

```
biblioteca_project > biblioteca > 🗂️ models.py > ...
1  from django.db import models
2  from django.core.validators import MinValueValidator, MaxValueValidator, MinLengthValidator
3  from django.core.exceptions import ValidationError
4  def validate_non_blank(value):
5      if value is None or not str(value).strip():
6          raise ValidationError('El campo no puede estar vacío ni contener sólo espacios.')
7
8  class Autor(models.Model):
9      nombre = models.CharField(max_length=100, validators=[validate_non_blank])
10     nacionalidad = models.CharField(max_length=100)
11
12    def __str__(self):
13        return self.nombre
14
15 class Libro(models.Model):
16     titulo = models.CharField(max_length=100)
17     autor = models.ForeignKey(Autor, on_delete=models.CASCADE, related_name='libros')
18     fecha_publicacion = models.DateField()
19     resumen = models.TextField(validators=[MinLengthValidator(30)])
20
21    def __str__(self):
22        return self.titulo
23
24 class Resena(models.Model):
25     libro = models.ForeignKey(Libro, on_delete=models.CASCADE, related_name='resenas')
26     texto = models.TextField()
27     calificacion = models.FloatField(validators=[MinValueValidator(0.0), MaxValueValidator(5.0)])
28     fecha = models.DateTimeField(auto_now_add=True)
29
30    def __str__(self):
31        return f'{self.libro.titulo} - {self.calificacion}/5'
```

```
biblioteca_project > biblioteca > 🗂️ serializers.py > 🗃️ AutorSerializer
1  from rest_framework import serializers
2  from .models import Autor, Libro, Resena
3  class ResenaSerializer(serializers.ModelSerializer):
4      class Meta:
5          model = Resena
6          fields = ['id', 'texto', 'calificacion', 'fecha']
7  class LibroSerializer(serializers.ModelSerializer):
8      author_name = serializers.ReadOnlyField(source='autor.nombre')
9      recent_reviews = serializers.SerializerMethodField()
10
11    class Meta:
12        model = Libro
13        fields = ['id', 'titulo', 'autor', 'author_name', 'fecha_publicacion', 'resumen', 'recent_reviews']
14
15    def get_recent_reviews(self, obj):
16        reviews = obj.resenas.order_by('-fecha')[:5]
17        return ResenaSerializer(reviews, many=True).data
18
19    def validate_resumen(self, value):
20        if value is None or len(str(value).strip()) < 30:
21            raise serializers.ValidationError('El resumen debe tener al menos 30 caracteres.')
22        return value
23  class AutorSerializer(serializers.ModelSerializer):
24      libros = LibroSerializer(many=True, read_only=True)
25      class Meta:
26          model = Autor
27          fields = ['id', 'nombre', 'nacionalidad', 'libros']
28
29    def validate_nombre(self, value):
30        if value is None or not str(value).strip():
31            raise serializers.ValidationError('El nombre no puede estar vacío ni contener sólo espacios.')
32        return value
```

```
 1  from rest_framework import viewsets, status
 2  from rest_framework.decorators import action
 3  from rest_framework.response import Response
 4  from django.db.models import Avg
 5  from .models import Autor, Libro, Resena
 6  from .serializers import AutorSerializer, LibroSerializer, ResenaSerializer
 7
 8  class AutorViewSet(viewsets.ModelViewSet):
 9      queryset = Autor.objects.all()
10      serializer_class = AutorSerializer
11
12  class LibroViewSet(viewsets.ModelViewSet):
13      queryset = Libro.objects.all()
14      serializer_class = LibroSerializer
15      filterset_fields = [ 'autor', 'fecha_publicacion' ]
16      ordering_fields = [ 'titulo', 'fecha_publicacion' ]
17
18      @action(detail=True, methods=['get'])
19      def average_rating(self, request, pk=None):
20          libro = self.get_object()
21          avg = libro.resenas.aggregate(Avg('calificacion'))['calificacion__avg']
22          return Response({ 'average_rating': avg })
23
24  class ResenaViewSet(viewsets.ModelViewSet):
25      queryset = Resena.objects.all()
26      serializer_class = ResenaSerializer
```

```
biblioteca_project > biblioteca_project > 🗃 urls.py > ...
10  Class-based views
11      1. Add an import: from other_app.views import Home
12      2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19 from rest_framework.routers import DefaultRouter
20 from biblioteca import views
21
22 router = DefaultRouter()
23 router.register(r'autores', views.AutorViewSet)
24 router.register(r'libros', views.LibroViewSet)
25 router.register(r'reseñas', views.ResenaViewSet)
26
27 urlpatterns = [
28     path('admin/', admin.site.urls),
29     path('api/', include(router.urls)),
30 ]
```

```
📝 poblar_datos.py ✘
biblioteca_project > 📝 poblar_datos.py > ...
1 import os
2 import django
3
4 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'biblioteca_project.settings')
5 django.setup()
6
7 from biblioteca.models import Autor, Libro, Resena
8 from django.utils import timezone
9
10 # Crear autores
11 autor1 = Autor.objects.create(nombre="Gabriel García Márquez", nacionalidad="Colombiano")
12 autor2 = Autor.objects.create(nombre="Isabel Allende", nacionalidad="Chilena")
13
14 # Crear libros
15 libro1 = Libro.objects.create(
16     titulo="Cien años de soledad",
17     autor=autor1,
18     fecha_publicacion=timezone.datetime(1967, 6, 5),
19     resumen="Una novela que narra la historia de la familia Buendía en el pueblo ficticio de Macondo."
20 )
21
22 libro2 = Libro.objects.create(
23     titulo="La casa de los espíritus",
24     autor=autor2,
25     fecha_publicacion=timezone.datetime(1982, 1, 1),
26     resumen="Una saga familiar que abarca varias generaciones en un país latinoamericano no identificado."
27 )
```

