



Universidad Autónoma de Baja California
Campus Tijuana
Facultad de contaduría y administración



Alumno:
García Bretado David Antonio

Grupo:
372

Base de datos Mysql

Programación en Python

Tijuana B.C. a 9 de octubre de 2018

Base de datos MySQL

Lo primero que realizamos es configurar la conexión de base de datos en nuestro proyecto creado anteriormente, para esto nos dirigimos a nuestro archivo `settings.py` y en la parte de `DATABASES` le indicamos que utilizaremos el motor de `mysql` junto con los campos que se requieren que son:

```
'ENGINE': 'django.db.backends.mysql',
'NAME': '',
'USER': '',
'PASSWORD': '',
'HOST': '',
'PORT': ''
```

Para llenar estos campos creamos una base de datos para nuestro proyecto y un usuario que tenga todos los privilegios para administrar esa base de datos, una vez creada se llena de la siguiente manera:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'KamiTweets',
        'USER': 'KamiTweets',
        'PASSWORD': 'User_KamiTeets72',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}
```

Después de realizar la conexión proseguimos a migrar todas las tablas que se necesitan para funciones de manera correcta nuestro proyecto con `mysql`, para ello utilizamos el comando **`python manage.py migrate`**.

```
(skynet) john@john-Latitude-E6540:~/KamiTweet$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, tweets
Running migrations:
  No migrations to apply.
```

Una vez que hayamos migrado las tablas que se necesiten, luego nos vamos a nuestro archivo `models.py` dentro de nuestra aplicación `tweets` y en ella creamos un modelo llamado `Tweet`, dentro de el pondremos el contenido que tendrá nuestro tweet, el usuario que creo el tweet y la fecha en la que fue creada, así como también creamos un método llamado `__str__` para que nuestro contenido del tweet pueda ser visto en nuestra página de admin dentro del navegador.

```
1 from django.db import models
2 from django.conf import settings
3 # Create your models here.
4
5
6 class Tweet(models.Model):
7     user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE, default=1)
8     content = models.CharField(max_length=250)
9     timestamp = models.DateTimeField(auto_now_add=True)
10    update = models.DateTimeField(auto_now_add=True)
11
12    def __str__(self):
13        return self.content
```

Una vez creado mandamos a registrar nuestro nuevo modelo para migrarlo a nuestra base de datos, para ello utilizamos el comando **python manage.py makemigrations** y ya que hallamos registrado nuestro modelo utilizamos el comando **python manage.py migrate** y con eso ya tenemos nuestro modelo registrado en la base de datos para guardar los registros de Tweet, y para comprobarlo simplemente abrimos nuestra base de datos y comprobamos que esta nuestra tabla creada.

```
(skynet) john@john-Latitude-E6540:~/KamITweet$ python manage.py makemigrations
```

```
(skynet) john@john-Latitude-E6540:~/KamITweet$ python manage.py migrate
```

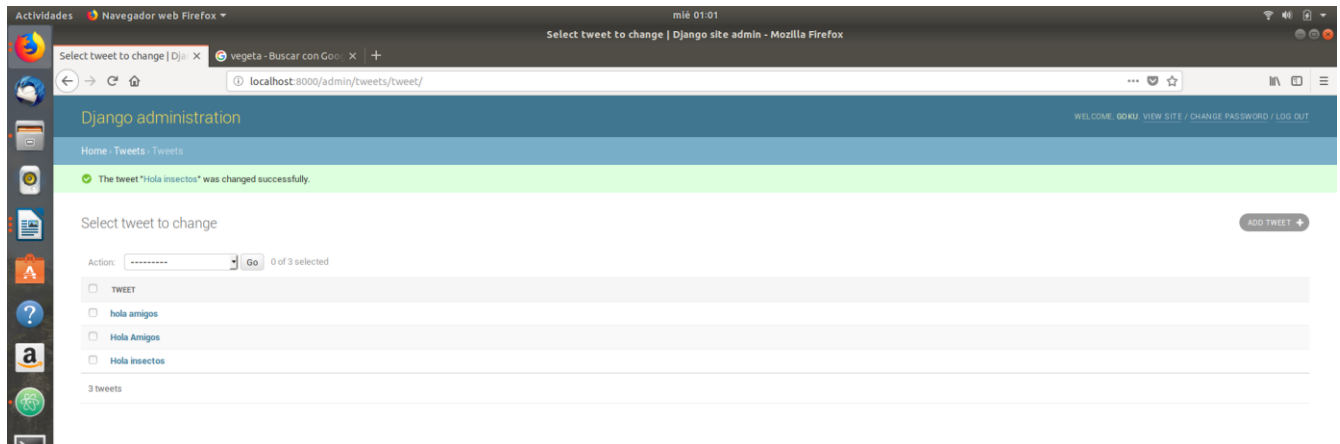
```
mysql> use KamITweets
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_KamITweets |
+-----+
| a                      |
| auth_group             |
| auth_group_permissions |
| auth_permission        |
| auth_user              |
| auth_user_groups       |
| auth_user_user_permissions |
| django_admin_log        |
| django_content_type     |
| django_migrations       |
| django_session          |
| tweets_tweet            |
+-----+
12 rows in set (0.00 sec)
```

Para probarlo se necesita registrarlo en el archivo `admin.py` de nuestra aplicación tweets, para ello utilizamos el comando `admin.site.register(Tweet)` donde el parámetro será el modelo que creamos anteriormente, y para que funcione necesitamos importar nuestro modelo

```
1 from django.contrib import admin
2
3 # Register your models here.
4 from .models import Tweet
5
6 admin.site.register(Tweet)
7
```

Y para finalizar corremos nuestro proyecto y nos dirigimos a localhost:8000/admin/ y dentro de el ya podremos crear nuestros tweets y guardarlas en la base de datos.



```
mysql> use KamiTweets
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from tweets_tweet;
+-----+-----+-----+-----+
| id | content      | timestamp                | update                |
| user_id |
+-----+-----+-----+-----+
| 1 | Hola insectos | 2018-09-29 00:41:43.582868 | 2018-09-29 00:41:43.582898 |
| 2 |              |                          |                          |
| 2 | Hola Amigos  | 2018-10-04 03:32:41.587134 | 2018-10-04 03:32:41.587164 |
| 1 |              |                          |                          |
| 3 | hola amigos  | 2018-10-04 03:55:33.862712 | 2018-10-04 03:55:33.862832 |
| 1 |              |                          |                          |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```