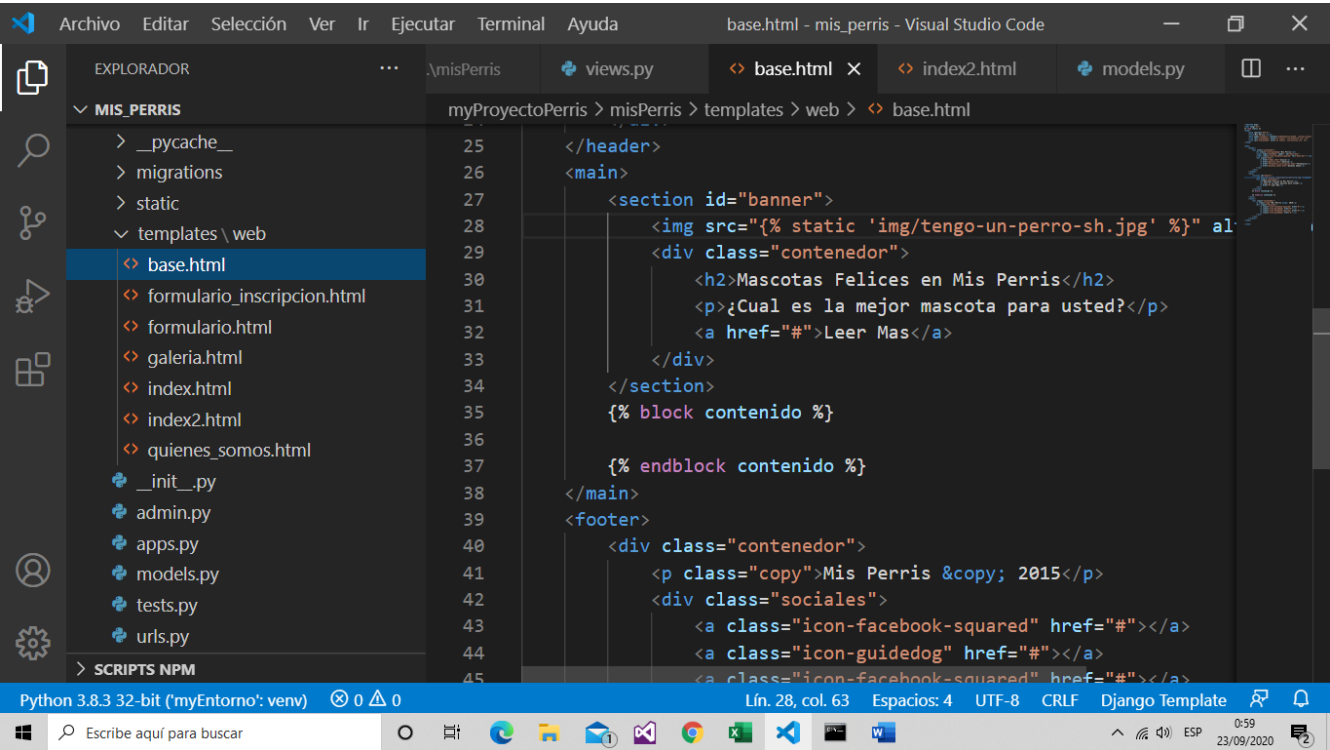


Generación de Plantillas en Templates de Django

Para crear una plantilla en django primero establecemos la estructura que deseamos repetir en las otras paginas y luego debemos incorporar sectores editables

