

Como empezar un proyecto

1. Creamos nuestro proyecto y el blog

- django-admin startproject FutbolJokalariak
- cd FutbolJokalariak
- python manage.py startapp Equipo

2. Arrancamos el servidor de desarrollo

- python manage.py runserver

Si da algun error por migrar(python manage.py migrate)

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
September 25, 2023 - 10:36:33
Django version 4.2.5, using settings 'FutbolJokalariak.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

PS C:\Users\parra.raul\Documents\GitHub\WebAplikazioakGarapena-Lander\FutbolJokalariak> python manage.py migrate
```

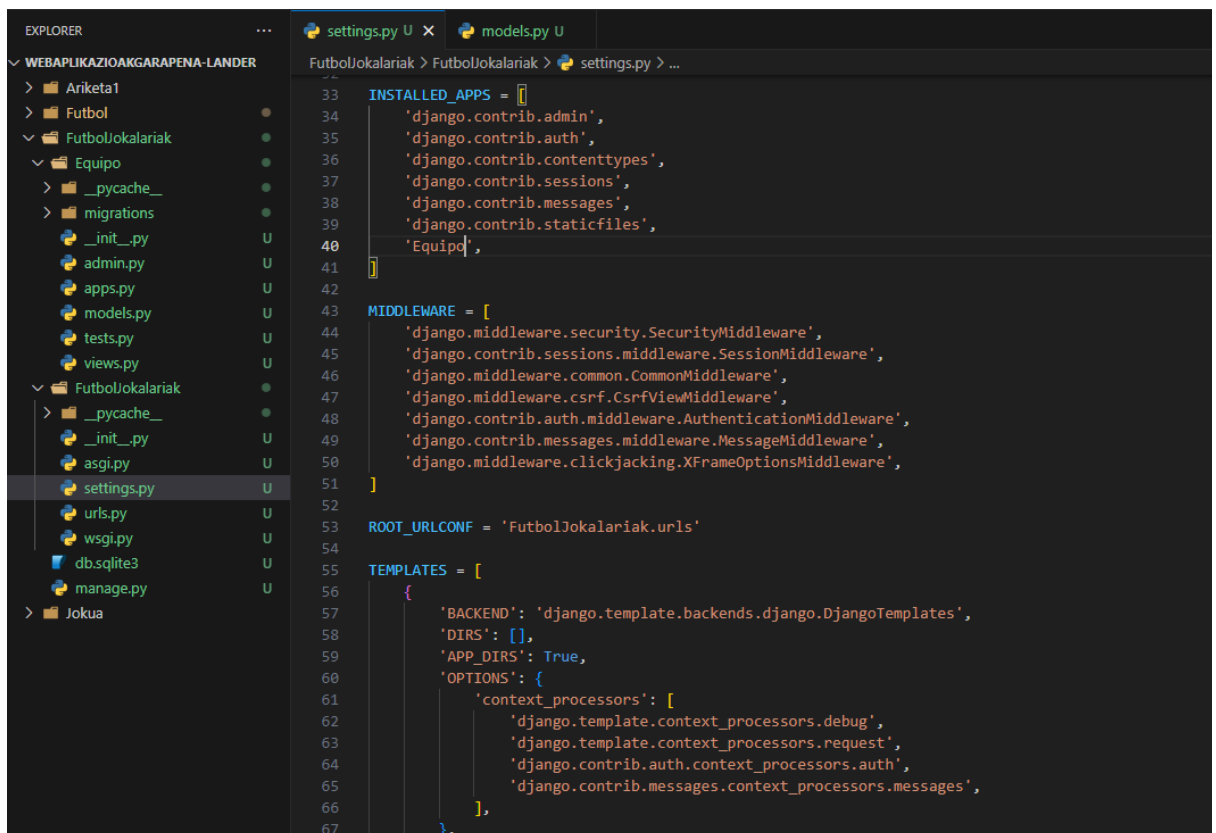
y entramos en el localhost(<http://localhost:8000/>)

3. Añadimos en settings.py del proyecto FutbolJokalariak,dentro de

Installed_apps=[....

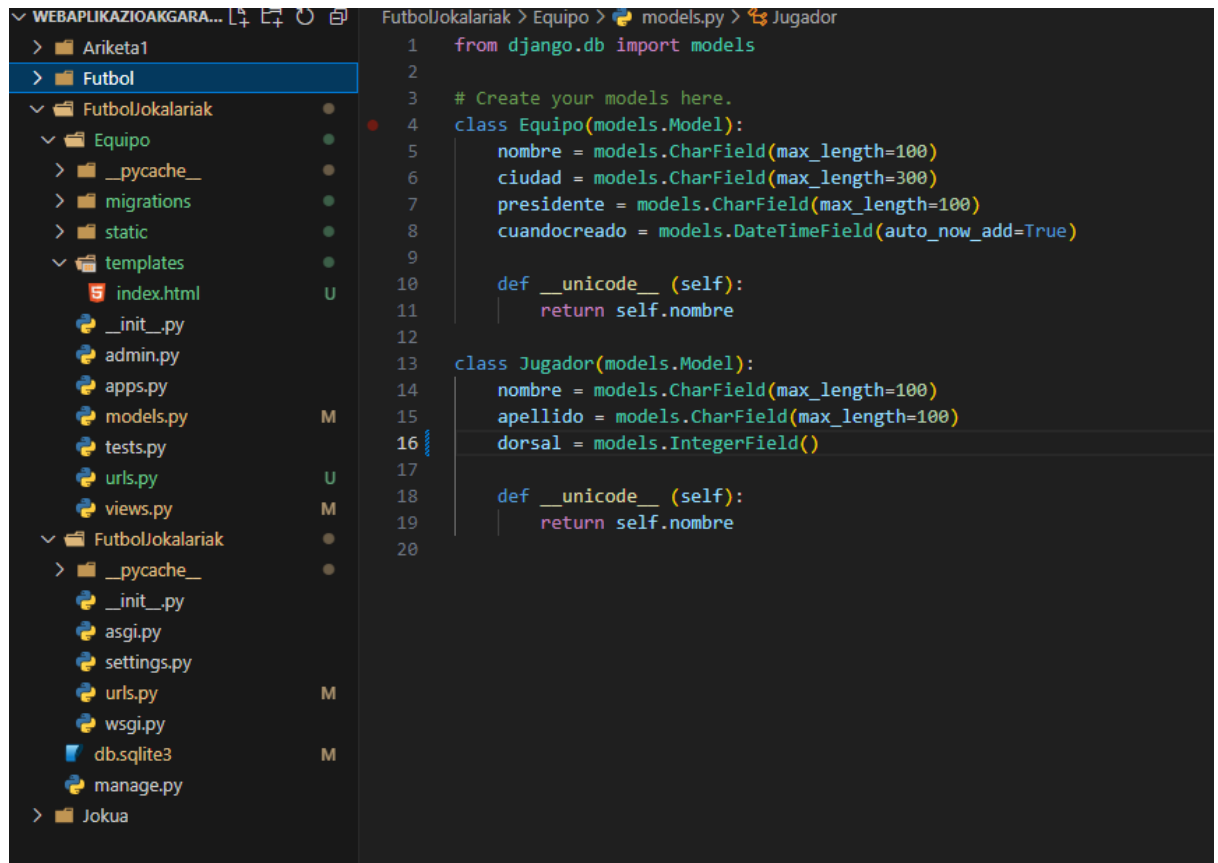
'Equipo',

]



