

CUE: EL SITIO ADMINISTRATIVO DE DJANGO I

Pablo Varas Salamanca

06 de noviembre de 2024, 13:23:02

[GitHub del proyecto](#)

REBOUND EXERCISE: GESTIÓN DEL PANEL ADMINISTRATIVO DE DJANGO

EJERCICIO:

Para esta revisión práctica del sitio de administración de Django se debe:

- Autorizar los archivos de configuración para habilitar el sitio administrativo de Django:

<http://localhost:8000/admin>

- Crear un superusuario.
- Registrar la aplicación Book en el sitio administrativo de Django.
- Configurar con la finalidad de que el sitio administrativo aparezca como Libro y Libros,

respectivamente.

- Agregar dos atributos especiales al modelo del tipo fecha: uno que se refiere a la fecha de creación (`fecha_creacion`) el otro a la fecha de modificación (`fecha_modificacion`).

DESARROLLO:

1. Autorizar los archivos de configuración

Verificar que el sitio administrativo de Django esté habilitado en el archivo de configuración del proyecto (settings.py). Comprobar de que 'django.contrib.admin' esté incluido en INSTALLED_APPS.

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'book.apps.BookConfig', # registro de la app en el proyecto
    'bootstrap5', # registro de bootstrap5 en el proyecto
    'crispy_forms', # registro de crispy forms en el proyecto
    'crispy_bootstrap5', # registro de crispy bootstrap5 en el proyecto
]

# https://github.com/django-crispy-forms/crispy-bootstrap5
# https://django-crispy-forms.readthedocs.io/en/latest
CRISPY_ALLOWED_TEMPLATE_PACKS = 'bootstrap5'
CRISPY_TEMPLATE_PACK = 'bootstrap5'

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
]

-:--- settings.py 23% L37 (Django company FlyC- WK SP yas ElDoc)
```

2. Crear un superusuario.

En la consola:

```
1 python manage.py createsuperuser
```

```
(pablo-env) pablo@debian:/media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S9/rebound2/book_/site_django$ py
thon manage.py createsuperuser
Nombre de usuario (leave blank to use 'pablo'): pablo
Dirección de correo electrónico: admin@gmail.com
Password:
Password (again):
La contraseña es demasiado similar a la de dirección de correo electrónico.
Esta contraseña es demasiado común.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

