







# comandos

|  |
| --- |
| python -m pip install Django  python -m pip install virtualenv  python -m pip freeze  python -m venv test19  cd test19  cd Scripts  activate  cd ..  cd test19  python -m pip install Django  python -m pip freeze  deactivate  pip freeze > requierements.txt  pip install –r requierements.txt  django-admin.py  # ojo en la nueva version de python no es django-admin.py es django-admin  # lo comprove haciendo un dir en el entorno virtual  django-admin startproject mascotas  django-admin startapp nombre  manage.py createsuperuser  manage.py makemigrations  manage.py migrate  manage.py runserver  c18778540 |

crear Proyecto

crear entorno virtual y activarlo

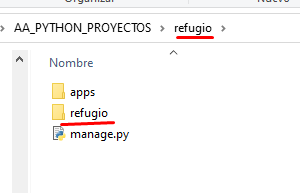
instalar dajango

# 4 crear aplicaciones

empezar de Nuevo desde el principio :S



django-admin startproject refugio



crear carpeta apps

crear en apps un archive \_\_init\_\_, se puede copiar uno del proyecto

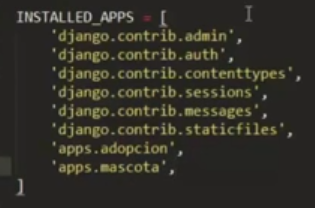


cd apps

django-admin startapp mascota

django-admin startapp adopcion

decirle a dejango que use las apps que creamos



settings.py

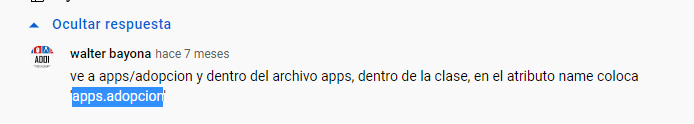
'apps.adopcion',

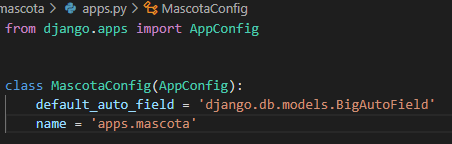
'apps. mascota',

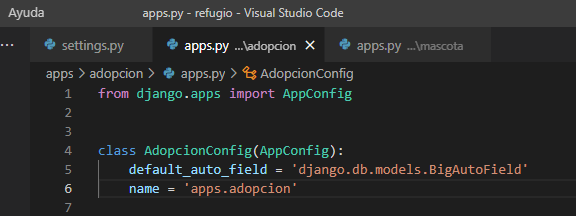
cd ..