Creación de una Plantilla



```
base.html - mis_perris - Visual Studio Code

EXPLORADOR
MIS_PERRIS
  > __pycache__
  > migrations
  > static
  > templates \ web
    <> base.html
    <> formulario_inscripcion.html
    <> formulario.html
    <> galeria.html
    <> index.html
    <> index2.html
    <> quienes_somos.html
  <+> _init_.py
  <+> admin.py
  <+> apps.py
  <+> models.py
  <+> tests.py
  <+> urls.py
  > SCRIPTS NPM

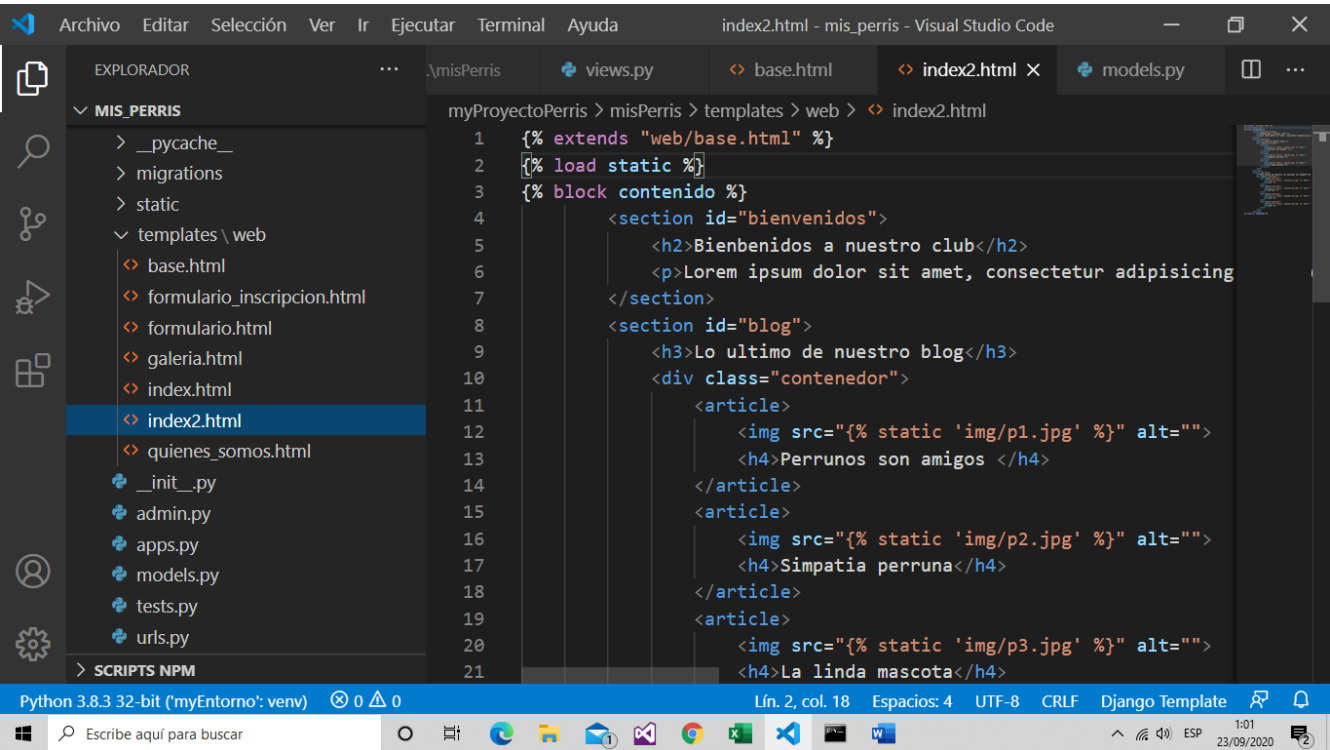
myProyectoPerris > misPerris > templates > web > <> base.html

25 </header>
26 <main>
27   <section id="banner">
28     
29     <div class="contenedor">
30       <h2>Mascotas Felices en Mis Perris</h2>
31       <p>¿Cual es la mejor mascota para usted?</p>
32       <a href="#">Leer Mas</a>
33     </div>
34   </section>
35   {% block contenido %}
36
37   {% endblock contenido %}
38 </main>
39 <footer>
40   <div class="contenedor">
41     <p class="copy">Mis Perris &copy; 2015</p>
42     <div class="sociales">
43       <a class="icon-facebook-squared" href="#"></a>
44       <a class="icon-guidedog" href="#"></a>
45       <a class="icon-facebook-squared" href="#"></a>
46     </div>
47   </div>
48 </footer>

Python 3.8.3 32-bit (myEntorno: venv) 0 0
Ln. 28, col. 63 Espacios: 4 UTF-8 CRLF Django Template
```

Para lograrlo debe agregar `{% block <nombre del bloque> %}` y finalizar con `{% endblock <nombre del block> %}`