```
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'Equipo',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
52
53 ROOT_URLCONF = 'FutbolJokalariak.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': [],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.auth',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     ],
69 ]
```

4. En el models.py dentro de equipo añadimos la clase Post, con los datos del equipo.(también añadimos los datos de Jugador)

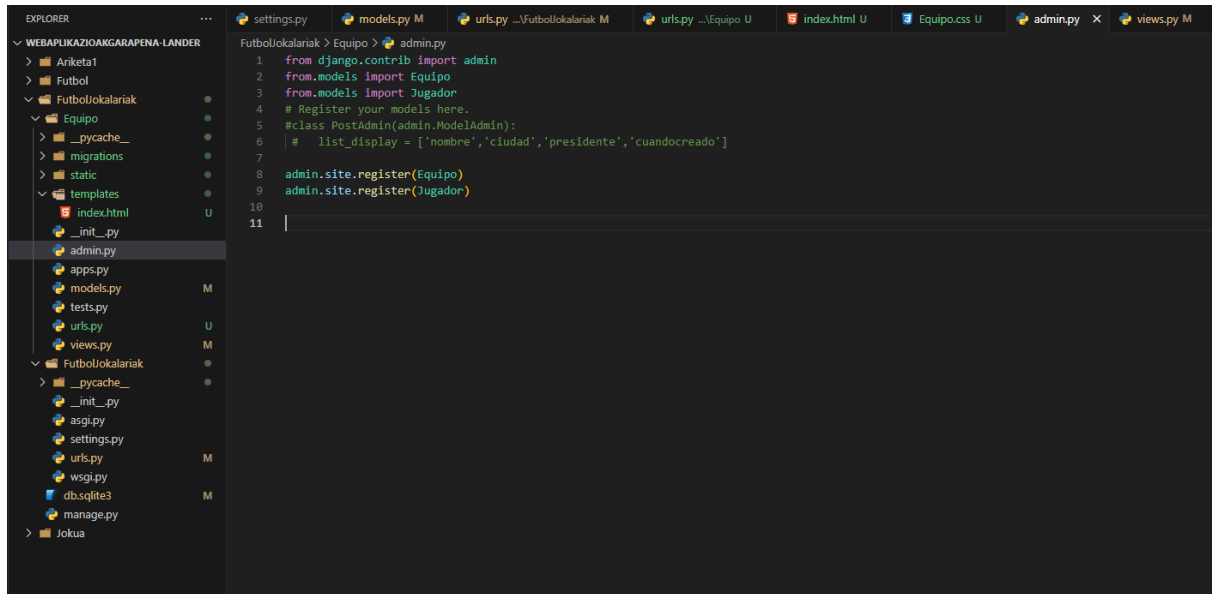


```
1 from django.db import models
2
3 # Create your models here.
4 class Equipo(models.Model):
5     nombre = models.CharField(max_length=100)
6     ciudad = models.CharField(max_length=300)
7     presidente = models.CharField(max_length=100)
8     cuandocreado = models.DateTimeField(auto_now_add=True)
9
10     def __unicode__(self):
11         return self.nombre
12
13 class Jugador(models.Model):
14     nombre = models.CharField(max_length=100)
15     apellido = models.CharField(max_length=100)
16     dorsal = models.IntegerField()
17
18     def __unicode__(self):
19         return self.nombre
20
```

5. Creamos la base de datos inicial con el makemigrations y despues añadimos los campos de la tabla de datos con el migrate.

