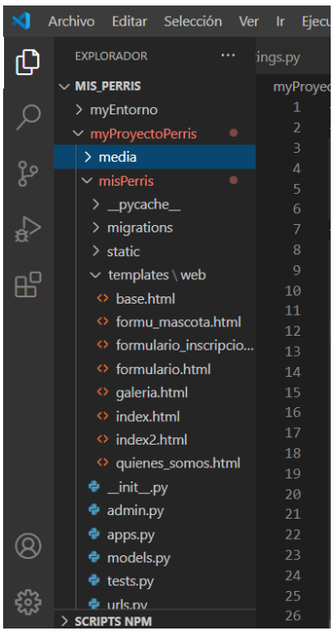
Manejo de Imágenes en Django

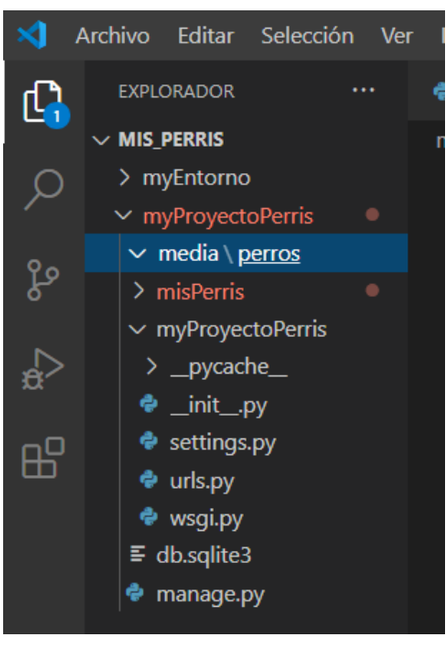
Para agregar imágenes y manipularlas sobre un modelo de base de datos, debemos realizar lo siguiente, primero debemos agregar un directorio llamado **media** sobre nuestro **proyecto**.




```
114
115   USE_TZ = True
116
117
118   # Static files (CSS, JavaScript, Images)
119   # https://docs.djangoproject.com/en/2.2/howto/static-files/
120
121   STATIC_URL = '/static/'
122
123   MEDIA_URL = '/media/'
124   MEDIA_ROOT = os.path.join(BASE_DIR,'media')
```

Definimos en nombre del directorio **media** y además indicamos su **ubicación** la cual se encuentra junto a nuestro proyecto.

Agregaremos a nuestro modelo un campo de tipo **imagen** y definimos un directorio en el interior de **media** en el cual alojaremos nuestras imágenes de perros.



Modificaremos entonces nuestro modelo de **mascotas**

settings.py | urls.py | base.html | index2.html | views.

myProyectoPerris > misPerris > models.py > Mascota

```python
from django.db import models

# Create your models here.
class Raza(models.Model):
    name=models.CharField(max_length=100, primary_key=True)
    annos=models.IntegerField()

    def __str__(self):
        return self.name

class Mascota(models.Model):
    name=models.CharField(max_length=100,primary_key=True)
    edad=models.IntegerField()
    descripcion=models.TextField()
    imagen=models.ImageField(upload_to='perros',null=True)
    raza=models.ForeignKey(Raza,on_delete=models.CASCADE)

    def __str__(self):
        return self.name
```

Para manipular imágenes en Django debemos instalar la extensión de **pillow** con el siguiente comando

PIP INSTALL PILLOW

```
System check identified 1 issue (0 silenced).

(myEntorno) C:\mis_perris\myProyectoPerris>pip install pillow
Collecting pillow
  Downloading Pillow-7.2.0-cp38-cp38-win32.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 2.2 MB/s
Installing collected packages: pillow
```

ahora debemos crear la migración al cambio realizado sobre la tabla, para ello primero utilizaremos el **makemigrations** para crear los archivos de migración y luego generar la migración(**migrate**) hacia la base de datos.