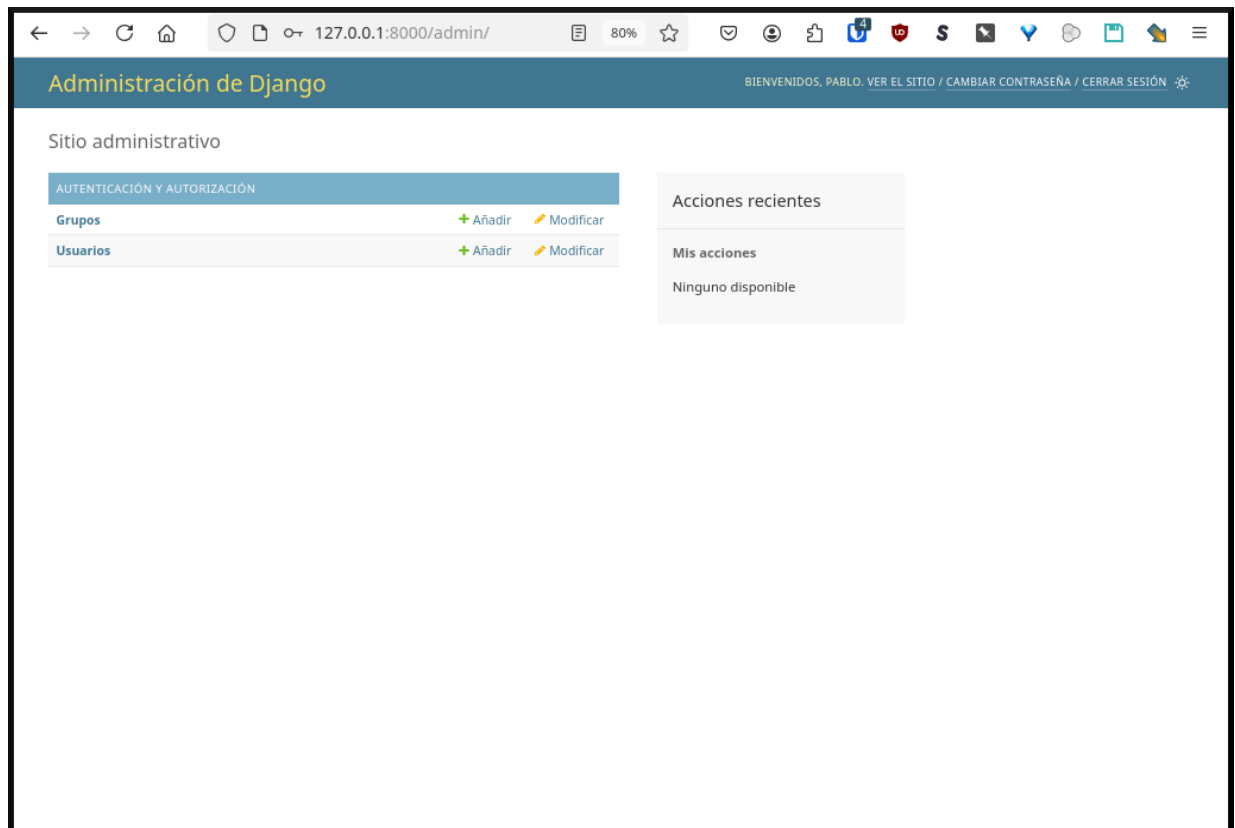
```
from django.contrib import admin
from .models import Book

# Register your models here.
# https://docs.djangoproject.com/en/5.1/ref/contrib/admin/#django.contrib.admin.ModelAdmin

admin.site.register(Book) # registro de los model separados por coma

-:--- admin.py All L1 (Django company FlyC- WK SP yas ElDoc)
```

3. Ingresar a la ruta del sitio administrativo en <http://127.0.0.1:8000/admin>



4. Registrar la aplicación Book

Revisar que en admin.py la el modelo Book esté registrada

- En el archivo admin.py de la aplicación Book, importar el modelo Book.
- Registrar el modelo en el sitio administrativo de Django:

```
1 from django.contrib import admin
2 from .models import Book
3
4 admin.site.register(Book)
```

```
from django.contrib import admin
from .models import Book

# Register your models here.
# https://docs.djangoproject.com/en/5.1/ref/contrib/admin/#django.contrib.admin.ModelAdmin

admin.site.register(Book) # registro de los model separados por coma

-:--- admin.py All L1 (Django company FlyC- WK SP yas ElDoc)
```

5. Configurar con la finalidad de que el sitio administrativo aparezca como Libro y Libros, respectivamente.

>app/admin.py

```
1 class Book(models.Model):
2     # campos del modelo
3
4     class Meta:
5         verbose_name = "Libro"
6         verbose_name_plural = "Libros"
```



```
class Meta:
    verbose_name = 'Libro'
    verbose_name_plural = 'Libros'
    permissions = [
        # ('permiso', 'descripcion')
        # https://docs.djangoproject.com/en/5.1/topics/auth/customizing/#custom-permissions
        ('book.view', 'Puede ver los libros'),
        ('development', 'Permiso como Desarrollador'),
        ('scrum_master', 'Permiso como Scrum Master'),
        ('product_owner', 'Permiso como Product Owner')
    ]
```

U:--- models.py 20% L7 (Django company FlyC- WK SP yas ElDoc)

6. Agregar dos atributos especiales al modelo del tipo fecha: uno que se refiere a la fecha de creación (fecha_creacion), y el otro a la fecha de modificación (fecha_modificacion).

