

# *Actividad: Extensión del proyecto biblioteca usando Django REST Framework*

## *Andrés Felipe Monsalve Pérez - 1152353*

### *Evidencias:*

- *ScreenShots del panel de admin:*

Django administration

Home > Biblioteca > Autores

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

BIBLIOTECA

- Autors [+ Add](#)
- Libros [+ Add](#)
- Resenas [+ Add](#)

«

Select autor to change

Action: ----- Go 0 of 3 selected

<input type="checkbox"/> NOMBRE	NACIONALIDAD	CANTIDAD LIBROS
Dante Alighieri	Italia	3
Homero	Grecia	3
John Green	Estados Unidos	3

3 autores

ADD AUTOR +

Django administration

Home > Biblioteca > Libros

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

BIBLIOTECA

- Autors [+ Add](#)
- Libros [+ Add](#)
- Resenas [+ Add](#)

«

Select libro to change

Action: ----- Go 0 of 9 selected

<input type="checkbox"/> TITULO	AUTOR
EL Teorema de Katherine	John Green
Ciudades de Papel	John Green
Bajo la Misma Estrella	John Green
De Monarchia	Dante Alighieri
Vita Nuova	Dante Alighieri
La Divina Comedia	Dante Alighieri
Himnos Homéricos	Homero
La Odisea	Homero
La Ilíada	Homero

9 libros

ADD LIBRO +

Django administration

Home > Biblioteca > Resenas

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

BIBLIOTECA

- Autors [+ Add](#)
- Libros [+ Add](#)
- Resenas [+ Add](#)

«

Select resena to change

Action: ----- Go 0 of 9 selected

<input type="checkbox"/> LIBRO	CALIFICACION
EL Teorema de Katherine	4.0
Ciudades de Papel	3.0
Bajo la Misma Estrella	5.0
De Monarchia	3.0

ADD RESENA +

## - Ejecutando script de carga

```
(venv) @ pipelinn ~/Documentos/Programacion/Web Projects/Practicas_Django  
/DRF_Actividad_1 [main] >> python3 manage.py shell < poblar_datos.py  
9 objects imported automatically (use -v 2 for details).
```

## - ScreenShots del código

```
18 class Autor(models.Model):  
19     nombre = models.CharField(  
20         max_length=100,  
21         validators=[validate_autor]  
22     )  
23     nacionalidad = models.CharField(max_length=100)  
24  
25     def __str__(self):  
26         return self.nombre  
27  
28 class Libro(models.Model):  
29     titulo = models.CharField(max_length=100)  
30     autor = models.ForeignKey(Autor, on_delete=models.CASCADE, related_name='libros')  
31     fecha_publicacion = models.DateField(null=True, blank=True)  
32     resumen = models.TextField(  
33         validators=[validate_libro]  
34     )  
35  
36     def __str__(self):  
37         return self.titulo  
38  
39 class Resena(models.Model):  
40     libro = models.ForeignKey(Libro, on_delete=models.CASCADE, related_name='resenas')  
41     texto = models.TextField()  
42     calificacion = models.FloatField(  
43         default=0.0,  
44         validators=[  
45             MinValueValidator(0.0),  
46             MaxValueValidator(5.0)  
47         ]
```

```
4     class AutorSerializer(serializers.ModelSerializer):  
5         class Meta:  
6             model = Autor  
7             fields = '__all__'  
8  
9     class ResenaSerializer(serializers.ModelSerializer):  
10        class Meta:  
11            model = Resena  
12            fields = '__all__'  
13  
14    class LibroSerializer(serializers.ModelSerializer):  
15        autor_name = serializers.ReadOnlyField(source="autor.nombre")  
16        recent_views = serializers.SerializerMethodField()  
17  
18        class Meta:  
19            model = Libro  
20            fields = ('titulo', 'autor_name', 'fecha_publicacion', 'resumen', 'recent_views')  
21  
22        def get_recent_views(self, obj):  
23            reviews = obj.resenas.order_by('-fecha')[::2]  
24            return ResenaSerializer(reviews, many=True).data
```

```
1 from rest_framework.decorators import action    No se ha podido resolver la impo
2 from rest_framework.response import Response    No se ha podido resolver la impo
3 from django.db.models import Avg      La importación "django.db.models" no se ha p
4 from rest_framework import viewsets   No se ha podido resolver la importación "re
5 from .models import Autor, Libro, Resena
6 from .serializers import AutorSerializer, LibroSerializer, ResenaSerializer
7
8 # Create your views here.
9 class AutorViewSet(viewsets.ModelViewSet):
10     queryset = Autor.objects.all()
11     serializer_class = AutorSerializer
12     filterset_fields = ['nacionalidad']
13     search_fields = ['nombre', 'nacionalidad']
14     ordering_fields = ['nombre']
15
16 class LibroViewSet(viewsets.ModelViewSet):
17     queryset = Libro.objects.all().select_related('autor')
18     serializer_class = LibroSerializer
19     filterset_fields = ['autor_nombre', 'autor_id', 'fecha_publicacion']
20     search_fields = ['titulo', 'autor_nombre', 'autor_id']
21     ordering_fields = ['titulo', 'fecha_publicacion']
22
23 class ResenaViewSet(viewsets.ModelViewSet):
24     queryset = Resena.objects.all().select_related('libro')
25     serializer_class = ResenaSerializer
26     filterset_fields = ['libro_titulo', 'libro_id', 'calificacion']
27     search_fields = ['libro_titulo', 'libro_id', 'fecha', 'calificacion']
28     ordering_fields = ['fecha', 'calificacion']
29
30     def get_queryset(self):
31         queryset = super().get_queryset()
32         calificacion_min = self.request.query_params.get('calificacion_minima')
33
34         if calificacion_min:
35             queryset = queryset.filter(calificacion__gte=calificacion_min)
36
37         return queryset
```

```
1 from rest_framework import routers    No se ha podido resolver la impo
2 from .views import AutorViewSet, LibroViewSet, ResenaViewSet
3
4 router = routers.DefaultRouter()
5 router.register(r'autores', AutorViewSet)
6 router.register(r'libros', LibroViewSet)
7 router.register(r'resenas', ResenaViewSet)
8
9 urlpatterns = router.urls
```

