

Creación de una landing page en Python



PYTHON

django

Misión

Ser capaces de desarrollar una aplicación web completa usando un framework MVC (Modelo, Vista, Controlador)

Planteamiento

Crear una prueba de vida de haber seguido el curso gratuito de Udemy sobre Django pero escribiendo el código en Python3 y subiendo a GitHub el estado de la aplicación tras cada capítulo.

Además, se pide un log de cabezazos. Es decir, cada vez que te encuentres un problema o una divergencia entre lo que ocurre en el vídeo y en la realidad, lo describas y describas también cómo lo has solucionado.

¿Qué se valora?

- Que los commit estén coherentemente distribuidos en el tiempo
- Que la presentación del trabajo esté bien ligada en un pdf o en una web (un site de Google por ejemplo) con pantallazos y explicaciones.

Empecemos por lo fundamental, ¿qué es...

PIP?

PIP (Pip Instalador de Paquetes o Pip Instalador de Python) es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python.

Python?

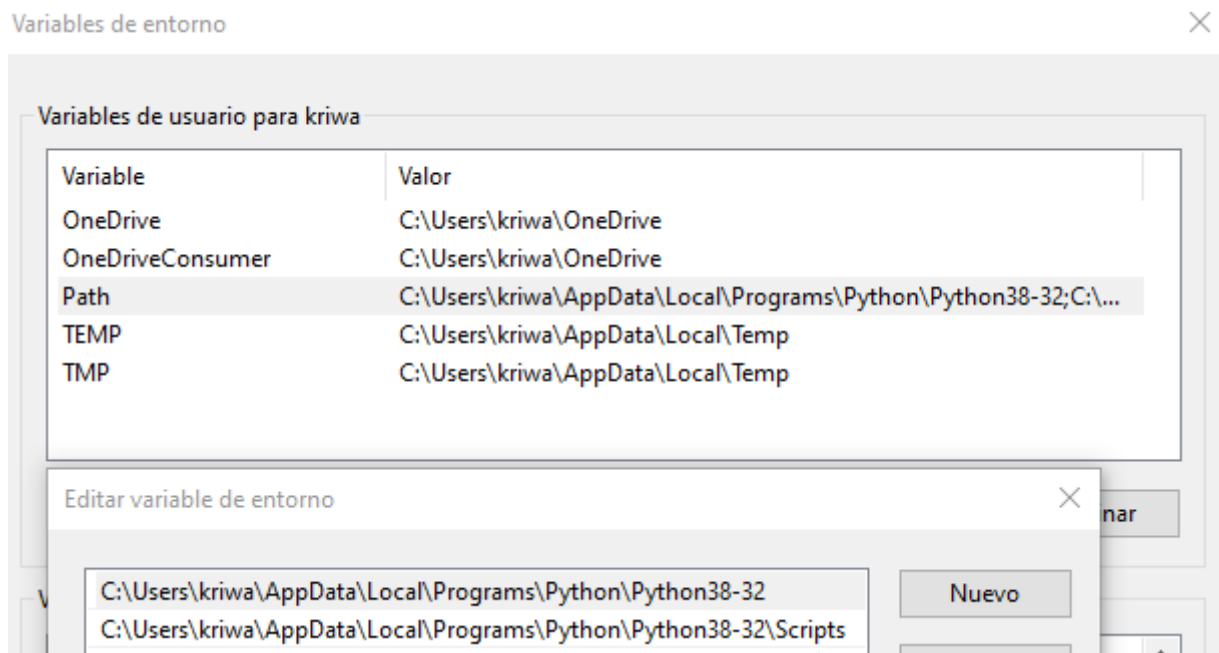
Es un lenguaje de programación interpretado, cuya filosofía hace hincapié en la legibilidad de su código y la orientación a objetos. Es un lenguaje interpretado, dinámico y multiplataforma, administrado por la Python Software Foundation.

Django?

Es un framework (o entorno de trabajo) de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC), un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. La meta fundamental de Django es facilitar la creación de sitios web complejos.