Luego debemos crear las paginas a partir de esta plantilla



```
index2.html - mis_perris - Visual Studio Code

EXPLORADOR
MIS_PERRIS
  > __pycache__
  > migrations
  > static
  > templates \ web
    <> base.html
    <> formulario_inscripcion.html
    <> formulario.html
    <> galeria.html
    <> index.html
    <> index2.html
    <> quienes_somos.html
  <+> _init_.py
  <+> admin.py
  <+> apps.py
  <+> models.py
  <+> tests.py
  <+> urls.py
  > SCRIPTS NPM

myProyectoPerris > misPerris > templates > web > <> index2.html

1 {% extends "web/base.html" %}
2 {% load static %}
3 {% block contenido %}
4   <section id="bienvenidos">
5     <h2>Bienvenidos a nuestro club</h2>
6     <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
7   </section>
8   <section id="blog">
9     <h3>Lo ultimo de nuestro blog</h3>
10    <div class="contenedor">
11      <article>
12        
13        <h4>Perrunos son amigos </h4>
14      </article>
15      <article>
16        
17        <h4>Simpatia perruna</h4>
18      </article>
19      <article>
20        
21        <h4>La linda mascota</h4>
22      </article>
23    </div>
24  </section>
25 </block contenido %>

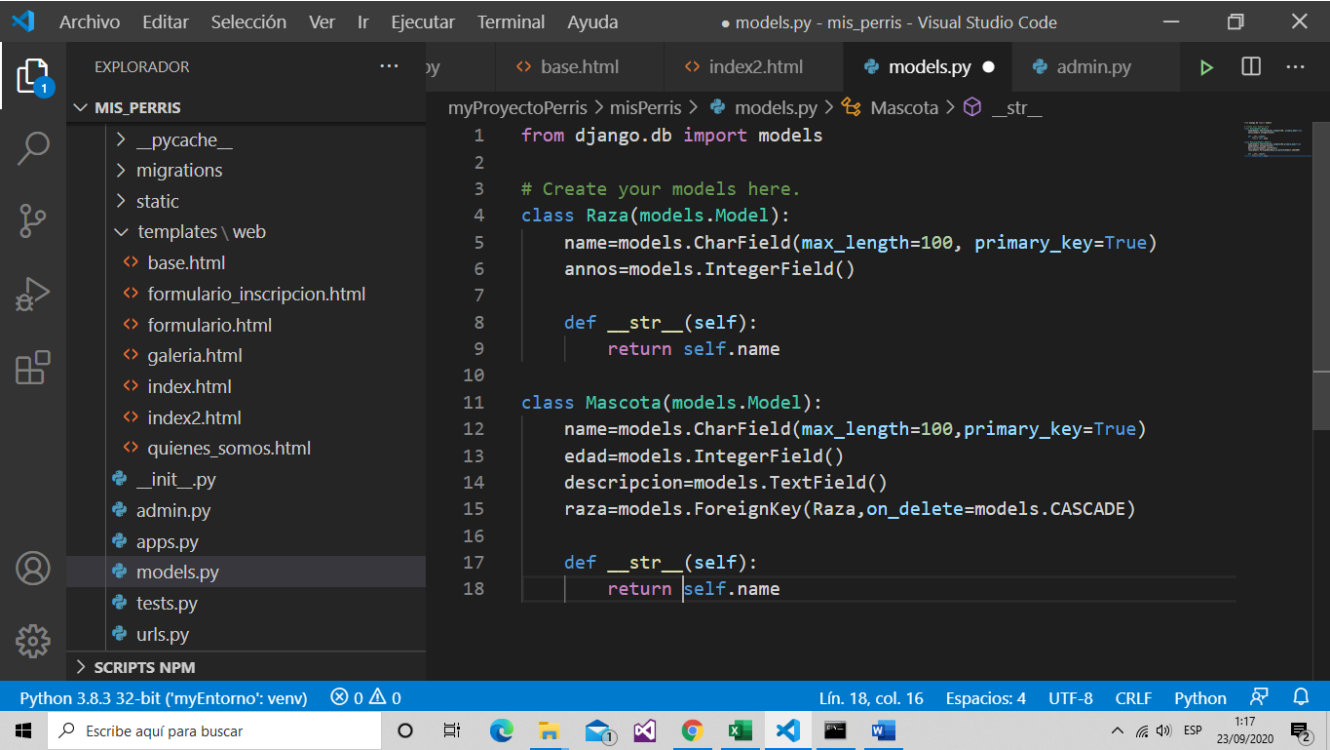
Python 3.8.3 32-bit (myEntorno: venv) 0 0
Ln. 2, col. 18 Espacios: 4 UTF-8 CRLF Django Template
```

Con `{% extends "nombre plantilla" %}` logramos vincular la plantilla y luego debemos agregar los sectores editables por medio de los `{% block <nombre> %}`

Creación de Modelos

Los modelos corresponden a las tablas las cuales podemos vincular y gestar en el interior de django