manage.py migrate

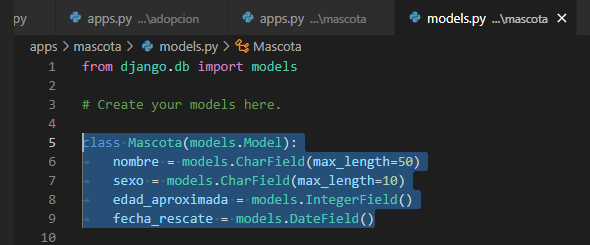
habia un error corregir con esto

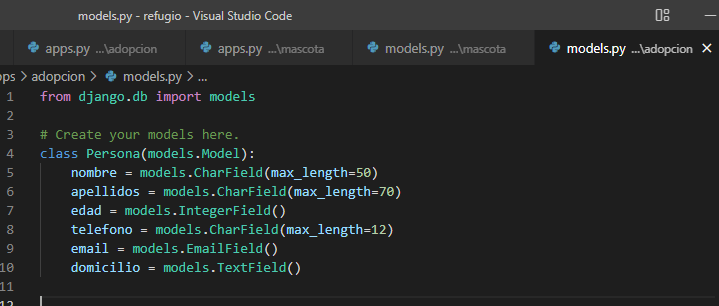






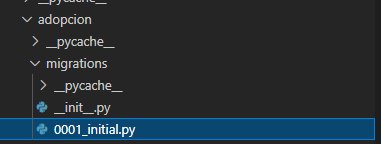
# 5 modelos y migraciones







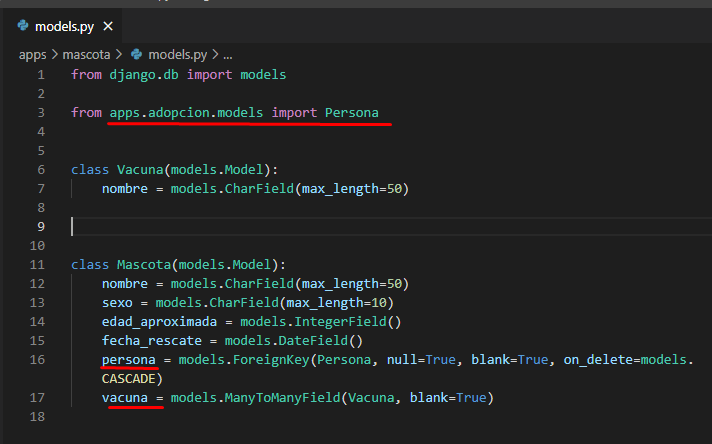
manage.py makemigrations





manage.py migrate

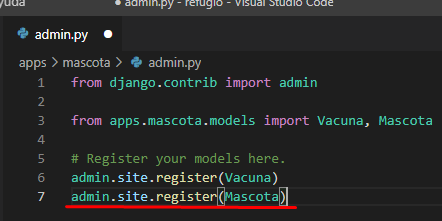
# 06.- Relaciones.mp4

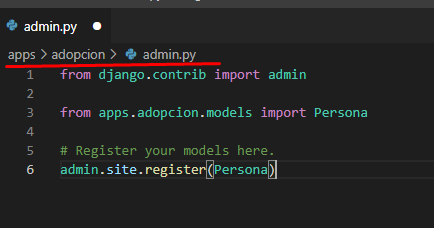


manage.py makemigrations

manage.py migrate

* registar modelos en el administrador





admin.site.register(nombreModelo)

manage.py createsuperuser

name : abraham

Email address: abrahamicm2@gmail.com

Password:c18778540

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): mascota.

Run 'python manage.py migrate' to apply them.