1. Instalación

Lo primero, descargarnos **Python** para **Windows** <https://www.python.org/downloads/>. Una vez hecho, tendremos que configurar las variables de entorno de nuestro sistema para que se pueda abrir *python* a través de la línea de comandos (cmd). Basta con averiguar la ruta donde hemos instalado nuestro *python* y crear una nueva variable de usuario (ver foto abajo)



Una vez configurado esto, podemos instalar *pip* desde cmd con el comando 'python -m pip install -U pip'

```
C:\Users\kriwa>python -m pip install -U pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/
/pip-21.0.1-py3-none-any.whl (1.5MB)
| 1.5MB 2.2MB/s
Installing collected packages: pip
  Found existing installation: pip 19.2.3
  Uninstalling pip-19.2.3:
    Successfully uninstalled pip-19.2.3
  Successfully installed pip-21.0.1
```

Con *pip* ya instalado, nuestro siguiente paso consiste en instalar **Virtual Enviroment**. Para ello, escribimos 'pip install virtualenv'

```
C:\Users\kriwa>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.4.4-py2.py3-none-any.whl (7.2 MB)
    | 7.2 MB 656 kB/s
Collecting six<2,>=1.9.0
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Collecting appdirs<2,>=1.4.3
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting distlib<1,>=0.3.1
  Downloading distlib-0.3.1-py2.py3-none-any.whl (335 kB)
    | 335 kB 819 kB/s
Collecting filelock<4,>=3.0.0
  Downloading filelock-3.0.12-py3-none-any.whl (7.6 kB)
Installing collected packages: six, filelock, distlib, appdirs, virtualenv
Successfully installed appdirs-1.4.4 distlib-0.3.1 filelock-3.0.12 six-1.15.0 virtualenv-20.4.4
```

La finalidad de crear un **Virtual Enviroment** es poder instalar **Django** dentro de él. A modo de prueba, creamos uno con 'virtualenv test_env' en nuestra carpeta personal. Lo activamos entrando en '.\Scripts\activate'.

Una vez activado, basta con escribir 'pip install django' y comenzará el proceso de instalación dentro de nuestro **Virtual Enviroment**.

```
(test_env) C:\Users\kriwa\test_env>pip install django
Collecting django
  Downloading Django-3.2-py3-none-any.whl (7.9 MB)
    | 7.9 MB 1.6 MB/s
Collecting pytz
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
    | 510 kB 1.3 MB/s
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.1-py3-none-any.whl (42 kB)
    | 42 kB 1.1 MB/s
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.3.4-py3-none-any.whl (22 kB)
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.3.4 django-3.2 pytz-2021.1 sqlparse-0.4.1
```

Vamos a crear un proyecto nuevo con **Django**, vamos a escribir lo siguiente 'python .\Scripts\django-admin.py startproject test_project'. Con esto indicamos que queremos iniciar un nuevo proyecto con su nombre. Para asegurarnos de su correcta instalación, podemos hacer un listado del contenido de nuestro **Virtual Enviroment**.

```
(test_env) C:\Users\kriwa\test_env>python .\Scripts\django-admin.py startproject test_project
.\Scripts\django-admin.py:17: RemovedInDjango40Warning: django-admin.py is deprecated in favor of django-admin.
warnings.warn(

(test_env) C:\Users\kriwa\test_env>dir
El volumen de la unidad C es Boot
El número de serie del volumen es: 6867-34E3

Directorio de C:\Users\kriwa\test_env

03/05/2021  12:58    <DIR>          .
03/05/2021  12:58    <DIR>          ..
03/05/2021  12:50             42 .gitignore
03/05/2021  12:50    <DIR>          lib
03/05/2021  12:50          416 pyenv.cfg
03/05/2021  12:52    <DIR>          Scripts
03/05/2021  12:58    <DIR>          test_project
                2 archivos          458 bytes
                5 dirs  19.267.997.696 bytes libres
```