>app/models.py

```
1 from django.db import models
2
3 class Book(models.Model):
4     # campos existentes
5
6     fecha_creacion = models.DateTimeField(auto_now_add=True)
7     fecha_modificacion = models.DateTimeField(auto_now=True)
```

```

from django.db import models
import datetime # paquete para manejar fechas

# Create your models here.
# https://docs.djangoproject.com/en/5.1/topics/db/models/
# https://www.w3schools.com/django/django_models.php
class Book(models.Model):
    # atributos
    titulo = models.CharField(max_length = 100, null = False)
    autor = models.CharField(max_length = 50, null = False)
    valoracion = models.IntegerField(help_text = 'Valoración entre 0 y 100')
    fecha_creacion = models.DateTimeField(default=datetime.datetime.now())
    fecha_modificacion = models.DateTimeField(default=datetime.datetime.now())

    class Meta:
        verbose_name = 'Libro'
        verbose_name_plural = 'Libros'
        permissions = [
            # ('permiso', 'descripcion')

```

U:--- models.py Top L13 (Django company FlyC- WK SP yas ElDoc)

7. Modificamos la función crear_libro en views.py para que guarde la fecha en que se modifica un libro

```

1 book = form.save(commit=False) # preguardado de los datos
2 book.fecha_modificacion = datetime.datetime.now() # guarda la fecha de modificación
3 book.save() # guarda los datos en la base de datos

```

```

@login_required
def editar_libro(request, libro_id):
    book = get_object_or_404(Book, pk = libro_id) # obteniendo el libro a editar en la db o status 404 not found
    if request.method == 'POST':
        form = BookForm(request.POST, instance=book) # instancia del formulario con los datos del libro a editar
        if form.is_valid(): # si el formulario es valido
            book = form.save(commit=False) # preguardado de los datos
            book.fecha_modificacion = datetime.datetime.now() # fecha de modificacion
            book.save() # guardando los datos en la base de datos
            messages.success(request, 'Libro modificado correctamente') # muestra un mensaje
            return redirect('lista_libros') # redirecciona a lista de libros
        else: # si el formulario no es valido
            messages.error(request, 'Modifica los datos de ingreso') # muestra un mensaje
            return HttpResponseRedirect(reverse('editar_libro', args=[book.id])) # redirecciona a la pagina
    else: # si el metodo es GET
        form = BookForm(instance=book) # instancia del formulario con los datos del libro a editar
        return render(request, 'editar_libro.html', {'form': form, 'libro_id': libro_id}) # renderiza la vista
    para editar el libro

```

U:--- views.py 39% L43 (Django company FlyC- WK SP yas ElDoc)

8. Aplicar migraciones ya que se modificó el modelo Book

- Activar el entorno virtual:

```

pablo@debian: /media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S10/rebound/book_/site_django$ source /home/pablo/pablo-env/bin/activate

```

- Aplicar las migraciones luego Ubicarse en la terminal a la altura de manage.py

```

1 python manage.py makemigrations

```

```
(pablo-env) pablo@debian:/media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S10/rebound/book_/site_django$ python manage.py makemigrations
Migrations for 'book':
  book/migrations/0005_alter_book_fecha_creacion_and_more.py
    - Alter field fecha_creacion on book
    - Alter field fecha_modificacion on book
(pablo-env) pablo@debian:/media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S10/rebound/book_/site_django$
```

1 python manage.py migrate

```
(pablo-env) pablo@debian:/media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S10/rebound/book_/site_django$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, book, contenttypes, sessions
Running migrations:
  Applying book.0005_alter_book_fecha_creacion_and_more... OK
(pablo-env) pablo@debian:/media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S10/rebound/book_/site_django$
```

- Iniciar el servidor

1 python manage.py runserver

```
(pablo-env) pablo@debian:/media/DATA/Descargas/BOOTCAMP/Bootcamp/VSC/M6_Django/S10/rebound/book_/site_django$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 06, 2024 - 07:44:33
Django version 4.2.16, using settings 'site_django.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

- Se despliega lo solicitado:

