

IES Fernando Aguilar Quignon

1º ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

KARLITA

José María Riol Sánchez

28 de abril de 2021

Índice

1. Introducción	2
2. Instalación - PIP, Python Django	2
3. Empezar Proyecto Nuevo	3
4. Primera migración	4
5. Crear superusuario + Administración de Django	5
6. Primera aplicación	7
7. Primer modelo	8
8. Crear Objetos en Python Shell + Registrar Model en Admin.	9
9. Personalizar Modelo en el Admin	11

1. Introducción

En este documento vamos a recoger todo el progreso que vayamos haciendo en la elaboración del curso PROBAR DJANGO | Crear una Aplicación Web en Udemty. Dividiremos el documento de ahora en adelante en secciones por cada vídeo, y dentro de cada sección tendremos subsecciones donde tendremos todo lo relacionado con el trabajo realizado, el log de cabezazos y demás información que se pida en la tarea.

Todos los commits de este curso podrán verse en el siguiente enlace → [Repositorio de Karlita](#)

2. Instalación - PIP, Python Django

En este vídeo nos enseña como instalar Python, en nuestro caso ya tenemos instalado Python3, como instalar pip, hacer un entorno virtual e instalar Django. Refrescamos como hacer un entorno virtual. Para crear uno, una vez estamos en el directorio deseado hacemos uso del comando:

```
python3 -m venv curso_django
```

Esto nos creará un directorio con la estructura de sistema de archivos de como Python guarda los módulos que se han instalado, de esa manera no afectarán a los otros módulos que tengamos instalados en nuestro sistema.

Para activarlo hacemos uso del comando:

```
source curso_django/bin/activate
```

```
joserisa@pc09-t3:/VMs/joserisa$ source curso_django/bin/activate
(curso_django) joserisa@pc09-t3:/VMs/joserisa$
```

Podemos ver que el prompt ha cambiado. Ahora dentro de este entorno virtual, con pip descargamos django con el siguiente comando:

```
pip install django
```

```
(curso_django) joserisa@pc09-t3:/VMs/joserisa$ pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/a8/9b/fe94c509e514f6c227308e81076506eb9d67f2bfb8061ce5cdfbde0432e3/Django-3.2-py3-none-any.whl (7.9MB)
    100% |#####| 7.9MB 164kB/s
Collecting asgiref<4,>=3.3.2 (from django)
  Downloading https://files.pythonhosted.org/packages/17/8b/05e225d11154b8f5358e6a6d277679c9741ec0339d1e451c9cef687a9170/asgiref-3.3.4-py3-none-any.whl
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/70/94/784178ca5dd892a98f113cdd923372024dc04b8d40abe77ca76b5fb90ca6/pytz-2021.1-py2.py3-none-any.whl (510kB)
    100% |#####| 512kB 1.8MB/s
Collecting sqlparse<=0.2.2 (from django)
  Downloading https://files.pythonhosted.org/packages/14/05/6e0eb62ca685b10e34051a80d7ea94b7137369d8c0be5c3b9d9b6e3f5dae/sqlparse-0.4.1-py3-none-any.whl (42kB)
    100% |#####| 51kB 2.7MB/s
Collecting typing-extensions; python_version < "3.8" (from asgiref<4,>=3.3.2->django)
  Downloading https://files.pythonhosted.org/packages/60/7a/e881b5abb54db0e6e71ab088d079c57ce54e8a01a3ca443f561ccadb37e/typing_extensions-3.7.4.3-py3-none-any.whl
Installing collected packages: typing-extensions, asgiref, pytz, sqlparse, django
Successfully installed asgiref-3.3.4 django-3.2 pytz-2021.1 sqlparse-0.4.1 typing-extensions-3.7.4.3
(curso_django) joserisa@pc09-t3:/VMs/joserisa$
```