2. Empezar proyecto nuevo

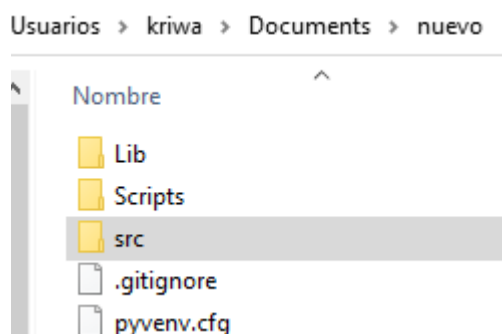
Para empezar un proyecto nuevo, abrimos nuestro cmd y creamos un nuevo directorio (en mi caso dentro de 'Documents' voy a crear una carpeta llamada 'nuevo'). Entramos en ella e instalamos el **Virtual Enviroment**. Después, lo activamos como hicimos en el ejemplo anterior (ver imagen inferior)

```
C:\Users\kriwa>cd Documents
C:\Users\kriwa\Documents>mkdir nuevo && cd nuevo
C:\Users\kriwa\Documents\nuevo>virtualenv .
created virtual environment CPython3.8.1.final.0-32 in 3981ms
creator CPython3Windows(dest=C:\Users\kriwa\Documents\nuevo, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\kriwa\AppData\Local\pypa\virtualenv)
added seed packages: pip==21.0.1, setuptools==56.0.0, wheel==0.36.2
activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
C:\Users\kriwa\Documents\nuevo>.\Scripts\activate
(nuevo) C:\Users\kriwa\Documents\nuevo>
```

Lo siguiente será instalar **Django** dentro de nuestro **Virtual Enviroment**. De nuevo, con el comando 'pip install django'

```
(nuevo) C:\Users\kriwa\Documents\nuevo>pip install django
Collecting django
  Using cached Django-3.2-py3-none-any.whl (7.9 MB)
Collecting pytz
  Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.1-py3-none-any.whl (42 kB)
Collecting asgiref<4,>=3.3.2
  Using cached asgiref-3.3.4-py3-none-any.whl (22 kB)
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.3.4 django-3.2 pytz-2021.1 sqlparse-0.4.1
```

Ya solo nos falta empezar un nuevo proyecto. Usamos 'python .\Scripts\django-admin.py startproject nuevo'. Como veréis, vamos a tener muchos directorios con el nombre 'nuevo'. Para no crear confusión, dentro de nuestro proyecto que creamos en 'Documents', vamos a renombrar la carpeta raíz del proyecto 'nuevo' a 'src'.