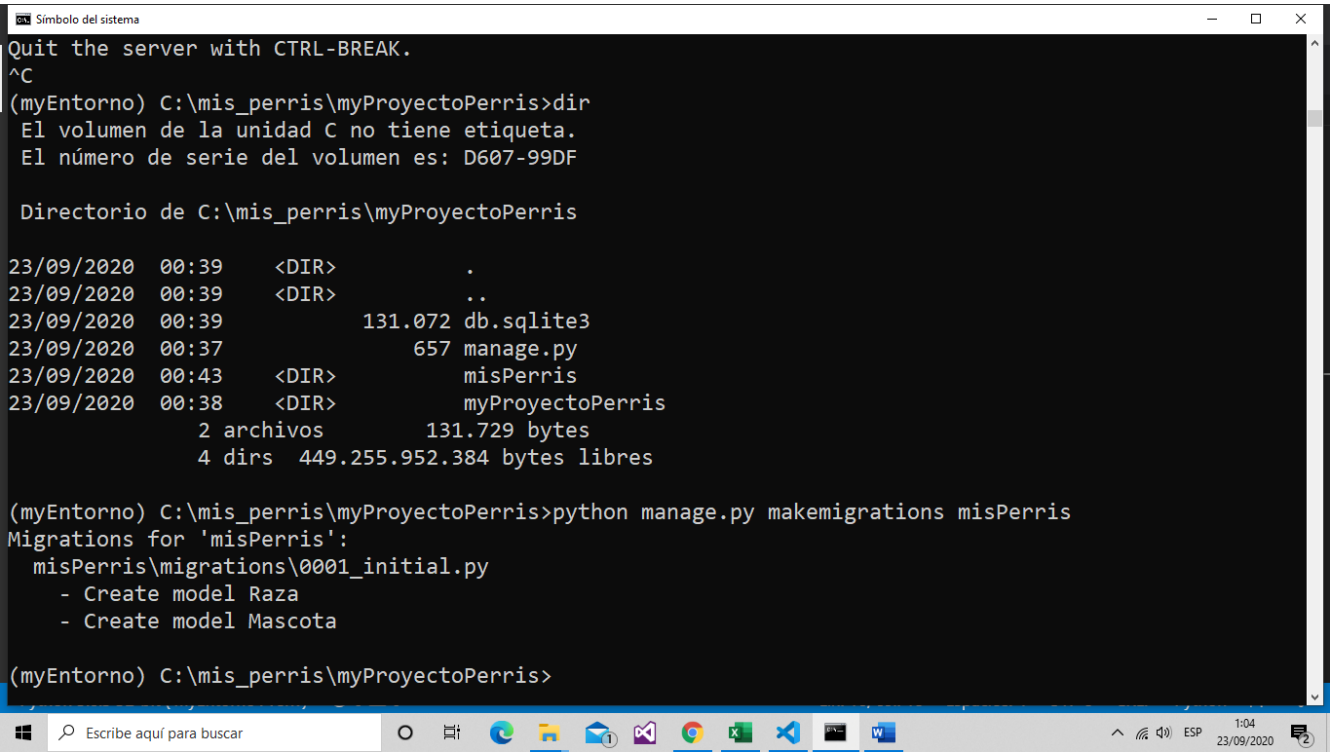
Aquí el diseño de dos tablas vinculadas



```
1 from django.db import models
2
3 # Create your models here.
4 class Raza(models.Model):
5     name=models.CharField(max_length=100, primary_key=True)
6     annos=models.IntegerField()
7
8     def __str__(self):
9         return self.name
10
11 class Mascota(models.Model):
12     name=models.CharField(max_length=100,primary_key=True)
13     edad=models.IntegerField()
14     descripcion=models.TextField()
15     raza=models.ForeignKey(Raza,on_delete=models.CASCADE)
16
17     def __str__(self):
18         return self.name
```

Luego de generarlas por medio de código debemos generar las tablas con la instrucción