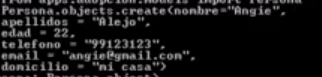
manage.py runserver

# 07.- Django Shell y Querysets.mp4

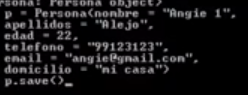




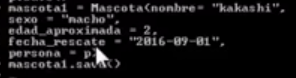
crear person forma numero 1



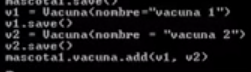
crear persona forma numero 2



asignar llave forenea



crear vacunas y asignarlas



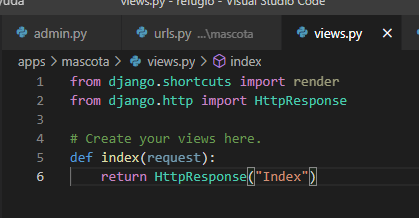
obtener todos los registros



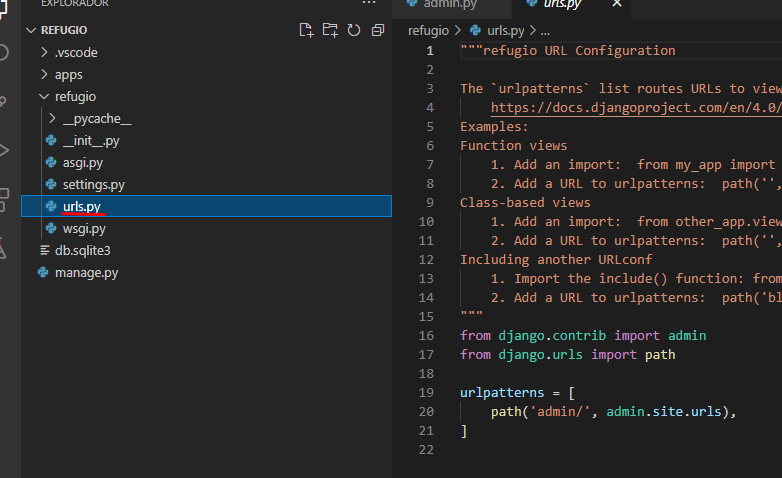




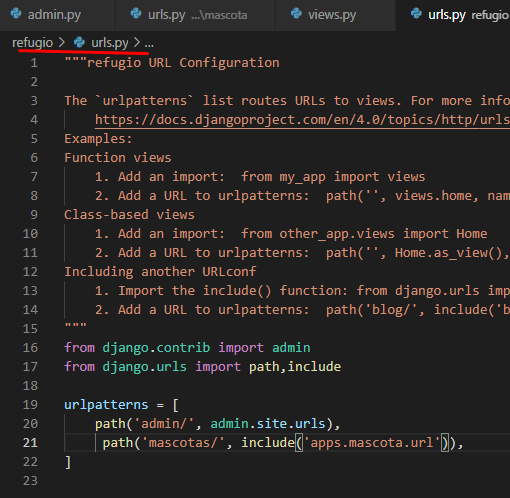
# 08.- Configurar URLs y primera view.mp4