```
PS C:\Users\parra.raul\Documents\GitHub\WebApplikazioakGarapena-Lander\FutbolJokalaria> python manage.py makemigrations
Migrations for 'blog':
  blog/migrations/0001_initial.py
    - Create model Post
PS C:\Users\parra.raul\Documents\GitHub\WebApplikazioakGarapena-Lander\FutbolJokalaria> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, contenttypes, sessions
Running migrations:
  Applying blog.0001_initial... OK
PS C:\Users\parra.raul\Documents\GitHub\WebApplikazioakGarapena-Lander\FutbolJokalaria>
```

6. Añadimos al admin.py lo que queremos meter en el post



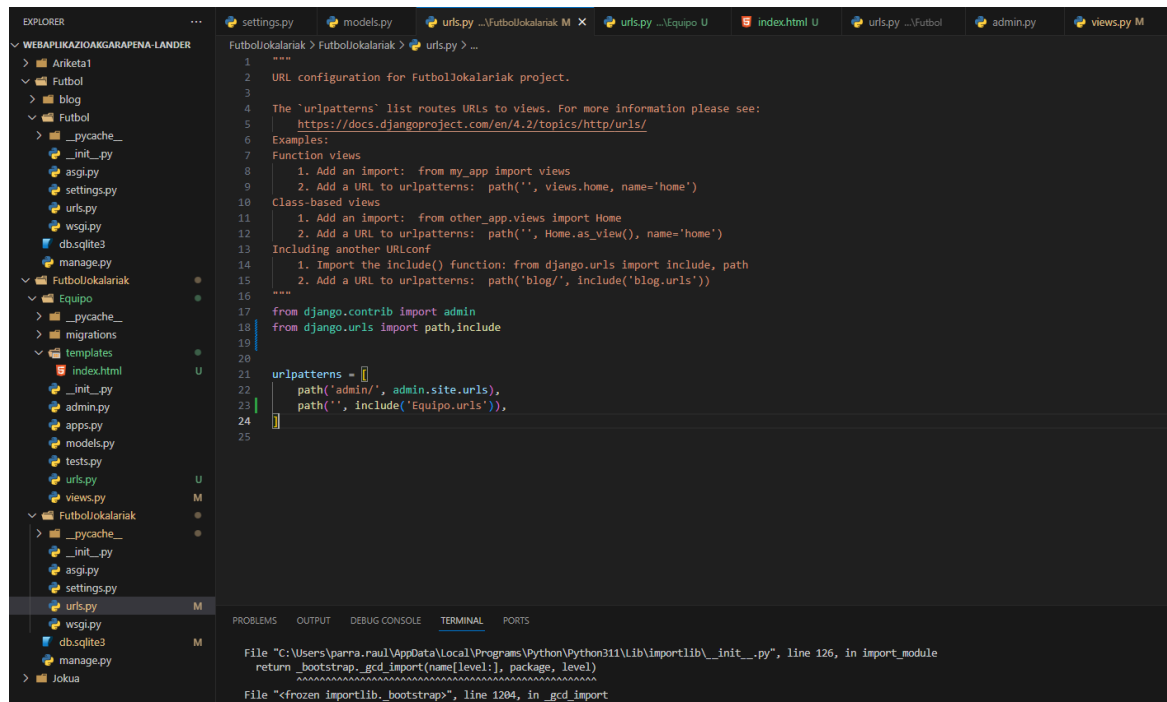
7. Creamos el superuser para ver lo que tenemos en la base de datos

