1) A mismo nivel manage.py

- Crear carpeta apps

- Crear carpeta templates

- Crear carpeta static

- Crear carpeta media

2) Ingresar a la carpeta principal del proyecto:

- Crear carpeta settings

- Dentro de settings creamos: base.py, local.py, production.py

3) Dentro de static

- Crear carpeta css, img y js

4) pasamos el contenido del settings.py a base.py y eliminamos settings.py.-

5) En el archivo base.py modificamos la configuración de la siguiente manera:

import os

from django.urls import reverse\_lazy

# AUTH

LOGIN\_REDIRECT\_URL = reverse\_lazy('home')

LOGOUT\_REDIRECT\_URL = reverse\_lazy('home')

LOGIN\_URL = reverse\_lazy('login')

# En TEMPLATES > DIRS :

'DIRS': [os.path.join(os.path.dirname(BASE\_DIR),'templates')],

LANGUAGE\_CODE = 'es-ar'

TIME\_ZONE = 'America/Argentina/Buenos\_Aires'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

STATIC\_URL = '/static/'

STATICFILES\_DIRS = (

os.path.join(os.path.dirname(BASE\_DIR),'static'),

)

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(os.path.dirname(BASE\_DIR),'media')

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

- Base Dejamos la configuración general que no cambia si está en local o en producción.

- Local (en desarrollo) Desarrollamos en localhost, con conexiones y contraseñas de prueba.

- Production Ya está subido a internet, con datos reales en DB.

6) de base.py a local.py:

from .base import \*

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Database

# <https://docs.djangoproject.com/en/4.2/ref/settings/#databases>

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

7) Modificar en manage.py la ruta en donde está ahora la configuración (local.py).

Linea 9 -> os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'prueba.settings.**local**')

Django está compuesto por aplicaciones. La idea es hacer las aplicaciones mas independientes posibles y luego implementarlas a todas.

La carpeta principal donde se encuentra el settings es nuestra aplicación principal.

Django funciona con el modelo MVT (Model View Template)

Navegador -> Activa urls.py manipula y checkea que vista (view) debe ejecutar -> Va al models para conectar y consultar con la DB -> y con esa información levanta un template -> lo manda al navegador.

Creamos en la app principal un archivo views y lo importamos en el urls.py:

from . import views

urlpatterns = [

path('admin/', admin.site.urls),

path('', views.home, name='index')

]

Herencia de HTMLs.

{% extends ‘base.html’ %}

{% load static %}

{% block content %}

{% endblock content %}

Creación de aplicaciones:

django-admin startapp noticias

Normalmente los nombres van en plural

Cuando se crea una aplicación, hay que agregarla a INSTALLED\_APPS en el settings/base.py

Luego agregar “apps.” al nombre de la aplicación ubicado en el archivo apps.py de la aplicación:

*class* NoticiasConfig(AppConfig):

    default\_auto\_field = '*django.db.models.BigAutoField*'

    name = '*apps.noticias*'

Y luego creamos un archivo urls.py en la aplicación con su plantilla default al no traer por defecto.

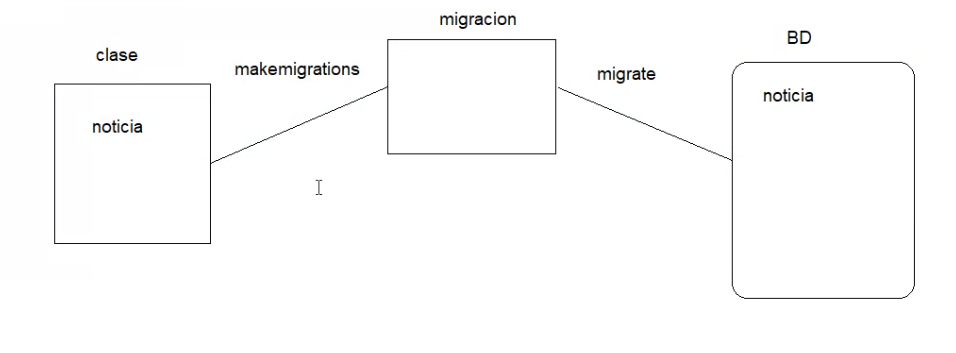
En urls.py pero de la aplicación principal hacemos referencia al archivo urls.py de la app en cuestión de la siguiente manera:

    path('*noticias/*', include('*apps.noticias.urls*')),

importando include de django.urls

En views.py de la aplicación hacemos referencia a una carpeta aplicacion/archivo.html y la creamos también en la carpeta templates para hacer referencia.

Los modelos es el modo de conectar nuestro proyecto a la base de datos.



Creamos el modelo  
python manage.py makemigrations  
Se pasó el modelo al stage de migrations.   
python manage.py migrate  
Impactó en la base de datos.

Para el admin:   
python manage.py createsuperuser  
user: scz  
pss: fscz