urls globales

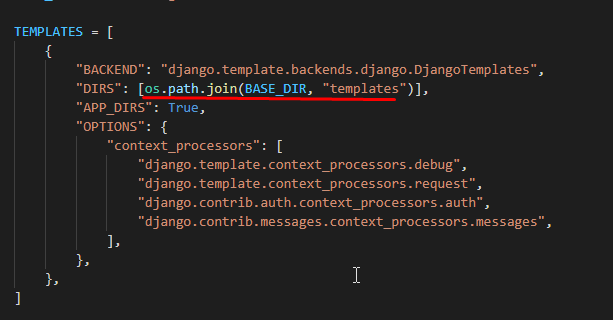


en mi aplicación crear un archivo urls.py



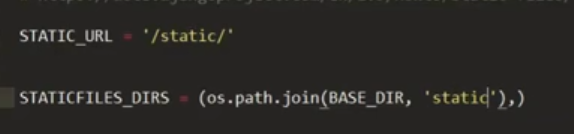
# 09.- Sistema Plantillas.mp4





# 10.- Herencia de plantillas.mp4

# 11.- Configurar archivos estáticos.mp4

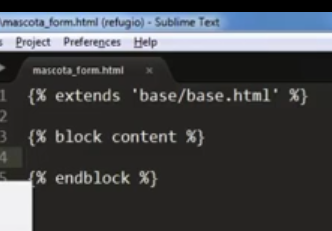




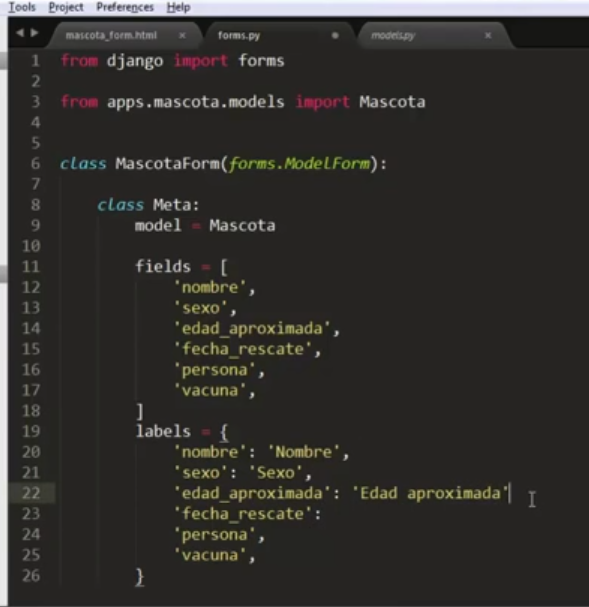


# 12.- Formularios y vistas basadas en funciones crear.mp4

1. crear plantilla para el formulario en la carpeta del template mascota



1. crear clase del formulario en la aplicación mascota

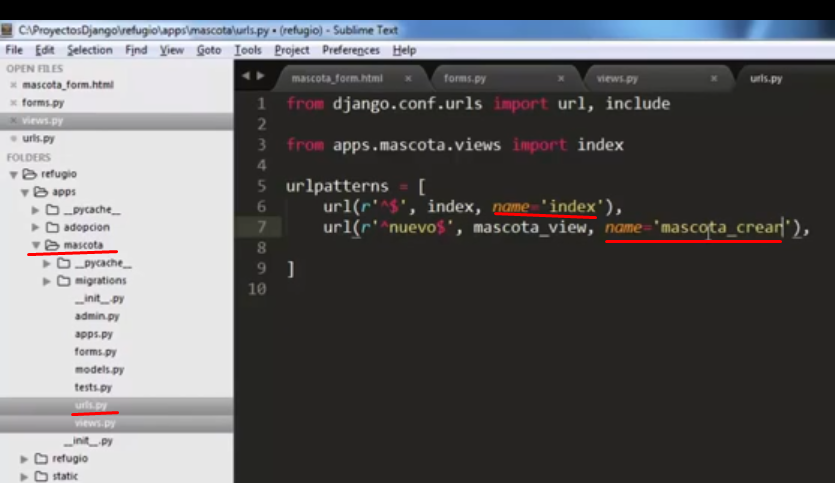




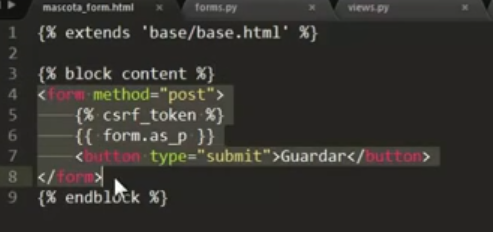
1. definir el método en la vista



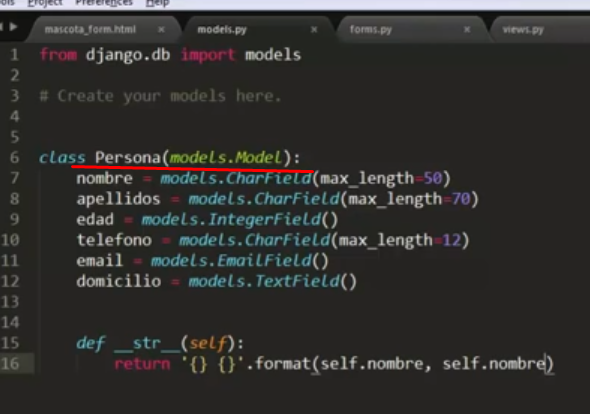
1. configurar url

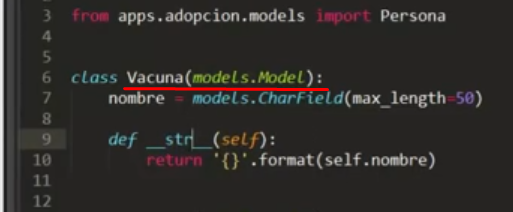


1. usar formulario en el template

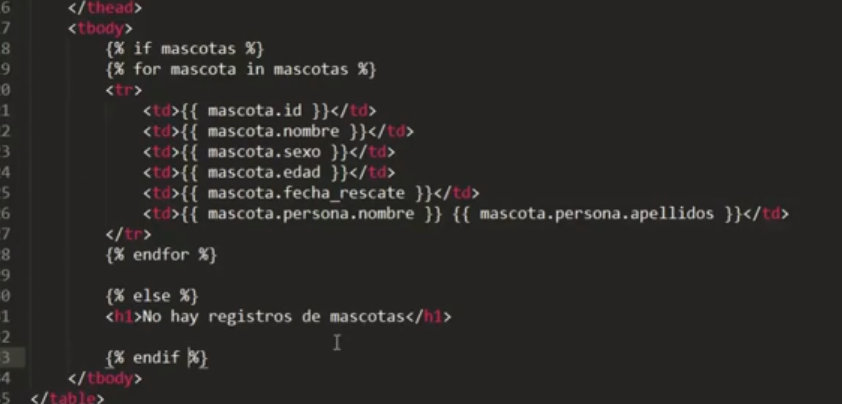


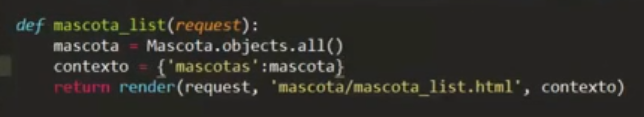
1. configurar los métodos \_\_str\_\_





# 13.- Listar registros (vistas basadas en funciones).mp4

1. crear témplate mascota listado
2. 
3. crear método en la view