```
PS C:\Users\parra.raul\Documents\GitHub\WebAplikazioakGarapena-Lander\FutbolJokalaria> python manage.py createsuperuser
Username (leave blank to use 'parra.raul'): raul
Email address: raul
Error: Enter a valid email address.
Email address: parra.raul@uni.eus
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS C:\Users\parra.raul\Documents\GitHub\WebAplikazioakGarapena-Lander\FutbolJokalaria>
```

8. Para añadir registros directamente desde la base de datos

<http://localhost:8000/admin>

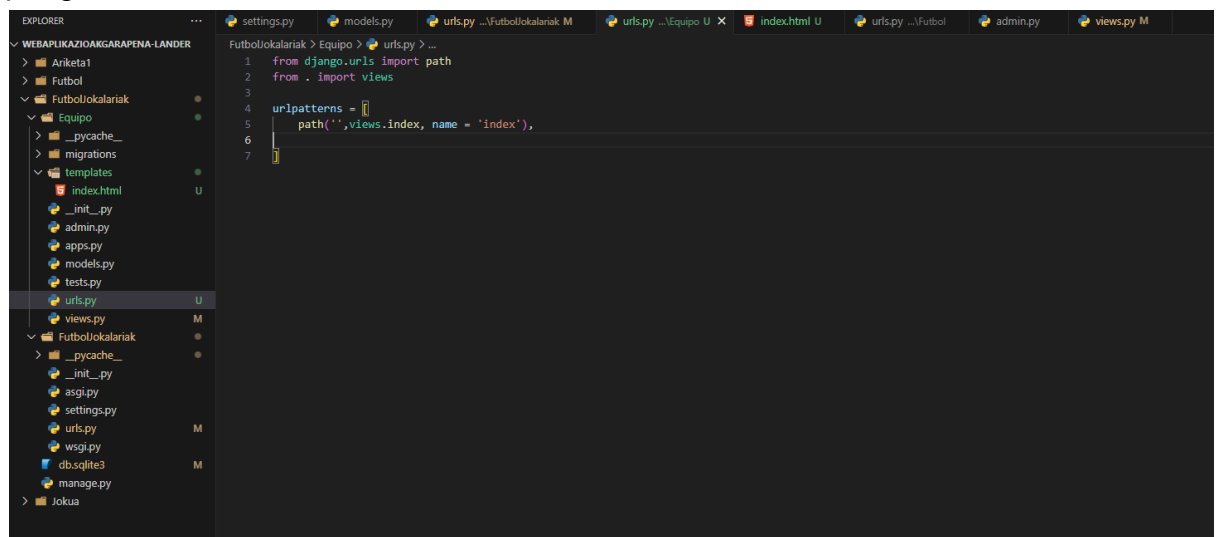
9. Modificamos las urls del proyecto general



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left and the Code editor on the right. The Explorer sidebar shows a project structure with folders like 'Ariketa1', 'Futbol', and 'Futbolokalarik'. The Code editor displays the 'urls.py' file for the 'Futbolokalarik' project. The file contains comments and code for URL configuration. The 'urlpatterns' list is modified to include the admin site and the 'Equipo' app's URLs.

```
1  """
2  URL configuration for Futbolokalarik project.
3
4  The 'urlpatterns' list routes URLs to views. For more information please see:
5      https://docs.djangoproject.com/en/4.2/topics/http/urls/
6  Examples:
7  Function views
8      1. Add an import: from my_app import views
9      2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('Equipo.urls')),
23 ]
```

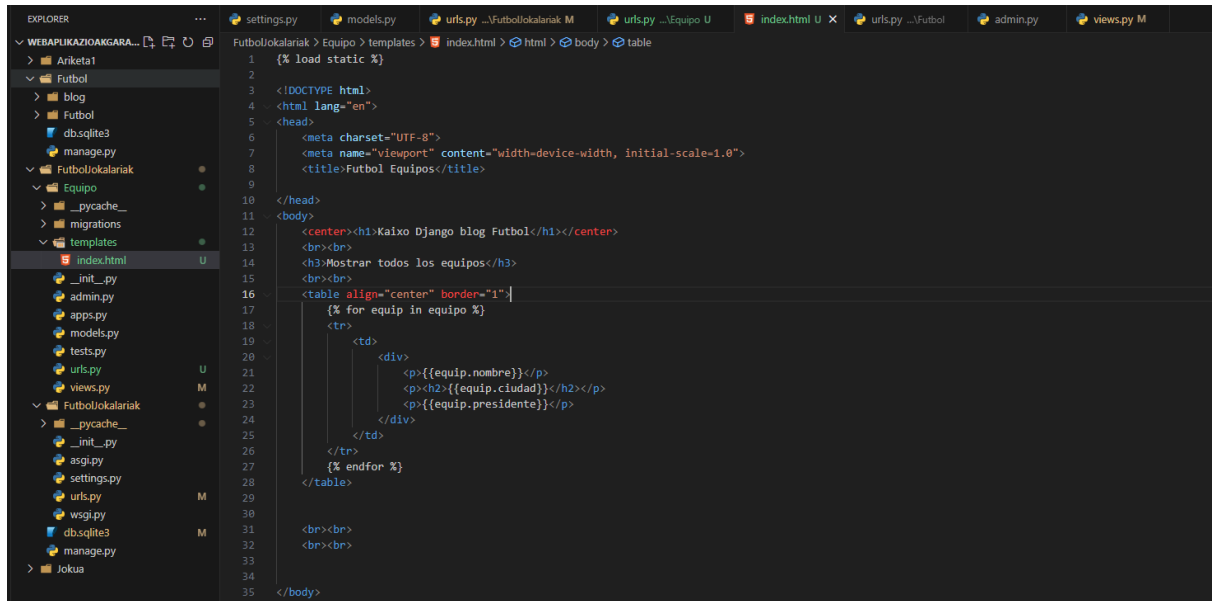
10. Y modificamos las urls de la aplicación para que vaya al index cuando no ponga nada



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left and the Code editor on the right. The Explorer sidebar shows the project structure with the 'Equipo' folder selected. The Code editor displays the 'urls.py' file for the 'Equipo' app. The file contains code for URL configuration. The 'urlpatterns' list is modified to include the 'index' view when no path is specified.