Volvemos al cmd y vamos a situarnos en nuestro 'src' para ejecutar el servidor de nuestro desarrollo de entorno. Escribiremos 'pyhton manage.py runserver'

```
(nuevo) C:\Users\kriwa\Documents\nuevo\src>python 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.
May 03, 2021 - 18:33:13
Django version 3.2, using settings 'nuevo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Con ello nuestro servidor empezará a funcionar. Simplemente copiamos la ruta del servidor y la añadimos a nuestro navegador web.

django

View [release notes](#) for Django 3.2



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

La instalación se instaló sin problemas. ¡Felicidades!

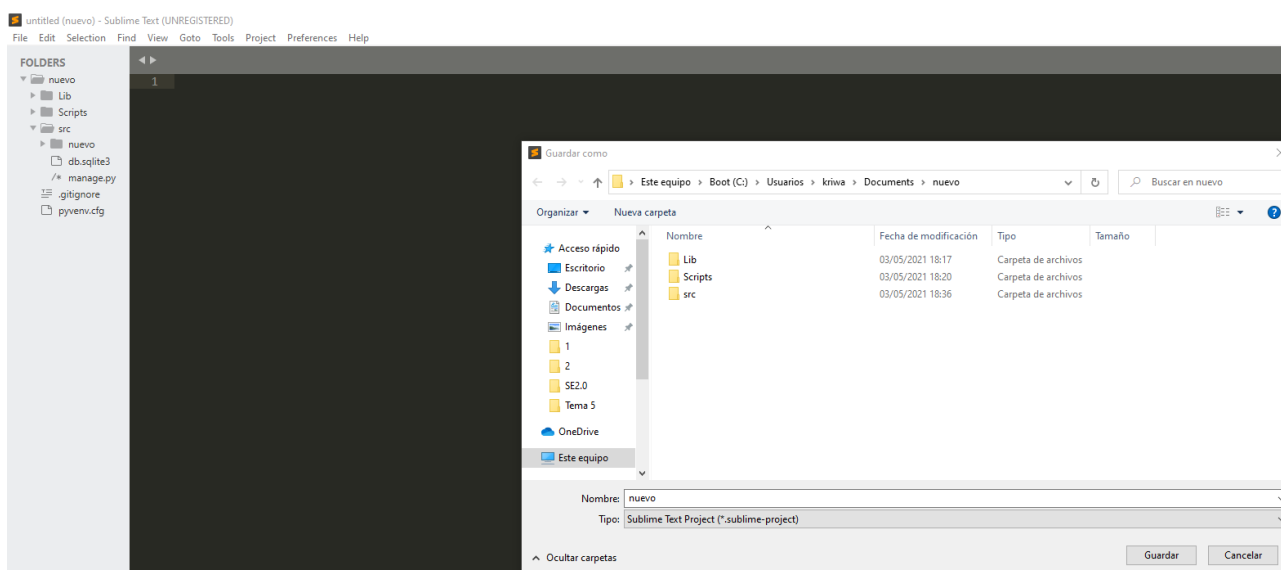
3. Primera migración

Ya tenemos nuestro proyecto funcionando. Ahora podemos empezar con las migraciones. Las migraciones nos sirven para comunicarnos con nuestra base de datos, es el vínculo entre la base de datos y nuestro proyecto.

En nuestro cmd, cerramos el servidor (Ctrl + D) y escribimos 'python manage.py migrate'.