PYTHON MANAGE.PY MAKEMIGRATIONS <NOMBRE DE LA APLIACION>



```
Símbolo del sistema
Quit the server with CTRL-BREAK.
^C
(myEntorno) C:\mis_perris\myProyectoPerris>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: D607-99DF

Directorio de C:\mis_perris\myProyectoPerris

23/09/2020  00:39    <DIR>        .
23/09/2020  00:39    <DIR>        ..
23/09/2020  00:39                131.072 db.sqlite3
23/09/2020  00:37                657 manage.py
23/09/2020  00:43    <DIR>        misPerris
23/09/2020  00:38    <DIR>        myProyectoPerris
                2 archivos        131.729 bytes
                4 dirs  449.255.952.384 bytes libres

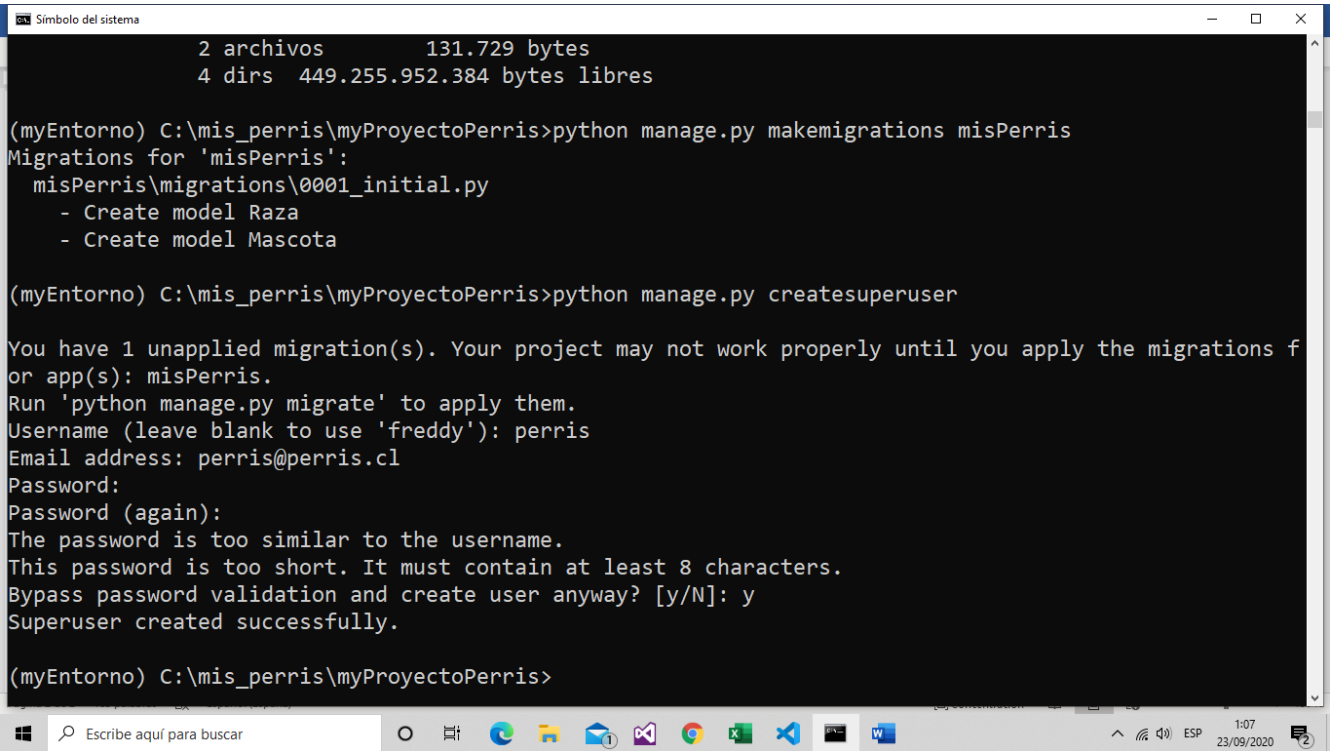
(myEntorno) C:\mis_perris\myProyectoPerris>python manage.py makemigrations misPerris
Migrations for 'misPerris':
  misPerris\migrations\0001_initial.py
    - Create model Raza
    - Create model Mascota

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