# API

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups + Add Change

Users + Add Change

BIBLIOTECA

Autors + Add Change

Libros + Add Change

Resenas + Add Change

Recent actions

My actions

None available

Django REST framework

Api Root / Autor List

Autor List

GET /api/autores/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 1,
            "nombre": "Gabriel García Márquez",
            "nacionalidad": "Colombiano"
        },
        {
            "id": 2,
            "nombre": "Isabel Allende",
            "nacionalidad": "Chilena"
        }
    ]
}
```

Raw data HTML form

Nombre

Nacionalidad

POST

Django REST framework

Api Root / Libro List

Libro List

GET /api/libros/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 1,
            "titulo": "Cien años de soledad",
            "autor_id": 1,
            "autor_nombre": "Gabriel García Márquez",
            "fecha_publicacion": "1967-06-05",
            "resumen": "Una novela que narra la historia de la familia Buendía en el pueblo ficticio de Macondo.",
            "recent_reviews": [
                {
                    "id": 2,
                    "texto": "Muy bueno, pero un poco confuso.",
                    "calificacion": 4.0,
                    "fecha": "2025-10-31T01:49:15.214299Z"
                },
                {
                    "id": 1,
                    "texto": "Una obra maestra de la literatura latinoamericana.",
                    "calificacion": 5.0,
                    "fecha": "2025-10-31T01:49:15.207320Z"
                }
            ]
        },
        {
            "id": 2,
            "titulo": "La casa de los espíritus",
            "autor_id": 2,
            "autor_nombre": "Isabel Allende",
            "fecha_publicacion": "1982-01-01",
            "resumen": "Una saga familiar que abarca varias generaciones en un país latinoamericano no identificado."
        }
    ]
}
```

Django REST framework jaderconterras

Api Root / Libro List

## Libro List

GET /api/libros/?autor=1

HTTP 200 OK  
Allow: GET, POST, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
{ "count": 1, "next": null, "previous": null, "results": [ { "id": 1, "titulo": "Cien años de soledad", "autor": "Gabriel García Márquez", "fecha_publicacion": "1967-06-05", "resumen": "Una novela que narra la historia de la familia Buendía en el pueblo ficticio de Macondo.", "recent_reviews": [ { "id": 2, "text": "Muy bueno, pero un poco confuso.", "calificacion": 4.0, "fecha": "2023-10-31T01:49:15.214299Z" }, { "id": 3, "text": "Una obra maestra de la literatura latinoamericana.", "calificacion": 5.0, "fecha": "2023-10-31T01:49:15.287320Z" } ] } ] }
```

Raw data HTML form

Django REST framework jaderconterras

Api Root / Libro List

## Libro List

GET /api/libros/?ordering=-fecha\_publicacion

HTTP 200 OK  
Allow: GET, POST, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
{ "count": 2, "next": null, "previous": null, "results": [ { "id": 2, "titulo": "La casa de los espíritus", "autor": "Isabel Allende", "fecha_publicacion": "1982-01-01", "resumen": "Una saga familiar que abarca varias generaciones en un país latinoamericano no identificado.", "recent_reviews": [ { "id": 3, "text": "Me encantó la historia y los personajes.", "calificacion": 4.5, "fecha": "2023-10-31T01:49:15.228933Z" } ] }, { "id": 1, "titulo": "Cien años de soledad", "autor": "Gabriel García Márquez", "fecha_publicacion": "1967-06-05", "resumen": "Una novela que narra la historia de la familia Buendía en el pueblo ficticio de Macondo.", "recent_reviews": [ { "id": 2, "text": "Muy bueno, pero un poco confuso.", "calificacion": 4.0, "fecha": "2023-10-31T01:49:15.214299Z" } ] } ] }
```

Django REST framework jaderconterras

Api Root / Libro List / Libro Instance / Average rating

## Average rating

GET /api/libros/1/average\_rating/

HTTP 200 OK  
Allow: GET, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
{ "average_rating": 4.5 }
```