PYTHON MANAGE.PY MAKEMIGRATIONS MISPERRIS

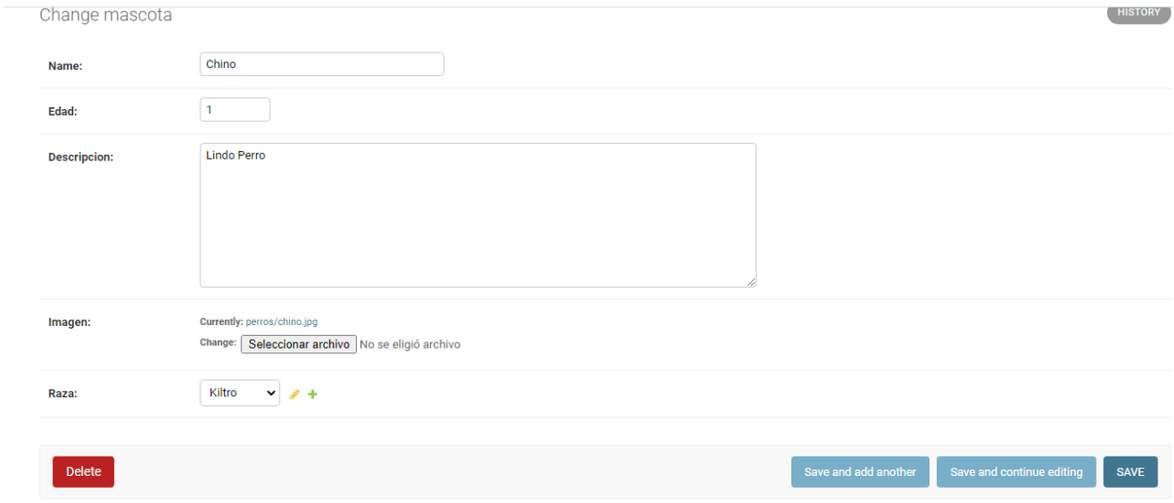
PYTHON MANAGE.PY MIGRATE

```
(myEntorno) C:\mis_perris\myProyectoPerris>python manage.py makemigrations misPerris
Migrations for 'misPerris':
  misPerris\migrations\0002_mascota_imagen.py
    - Add field imagen to mascota

(myEntorno) C:\mis_perris\myProyectoPerris>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, misPerris, sessions
Running migrations:
  Applying misPerris.0002_mascota_imagen... OK

(myEntorno) C:\mis_perris\myProyectoPerris>
```

Iniciaremos el servidor y entraremos al entorno de administración. Y agregaremos una nueva imagen sobre un registro existente.



Una vez agregado, cuando deseemos verla pinchando el vinculo asociado a la imagen nos presentara un error, debido a que no conoce la ubicación real del archivo sobre el directorio media, para ello se debe agregar en **urls.py** del **proyecto** la configuración de la ruta estática en donde se encuentra la imagen.

```python
from django.contrib import admin
from django.urls import path,include
from django.conf.urls.static import static #importar el uso de directorios estaticos
from django.conf import settings # importar el archivo de configuracion (uso de variable MEDIA)

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('misPerris.urls')),
]
#incluir en el interior del "path" la ubicacion de los directorios MEDIA
if settings.DEBUG:
    urlpatterns+=static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

Ahora podemos modificar nuestra aplicación para que pueda recibir y procesar imágenes para luego dejarlas en la galería.

```
<form action="" method="post" class="form-register">
    {% csrf_token %}
    <h2 class="form-titulo">Ingreso de una nueva mascota</h2>
    <div class="contenedor-inputs">
        <input type="text" name="nombre" placeholder="ingrese nombres" class="input-48" require
        <input type="number" name="Edad" class="input-48" required>
        <input type="text" name="descripcion" placeholder="ingrese descripcion" class="input-10
        <select class="input-100" name="raza">
            {% for raz in razas %}
                <option>{{raz.name}}</option>
            {% endfor %}
        </select>
        <input type="file" name="txtImagen" class="input-100">

        <input type="submit" value="Registrar" name="accion" class="btn-enviar">
        <input type="submit" value="Elimimnar" name="accion" class="btn-enviar">
        <p class="form-link">¿Ya tienes una cuenta? <a href="#">Ingresa Aqui</a></p>
    </div>
</form>
<h2>{{mensaje}}</h2>

{% endblock contenido %}
```

En la **view** debemos **capturar la imagen** empleando el **REQUEST.FILES.GET(“TXTIMAGEN”)**

```
raza=Raza.objects.all() # recupera todas las
if request.POST:
    accion=request.POST.get("accion")
    if accion=='Registrar':
        nombre=request.POST.get("nombre")
        edad=request.POST.get("Edad")
        desc=request.POST.get("descripcion")
        raza=request.POST.get("raza")
        imagen=request.FILES.get("txtImagen")
        obj_raza=Raza.objects.get(name=raza)
        mascota=Mascota(
            name=nombre,
            edad=edad,
            descripcion=desc,
            raza=obj_raza
        )
        mascota.save()
```