```
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5     path('', views.index, name = 'index'),
6 ]
```

11. Después creamos una carpeta templates para meter el index.html y mostrar los datos(Podemos añadir también el css si queremos).

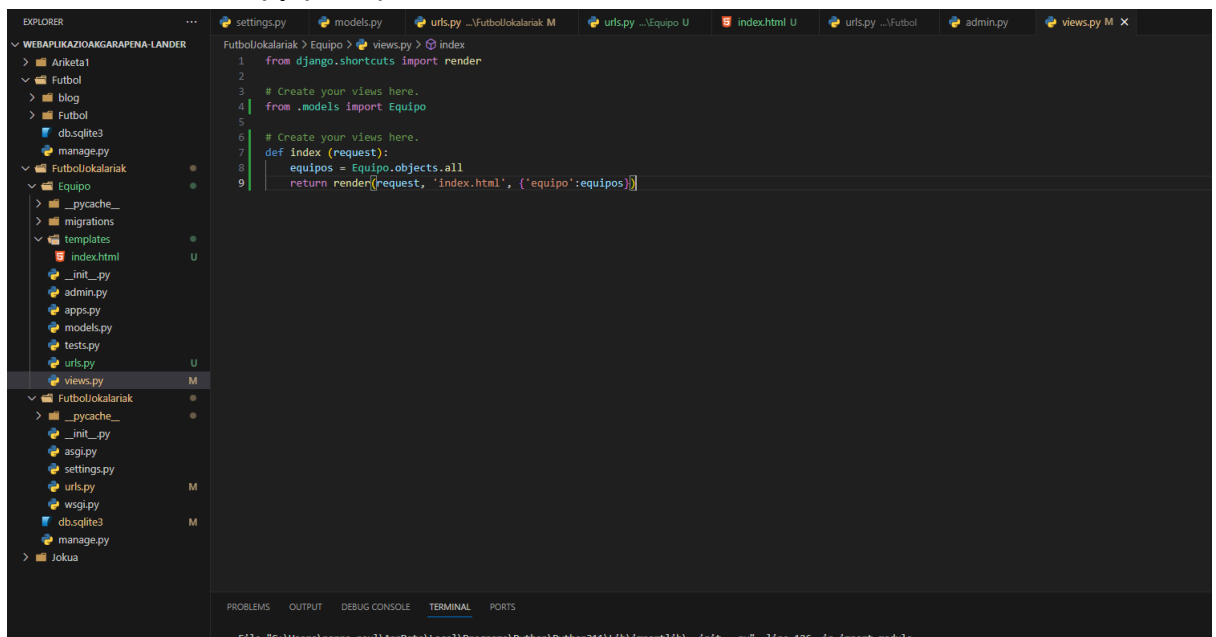