De esta forma están incorporadas como 2 nuevos modelos creados

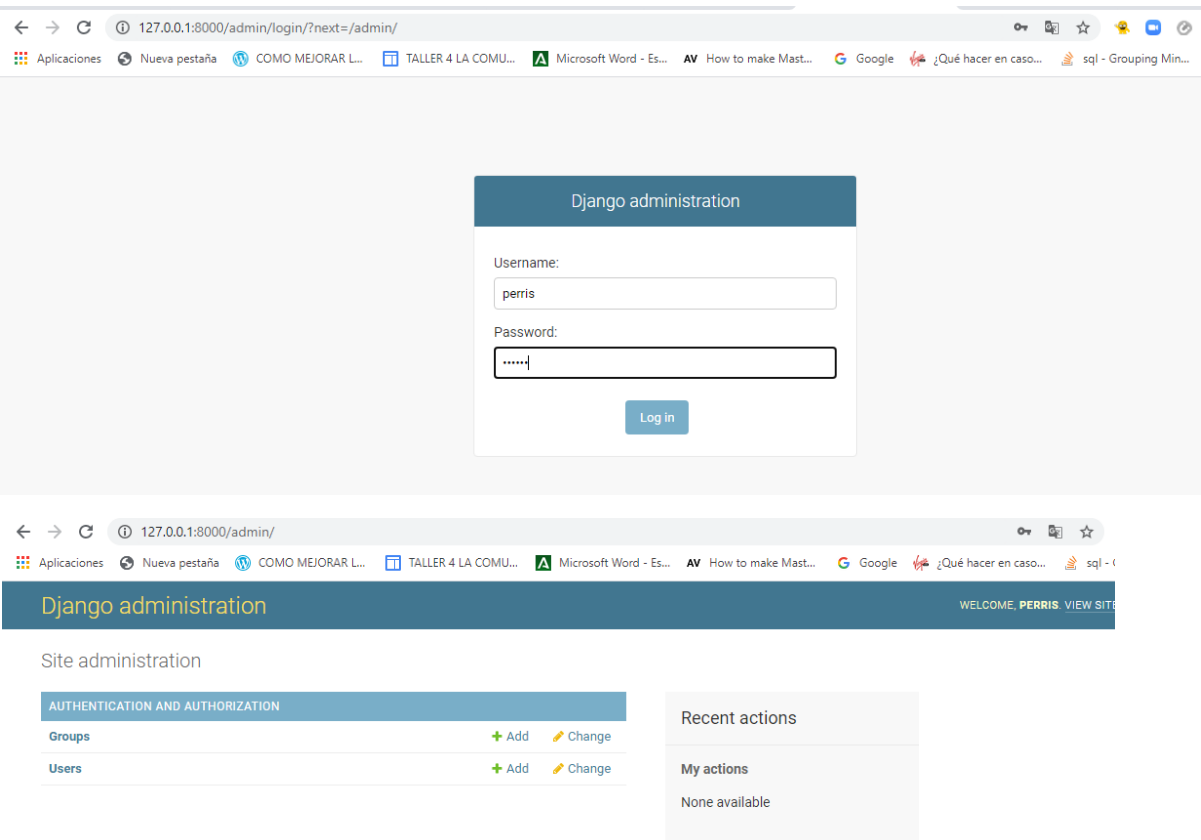
Luego para poder administrarlas debemos crear un super usuario por medio de la instrucción

PYTHON MANAGE.PY CREATESUPERUSER

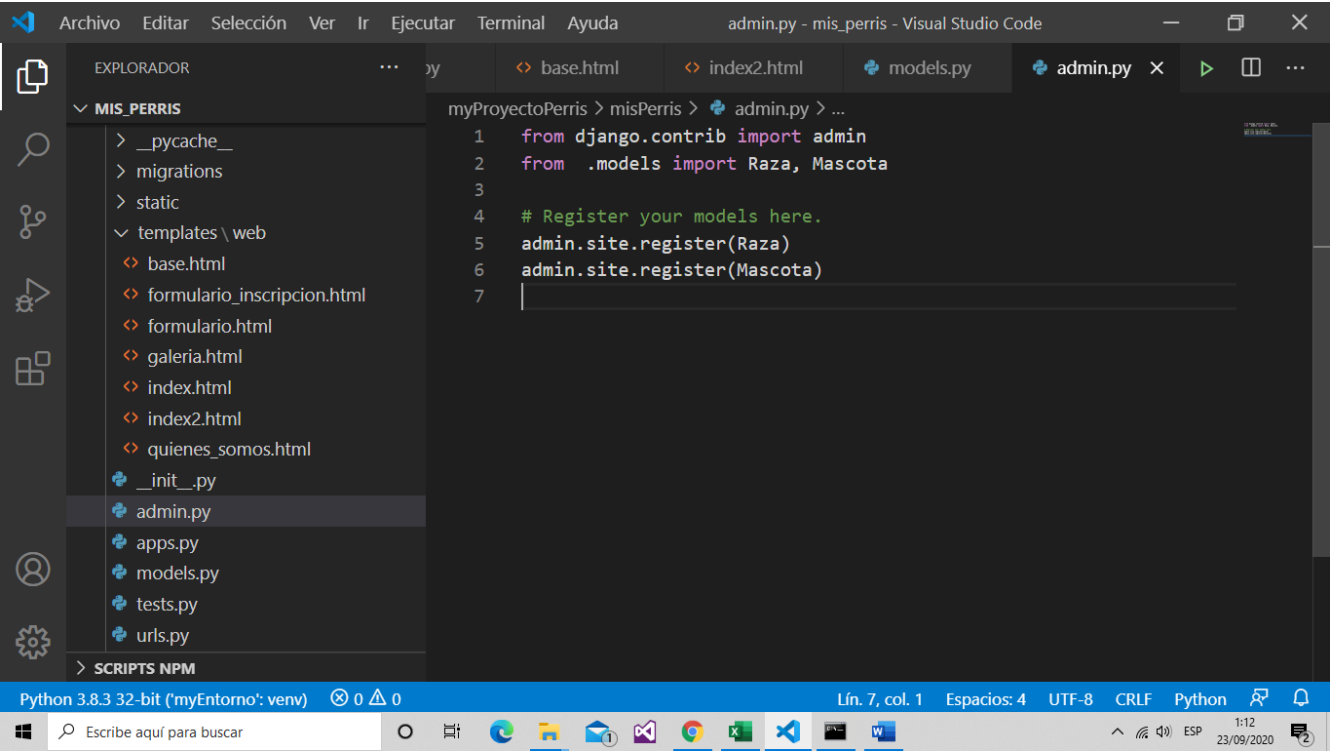


Ahora debemos arrancar el servidor y probar este usuario

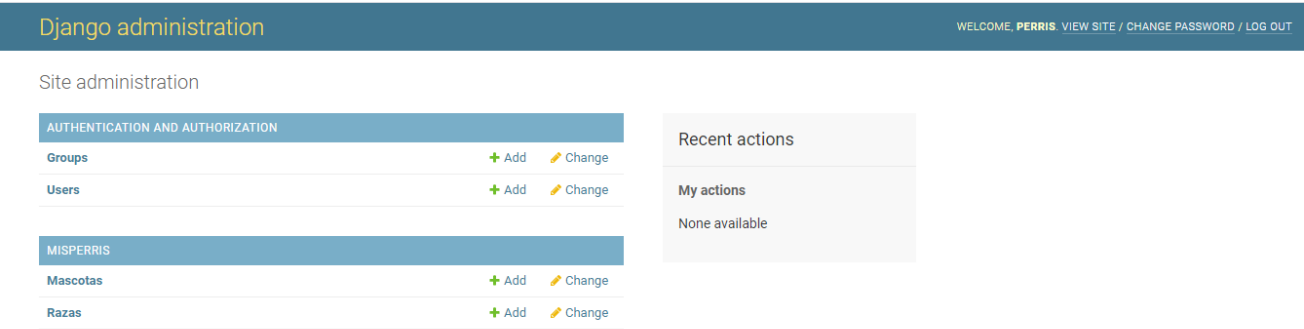
PYTHON MANAGE.PY RUNSERVER



Para que aparezcan las tablas debemos incorporarlas en el archivo **ADMIN.PY**

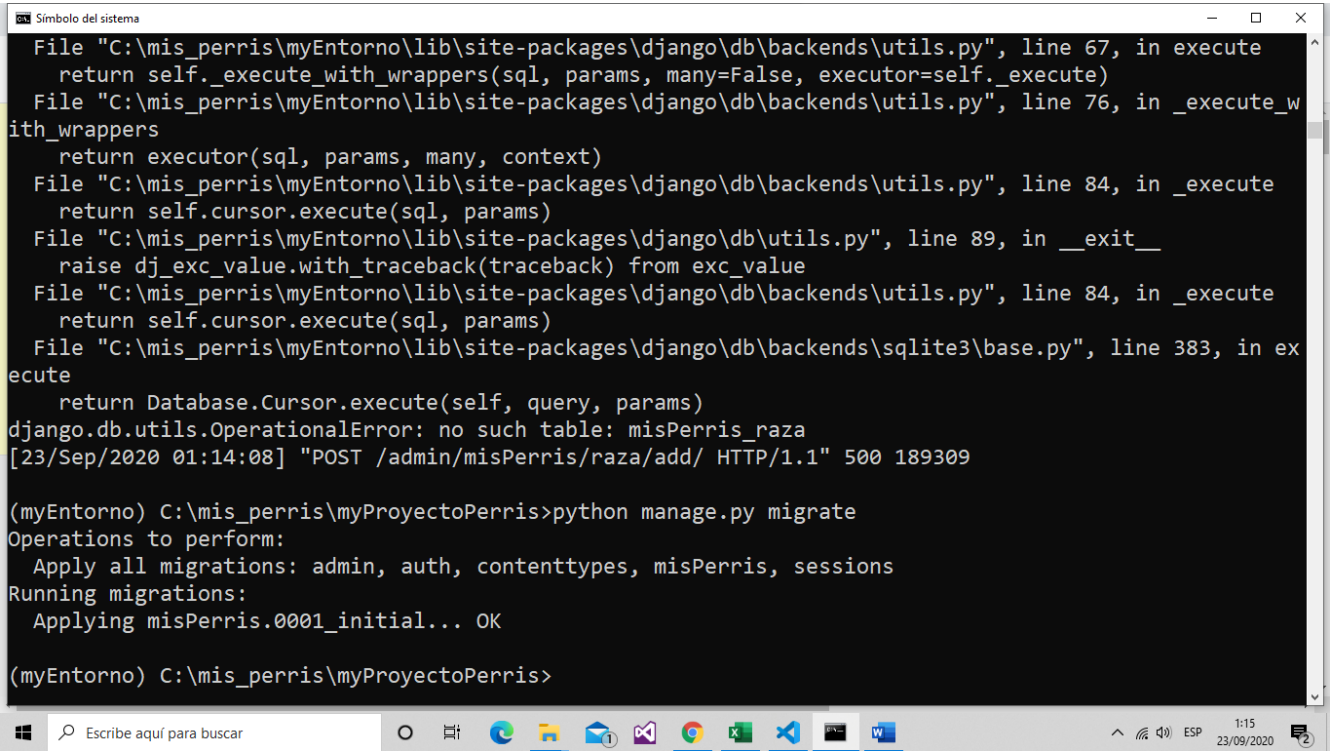


Ahora puede actualizar el **Admin**



Al intentar ingresar datos tendremos un problema debido a que no hemos migrado estas nuevas tablas, para ello debe ingresar

PYTHON MANAGE.PY MIGRATE



Podemos crear un formulario que permita el ingreso de estos datos

Creamos primero el formulario

Crea una Cuenta

ingrese nombres

ingrese descripcion

registrar

¿Ya tienes una cuenta? [Ingresa Aqui](#)

Luego pasamos datos desde el modelo