## - Uso de la API

GET ▾ http://127.0.0.1:8000/api/libros/ Send ▾ 200 OK 18 ms 960 B

Params Body (1) Auth Headers (4) Scripts Doc Preview Headers (10) Cookies Tests 0/0 → Mock Console

URL PREVIEW  
http://127.0.0.1:8000/api/libros/

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value

Preview ▾

```
1 {  
2   "count": 9,  
3   "next": "http://127.0.0.1:8000/api/libros/?page=2",  
4   "previous": null,  
5   "results": [  
6     {  
7       "titulo": "La Iliada",  
8       "autor_name": "Homero",  
9       "fecha_publicacion": "0800-01-01",  
10      "resumen": "Relato épico de la guerra de Troya y el papel de Aquiles.",  
11      "recent_views": [  
12        {  
13          "id": 1,  
14          "texto": "Una obra épica impresionante y muy profunda.",  
15          "calificacion": 5.0,  
16          "fecha": "2025-11-22T23:30:03.258975Z",  
17          "libro": 1  
18        }  
19      ]  
20    }  
21  ]  
22 }
```

POST My first request × +

POST ▾ http://127.0.0.1:8000/api/autores/ Send ▾ 201 Created 22 ms 62 B

Params Body (1) Auth Headers (4) Scripts Doc Preview Headers (10) Cookies Tests 0/0

JSON ▾

Preview ▾

```
1 {  
2   "nombre": "Andrés Monsalve",  
3   "nacionalidad": "Colombia"  
4  
5 }
```

```
1 {  
2   "id": 4,  
3   "nombre": "Andrés Monsalve",  
4   "nacionalidad": "Colombia"  
5 }
```

GET ▾ http://127.0.0.1:8000/api/autores/ Send ▾ 200 OK 12 ms 264 B

Params Body (1) Auth Headers (4) Scripts Doc Preview Headers (10) Cookies Tests 0/0 → Mock Console

URL PREVIEW  
http://127.0.0.1:8000/api/autores/

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value

PATH PARAMETERS

Path parameters are url path segments that start with a colon

Preview ▾

```
1 {  
2   "count": 4,  
3   "next": "http://127.0.0.1:8000/api/autores/?page=2",  
4   "previous": null,  
5   "results": [  
6     {  
7       "id": 1,  
8       "nombre": "Homero",  
9       "nacionalidad": "Grecia"  
10     },  
11     {  
12       "id": 2,  
13       "nombre": "Dante Alighieri",  
14       "nacionalidad": "Italia"  
15     },  
16     {  
17       "id": 3,  
18       "nombre": "John Green",  
19       "nacionalidad": "Estados Unidos"  
20     }  
21   ]  
22 }
```

GET ▾ http://127.0.0.1:8000/api/libros/?search=Green Send 200 OK 15 ms 1001 B

Params Body Auth Headers 4 Scripts Docs Preview Headers 10 Cookies Tests 0/0 → Mock Console