```
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6   <meta charset="UTF-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Futbol Equipos</title>
9
10 </head>
11 <body>
12   <center><h1>Kaixo Django blog Futbol</h1></center>
13   <br><br>
14   <h3>Mostrar todos los equipos</h3>
15   <br><br>
16   <table align="center" border="1">
17     {% for equip in equipo %}
18     <tr>
19       <td>
20         <div>
21           <p>{{equip.nombre}}</p>
22           <p><h2>{{equip.ciudad}}</h2></p>
23           <p>{{equip.presidente}}</p>
24         </div>
25       </td>
26     </tr>
27     {% endfor %}
28   </table>
29
30   <br><br>
31   <br><br>
32
33
34
35 </body>
```

poniendo esto debajo del <title>

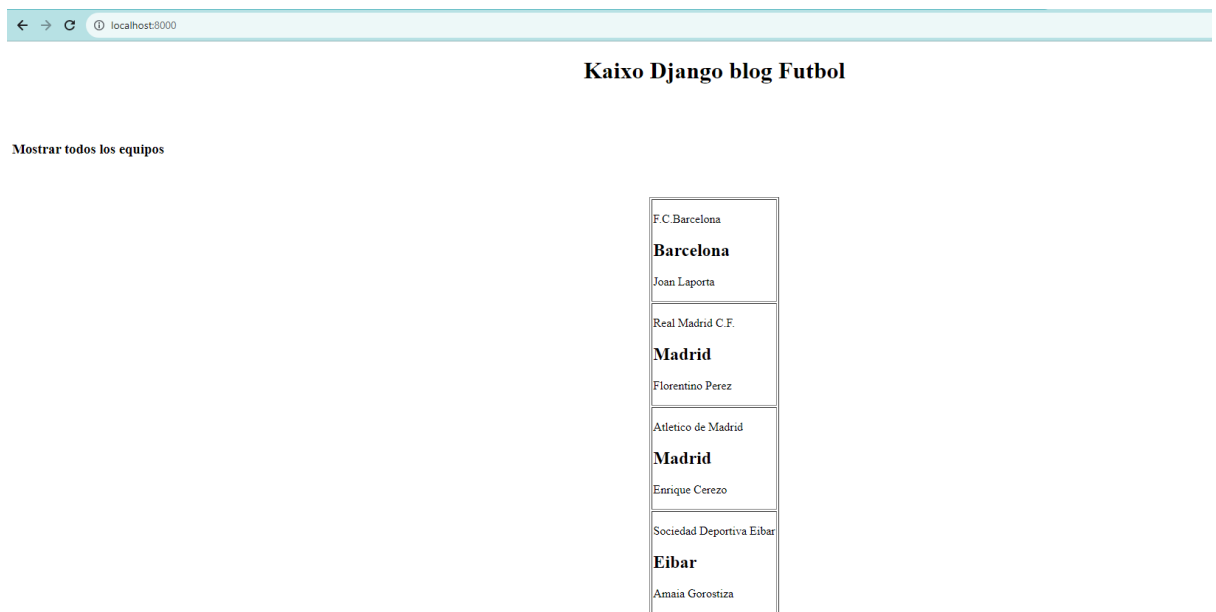
```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Futbol Equipos</title>
  <link rel="stylesheet" href='{% static "css/Equipo.css" %}'>
</head>
```

12. Y editamos views.py para poder mostrarlos

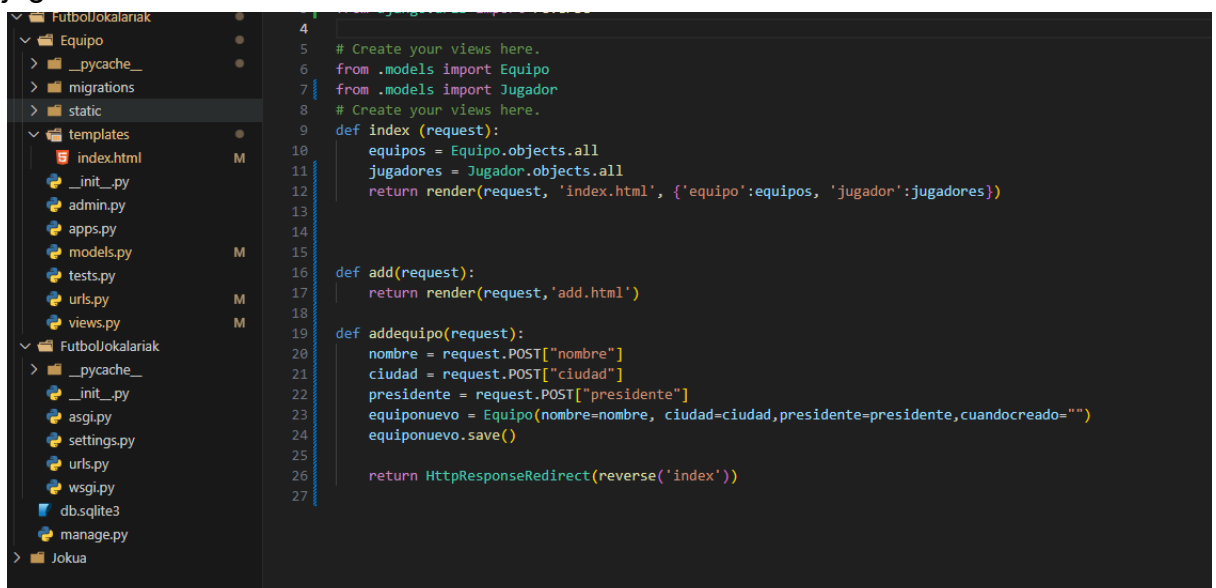


