

DJANGO REST FRAMEWORK

PROYECTO BIBLIOTECA

Presentado por:

1152358 - Israel Bulla Rey

Capturas del código

Modelos

```
1  from django.db import models
2  from .validators import *
3  from django.core.validators import MinValueValidator, MaxValueValidator
4
5  # Create your models here.
6  class Autor(models.Model):
7      nombre = models.CharField(max_length=100, validators=[validate_blank])
8      nacionalidad = models.CharField(max_length=100)
9
10     def __str__(self):
11         return self.nombre
12
13 class Libro(models.Model):
14     titulo = models.CharField(max_length=100)
15     autor = models.ForeignKey(Autor, on_delete=models.CASCADE, related_name='libros')
16     fecha_publicacion = models.DateField()
17     resumen = models.TextField(validators=[validate_character])
18
19     def __str__(self):
20         return self.titulo
21
22 class Resena(models.Model):
23     libro = models.ForeignKey(Libro, on_delete=models.CASCADE, related_name='resenas')
24     texto = models.TextField()
25     calificacion = models.IntegerField(validators=[validate_calification])
26     fecha = models.DateTimeField(auto_now_add=True)
27     rating = models.FloatField(default=0, validators=[MinValueValidator(0.0), MaxValueValidator(5.0)])
28
29     def __str__(self):
30         return f'{self.libro.titulo} - {self.calificacion}/5'
```

Serializadores

```
from rest_framework import serializers
from .models import *

class AutorSerializer(serializers.ModelSerializer):
    class Meta:
        model = Autor
        fields = '__all__'

class LibroSerializer(serializers.ModelSerializer):
    recent_reviews = serializers.SerializerMethodField()
    author_name = serializers.ReadOnlyField(source='autor.nombre')

    def get_recent_reviews(self, obj):
        reviews = obj.resenas.order_by('-fecha')[:5]
        return ResenaSerializer(reviews, many=True).data

    class Meta:
        model = Libro
        fields = '__all__'

class ResenaSerializer(serializers.ModelSerializer):
    class Meta:
        model = Resena
        fields = '__all__'
```

ViewSets

Inicio hasta AutorViewSet

```
from .models import *
from .serializers import *
from rest_framework import viewsets
from django.db.models import Avg
from rest_framework.decorators import action
from rest_framework.response import Response
from django_filters.rest_framework import DjangoFilterBackend
from rest_framework.filters import OrderingFilter

class AutorViewSet(viewsets.ModelViewSet):
    queryset = Autor.objects.all()
    serializer_class = AutorSerializer
    filter_backends = [DjangoFilterBackend, OrderingFilter]
    #filterset_fields = ['nombre', 'nacionalidad']
    ordering_fields = ['nombre', 'nacionalidad']

    def get_queryset(self):
        queryset = Autor.objects.all()
        nombreautor = self.request.query_params.get('nombre')
        nacionalidad = self.request.query_params.get('nacionalidad')
        if nombreautor:
            queryset = queryset.filter(nombre_icontains=nombreautor)
        if nacionalidad:
            queryset = queryset.filter(nacionalidad_icontains=nacionalidad)
        return queryset

    def perform_create(self, serializer):
        autor = serializer.save()
        print(f'Nuevo autor: {autor.nombre}')
```

LibroViewSet

```
class LibroViewSet(viewsets.ModelViewSet):
    queryset = Libro.objects.all()
    serializer_class = LibroSerializer
    filter_backends = [DjangoFilterBackend, OrderingFilter]
    filterset_fields = ['autor', 'fecha_publicacion']
    ordering_fields = ['titulo', 'fecha_publicacion']

    def get_queryset(self):
        if self.request.query_params.get('reciente'):
            return Libro.objects.order_by('-fecha_publicacion')[10:]
        return Libro.objects.all()

    @action(detail=True, methods=['get'])
    def rating_aprox(self, request, pk=None):
        libro = self.get_object()
        aprox = libro.resenas.aggregate(rating_aprox=Avg('rating'))['rating_aprox']
        return Response({'rating_aprox': aprox})
```

ResenaViewSet

```
class ResenaViewSet(viewsets.ModelViewSet):
    queryset = Resena.objects.all()
    serializer_class = ResenaSerializer
    filter_backends = [DjangoFilterBackend, OrderingFilter]
    filterset_fields = ['calificacion', 'rating']
    ordering_fields = ['calificacion', 'rating']

    def get_queryset(self):
        if self.request.query_params.get('reciente'):
            return Resena.objects.order_by('-fecha')
        return Resena.objects.all()

    def perform_create(self, serializer):
        resena = serializer.save()
        print(f'Reseña creada para el libro: {resena.libro.titulo}')
```

