Pasos Django

1º Paso: Creación del proyecto

```
$ django-admin.py startproject <project-name>
<PROJECT_ROOT>
   manage.py
   <PROJECT_DIR>
      __init__.py
      settings.py
      urls.py
      wsgi.py
```

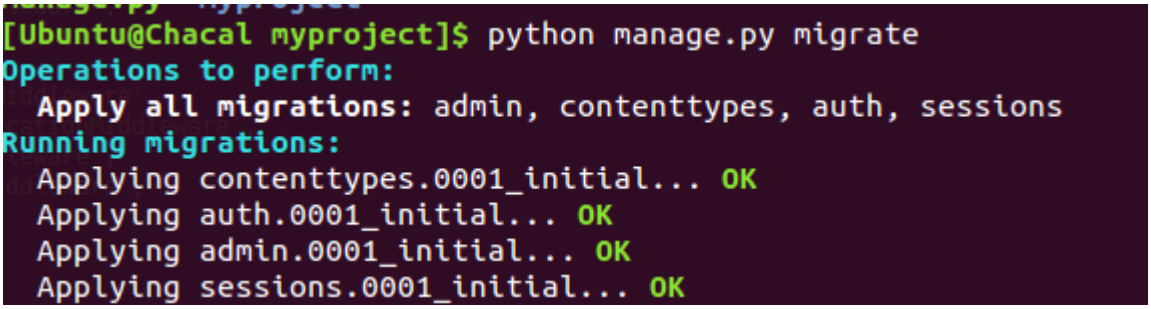
2º Paso: startapp

```
[Ubuntu@Chacal myproject]$ python manage.py startapp polls
buntu@Chacal myproject]$
```

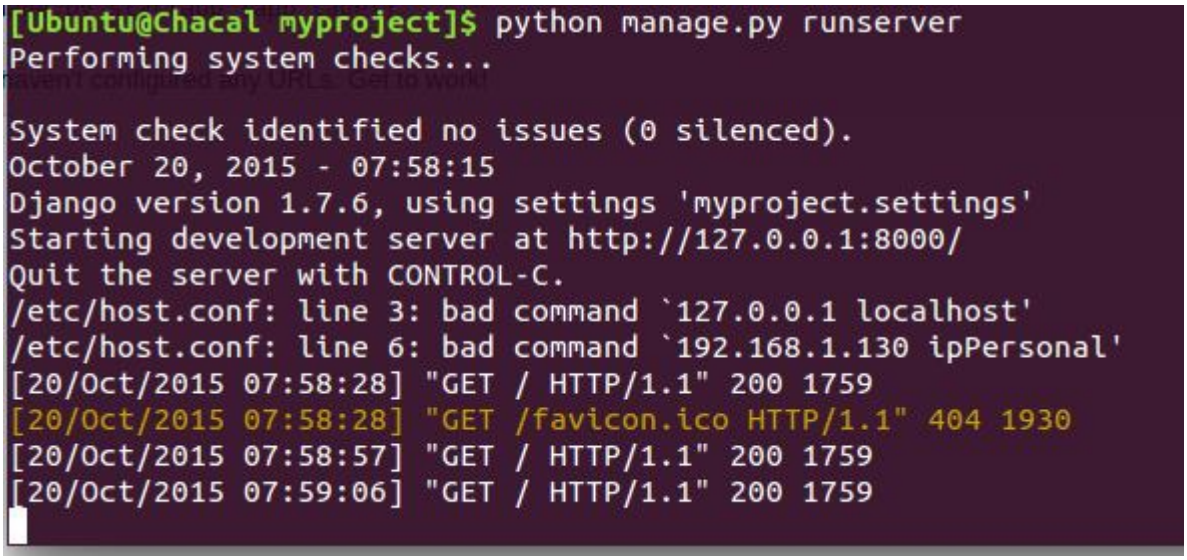
2.2º Paso: en archivo setting.py añadimos el nombre de la aplicación con la coma alfinal.

```
# Application definition

INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'noticias',
)
```