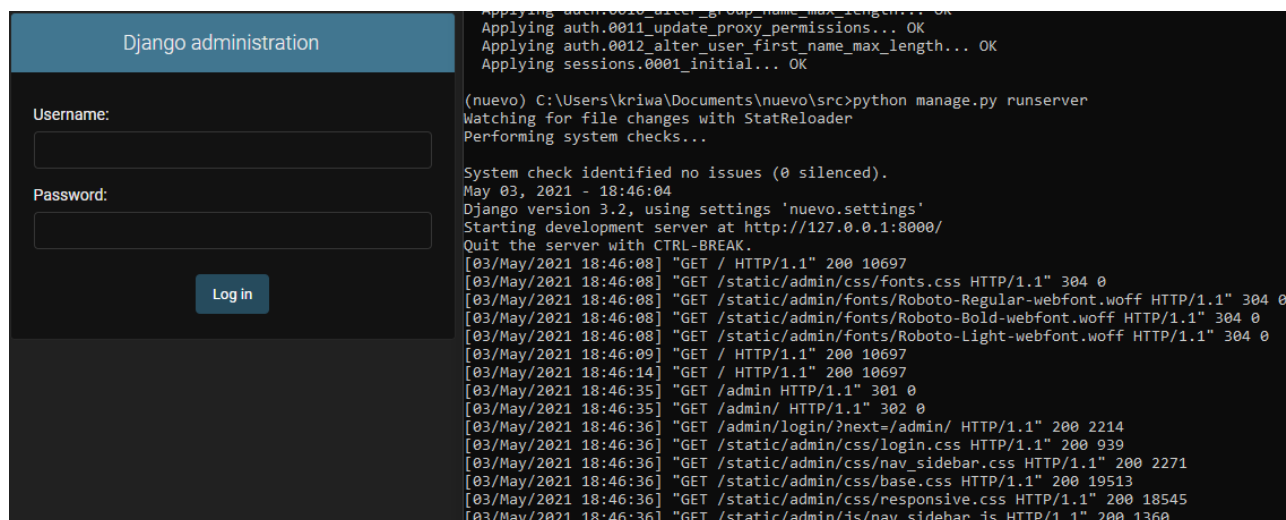
```
(nuevo) C:\Users\kriwa\Documents\nuevo\src>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... 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
```

A continuación, abrimos nuestro editor de texto (en mi caso Sublime Text) y vamos a trabajar con nuestro proyecto. Pinchamos en 'Project' y añadimos la carpeta del proyecto 'nuevo'. Una vez abierto, lo guardamos dentro de nuestra carpeta 'raíz'.



4. Superusuarios y Administración de Django

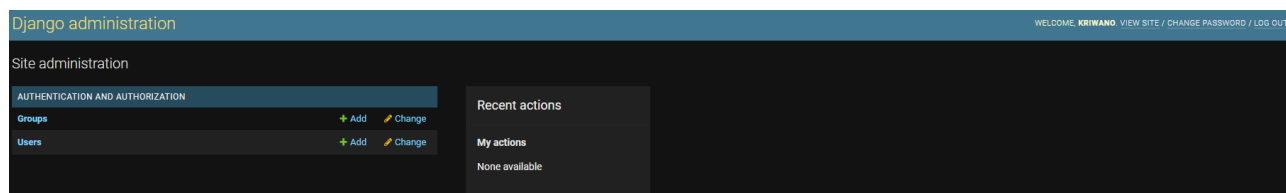
Los proyectos de **Django** vienen automáticamente con una interfaz ya escrita. Volvemos a activar nuestro servidor y recargamos en nuestro navegador web.



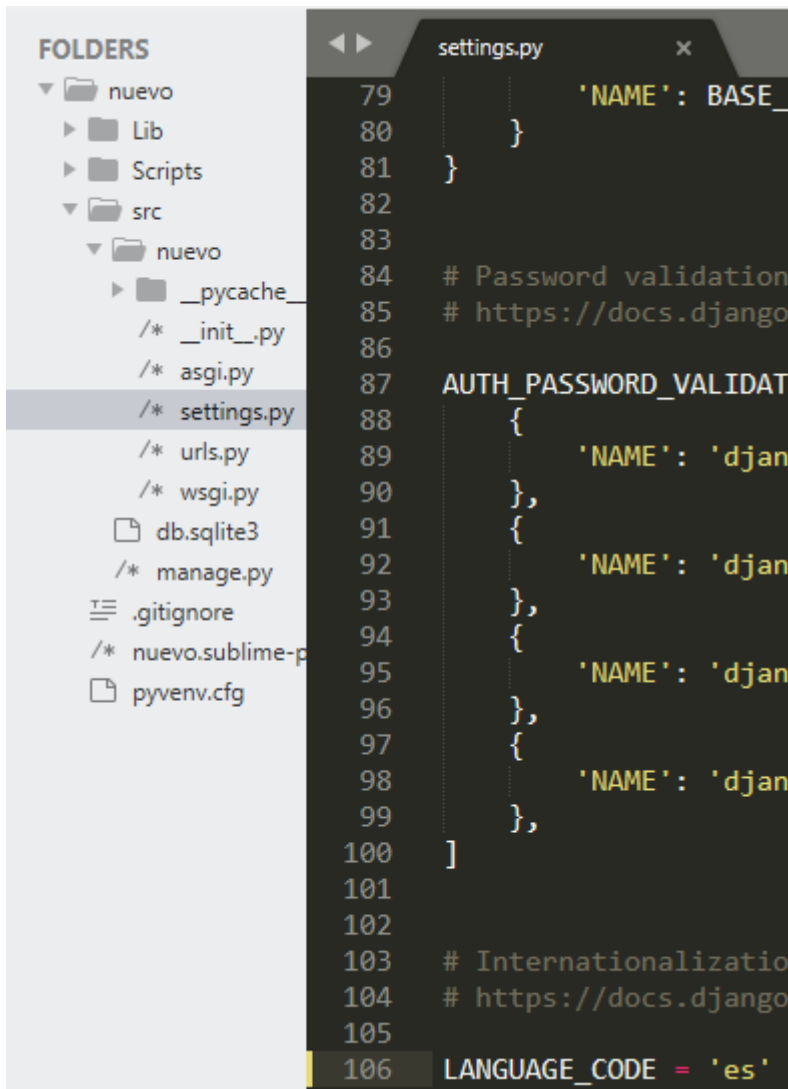
Para poder iniciar sesión dentro de **Django**, cerramos servidor y vamos a crear un *superusuario*, capaz de entrar. Para ello, ponemos en nuestro cmd 'python manage.py createsuperuser' y añadimos nuestro usuario, email y contraseña.