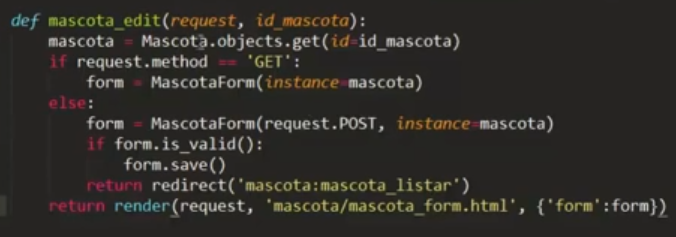


1. modificar url

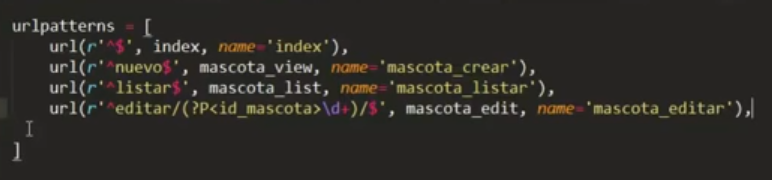


# 14.- Vistas basadas en funciones actualizar y eliminar.mp4

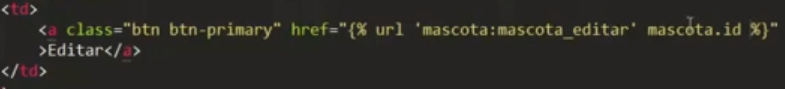
1. crear método editar



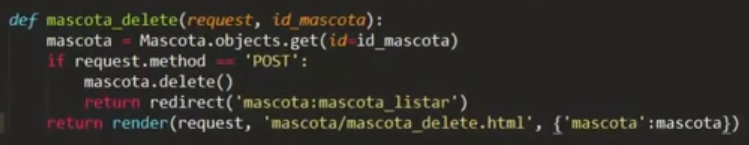
1. modificar url



1. agregar acción de editar en la vista



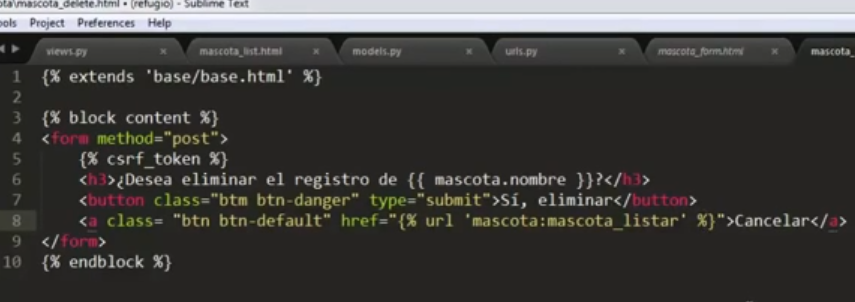
1. configurar método delete



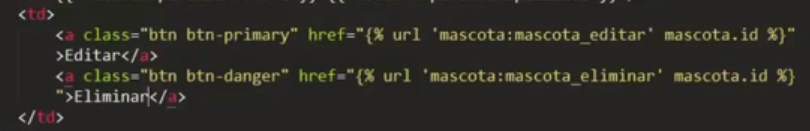
1. configurar url método delete



1. crear template para el método delete



1. añadir método delete al listado



# 15.- Vistas basadas en clases, ListView y CreateView.mp4

# 16.- UpdateView y DeleteView, vistas basadas en clases.mp4

# 17.- CRUD con dos formularios Parte 1.mp4

# 18.- CRUD con dos formularios Parte 2.mp4

# 19.- Crear Registro de Usuarios (modelo).mp4

# 20.- Crear login (facilito).mp4

# 21.- Recuperar contraseña por correo (facilito).mp4

# 22.- Decorador login required.mp4

# 23.- Serializar objetos.mp4

# 24.- Paginación.mp4

# 25.- Restframework ModelSerializer.mp4