```
views.py
1 from django.shortcuts import render
2 from .models import Raza
3 # Create your views here.
4 def index(request):
5     return render(request, 'web/index2.html')
6
7 def formulario(request):
8     raza=Raza.objects.all() # recupera todas las razas
9     return render(request, 'web/formulario.html',{'razas':raza})# se pasan a la pagina web
```

Se incorpora el código a la pagina web

```
formulario.html
10 </head>
11
12 <body>
13     <h1>Formulario de Registro</h1>
14     <form action="registrar.php" method="post" class="form-register">
15         <h2 class="form-titulo">Crea una Cuenta</h2>
16         <div class="contenedor-inputs">
17             <input type="text" name="nombre" placeholder="ingrese nombres" class="input-48" required>
18             <input type="number" name="Edad" class="input-48" required>
19             <input type="text" name="descripcion" placeholder="ingrese descripcion" class="input-48">
20             <select class="input-100" name="raza">
21                 {% for raz in razas %}
22                 <option>{{raz.name}}</option>
23                 {% endfor %}
24             </select>
25             <input type="submit" value="registrar" class="btn-enviar">
26             <p class="form-link">¿Ya tienes una cuenta? <a href="#">Ingresa Aqui</a></p>
27         </div>
28     </form>
29 </body>
```

El formulario debemos modificarlo si deseamos almacenar datos , quedara así

