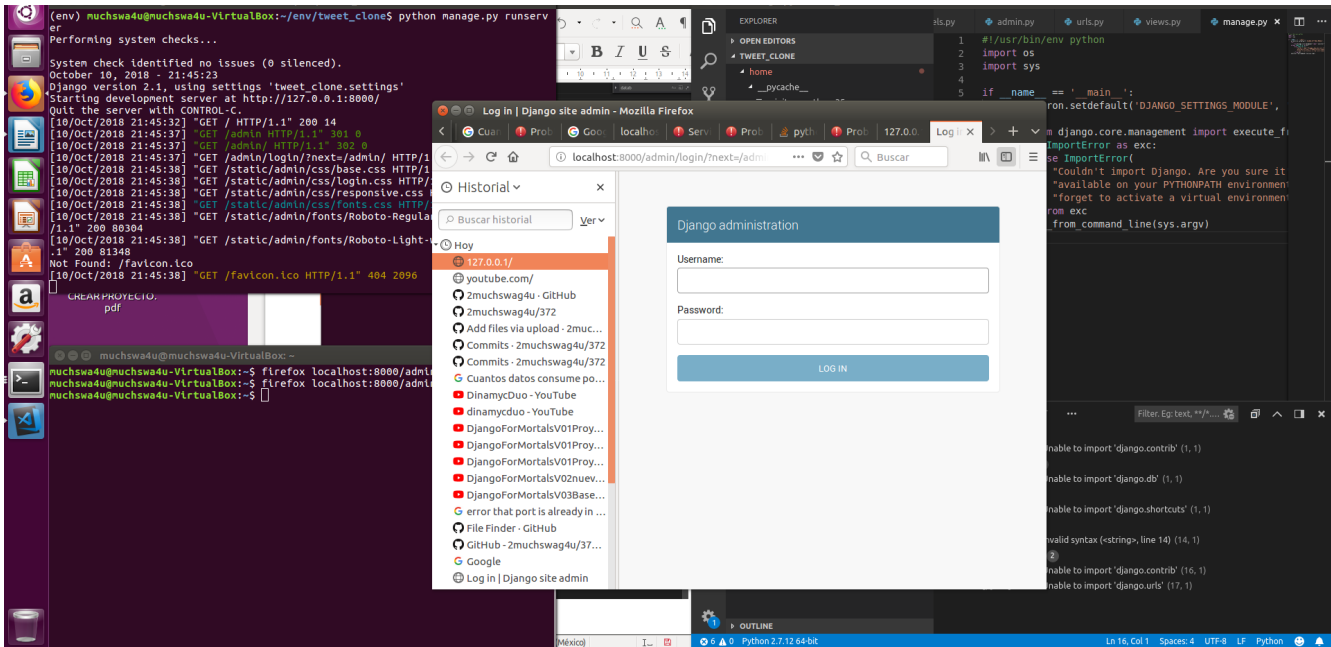
from .models import Tweet

EJEMPLO:

```
admin.py - tweet_clone - Visual Studio Code
1 from django.contrib import admin
2 from .models import Tweet
3 # Register your models here.
4 admin.site.register(Tweet)
```

despues ingresaremos como admin , entraremos a la terminal y ejecutaremos este comando
abriremos nustr navegador

firefox localhost:8000/admin
nos abra el navegador en la pantalla de admin



para poder acceder al panel previamene se debio de haber registrado a un superusuario

con el siguiente comando

python manage.py createsuperuser
ingresaras username en nuestro caso inmortal
email addres: admin@admin.com
password: hola123qwe

y te aparesera un mensaje de éxito.

Bueno una vez poniendo las credenciales les parecera una pantalla como la siguiente

