

## TEXT CLASS REVIEW

### TEMAS A TRATAR EN EL CUE

- El sitio administrativo de Django.
- Utilizando manage.py.
- Creando un super usuario.
- Limitando el acceso al sitio administrativo.
- Conociendo el sitio de administración de Django.
- Creación de usuarios en el sistema.

### EL SITIO ADMINISTRATIVO DE DJANGO

Django ofrece una Interfaz de administración muy útil para la gestión web de proyectos y aplicaciones. Este sitio administrativo lee metadatos de los modelos para proporcionar una interfaz rápida, la cual está centrada en el modelo donde los usuarios confiables pueden administrar el contenido de su sitio.

El sitio administrativo de Django usa los modelos creados en un proyecto para construir automáticamente un área dentro del sitio para crear, consultar, actualizar y borrar registros. Esto puede ahorrar mucho tiempo de desarrollo, pues es una manera sencilla de probar los modelos, y de verificar si los datos son correctos. También puede ser útil para manejar datos en producción, dependiendo del estilo del sitio web.

Se recomienda para la gestión de datos internos (sólo para uso de administradores, o personas internas de una organización), ya que como enfoque centrado en el modelo no es la mejor interfaz para todos los usuarios, pues expone una gran cantidad de detalles innecesarios.

### UTILIZANDO MANAGE.PY

`manage.py` es un script que se genera automáticamente al crear un proyecto de Django, y se utiliza para administrar dicho proyecto. Debe ejecutarse mediante comandos de Python, y acepta los comandos integrados proporcionados por Django, tales como:

- **check:** evalúa todo el proyecto en búsqueda de problemas comunes. Por defecto, verifica todas las aplicaciones. Si se desea verificar solo una en particular se debe especificar el nombre de ésta.
- **dbshell:** ejecuta el cliente de línea de comandos para el motor de base de datos especificado en la configuración de motor, con los parámetros de conexión especificados en su configuración de usuario, contraseña, entre otros. Para PostgreSQL, esto ejecuta el cliente de línea de comandos psql. Para MySQL, esto ejecuta el cliente de línea de comandos mysql.
- **diffsettings:** muestra las diferencias entre el archivo de configuración actual, y la configuración predeterminada de Django.
- **Flush:** elimina todos los datos de la base de datos. Las migraciones aplicadas no se borran.
- **Makemigrations:** crea una nueva migración basada en el modelo detectado. El papel de la migración consiste más en registrar el funcionamiento de la base de datos en forma de archivo, lo cual es conveniente para su posterior inspección y recuperación.
- **Migrate:** sincroniza el estado de la base de datos con el conjunto actual de modelos y migraciones. El comportamiento de este comando varía dependiendo de los argumentos suministrados.
- **Runserver:** inicia un servidor web de desarrollo ligero en la máquina local. De forma predeterminada, dicho servidor se ejecuta en el puerto 8000 en la dirección IP 127.0.0.1. Puede cambiar estos parámetros suministrándolos de forma explícita.
- **Shell:** inicia el intérprete interactivo de Python con el entorno Django, que es el entorno de línea de comandos. El intérprete interactivo básico de Python se usa por defecto.
- **Startapp:** crea una estructura de directorio de aplicaciones de Django para el nombre de la aplicación dada, en el directorio actual o el destino dado. De forma predeterminada, se crean una serie de plantillas de archivo en este nuevo directorio de aplicaciones, como models.py, views.py, admin.py, entre otras.
- **Startproject:** crea una estructura de directorio de proyectos de Django para el nombre del proyecto dado en el directorio actual, o el destino dado. De forma predeterminada, el nuevo directorio contiene manage.py, y un paquete de proyecto (que contiene settings.py y otros archivos).
- **Test:** ejecuta pruebas para todas las aplicaciones instaladas.

En realidad, hay tres formas de ejecutar comandos integrados proporcionados por Django en una ventana cmd, y manage.py es una de ellas:

```
1 $ python manage.py <command> [options]
```

## CREANDO UN SUPER USUARIO

Para iniciar sesión en el sitio de administración, se necesita una cuenta de usuario cuyo estado sea Staff habilitado. Para ver y crear registros también necesitamos que este usuario tenga permisos para administrar todos los objetos. Puedes crear una cuenta «administrador», que tenga acceso total al sitio y a todos los permisos necesarios usando el comando [manage.py](#).