urls

```
"""
URL configuration for biblioteca_project project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.2/topics/http/urls/
Examples:
Function views
    1. Add an import: from my_app import views
    2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
    1. Add an import: from other_app.views import Home
    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""

from django.contrib import admin
from biblioteca.modelviewsets import *
from rest_framework.routers import DefaultRouter
from django.urls import path, include

router = DefaultRouter()
router.register('libros', LibroViewSet)
router.register('autores', AutorViewSet)
router.register('resenas', ResenaViewSet)

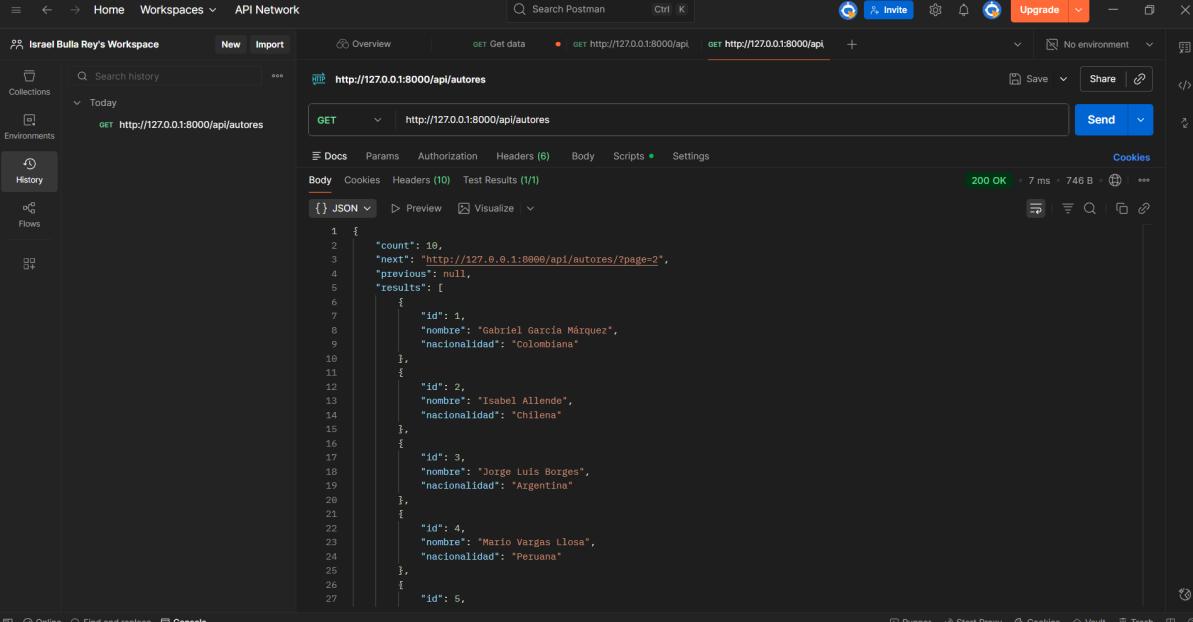
urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include(router.urls)),
]
```

Evidencias del uso de la API

Comprobado mediante el uso de Postman

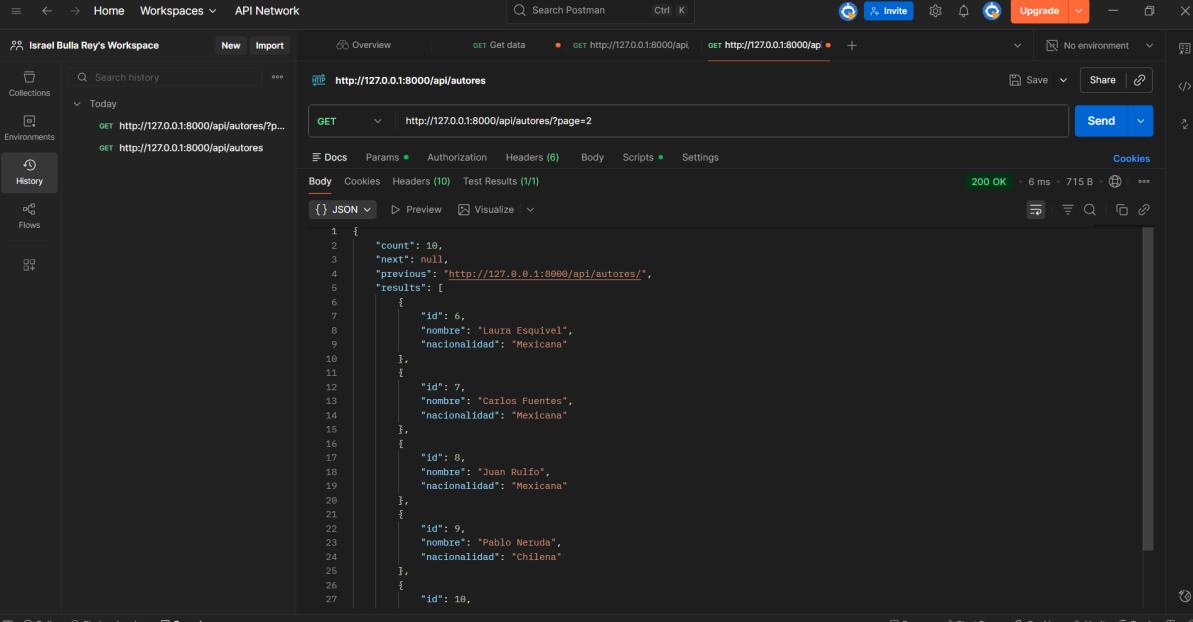
Mostrar autores:

Solo se muestran 5 por página.



The screenshot shows a Postman interface with a collection named "Israel Bulla Rey's Workspace". A GET request is made to `http://127.0.0.1:8000/api/autores`. The response status is 200 OK, with a response time of 7 ms and a size of 746 B. The response body is a JSON object with a "count" field of 10, a "next" field pointing to `http://127.0.0.1:8000/api/autores/?page=2`, and a "previous" field pointing to null. The "results" array contains five objects, each representing an author with fields "id", "nombre", and "nacionalidad".

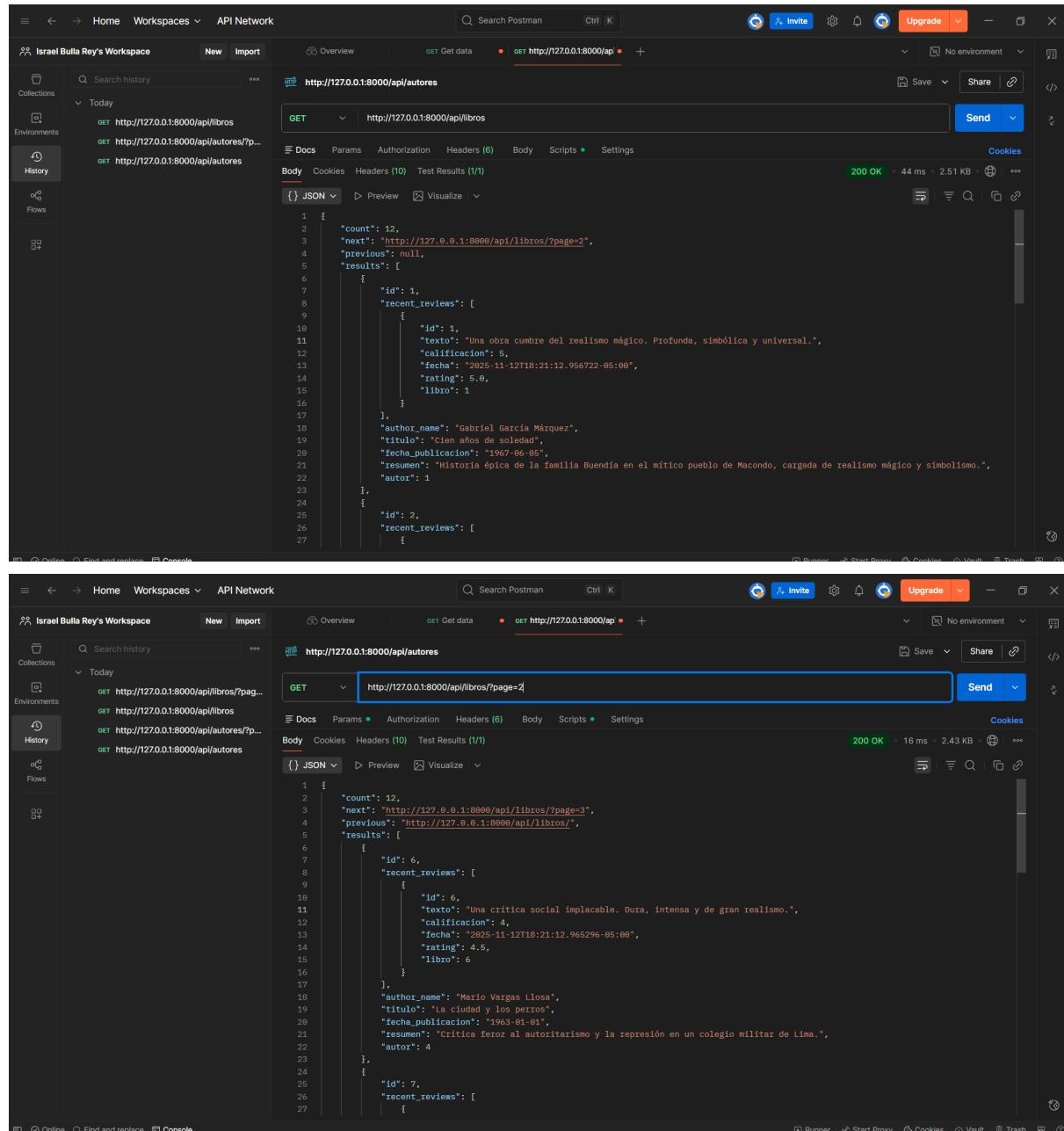
```
1 {
2     "count": 10,
3     "next": "http://127.0.0.1:8000/api/autores/?page=2",
4     "previous": null,
5     "results": [
6         {
7             "id": 1,
8             "nombre": "Gabriel García Márquez",
9             "nacionalidad": "Colombiana"
10        },
11        {
12            "id": 2,
13            "nombre": "Isabel Allende",
14            "nacionalidad": "Chilena"
15        },
16        {
17            "id": 3,
18            "nombre": "Jorge Luis Borges",
19            "nacionalidad": "Argentina"
20        },
21        {
22            "id": 4,
23            "nombre": "Mario Vargas Llosa",
24            "nacionalidad": "Peruana"
25        },
26        {
27            "id": 5,
```



The screenshot shows a Postman interface with a collection named "Israel Bulla Rey's Workspace". A GET request is made to `http://127.0.0.1:8000/api/autores/?page=2`. The response status is 200 OK, with a response time of 6 ms and a size of 715 B. The response body is a JSON object with a "count" field of 10, a "next" field pointing to null, a "previous" field pointing to `http://127.0.0.1:8000/api/autores/`, and a "results" array containing five objects, each representing an author with fields "id", "nombre", and "nacionalidad".

```
1 {
2     "count": 10,
3     "next": null,
4     "previous": "http://127.0.0.1:8000/api/autores/",
5     "results": [
6         {
7             "id": 6,
8             "nombre": "Laura Esquivel",
9             "nacionalidad": "Mexicana"
10        },
11        {
12            "id": 7,
13            "nombre": "Carlos Fuentes",
14            "nacionalidad": "Mexicana"
15        },
16        {
17            "id": 8,
18            "nombre": "Juan Rulfo",
19            "nacionalidad": "Mexicana"
20        },
21        {
22            "id": 9,
23            "nombre": "Pablo Neruda",
24            "nacionalidad": "Chilena"
25        },
26        {
27            "id": 10,
```

Mostrar libros:



The image shows two screenshots of the Postman application interface, both displaying the same API endpoint: `http://127.0.0.1:8000/api/libros`.

Screenshot 1 (Top): This screenshot shows the initial response from the API. The JSON output includes a header with `count: 12`, a `next` URL (`http://127.0.0.1:8000/api/libros/?page=2`), and a `previous` URL (`http://127.0.0.1:8000/api/libros/`). The `results` array contains three book entries, each with an `id` and a `recent_reviews` array. The first book's details are as follows:

```
1 {
2     "count": 12,
3     "next": "http://127.0.0.1:8000/api/libros/?page=2",
4     "previous": null,
5     "results": [
6         {
7             "id": 1,
8             "recent_reviews": [
9                 {
10                     "id": 1,
11                     "texto": "Una obra cumbre del realismo m\u00e1gico. Profunda, simb\u00f3lica y universal.",
12                     "calificacion": 5,
13                     "fecha": "2025-11-12T18:21:12.956722-05:00",
14                     "rating": 5.0,
15                     "libro": 1
16                 ],
17                 "author_name": "Gabriel Garc\u00eda M\u00e1rquez",
18                 "titulo": "Cien a\u00f1os de soledad",
19                 "fecha_publicaci\u00f3n": "1967-06-05",
20                 "resumen": "Historia \u00e9pica de la familia Buend\u00eda en el m\u00f3tico pueblo de Macondo, cargada de realismo m\u00e1gico y simbolismo.",
21                 "autor": 1
22             ],
23             "id": 2,
24             "recent_reviews": [
25                 {
26                     "id": 2,
27                     "recent_reviews": []
28                 }
29             ]
30         }
31     ]
32 }
```

Screenshot 2 (Bottom): This screenshot shows the response for page 2 of the API. The JSON output is identical to the first one, except for the `next` URL, which now points to `http://127.0.0.1:8000/api/libros/?page=3`. The `results` array contains three book entries, each with an `id` and a `recent_reviews` array. The first book's details are as follows:

```
1 {
2     "count": 12,
3     "next": "http://127.0.0.1:8000/api/libros/?page=3",
4     "previous": "http://127.0.0.1:8000/api/libros/",
5     "results": [
6         {
7             "id": 6,
8             "recent_reviews": [
9                 {
10                     "id": 6,
11                     "texto": "Una critica social implacable. Dura, intensa y de gran realismo.",
12                     "calificacion": 4,
13                     "fecha": "2025-11-12T18:21:12.965296-05:00",
14                     "rating": 4.5,
15                     "libro": 6
16                 ],
17                 "author_name": "Mario Vargas Llosa",
18                 "titulo": "La ciudad y los perros",
19                 "fecha_publicaci\u00f3n": "1963-01-01",
20                 "resumen": "Critica feroz al autoritarismo y la represi\u00f3n en un colegio militar de Lima.",
21                 "autor": 4
22             ],
23             "id": 7,
24             "recent_reviews": [
25                 {
26                     "id": 7,
27                     "recent_reviews": []
28                 }
29             ]
30         }
31     ]
32 }
```

Home Workspaces API Network

Overview GET Get data GET http://127.0.0.1:8000/api/autos +

http://127.0.0.1:8000/api/autos

GET http://127.0.0.1:8000/api/libros/?page=3

Docs Params Authorization Headers (6) Body Scripts Settings Cookies

Body Cookies Headers (10) Test Results (1/1)

```
{
  "count": 12,
  "next": null,
  "previous": "http://127.0.0.1:8000/api/libros/?page=2",
  "results": [
    {
      "id": 11,
      "recent_reviews": [
        {
          "id": 11,
          "texto": "Versos de una intensidad emocional incomparable. Neruda en su máxima expresión.",
          "calificacion": 5,
          "fecha": "2025-11-12T10:21:12.975785-05:00",
          "rating": 4.9,
          "libro": 11
        }
      ],
      "author_name": "Pablo Neruda",
      "titulo": "Veinte poemas de amor y una canción desesperada",
      "fecha_publicacion": "1924-01-01",
      "resumen": "Obra poética que exalta el amor, la pasión y la melancolía con versos intensos y sensuales.",
      "autor": 9
    },
    {
      "id": 12,
      "recent_reviews": [
        ...
      ]
    }
  ]
}
```

200 OK 10 ms 1.24 KB

Online Find and replace Console Runner Start Proxy Cookies Vault Trash

Mostrar reseñas:

Home Workspaces API Network

Overview GET Get data GET http://127.0.0.1:8000/api/resenas +

http://127.0.0.1:8000/api/resenas

GET http://127.0.0.1:8000/api/autos/?page=3

GET http://127.0.0.1:8000/api/libros/?page=3

GET http://127.0.0.1:8000/api/autos/?page=3

GET http://127.0.0.1:8000/api/autos

Docs Params Authorization Headers (6) Body Scripts Settings Cookies

Body Cookies Headers (10) Test Results (1/1)

```
{
  "count": 12,
  "next": "http://127.0.0.1:8000/api/resenas/?page=2",
  "previous": null,
  "results": [
    {
      "id": 1,
      "texto": "Una obra cumbre del realismo mágico. Profunda, simbólica y universal.",
      "calificacion": 5,
      "fecha": "2025-11-12T10:21:12.956722-05:00",
      "rating": 5.0,
      "libro": 1
    },
    {
      "id": 2,
      "texto": "Uma historia sobre el amor eterno, contada con una sensibilidad extraordinaria.",
      "calificacion": 5,
      "fecha": "2025-11-12T10:21:12.958429-05:00",
      "rating": 4.8,
      "libro": 2
    },
    {
      "id": 3,
      "texto": "Fascinante mezcla entre historia y fantasía. Narrativa envolvente y emotiva.",
      "calificacion": 4,
      "fecha": "2025-11-12T10:21:12.959961-05:00",
      "rating": 4.6,
      "libro": 3
    }
  ]
}
```

200 OK 27 ms 1.3 KB

Online Find and replace Console Runner Start Proxy Cookies Vault Trash

```

{
  "count": 12,
  "next": "http://127.0.0.1:8000/api/resenas/?page=3",
  "previous": "http://127.0.0.1:8000/api/resenas/",
  "results": [
    {
      "id": 6,
      "texto": "Una critica social implacable. Dura, intensa y de gran realismo.",
      "calificacion": 4,
      "fecha": "2025-11-12T18:21:12.965296-05:00",
      "rating": 4.5,
      "libro": 6
    },
    {
      "id": 7,
      "texto": "Una obra que reinventa la lectura. Genial y desafiante.",
      "calificacion": 5,
      "fecha": "2025-11-12T18:21:12.967362-05:00",
      "rating": 4.7,
      "libro": 7
    },
    {
      "id": 8,
      "texto": "Encantadora y mágica. La combinación entre cocina y emociones es perfecta.",
      "calificacion": 4,
      "fecha": "2025-11-12T18:21:12.969450-05:00",
      "rating": 4.4,
      "libro": 8
    }
  ]
}

```

```

{
  "count": 12,
  "next": null,
  "previous": "http://127.0.0.1:8000/api/resenas/?page=2",
  "results": [
    {
      "id": 11,
      "texto": "Versos de una intensidad emocional incomparable. Neruda en su máxima expresión.",
      "calificacion": 5,
      "fecha": "2025-11-12T18:21:12.975785-05:00",
      "rating": 4.9,
      "libro": 11
    },
    {
      "id": 12,
      "texto": "Oscura y absorbente. Una mirada desgarradora a la mente humana.",
      "calificacion": 4,
      "fecha": "2025-11-12T18:21:12.977360-05:00",
      "rating": 4.8,
      "libro": 12
    }
  ]
}

```

Ordenar por nombre los autores:

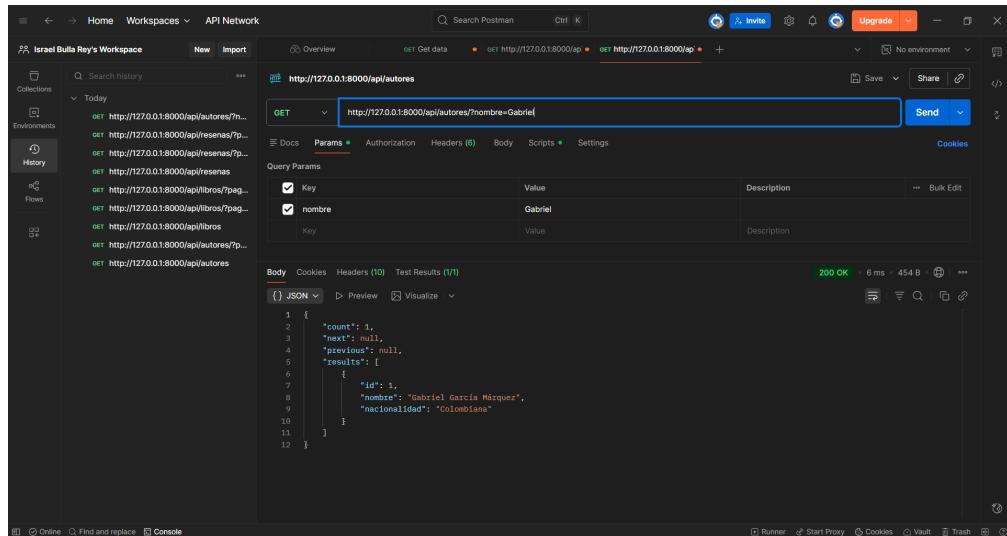
Key	Value	Description
ordering	nombre	

```

{
  "count": 10,
  "next": "http://127.0.0.1:8000/api/autores/?ordering=nombre&page=2",
  "previous": null,
  "results": [
    {
      "id": 7,
      "nombre": "Carlos Fuentes",
      "nacionalidad": "Mexicana"
    },
    {
      "id": 10,
      "nombre": "Ernesto Sabato",
      "nacionalidad": "Argentina"
    },
    {
      "id": 1,
      "nombre": "Gabriel García Márquez",
      "nacionalidad": "Colombiana"
    }
  ]
}

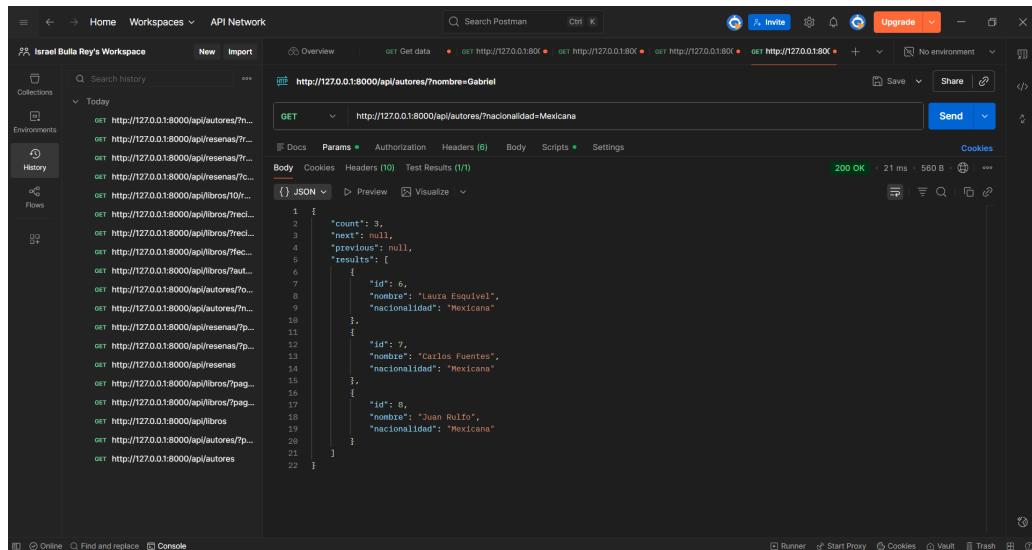
```

Filtrar autores por nombre:



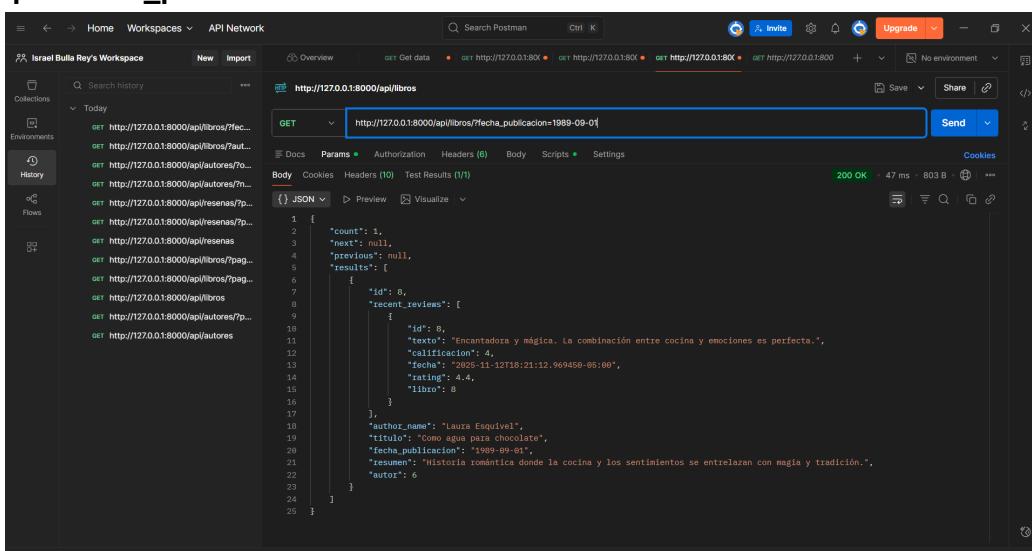
The screenshot shows the Postman interface with a search history entry for 'Israel Bulla Rey's Workspace'. A GET request is made to `http://127.0.0.1:8000/api/autores/?nombre=Gabriel`. The 'Params' tab shows a key 'nombre' with the value 'Gabriel'. The response body is displayed as JSON, showing a single result with id 1, name 'Gabriel Garcia Márquez', and nationality 'Colombiana'.