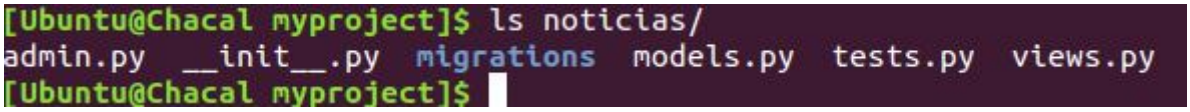
3º Paso: migrate



4º Paso: runserver



5º Paso: Ir a carpeta del proyecto y en models.py añadir las clases necesarias



6º Paso:python manage.py makemigrations noticias

```
[Ubuntu@Chacal myproject]$ python manage.py makemigrations noticias
Migrations for 'noticias':
  0001_initial.py:
    - Create model Autor
    - Create model Noticia
    - Create model Seccion
[Ubuntu@Chacal myproject]$ python manage.py sqlmigrate noticias 0001
BEGIN;
CREATE TABLE "noticias_autor" ("name" varchar(100) NOT NULL PRIMARY KEY);
CREATE TABLE "noticias_noticia" ("ID" varchar(20) NOT NULL PRIMARY KEY, "name" v
archar(100) NOT NULL);
CREATE TABLE "noticias_seccion" ("tipo" varchar(100) NOT NULL PRIMARY KEY);

COMMIT;
[Ubuntu@Chacal myproject]$
```

***Si en este paso nos ocurre algo como esto:

```
joseantonio@joseantonio ~/Documentos/PW/Prácticas/equipoFutbol $ python manage.py makemigrations
Migrations for 'gestion':
  0001_initial.py:
    - Create model Equipo
    - Create model Jugador
Traceback (most recent call last):
  File "manage.py", line 10, in <module>
    execute_from_command_line(sys.argv)
  File "/usr/local/lib/python2.7/dist-packages/django/core/management/__init__.py", line 385, in execute_from_command_line
    utility.execute()
  File "/usr/local/lib/python2.7/dist-packages/django/core/management/__init__.py", line 377, in execute
    self.fetch_command(subcommand).run_from_argv(self.argv)
  File "/usr/local/lib/python2.7/dist-packages/django/core/management/base.py", line 288, in run_from_argv
    self.execute(*args, **options.__dict__)
  File "/usr/local/lib/python2.7/dist-packages/django/core/management/base.py", line 338, in execute
    output = self.handle(*args, **options)
  File "/usr/local/lib/python2.7/dist-packages/django/core/management/commands/makemigrations.py", line 124, in handle
    self.write_migration_files(changes)
  File "/usr/local/lib/python2.7/dist-packages/django/core/management/commands/makemigrations.py", line 143, in write_migration_files
    migrations_directory = os.path.dirname(writer.path)
  File "/usr/local/lib/python2.7/dist-packages/django/db/migrations/writer.py", line 204, in path
    return os.path.join(basedir, self.filename)
  File "/usr/lib/python2.7/posixpath.py", line 80, in join
    path += '/' + b
UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 34: ordinal not in range(128)
```

debemos de introduccir en models.py, debajo de from django.db import models lo siguiente:

```
from django.db import models
import sys

reload(sys)
sys.setdefaultencoding('utf8')
```

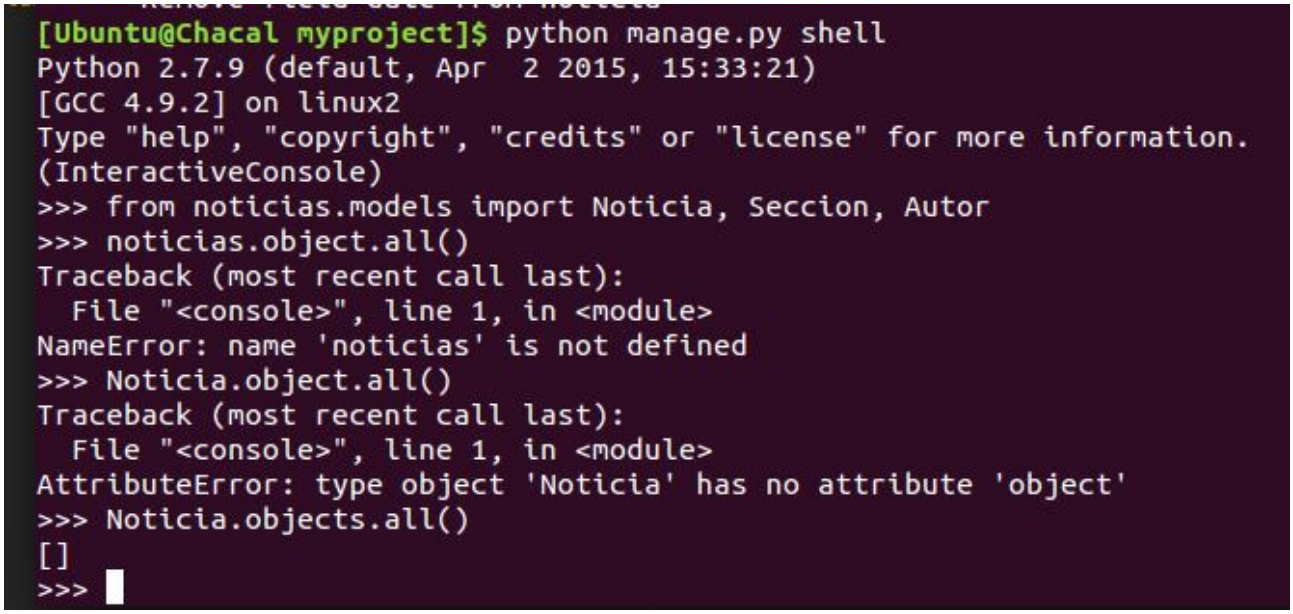
7º Paso: python manage.py migrate

[creando así la base de datos. Esto debemos de hacerlo siempre que hagamos un cambio en el fichero models.py]

Si hacemos cambios en models.py, debemos de borrar el archivo db.sqlite3 y dentro de la carpeta de la aplicación, debemos de borrar la carpeta migrations.