```
(nuevo) C:\Users\kriwa\Documents\nuevo\src>
(nuevo) C:\Users\kriwa\Documents\nuevo\src>python manage.py createsuperuser
Username (leave blank to use 'kriwa'): kriwano
Email address: ivanperezprofesional@hotmail.com
Password:
Password (again):
Superuser created successfully.
```

Volvemos a iniciar el servidor y ya podemos acceder a nuestra pagina. Esta es la pagina principal de la interfaz administrativa..



Si queremos que la pagina esté en ingles, podemos entrar en el archivo 'settings.py' y donde pone 'LANGUAGE_CODE' cambiar el idioma al español.



```
79     'NAME': BASE_
80     }
81 }
82
83
84 # Password validation
85 # https://docs.django
86
87 AUTH_PASSWORD_VALIDAT
88 {
89     'NAME': 'djan
90 },
91 {
92     'NAME': 'djan
93 },
94 {
95     'NAME': 'djan
96 },
97 {
98     'NAME': 'djan
99 },
100 ]
101
102
103 # Internationalizatio
104 # https://docs.django
105
106 LANGUAGE_CODE = 'es'
```

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos	+ Añadir	✎ Modificar
Usuarios	+ Añadir	✎ Modificar

Acciones recientes

Mis acciones

Ninguno disponible

Si, por ejemplo, queremos crear otro usuario, podemos entrar en la sección de ‘Usuarios’ y ‘Añadir usuario’, con un menú como el de la imagen de abajo.

Añadir usuario

Primero, ingrese un nombre de usuario y contraseña. Luego, podrá editar más opciones del usuario.

Nombre de usuario:
Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @/./+/_

Contraseña:
Su contraseña no puede asemejarse tanto a su otra información personal.
Su contraseña debe contener al menos 8 caracteres.
Su contraseña no puede ser una clave utilizada comúnmente.
Su contraseña no puede ser completamente numérica.

Contraseña (confirmación):
Para verificar, introduzca la misma contraseña anterior.

Si volvemos a ‘Usuarios’ podremos ver que ya tenemos el usuario creado. No es ‘personal’, no tendrá todos los permisos como nuestro *superusuario*.

Seleccione usuario a modificar

ÁÑADIR USUARIO +

Buscar

Acción: Ir seleccionados 0 de 2

	NOMBRE DE USUARIO	DIRECCIÓN DE CORREO ELECTRÓNICO	NOMBRE	APELLIDOS	ES STAFF
<input checked="" type="checkbox"/>	kriwano	ivanperezprofesional@hotmail.com			✓
<input checked="" type="checkbox"/>	usuario				✗

2 usuarios

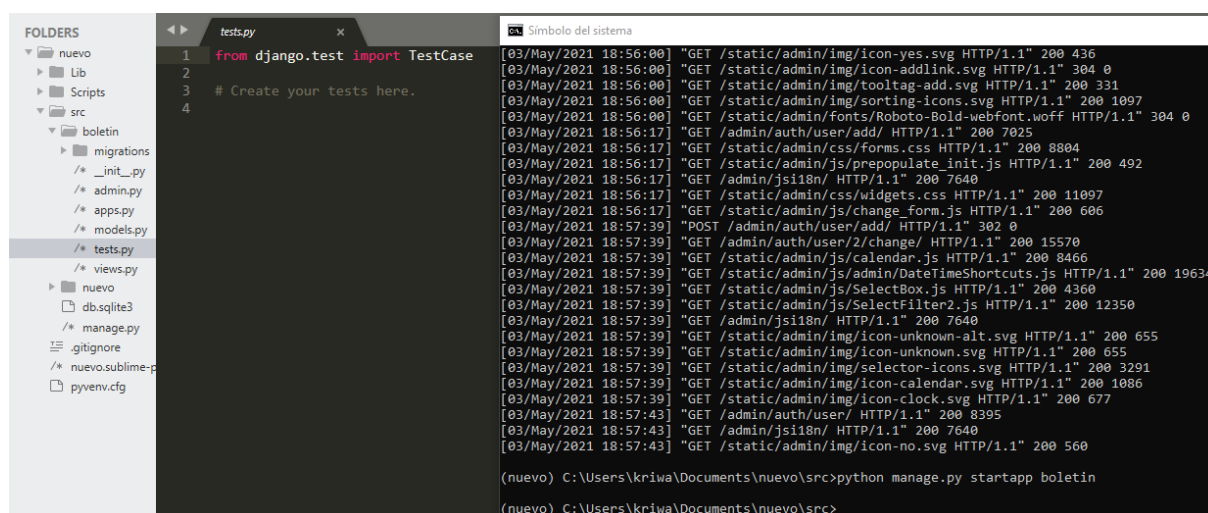
FILTRO
Por es staff
Todo
Sí
No
Por estado de superusuario
Todo
Sí
No
Por activo
Todo
Sí
No

5. Primera aplicación

Vamos a crear nuestra primera aplicación dentro de nuestro proyecto. La diferencia entre un sitio web y una aplicación web es muy subjetiva, un sitio web puede ser escrito solo en HTML, pero una aplicación web añade una interacción con el usuario.

Nuestra aplicación web sera una pagina de aterrizaje o *landing page*, pero además dentro vamos a crear aplicaciones. Una aplicación de nuestro proyecto va a ser un boletín y solo tendrá que ver con cosas de ese boletín. Hay que crear una aplicación para cada una de las funciones de nuestro proyecto.

Para crear nuestro boletín, cerramos el servidor y añadimos 'python manage.py startapp boletin'. Si volvemos a nuestro editor de texto, observaremos que se ha creado un nuevo directorio 'migrations' con varios archivos de *python*.