Filtrar autores por su nacionalidad:



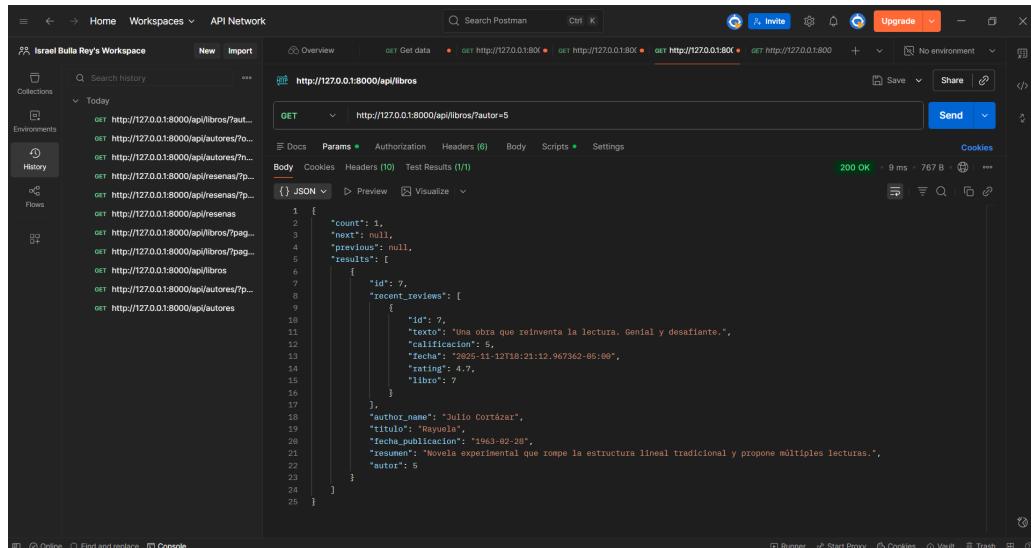
The screenshot shows the Postman interface with a search history entry for 'Israel Bulla Rey's Workspace'. A GET request is made to `http://127.0.0.1:8000/api/autores/?nacionalidad=Mexicana`. The response body is displayed as JSON, showing three results with id 6, 7, and 8, all from Mexico.

Filtrar por fecha_publicacion los libros:



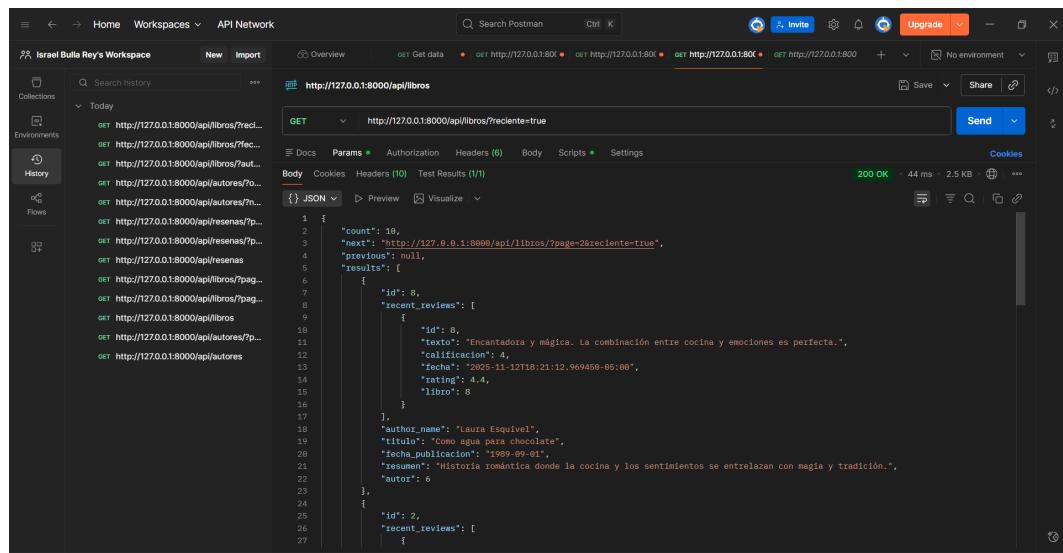
The screenshot shows the Postman interface with a search history entry for 'Israel Bulla Rey's Workspace'. A GET request is made to `http://127.0.0.1:8000/api/libros/?fecha_publicacion=1989-09-01`. The response body is displayed as JSON, showing one result with id 8, title 'Como agua para chocolate', author 'Laura Esquivel', publication date '1989-09-01', summary 'Historia rom\u00e1ntica donde la cocina y los sentimientos se entrelazan con magia y tradici\u00f3n.', rating 4.4, and a recent review with id 8 and text 'Encantadora y m\u00e1gica. La combinaci\u00f3n entre cocina y emociones es perfecta.'

Filtrar libros por el autor (ID)