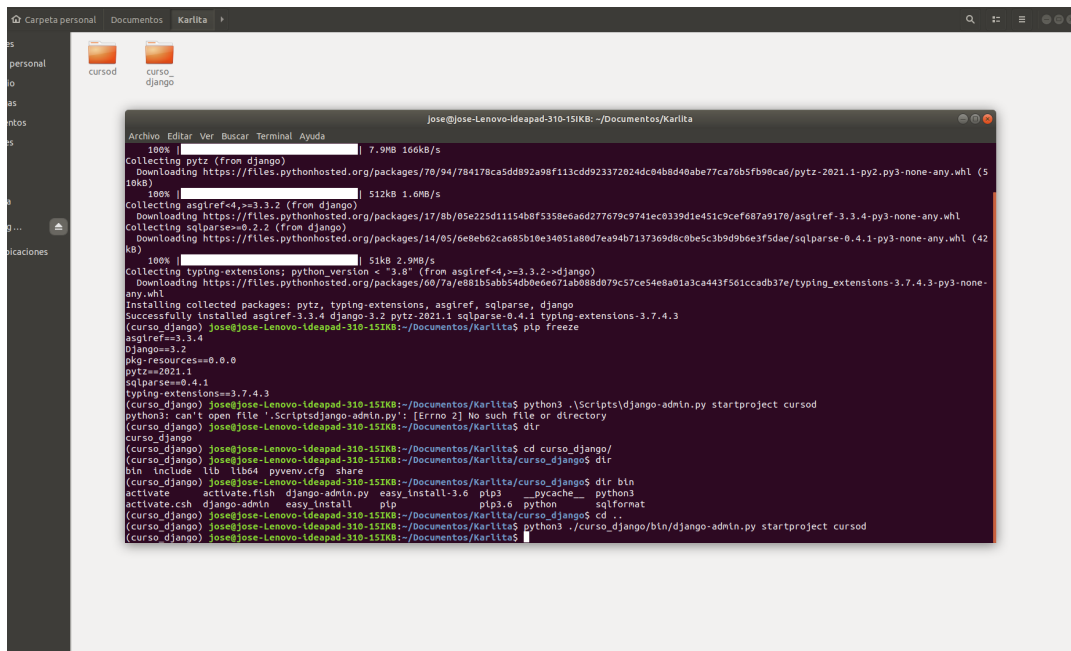
Con estas configuraciones ya estamos preparados para empezar un nuevo proyecto.

3. Empezar Proyecto Nuevo

En este capítulo nos muestra como crear un entorno virtual para el curso, nosotros ya lo tenemos creado del vídeo anterior.

Procedemos a crear el nuevo proyecto en nuestro caso usando el siguiente comando dentro de nuestra carpeta Karlita

```
python3 ./curso_django/bin/django-admin.py startproject cursod
```

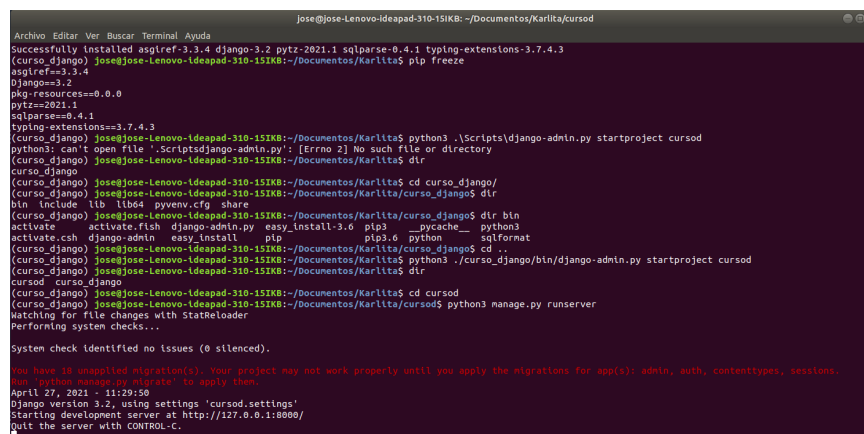


```
Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita
Archivo Editor Ver Buscar Terminal Ayuda
100% | 7.9MB 166kB/s
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/70/94/784178ca5dd892a98f113cdd923372024dc04b8d40abe77ca76b5fb98ca6/pytz-2021.1-py2.py3-none-any.whl (510kB)
100% | 512kB 1.6MB/s
Collecting asgiref<4,=>3.3.2 (from django)
  Downloading https://files.pythonhosted.org/packages/17/8b/05e225d1154b6f5358e6a0d277679c9741ec0339d1e451c9cf687a9170/asgiref-3.3.4-py3-none-any.whl
Collecting sqlparse<0.2,=>0.2 (from django)
  Downloading https://files.pythonhosted.org/packages/14/05/6e8eb2ca685b10e34051a80d7ea94b7137369d8c0be5c3b9d9b6e3f5dae/sqlparse-0.4.1-py3-none-any.whl (42kB)
100% | 51kB 2.9MB/s
Collecting typing-extensions; python_version < "3.8" (from asgiref<4,=>3.3.2->django)
  Downloading https://files.pythonhosted.org/packages/60/7a/e881b5abb54db0e0e71ab088d079c57ce54e8a01a3ca443f561ccadb37e/typing_extensions-3.7.4.3-py3-none-any.whl
Installing collected packages: pytz, typing-extensions, asgiref, sqlparse, django
Successfully installed asgiref-3.3.4 django-3.2 pytz-2021.1 sqlparse-0.4.1 typing-extensions-3.7.4.3
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ python3 ./Scripts/django-admin.py startproject cursod
python3: can't open file './Scripts/django-admin.py': [Errno 2] No such file or directory
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ dir
cursod.django
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ cd cursod.django/
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod.django$ dir
bin include lib lib64 pyenv.cfg share
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod.django$ dir bin
activate activate.fish django-admin.py easy_install-3.6 pip3 _pycache_ python3
activate.csh django-admin easy_install pip pip3.6 python sqlformat
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod.django$ cd ..
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ python3 ./cursod.django/bin/django-admin.py startproject cursod
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$
```