```
1 from django.shortcuts import render
2
3 # Create your views here.
4 from .models import Equipo
5
6 # Create your views here.
7 def index(request):
8     equipos = Equipo.objects.all()
9     return render(request, 'index.html', {'equipo': equipos})
```

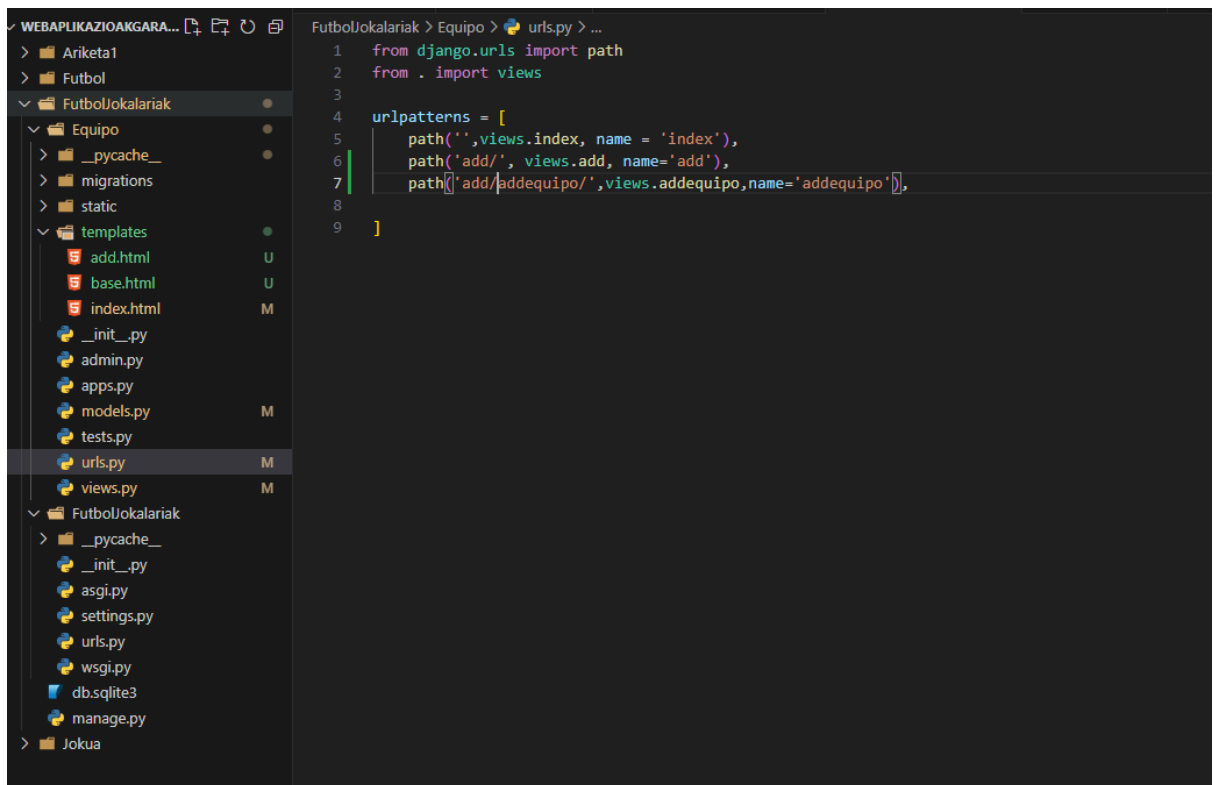
13. Ahora al buscar en localhost debería salir la pagina con los datos de la DB



14. Creamos en la clase views los metodos para añadir nuevo equipos y jugadores

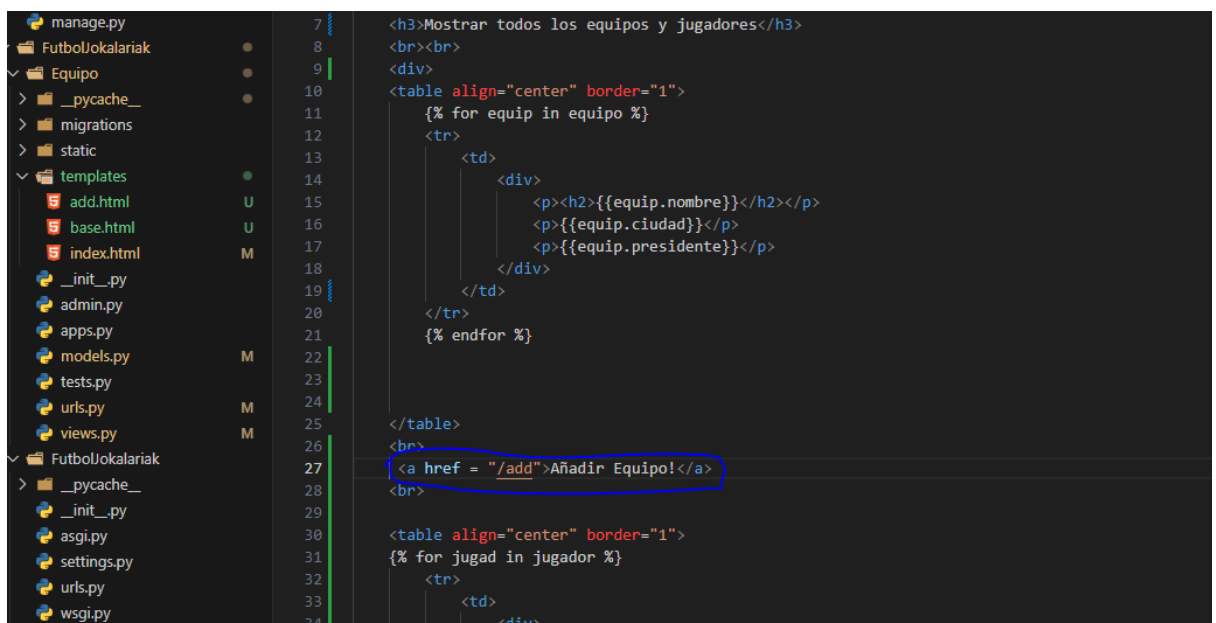


15. Y en los urls de la aplicación añadimos los links



```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name = 'index'),
6     path('add/', views.add, name='add'),
7     path('add/addequipo/', views.addequipo, name='addequipo'),
8 ]
```

16. Añadimos la url al index con el href



```
7 <h3>Mostrar todos los equipos y jugadores</h3>
8 <br><br>
9 <div>
10 <table align="center" border="1">
11     {% for equip in equipo %}
12     <tr>
13         <td>
14             <div>
15                 <p><h2>{{equip.nombre}}</h2></p>
16                 <p>{{equip.ciudad}}</p>
17                 <p>{{equip.presidente}}</p>
18             </div>
19         </td>
20     </tr>
21     {% endfor %}
22 </table>
23 <br>
24 <a href = "/add">Añadir Equipo!</a>
25 <br>
26 <table align="center" border="1">
27     {% for jugad in jugador %}
28     <tr>
29         <td>
30             <div>
```

17. Y creamos el add.html donde se van a mostrar los datos