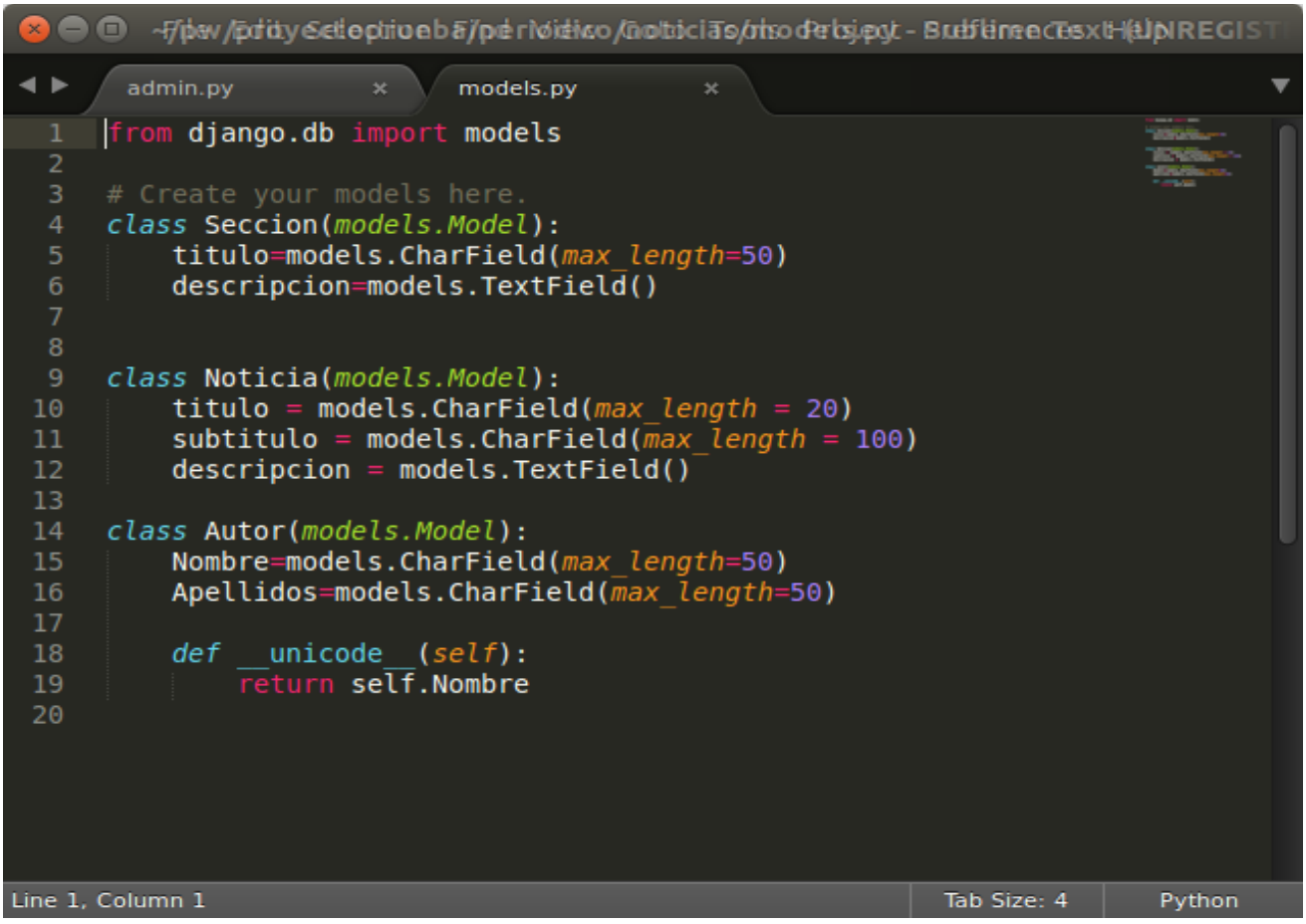
Ahora, volvemos hacer el paso 6 y 7.

8º Paso: Para entrar en el shell de python-> python manage.py shell



Pasos para trabajar con BBDD en Django:

9º Paso: debemos de editar el archivo models.py con las clases que deseemos tener.



10º Paso: from noticias.models import Noticia, Autor, Seccion

Con este paso podemos editar nuestra base de datos, añadiendo, filtrando o borrando cualquier dato de nuestra base de datos mediante python.

```
joseantonio@joseantonio ~/Documentos/PW/Teoria/periodico $ python manage.py shell
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from noticias.models import Noticia, Autor, Secciond
Traceback (most recent call last):
  File "<console>", line 1, in <module>
ImportError: cannot import name Secciond
>>> from noticias.models import Noticia, Autor, Seccion
>>> print a
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'a' is not defined
>>> a = Autor(name="Pepe")
>>> print a
Pepe
>>> a.save()
>>> Autor.objects.all()
[<Autor: Pepe Perez>, <Autor: Pepe>]
>>>
```

11º Paso:

RELACIONES 1-N y N-N

```python
from django.db import models


class Equipo(models.Model):
    name = models.CharField(max_length = 100, primary_key = True)
    liga = models.CharField(max_length = 100)


    def __unicode__(self):
        return self.name

class Jugador(models.Model):
    ID = models.CharField(max_length = 100, primary_key = True)
    name = models.CharField(max_length = 100)
    dorsal = models.CharField(max_length = 100, blank = True)
    nacionalidad = models.CharField(max_length = 100)
    equipo = models.ForeignKey(Equipo, related_name = "Equipo Actual", blank = True, null = True)
    equipos = models.ManyToManyField(Equipo, related_name = "Equipos Anteriores", blank = True, null = True)


    def __unicode__(self):
        return self.name + ", Dorsal: " + self.dorsal
```

```
>>> from gestion.models import Equipo, Jugador
>>> equipo1 = Equipo.objects.create(name = "Cordoba", liga = "Española")
>>> equipo1.save()
>>> equipo2 = Equipo.objects.create(name = "Sevilla", liga = "Española")
>>> equipo2.save()
>>> equipo3 = Equipo.objects.create(name = "Malaga", liga = "Española")
>>> equipo3.save()
>>> equipo4 = Equipo.objects.create(name = "Espanyol", liga = "Española")
>>> equipo4.save()
>>> Equipo.objects.all()
[<Equipo: Cordoba>, <Equipo: Sevilla>, <Equipo: Malaga>, <Equipo: Espanyol>]
>>>
```

Creamos un jugador = jugador1

```
[Ubuntu@Chacal equipoFutbol]$ python manage.py shell
Python 2.7.9 (default, Apr  2 2015, 15:33:21)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from gestion.models import Equipo, Jugador
>>> Jugador.objects.all()
[<Jugador: Jorge Molina, Dorsal: >]
>>> e = Equipo.objects.get(name = "Cordoba")
>>> e.save()
>>> jugador1.equipo = e
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'jugador1' is not defined
>>> j = Jugador.objects.get(pk=1)
>>> j.equipo = e
>>> j.save()
>>> Jugador.objects.all()
[<Jugador: Jorge Molina, Dorsal: >]
>>> Jugador.objects.all().filter(equipo.name = "Cordoba")
  File "<console>", line 1
SyntaxError: keyword can't be an expression
>>> Jugador.objects.all().filter(equipo = "Cordoba")
[<Jugador: Jorge Molina, Dorsal: >]
>>> Jugador.objects.all().filter(equipo = "Sevilla")
[]
>>>
```

Una doble consulta: devuelve los jugadores cuyo equipo sea Sevilla

```
joseantonio@joseantonio ~/Documentos/PW/Prácticas/equipoFutbol $ python manage.py shell
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from gestion.models import Jugador,Equipo
>>> Jugador.objects.all().filter(equipo=Equipo.objects.all().filter(name="Sevilla"))
[<Jugador: Pepito>]
>>>
```