Se emplea el siguiente comando, en el mismo directorio del proyecto Django donde se encuentra el módulo `manage.py`, para crear el usuario administrador. Se debe indicar un nombre de usuario, y una dirección de correo electrónico, así:

```
1 $ python3 manage.py createsuperuser --username admin --email admin@mail.com
```

Cuando se ejecuta de forma interactiva, este comando solicitará una contraseña para la nueva cuenta de super usuario. Pero, cuando se ejecuta de forma no interactiva, puede proporcionar una contraseña configurando la variable de entorno `DJANGO_SUPERUSER_PASSWORD`. De lo contrario, no se establecerá ninguna contraseña, y la cuenta de super usuario no podrá iniciar sesión hasta que se establezca una manualmente.

## LIMITANDO EL ACCESO AL SITIO ADMINISTRATIVO

Django admin se puede asegurar para limitar el acceso a éste haciendo uso de SSL, cambiar la URL de acceso, utilizando contraseñas fuertes, entre otras. Posteriormente, se listan varias acciones que pueden emplearse para limitar el acceso al sitio administrativo de Django:

- **Habilitar SSL:** es importante que el sitio web quede detrás de HTTPS, de lo contrario, es vulnerable a que la contraseña pueda ser capturada.
- **Cambiar la URL:** de forma predeterminada, la URL de administración es `/admin/`. Es importante cambiarla para despistar a un posible atacante del sitio web:

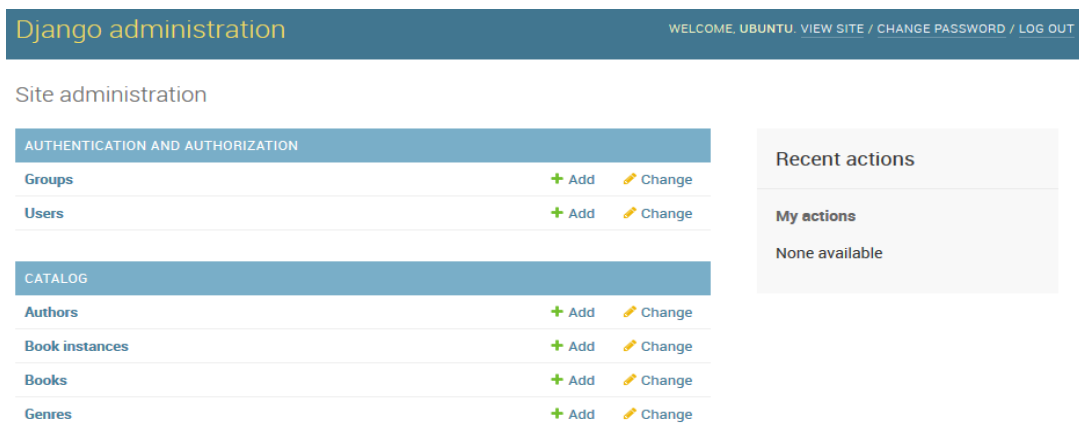
```
1 patrones de URL = [ruta ('mi-admin-secreto/', admin.sitio.urls),]
```

También sería importante, para brindar seguridad adicional, alojar al administrador en un dominio completamente diferente. Si se necesita aún más seguridad, puede ubicarse al administrador detrás de una Vpn.