```
1 from django.test import TestCase
2
3 # Create your tests here.
4
```

```
[03/May/2021 18:56:00] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436
[03/May/2021 18:56:00] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 304 0
[03/May/2021 18:56:00] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331
[03/May/2021 18:56:00] "GET /static/admin/img/sorting-icons.svg HTTP/1.1" 200 1097
[03/May/2021 18:56:00] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
[03/May/2021 18:56:17] "GET /admin/auth/user/add/ HTTP/1.1" 200 7025
[03/May/2021 18:56:17] "GET /static/admin/css/forms.css HTTP/1.1" 200 8804
[03/May/2021 18:56:17] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 492
[03/May/2021 18:56:17] "GET /admin/jsi18n/ HTTP/1.1" 200 7640
[03/May/2021 18:56:17] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11097
[03/May/2021 18:56:17] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606
[03/May/2021 18:57:39] "POST /admin/auth/user/add/ HTTP/1.1" 302 0
[03/May/2021 18:57:39] "GET /admin/auth/user/2/change/ HTTP/1.1" 200 15570
[03/May/2021 18:57:39] "GET /static/admin/js/calendar.js HTTP/1.1" 200 8466
[03/May/2021 18:57:39] "GET /static/admin/js/admin/DateTimeShortcuts.js HTTP/1.1" 200 19634
[03/May/2021 18:57:39] "GET /static/admin/js/SelectBox.js HTTP/1.1" 200 4360
[03/May/2021 18:57:39] "GET /static/admin/js/SelectFilter2.js HTTP/1.1" 200 12350
[03/May/2021 18:57:39] "GET /admin/jsi18n/ HTTP/1.1" 200 7640
[03/May/2021 18:57:39] "GET /static/admin/img/icon-unknown-alt.svg HTTP/1.1" 200 655
[03/May/2021 18:57:39] "GET /static/admin/img/icon-unknown.svg HTTP/1.1" 200 655
[03/May/2021 18:57:39] "GET /static/admin/img/selector-icons.svg HTTP/1.1" 200 3291
[03/May/2021 18:57:39] "GET /static/admin/img/icon-calendar.svg HTTP/1.1" 200 1086
[03/May/2021 18:57:39] "GET /static/admin/img/icon-clock.svg HTTP/1.1" 200 677
[03/May/2021 18:57:43] "GET /admin/auth/user/ HTTP/1.1" 200 8395
[03/May/2021 18:57:43] "GET /admin/jsi18n/ HTTP/1.1" 200 7640
[03/May/2021 18:57:43] "GET /static/admin/img/icon-no.svg HTTP/1.1" 200 560

(nuevo) C:\Users\kriwa\Documents\nuevo\src>python manage.py startapp boletin

(nuevo) C:\Users\kriwa\Documents\nuevo\src>
```

Vemos que ya tenemos nuestra aplicacion 'boletín' pero nos falta registrarlo en nuestro archivo 'settings.py'. Buscamos el campo de 'INSTALLED_APPS' y lo añadimos.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'boletin',
]
```

6.