```
myProyectoPerris > misPerris > templates > web > formulario.html
12 <body>
13 <h1>Formulario de Registro</h1>
14
15 <form action="" method="post" class="form-register">
16     {% csrf_token %}
17     <h2 class="form-titulo">Crea una Cuenta</h2>
18     <div class="contenedor-inputs">
19         <input type="text" name="nombre" placeholder="ingrese nombres" class="input-48" requi
20         <input type="number" name="Edad" class="input-48" required>
21         <input type="text" name="descripcion" placeholder="ingrese descripcion" class="input-
22         <select class="input-100" name="raza">
23             {% for raz in razas %}
24                 <option>{{raz.name}}</option>
25             {% endfor %}
26         </select>
27     </div>
28     <input type="submit" value="Registrar" name="accion" class="btn-enviar">
29     <input type="submit" value="Elimimnar" name="accion" class="btn-enviar">
30     <p class="form-link">¿Ya tienes una cuenta? <a href="#">Ingresa Aqui</a></p>
31 </div>
32 </form>
```

Y la definición en la vista del formulario quedara así

```
myProyectoPerris > misPerris > views.py > formulario
1 from django.shortcuts import render
2 from .models import Raza,Mascota
3 # Create your views here.
4 def index(request):
5     return render(request,'web/index2.html')
6
```

```
7 def formulario(request):
8     raza=Raza.objects.all() # recupera todas las razas
9     if request.POST:
10         accion=request.POST.get("accion")
11         if accion=='Registrar':
12             nombre=request.POST.get("nombre")
13             edad=request.POST.get("Edad")
14             desc=request.POST.get("descripcion")
15             raza=request.POST.get("raza")
16             obj_raza=Raza.objects.get(name=raza)
17             mascota=Mascota(
18                 name=nombre,
19                 edad=edad,
20                 descripcion=desc,
21                 raza=obj_raza
22             )
23             mascota.save()
24             return render(request,'web/formulario.html',{'razas':raza,'mensaje':'grabada'})
25     return render(request,'web/formulario.html',{'razas':raza})# se pasan a la pagina web
```