Podemos observar que se nos creó la carpeta cursod. Nos movemos a esa carpeta y ejecutamos:

```
python3 manage.py runserver
```

Y tendremos el siguiente efecto:



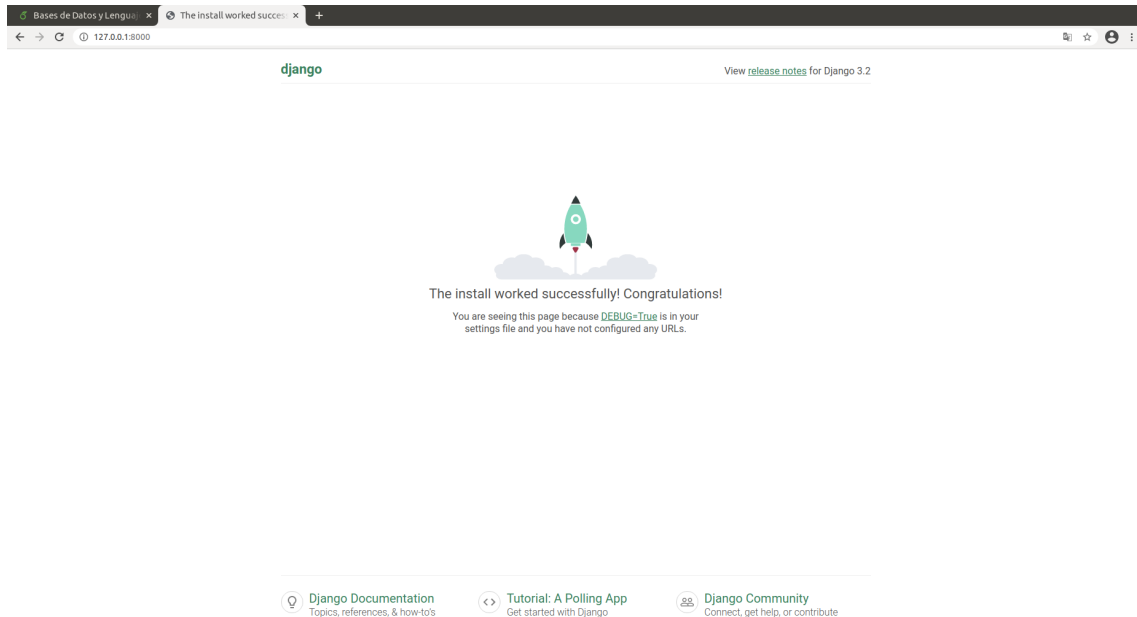
```
Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod
Archivo Editor Ver Buscar Terminal Ayuda
Successfully installed asgiref-3.3.4 django-3.2 pytz-2021.1 sqlparse-0.4.1 typing-extensions-3.7.4.3
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ pip freeze
asgiref==3.3.4
Django==3.2
pkg-resources==0.0.0
pytz==2021.1
sqlparse==0.4.1
typing-extensions==3.7.4.3
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ python3 ./Scripts/django-admin.py startproject cursod
python3: can't open file './Scripts/django-admin.py': [Errno 2] No such file or directory
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ dir
cursod.django
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ cd cursod.django/
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod.django$ dir
bin include lib lib64 pyenv.cfg share
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod.django$ dir bin
activate activate.fish django-admin.py easy_install-3.6 pip3 _pycache_ python3
activate.csh django-admin easy_install pip pip3.6 python sqlformat
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod.django$ cd ..
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ python3 ./cursod.django/bin/django-admin.py startproject cursod
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ dir
cursod cursod.django
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita$ cd cursod
(cursod.django) Jose@Jose-Lenovo-Ideapad-310-15IKB: ~/Documentos/Karlita/cursod$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

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.

April 27, 2021 - 11:29:50
Django version 3.2, using settings 'cursod.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Nos interesa la IP, `http://127.0.0.1:8000/` la copiamos, abrimos el navegador y la pegamos, nos aparece la siguiente pantalla:



Tenemos pues el primer proyecto funcionando.

No hacemos ningún commit porque no hemos hecho nada de código, solamente hemos configurado un par de cosas y comprobado que funciona.

4. Primera migración

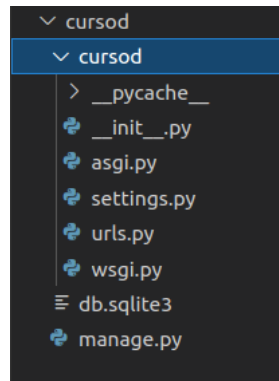
Debido al mensaje: `You have 18 unapplied migration(s)` debemos empezar a hacer nuestras migraciones, para que esté todo sincronizado y funcionando bien. Debemos crear el vínculo entre la BD y nuestro proyecto, para ello cerramos el server y ejecutamos:

```
python3 manage.py migrate
```

y nos lleva a cabo lo siguiente:

```
^C(curso_django) jose@jose-Lenovo-Ideapad-310-15IKB:~/Documentos/Karlita/cursod$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(curso_django) jose@jose-Lenovo-Ideapad-310-15IKB:~/Documentos/Karlita/cursod$ ^C
(curso_django) jose@jose-Lenovo-Ideapad-310-15IKB:~/Documentos/Karlita/cursod$
```

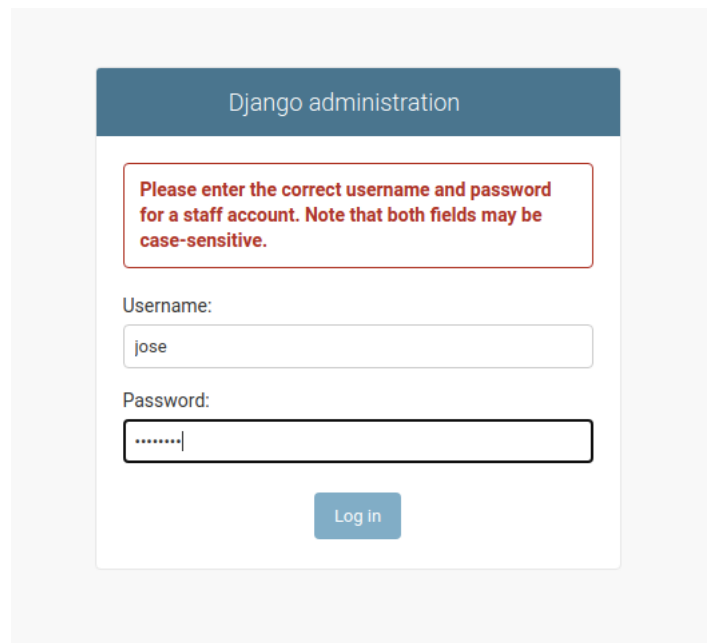
En nuestro caso usamos CODE para añadir la carpeta de trabajo cursod y supongo que trabajar desde ahí.



En este caso tampoco hacemos ningún commit, no hubo cambio alguno.

5. Crear superusuario + Administración de Django

Debido a que cuando arrancamos el server y procedemos a hacer login dice que nuestro usuario no está registrado.



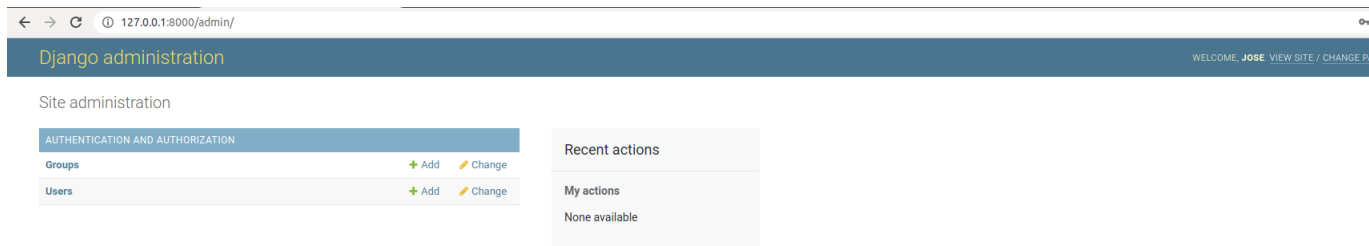
Por tanto, volvemos a cerrar el servidor y procedemos a crear un super usuario que nos permita entrar, ejecutamos el comando:

```
python3 manage.py createsuperuser
```

y completamos con la información que nos pida.

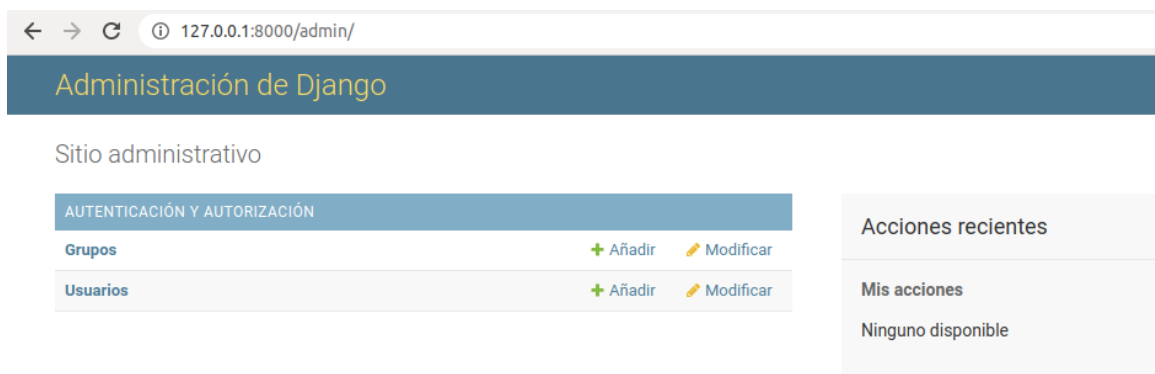
```
(curso_django) jose@jose-Lenovo-ideapad-310-15IKB:~/Documentos/Karlita/cursod$ python3 manage.py createsuperuser
Username (leave blank to use 'jose'):
Email address: josemaria.riol.sanchez.alu@iesfernandoaguiar.es
Password:
Password (again):
Superuser created successfully.
(curso_django) jose@jose-Lenovo-ideapad-310-15IKB:~/Documentos/Karlita/cursod$
```

Si volvemos a ejecutar nuestro server e intentamos entrar:



Podemos ver que ahora sí nos deja entrar.

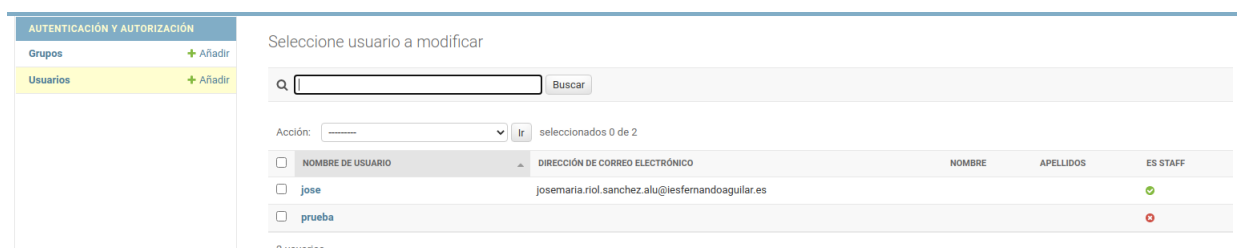
Vamos a poner ahora la página en español, nos vamos a settings.py y nos vamos a la línea LANGUAGE_CODE = 'es' y al recargar nos saldrá la página en español.



Navegamos dentro de grupos y usuarios y podemos ver la lista de los mismos.



Podemos ver que podemos añadir usuarios, grupos. Si le damos a añadir usuario y creamos un usuario cualquiera podremos ver que la lista de usuarios se actualizará.



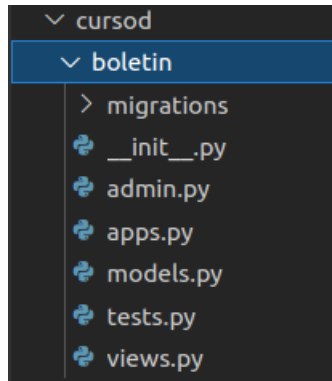
6. Primera aplicación

En esta ocasión crearemos nuestra primera aplicación mediante un comando. Cerramos el server si está abierto y escribimos:

```
python3 manage.py startapp boletin
```

```
^C(curso_django) jose@jose-Lenovo-ideapad-310-15IK8:~/Documentos/Karlita/cursod$ python3 manage.py startapp boletin
(curso_django) jose@jose-Lenovo-ideapad-310-15IK8:~/Documentos/Karlita/cursod$
```

Se nos creará una carpeta con el nombre boletín dentro de nuestro "cursod" con varios ficheros.



Ahora tenemos que registrarlo en el settings.py de nuestro proyecto. Entramos en settings.py y añadimos boletin de la siguiente forma:

```
29
30
31 # Application definition
32
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     'boletin',
41 ]
42
```


7. Primer modelo

Procedemos a modificar models.py de boletín.

```
curso_django > boletin > models.py > ...
1  from __future__ import unicode_literals
2
3  from django.db import models
4
5  # Create your models here.
6  class Resgistrado(models.Model):
7      nombre= models.CharField(max_length=100, blank=True, null=True)
8      email= models.EmailField()
9      timestamp= models.DateTimeField(auto_now_add=True, auto_now=False)
10
11     def __unicode__(self):
12         return self.email
13
14     def __str__(self):
15         return self.email
```

Una vez hayamos terminado con models.py debemos hacer una migración, haciendo uso de dos comandos:

```
python3 manage.py makemigrations
```

Con esto empaquetaremos los cambios.

```
(curso_django) jose@jose-Lenovo-ideapad-310-15IKB:~/Documentos/Karlita/cursod$ python3 manage.py makemigrations
Migrations for 'boletin':
  boletin/migrations/0001_initial.py
    - Create model Resgistrado
(curso_django) jose@jose-Lenovo-ideapad-310-15IKB:~/Documentos/Karlita/cursod$
```

El siguiente comando sería python3 manage.py migrate

```
(curso_django) jose@jose-Lenovo-ideapad-310-15IKB:~/Documentos/Karlita/cursod$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, boletin, contenttypes, sessions
Running migrations:
  Applying boletin.0001_initial... OK
(curso_django) jose@jose-Lenovo-ideapad-310-15IKB:~/Documentos/Karlita/cursod$
```

Y ya habríamos terminado por aquí.

8. Crear Objetos en Python Shell + Registrar Model en Admin.

Vamos a crear un objeto de Python a partir del shell, hacemos uso del comando:

```
(curso_django) jose@jose-Lenovo-Ideapad-310-15IKB:~/Documentos/Karlita/cursod$ python3 manage.py shell
Python 3.6.9 (default, Jan 26 2021, 15:33:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>>
```

Hacemos una importación desde nuestro modelo, y posteriormente ponemos una variable nueva "gente", creado como un set de objetos.