URL PREVIEW  
http://127.0.0.1:8000/api/libros/?search=Green

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value
------	-------

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id' ×

Preview

```
1▼ {
2   "count": 3,
3   "next": null,
4   "previous": null,
5▼   "results": [
6▼     {
7       "titulo": "Bajo la Misma Estrella",
8       "autor_name": "John Green",
9       "fecha_publicacion": "2012-01-10",
10      "resumen": "Historia de amor entre dos adolescentes que luchan contra el cáncer.",
11      "recent_views": [
12        {
13          "id": 7,
14          "texto": "Emocionante, triste y muy humana.",
15          "calificacion": 5.0,
16          "fecha": "2025-11-22T23:30:03.268066Z",
17          "libro": 7
18        }
19      ],
20    },
21    {
22      "titulo": "Ciudades de Papel",
23      "autor_name": "John Green",
24      "fecha_publicacion": "2008-10-16",
25      "resumen": "Un joven busca a una chica misteriosa que desaparece repentinamente.",
26      "recent_views": [
27        ...
28      ]
29    }
30  ]
31}
```

GET ▾ http://127.0.0.1:8000/api/resenas/promedio/?libro\_id=4 Send 200 OK 17 ms 60 B

Params Body Auth Headers 4 Scripts Docs Preview Headers 10 Cookies Test

URL PREVIEW  
http://127.0.0.1:8000/api/resenas/promedio/?libro\_id=4

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value
------	-------

Preview

```
1▼ {
2   "libro_id": "4",
3   "rating_promedio": {
4     "calificacion_avg": 5.0
5   }
6 }
```

GET ▾ http://127.0.0.1:8000/api/resenas/?calific Send 200 OK 17 ms 434 B

Params Body Auth Headers 4 Scripts Docs Preview Headers 10 Cookies Tests 0/0 → Mock Console

URL PREVIEW  
http://127.0.0.1:8000/api/resenas/?calificacion\_minima=5

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value
------	-------

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id' ×

Preview

```
1▼ {
2   "count": 3,
3   "next": null,
4   "previous": null,
5▼   "results": [
6▼     {
7       "id": 1,
8       "texto": "Una obra épica impresionante y muy profunda.",
9       "calificacion": 5.0,
10      "fecha": "2025-11-22T23:30:03.258975Z",
11      "libro": 1
12    },
13    {
14      "id": 4,
15      "texto": "Una obra maestra de la literatura universal.",
16      "calificacion": 5.0,
17      "fecha": "2025-11-22T23:30:03.263669Z",
18      "libro": 4
19    },
20    {
21      "id": 7,
22      "texto": "Emocionante, triste y muy humana.",
23      "calificacion": 5.0,
24      "fecha": "2025-11-22T23:30:03.268066Z",
25      "libro": 7
26    }
27  ]
28}
```

GET ▾ http://127.0.0.1:8000/api/autores/?nacionalidad=Colombia

Send ▾ 200 OK 10 ms 114 B

Params Body (1) Auth Headers (4) Scripts Docs

Preview Headers (10) Cookies Tests 0 / 0

**URL PREVIEW**  
http://127.0.0.1:8000/api/autores/?nacionalidad=Colombia

**QUERY PARAMETERS**

+ Add Delete all Description

name	value

Import from URL Bulk Edit

**Preview**

```
1▼ {  
2   "count": 1,  
3   "next": null,  
4   "previous": null,  
5   "results": [  
6     {  
7       "id": 4,  
8       "nombre": "Andrés Monsalve",  
9       "nacionalidad": "Colombia"  
10    }  
11  ]  
12 }
```

GET ▾ http://127.0.0.1:8000/api/libros/?search=Divi

Send ▾ 200 OK 15 ms 378 B

Params Body (1) Auth Headers (4) Scripts Docs

Preview Headers (10) Cookies Tests 0 / 0 → Mock Console

**URL PREVIEW**  
http://127.0.0.1:8000/api/libros/?search=Divi

**QUERY PARAMETERS**

+ Add Delete all Description

name	value

Import from URL Bulk Edit

**PATH PARAMETERS**

Path parameters are url path segments that start with a colon ':', e.g. ':id'

**Preview**

```
1▼ {  
2   "count": 1,  
3   "next": null,  
4   "previous": null,  
5   "results": [  
6     {  
7       "titulo": "La Divina Comedia",  
8       "autor_name": "Dante Alighieri",  
9       "fecha_publicacion": "1320-01-01",  
10      "resumen": "Dante viaja por el Infierno, Purgatorio y Paraíso guiado por Virgilio.",  
11      "recent_views": [  
12        {  
13          "id": 4,  
14          "texto": "Una obra maestra de la literatura universal.",  
15          "calificacion": 5.0,  
16          "fecha": "2025-11-22T23:30:03.263669Z",  
17          "libro": 4  
18        }  
19      ]  
20    }  
21  ]  
22 }
```