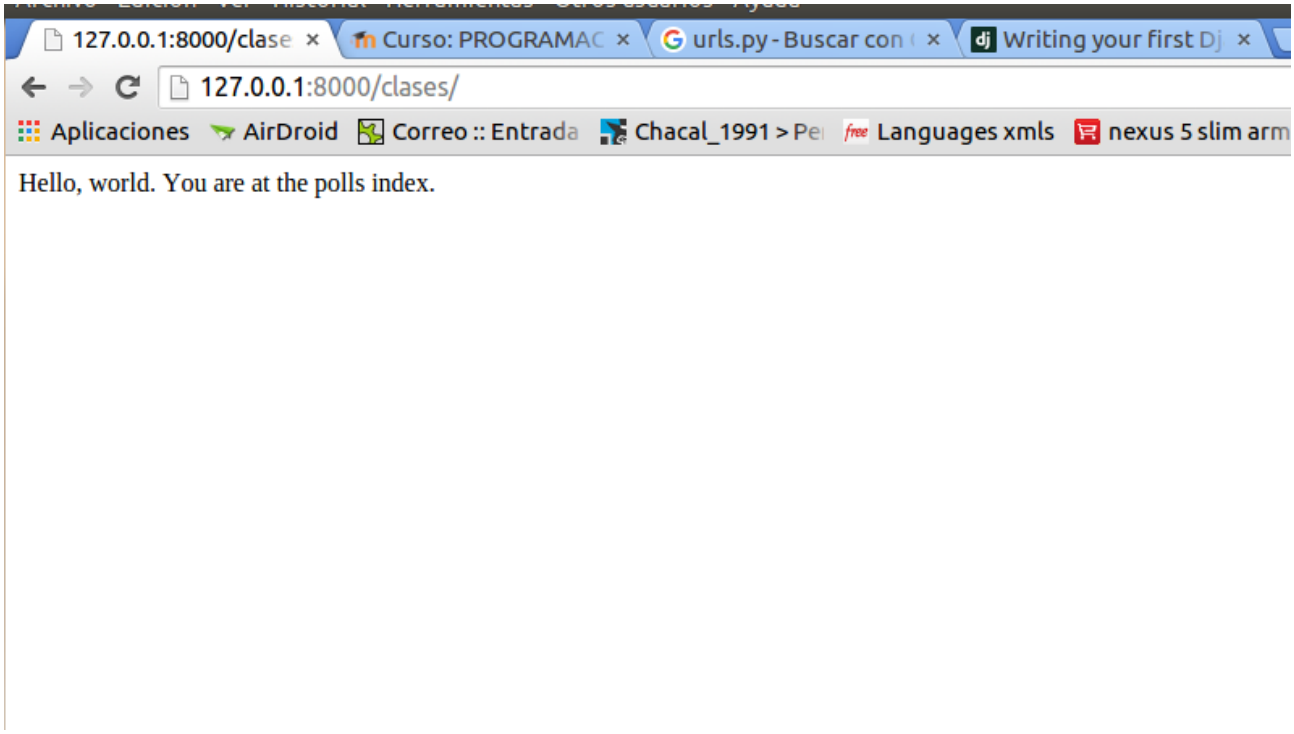
urls.py

views.py | urls.py - clases | urls.py - /home/chacal/Drop...

```python
#/project/clases/views.py

from django.shortcuts import render

from django.http import HttpResponse
def index(request):
    return HttpResponse("Hello, world. You are at the polls index.")

# Create your views here.

```

```python
#project/clases/urls.py

from django.conf.urls import patterns, include, url
from django.contrib import admin
from clases import views


urlpatterns = patterns('',
    url(r'^$', views.index, name = 'index'),
)

```

```python
# /project/urls.py

from django.conf.urls import patterns, include, url
from django.contrib import admin

urlpatterns = patterns('',
    url(r'^clases/', include('clases.urls')),
    url(r'^admin/', include(admin.site.urls)),
)

```


127.0.0.1:8000/clase
Curso: PROGRAMAC
urls.py - Buscar con
Writing your first Dj
127.0.0.1:8000/clases/
Aplicaciones
AirDroid
Correo :: Entrada
Chacal_1991 > Pe
Languages xmls
nexus 5 slim arm
Hello, world. You are at the polls index.

template e imprimir lista

views.py | models.py | .bashrc | urls.py — noticias

```python
from django.shortcuts import render
from django.http import HttpResponse
from noticias.models import Noticia, Autor, Seccion
from django.template import RequestContext, loader

# Create your views here.

#def index(request):
#    return HttpResponse("Hello, world. You're at the noticias index.")

def detail(request, noticia_id):
    return HttpResponse("You're looking at noticia %s. " % noticia_id)

def index(request):
    latest_noticia_list = Noticia.objects.order_by('-id')[:5]
    template = loader.get_template('noticias/index.html')
    context = RequestContext(request, {
    'latest_noticia_list': latest_noticia_list,
    })
    return HttpResponse(template.render(context))


```

views.py | models.py | urls.py — noticias | index.html

```
{% if latest_noticia_list %}
    <ul>
    {% for noticia in latest_noticia_list %}
        <li><a href="/noticias/{{ noticia.id }}/">{{ noticia.name }}</a></li>
    {% endfor %}
    </ul>
{% else %}
    <p>No hay noticias disponibles.</p>
{% endif %}
```

127.0.0.1:8000/noticias/

- Noticia 45
- Noticia 4
- Noticia 2
- Noticia 1
- Llueve

ADMINISTRADOR

Para crear crear un usuario administrador, debemos de irnos a la carpeta raíz de nuestro proyecto y escribir lo siguiente:

$ python manage.py createsuperuser

te pedirá que introduzcas un nombre de usuario, correo electrónico y contraseña de la cuenta admin que hemos creado.