```
{% extends 'base.html' %}

{% block content %}
<br><br>

<form action="addequipo/" method="POST">
    {% csrf_token %}

    Nombre del equipo: <input type="text" name="nombre">
    Nombre de la ciudad: <input type="text" name="ciudad">
    Nombre del presidente: <input type="text"
name="presidente">
    <button type = "submit" value="Submit">Crear
equipo</button>
</form>
{% endblock content %}
```

18. Para hacer el delete tenemos que crear en la clase views.py el metodo delete

```
def deleteequipo(request, id):
    equipoeliminar = Equipo.objects.get(id = id) #pasar el equipo
    Equipo.delete(equipoeliminar) #ezabatu
```

19. Seguido añadimos el método en las urls

```
urlpatterns = [
    path(' ', views.index, name = 'index'),
    path('add/', views.add, name='add'),
    path('add/addequipo/', views.addequipo, name='addequipo'),
    path('add/addjugador/', views.addjugador, name='addjugador'),
    #path('ezabatu/', views.deleteequipo, name='deleteequipo'),
    path('ezabatu/<int:id>/', views.deleteequipo,
name='deleteequipo'),
]
```

20. Y metemos el href en el index.html

```
<center><h1>Kaixo Django blog Futbol</h1></center>
```



```

<br><br>
<h3 style="text-align: center;">Mostrar todos los equipos y
jugadores</h3>
<br><br>
<div style="float: left;">
<table align="center" border="1">
    {% for equip in equipo %}
    <tr>
        <td>
            <div>
                <p><h2>{{equip.nombre}}</h2></p>
                <p>{{equip.ciudad}}</p>
                <p>{{equip.presidente}}</p>
            </div>
        </td>
        <td>
            <a href="ezabatu/{{equip.id}}"
color="red">Eliminar</a>
        </td>
    </tr>
    {% endfor %}

</table>

```

21. Ahora hacemos lo mismo para el jugador

22. Despues de tener creado el Add y eliminar vamos a hacer el editar, para eso creamos 2 post en la clase views.py, uno para que vaya al html y otro para el formulario

```

def updateequipo(request, id):
    equipoUpdate = Equipo.objects.get(id = id)
    return render(request, 'update.html', {'equip':equipoUpdate})

def editarequipo(request, id):

    equipoeditar = Equipo.objects.get(id = id) #pasar el id del
equipo

    nombre = request.POST["nombre"]
    ciudad = request.POST["ciudad"]
    presidente = request.POST["presidente"]

```

```

equipoeditar.nombre = nombre
equipoeditar.ciudad = ciudad
equipoeditar.presidente = presidente
equipoeditar.save()

return HttpResponseRedirect(reverse(index))

```

23. Y lo añadimos a las urls

```

urlpatterns = [
    path('', views.index, name = 'index'),
    path('add/', views.add, name='add'),
    path('add/addequipo/', views.addequipo, name='addequipo'),
    path('add/addjugador/', views.addjugador, name='addjugador'),
    #path('eliminar/', views.deleteequipo, name='deleteequipo'),
    path('eliminar equipo/<int:id>/', views.deleteequipo,
name='deleteequipo'),
    path('eliminarjugador/<int:id>/', views.deletejugador,
name='deletejugador'),
    path('updateequipo/<int:id>/', views.updateequipo,
name='updateequipo'),
    path('editarequipo/<int:id>/', views.editarequipo,
name='editarequipo'),
]

```

24. Añadimos el href al index

```

<center><h1>Kaixo Django blog Futbol</h1></center>
<br><br>
<h3 style="text-align: center;">Mostrar todos los equipos y
jugadores</h3>
<br><br>
<div style="float: left;">
<table align="center" border="1">
    {% for equip in equipo %}
    <tr>
        <td>
            <div>
                <p><h2>{{equip.nombre}}</h2></p>
                <p>{{equip.ciudad}}</p>
                <p>{{equip.presidente}}</p>
            </div>

```

```

        </td>
        <td>
            <a href="eliminar equipo/{{equip.id}}"
color="red">Eliminar</a>
        </td>
        <td>
            <a href="updateequipo/{{equip.id}}"
color="red">Editar</a>
        </td>
    </tr>
    {% endfor %}

</table>

```

25. Finalmente creamos un html que contenga un formulario para meter los datos a editar Update.html

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel = "stylesheet" href='{%static "css/blog.css" %}'>
    <title>Nuevos Datos</title>
</head>
<body>
    <center><h1>Mete los nuevos datos</h1></center>
    <br>

    <form action="/editarequipo/{{equip.id}}/" method="post">
        {% csrf_token %}
        Nombre del equipo: <input type="text" name="nombre"
value="{{equip.nombre}} ">
        Ciudad del equipo: <input type="text" name="ciudad"
value="{{equip.ciudad}} ">

```

```
    Presidente del equipo: <input type="text"
name="presidente" value="{{equip.presidente}}" ">
    <button type = submit>Editar Equipo</button>
</form>

</body>
</html>
```

26. Ahora hacemos lo mismo para el jugador