- **Utilizar django-admin-honeypot:** después de mover el sitio de administración a una nueva URL (o incluso de haber decidido alojarlo en su propio dominio), instale la biblioteca `django-admin-honeypot` en la antigua `/admin/` URL, para capturar los intentos de piratear el sitio. `Django-admin-honeypot` generará una pantalla de inicio de sesión de administrador falsa. Luego, cada vez que alguien intente iniciar sesión en la antigua `/admin/` URL, éste enviará un correo electrónico a los administradores del sitio. El correo electrónico generado por `django-admin-honeypot` contendrá la dirección IP del atacante. Entonces, para mayor seguridad, si notamos repetidos intentos de inicio de sesión desde la misma dirección IP, podemos bloquearla.
- **Requerir contraseñas más seguras:** habilitar la validación de contraseña garantizará que los usuarios seleccionen contraseñas más seguras, lo que a su vez aumentará la seguridad de sus datos.
- **Utilizar autenticación de dos factores:** configurar un 2FA para el administrador de Django proporcionará más seguridad. Aquí hay tres formas en que se puede habilitar 2FA en el sitio:
  - **2FA** con SMS, donde enviamos un código de inicio de sesión. Esto es mejor que requerir solo una contraseña; pero los mensajes SMS son fáciles de interceptar.
  - **2FA** con una aplicación como Google Authenticator. Normalmente genera códigos de inicio de sesión únicos para cualquier servicio que registremos en él. Para configurar estas aplicaciones, los usuarios deberán escanear un código QR en el sitio para registrarlo con su aplicación. Luego, la aplicación le generará el código de inicio de sesión que pueden usar para tener acceso.
  - **2FA** con `YubiKey`, es una de las formas más seguras de habilitar **2FA** en el sitio. En este método, los usuarios deben tener un dispositivo físico, una `YubiKey`, que conectan a un puerto USB cuando intentan iniciar sesión.
  - La biblioteca `Django-two-factor-auth` puede ayudar a habilitar cualquiera de los métodos **2FA** anteriores.
- Utilizar la última versión de Django: es importante para mantener al día las actualizaciones de seguridad y las correcciones de errores.

## CONOCIENDO EL SITIO DE ADMINISTRACIÓN DE DJANGO

Para acceder al sitio administrativo de Django, esto se logra a través de la URL configurada para `/admin`, y las credenciales del super usuario creado. En la página de inicio el sitio muestra todos los modelos agrupados por aplicación instalada. Desde ahí se puede hacer clic sobre cualquiera de los modelos para obtener más detalles, tales como todos los registros asociados, y también editarlos al hacer clic sobre cualquiera de ellos. En esta misma interfaz puedes hacer clic sobre **Add**, seguido de cada modelo para comenzar a crear un registro de ese tipo.



Considerando el ejemplo de la imagen anterior, si tenemos una aplicación relacionada con libros, se puede agregar uno nuevo haciendo clic sobre **Add**, e introduciendo la información solicitada. Finalmente, se escoge la opción para guardar que sea la más conveniente para la siguiente acción que se desee realizar.

**Django administration** WELCOME, UBUNTU / [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) > [Catalog](#) > [Books](#) > [Add book](#)

### Add book

**Title:**

**Author:**  ▼ ✚ ✕

**Summary:**

Enter a brief description of the book

**ISBN:**   
13 Character ISBN number

**Genre:**

Science Fiction  
Fantasy  
Western  
French Poetry

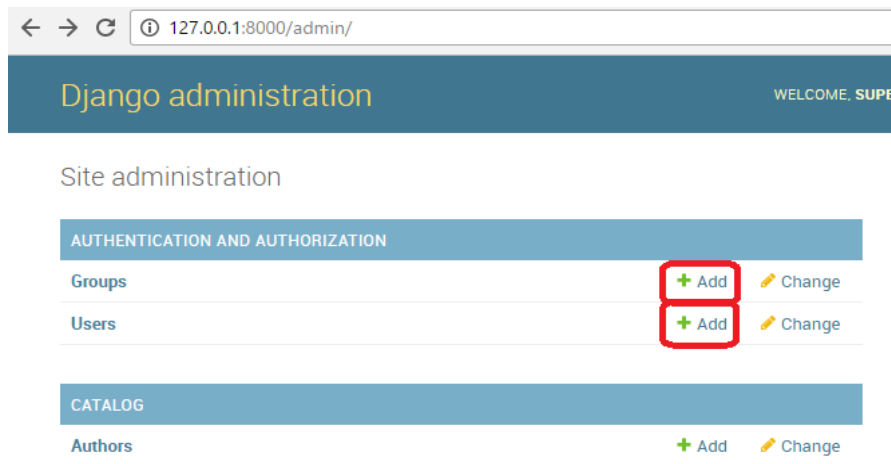
+

  
Select a genre for this book. Hold down "Control", or "Command" on a Mac, to select more than one.