```
>>> from boletin.models import Registrado
>>> gente = Registrado.objects.all()
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: type object 'Registrado' has no attribute 'objects'
>>> gente = Registrado.objects.all()
>>> gente
<QuerySet []>
>>>
```

Si queremos añadir una persona, lo haríamos así:

```
>>> persona1 = Registrado.objects.create(nombre='joselito', email='j@email.com')
>>> persona1
<Registrado: j@email.com>
>>>
```

El problema es que tal y como hemos hecho esto, no hay forma de ver los cambios fuera del shell, así que vamos a llevarlo a cabo a través de la administración de Django. Nos metemos en admin.py de boletin y registramos nuestro modelo de la siguiente forma:

```
cursod > boletin > admin.py > ...
1  from django.contrib import admin
2
3  # Register your models here.
4  from .models import Registrado
5
6  admin.site.register(Registrado)
```

Volvemos a ejecutar nuestro servidor y entramos en él, observamos que tenemos boletin, Registrados y dentro de Registrados 1 registrado.



Administración de Django

Inicio › Boletín › Registrados

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos + Añadir

Usuarios + Añadir

BOLETIN

Registrados + Añadir

Seleccione registrado a modificar

Acción: ----- Ir seleccionados 0 de 1

☐ REGISTRADO

☐ j@email.com

1 registrado

Viene solamente el email por el unicode que escribimos, si lo cambiamos a nombre se reflejará el cambio.

Administración de Django

Inicio › Boletín › Registrados

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos + Añadir

Usuarios + Añadir

BOLETIN

Registrados + Añadir

Seleccione registrado a modificar

Acción: ----- Ir seleccionados 0 de 1

☐ REGISTRADO

☐ joselito

1 registrado

Al pinchar en el nombre vemos los dos campos, uno en negrita y otro no, porque pusimos que uno sí fuera obligatorio.

Administración de Django

Inicio › Boletín › Registrados › joselito

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos + Añadir

Usuarios + Añadir

BOLETIN

Registrados + Añadir

Modificar registrado

joselito

Nombre: joselito

Email: j@email.com

Eliminar

En el unicode no es recomendable poner un campo que no sea obligatorio, puesto que al registrarse alguien, si deja un campo en blanco se vería en blanco en la página y no es lo idóneo. Dejamos models.py como estaba con el email, aunque a nosotros nos da error y no se registra el usuario, pero dejando en el unicode el email, va bien.



Django de forma automática de un id aunque nosotros no lo pusiéramos en el código.

9. Personalizar Modelo en el Admin

Queremos personalizar el display de la administración, lo debemos hacer en el código de admin.py.

```
admin.py • models.py
cursod > boletin > admin.py > ...
1  from django.contrib import admin
2
3  # Register your models here.
4  from .models import Registrado
5
6  class AdminRegistrado(admin.ModelAdmin):
7      list_display = ["__unicode__", "nombre", "timestamp"]
8      class Meta:
9          model = Registrado
10
11  admin.site.register(Registrado, AdminRegistrado)
```

Volvemos a la página, recargamos y tenemos:

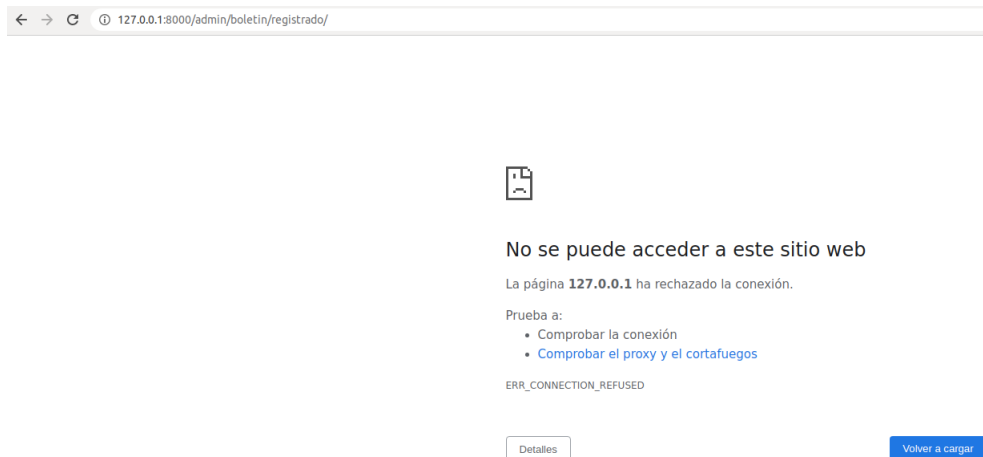


Podemos cambiar a nombre en list_display y si volvemos a recargar la página nos parecerán los nombres, pero lo vamos a dejar con el email puesto.

Añadimos en admin.py las líneas:

```
list_filter = ["timestamp"]
list_editable = [".email"]
search_fields = [".email", "nombre"]
```

Cuando recargamos la página nos da error.



Miramos la consola y vemos lo siguiente.

```
fn(*args, **kwargs)
File "/home/jose/Documentos/Karlita/curso_django/lib/python3.6/site-packages/django/core/management/commands/runserver.py", line 118, in inner_r
un
    self.check(display_num_errors=True)
File "/home/jose/Documentos/Karlita/curso_django/lib/python3.6/site-packages/django/core/management/base.py", line 469, in check
    raise SystemCheckError(msg)
django.core.management.base.SystemCheckError: SystemCheckError: System check identified some issues:

ERRORS:
<class 'boletin.admin.AdminRegistrado':>: (admin.E124) The value of 'list_editable[0]' refers to the first field in 'list_display' ('email'), which
cannot be used unless 'list_display_links' is set.

System check identified 1 issue (0 silenced).

/home/jose/Documentos/Karlita/curso_django/lib/python3.6/site-packages/django/core/management/commands/runserver.py changed, reloading.
Watching for file changes with StatReloader
Performing system checks...

Exception in thread django-main-thread:
Traceback (most recent call last):
  File "/usr/lib/python3.6/threading.py", line 916, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.6/threading.py", line 864, in run
    self._target(*self._args, **self._kwargs)
  File "/home/jose/Documentos/Karlita/curso_django/lib/python3.6/site-packages/django/utils/autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "/home/jose/Documentos/Karlita/curso_django/lib/python3.6/site-packages/django/core/management/commands/runserver.py", line 118, in inner_r
un
    self.check(display_num_errors=True)
File "/home/jose/Documentos/Karlita/curso_django/lib/python3.6/site-packages/django/core/management/base.py", line 469, in check
    raise SystemCheckError(msg)
django.core.management.base.SystemCheckError: SystemCheckError: System check identified some issues:

ERRORS:
<class 'boletin.admin.AdminRegistrado':>: (admin.E124) The value of 'list_editable[0]' refers to the first field in 'list_display' ('email'), which
cannot be used unless 'list_display_links' is set.
```

No podemos tener email en editable si es el primer elemento de la lista display, porque contiene un enlace hacia el usuario, por tanto es algo que no se puede modificar. Si ponemos nombre en editable, no hay problema y nos dejaría editarlo.

<input type="checkbox"/>	EMAIL	NOMBRE
<input type="checkbox"/>	lu@email.com	
<input type="checkbox"/>	p@email.com	k
<input type="checkbox"/>	j@email.com	joeslito

3 registrados

Si añadimos `list_display_links` podemos solventar de otra forma este error y además podemos conservar en editable el email, haremos que el link esté en el nombre.

<input type="checkbox"/>	EMAIL	NOMBRE
<input type="checkbox"/>	<input type="text" value="lu@email.com"/>	-
<input type="checkbox"/>	<input type="text" value="p@email.com"/>	k
<input type="checkbox"/>	<input type="text" value="j@email.com"/>	josecito

3 registrados

Vemos que podemos editar ahora el email y el enlace es el nombre ahora, podemos observar que quien no tiene nombre le añade un guión. En el vídeo no le aparece guión y explica como solventarlo, aunque a nosotros no nos haga falta cambiar nada, lo haremos por continuar el curso.