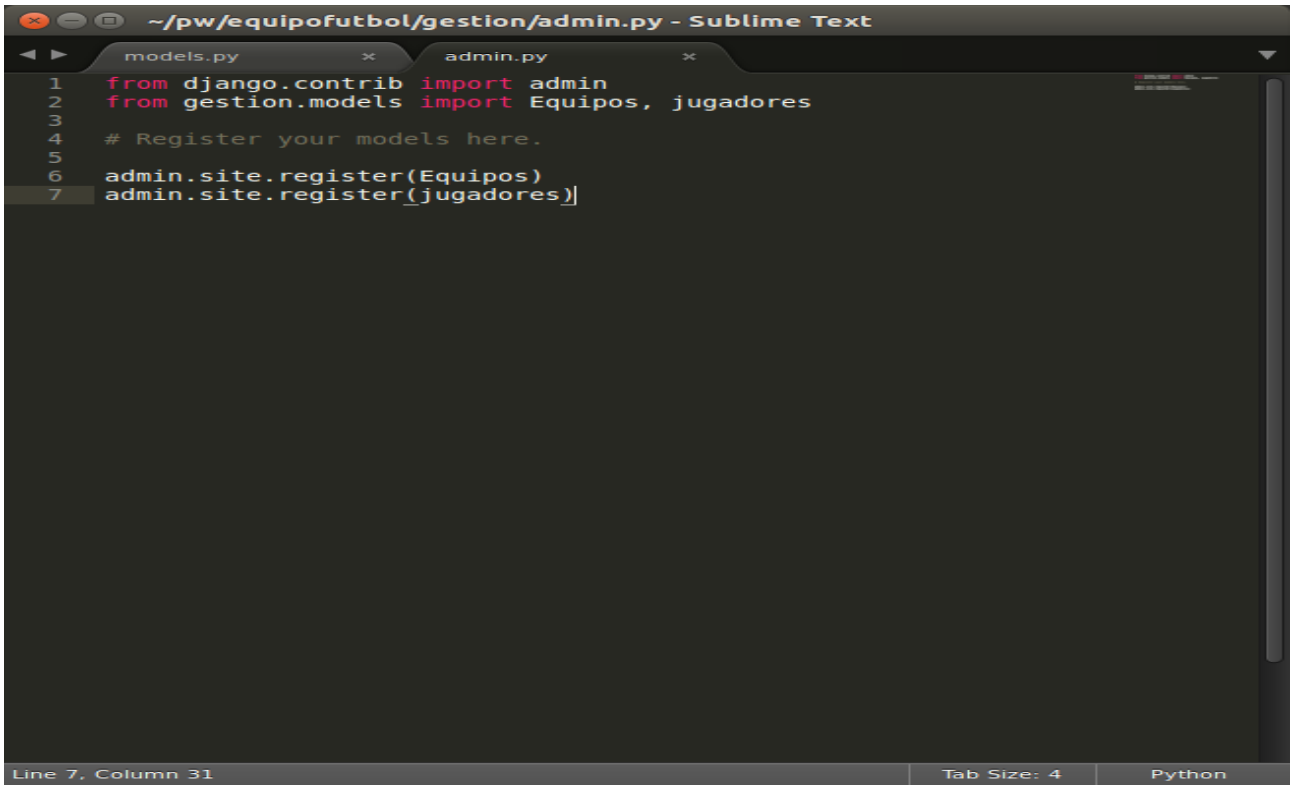
El siguiente paso es arrancar el servidor para poder logearnos con el nuevo usuario creado.

$ python manage.py runserver

Ahora nos vamos al navegador y abrimos la siguiente direccion:

http://127.0,0,1:8000/admin

Por ultimo, nos debemos de ir al archivo admin.py que se ha creado en nuestra raíz del proyecto y debemos editarlo, introduciendo las siguientes lineas:



Importando así los campos introducidos previamente en el archivo models.py
Ahora actualizamos la pagina anterior y nos aparecerán estos campos.