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Desde la interfaz de inicio, y haciendo clic sobre el modelo de Libros, se listan los libros creados. Desde ahí pueden editarse, eliminarse, o agregar uno nuevo. La interfaz del sitio administrativo puede personalizarse según las necesidades del usuario. Algunas de las cosas que se pueden hacer son:

- **Vistas de lista:**
  - Añadir campos, e información adicional desplegada para cada registro.
  - Añadir filtros para seleccionar qué registros se listan, basados en fechas u otros tipos de valores (ej. estado de préstamo del libro).
  - Añadir opciones adicionales al menú **Action** en las vistas de lista, y elegir en qué lugar del formulario se despliega este menú.
- **Vistas de detalle:**
  - Elegir qué campos desplegar (o excluir), junto con su orden, agrupamiento, si son editables, el tipo de control a usarse, orientación, entre otros.
  - Añadir campos relacionados a un registro para permitir la edición en cadena; ejemplo: proveer la capacidad de añadir y editar registros de libros mientras estás creando su registro de autor.



## CREANDO USUARIOS EN EL SISTEMA

Para crear usuarios y grupos en el sistema haciendo uso del sitio administrativo de Django, en primer lugar, se debe iniciar el servidor de desarrollo y acceder al sitio de administración a través de la URL configurada. Se ingresa a éste con las credenciales de la cuenta de super usuario.

El nivel superior del sitio de administración "Admin site" muestra todos tus modelos, ordenados por la aplicación por defecto de Django **"django application"**. Desde la sección de Autenticación y Autorización, puedes acceder a los enlaces de Usuarios **"Users"** y Grupos **"Groups"** para ver todos sus registros existentes.

Es importante crear grupos de usuarios para que, al momento de necesitar asignar permisología a varios usuarios, sea más cómodo hacerlo a través del grupo al que pertenecen que individualmente. Por lo tanto, se recomienda crear en primer lugar un grupo de usuarios, para ello se accede a la interfaz a través del botón **"Add"**, seguido de **"Groups"** y se ingresa el nombre escogido para el grupo. Puede ser que se requiera agregar la permisología si ya se tiene determinada, o de lo contrario se guarda el grupo solamente.

Home > Authentication and Authorization > Groups > Add group

## Add group

Name:

Permissions:

Available permissions ⓘ	Chosen permissions ⓘ
<div><input type="text" value="Filter"/></div> <div><ul style="list-style-type: none"><li>admin   log entry   Can add log entry</li><li>admin   log entry   Can change log entry</li><li>admin   log entry   Can delete log entry</li><li>auth   group   Can add group</li><li>auth   group   Can change group</li><li>auth   group   Can delete group</li><li>auth   permission   Can add permission</li><li>auth   permission   Can change permission</li><li>auth   permission   Can delete permission</li><li>auth   user   Can add user</li><li>auth   user   Can change user</li><li>auth   user   Can delete user</li></ul></div>	<div></div> <div><a href="#">Remove all</a></div>

[Choose all ⓘ](#)

Hold down "Control", or "Command" on a Mac, to select more than one.

[Save and add another](#)

[Save and continue editing](#)

[SAVE](#)

Luego de haberlo creado, se accede a la generación del usuario a través de **"Add"**, seguido de **Users** en la página de inicio del sitio administrativo. En la interfaz en cuestión se completa la información solicitada, y se guarda para que el usuario sea creado.

El sitio de administrador creará el nuevo usuario, e inmediatamente muestra la pantalla de **Change user** (Cambios del usuario) donde se pueden realizar cambios relacionados con el nombre de usuario, y agregar información para los campos opcionales. Estos incluyen: el primer nombre, el apellido, la dirección de correo electrónico, los estados de los usuarios y sus permisos (solo el indicador Active debería ser activado). Más abajo se puede especificar los grupos y permisos del usuario, y ver datos importantes relacionados a éste, tanto la fecha en que se incluyó como la fecha del último inicio de sesión.



## Django administration

WELCOME, **SUPERMAN**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) > [Authentication and Authorization](#) > [Users](#) > [Add user](#)

### Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

Password confirmation:

Enter the same password as before, for verification.

[Save and add another](#)[Save and continue editing](#)[SAVE](#)

Posteriormente, en la sección de Grupos, se selecciona el grupo creado anteriormente de la lista de los disponibles, y se agrega a la caja de la derecha correspondiente a los grupos escogidos. Se guarda la acción realizada, y de esta manera concluye la creación de un usuario en el sistema.

Groups:

Available groups ?



Chosen groups ?

Nombre del grupo



Remove all

[Choose all](#) ?

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.