```
imagen=request.FILES.get("txtI
obj_raza=Raza.objects.get(name
mascota=Mascota(
    name=nombre,
    edad=edad,
    descripcion=desc,
    raza=obj_raza,
    imagen=imagen
)
mascota.save()
```

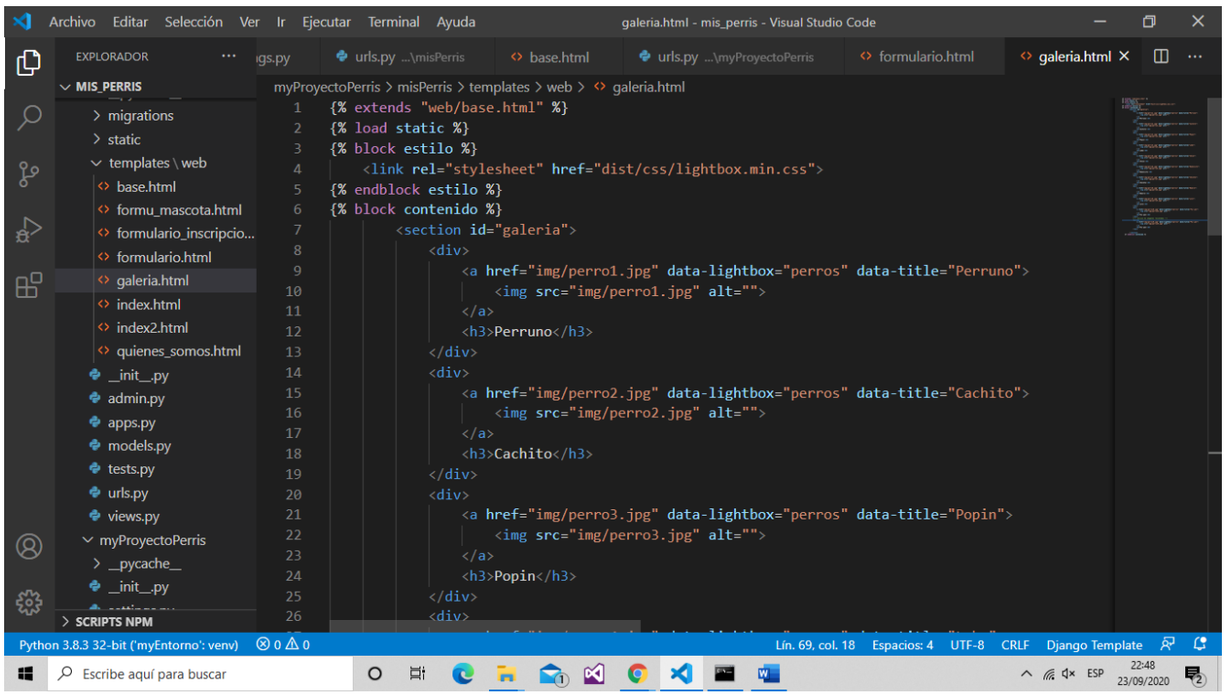
Para definir el envio de estos datos por medio del formulario html debemos agregar como argumento la **enctype=”multipart/form-data”** con esto permitimos la carga de archivos mas grandes

```
<form action="" method="post" class="form-register" enctype="multipart/form-data">
    {% csrf_token %}
```

Ahora es tiempo de probarlo con nuestro servidor corriendo.

Cargar las fichas de mascotas en la galería

Para ello debemos crear una galería que permita la carga de estos datos, crearemos una galería alterna



Pasaremos a esta galería todas las mascotas, para ello es necesario modificar la **view**

```
def galeria(request):
    mascotas=Mascota.objects.all()
    return render(request,'web/galeria.html',{'mascotas':mascotas})
```

Luego modificaremos la galería

```
<!-- seccion de imagenes rescatadas -->
{% for mas in mascotas %}
<div>
    <a href="{{mas.imagen.url}}" data-lightbox="perros" data-title="{{mas.descripcion}}">
        <img src="{{mas.imagen.url}}" alt="">
    </a>
    <h3>{{mas.name}}</h3>
</div>
{% endfor %}
```

En caso de querer eliminar debemos modificar la **view** y anotar lo siguiente