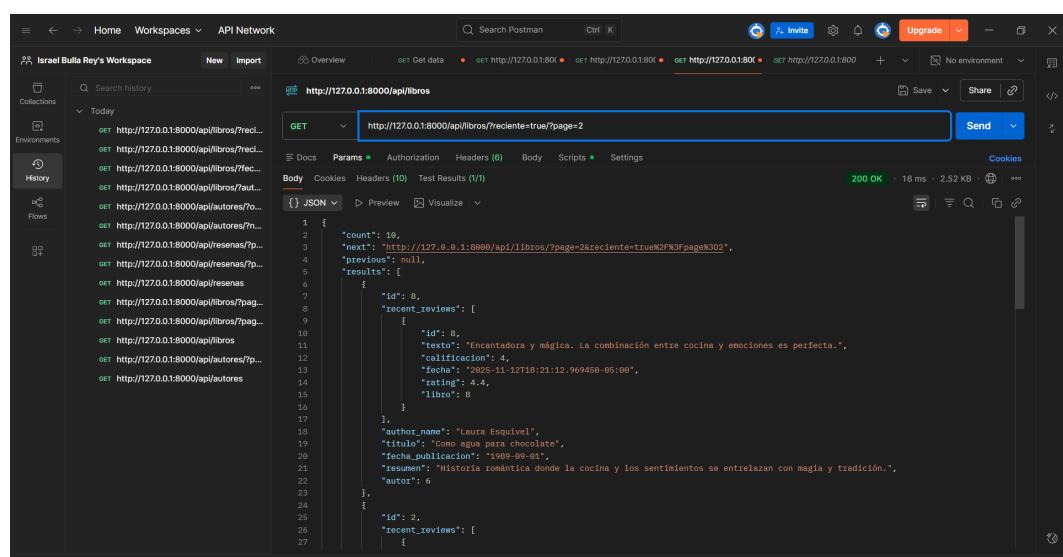


```
1 {
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 7,
      "recent_reviews": [
        {
          "id": 7,
          "texto": "Una obra que reinventa la lectura. Genial y desafiante.",
          "calificacion": 5,
          "fecha": "2025-11-12T18:21:12.967362-06:00",
          "rating": 4.7,
          "libro": 7
        }
      ],
      "author_name": "Julio Cortázar",
      "titulo": "Rayuela",
      "fecha_publicacion": "1963-02-28",
      "resumen": "Novela experimental que rompe la estructura lineal tradicional y propone múltiples lecturas.",
      "autor": 5
    }
  ]
}
```

Mostrar los libros más recientes por fecha de publicación:

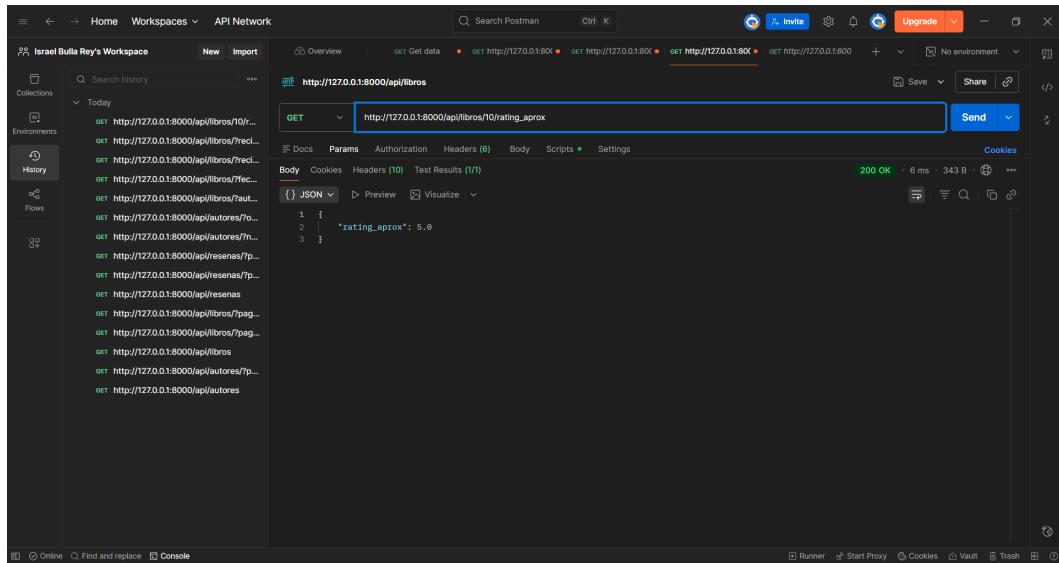


```
1 {
  "count": 10,
  "next": "http://127.0.0.1:8000/api/libros/?page=2&reciente=true",
  "previous": null,
  "results": [
    {
      "id": 9,
      "recent_reviews": [
        {
          "id": 8,
          "texto": "Encantadora y mágica. La combinación entre cocina y emociones es perfecta.",
          "calificacion": 4,
          "fecha": "2025-11-12T18:21:12.969458-06:00",
          "rating": 4.4,
          "libro": 8
        }
      ],
      "author_name": "Laura Esquivel",
      "titulo": "Como agua para chocolate",
      "fecha_publicacion": "1989-09-01",
      "resumen": "Historia romántica donde la cocina y los sentimientos se entrelazan con magia y tradición.",
      "autor": 6
    },
    {
      "id": 2,
      "recent_reviews": [
        {
          "id": 1,
          "texto": "Encantadora y mágica. La combinación entre cocina y emociones es perfecta.",
          "calificacion": 4,
          "fecha": "2025-11-12T18:21:12.969458-06:00",
          "rating": 4.4,
          "libro": 8
        }
      ],
      "author_name": "Laura Esquivel",
      "titulo": "Historia romántica",
      "fecha_publicacion": "2025-11-12T18:21:12.969458-06:00",
      "resumen": "Historia romántica donde la cocina y los sentimientos se entrelazan con magia y tradición.",
      "autor": 6
    }
  ]
}
```



```
1 {
  "count": 10,
  "next": "http://127.0.0.1:8000/api/libros/?page=3&reciente=true",
  "previous": "http://127.0.0.1:8000/api/libros/?page=1&reciente=true",
  "results": [
    {
      "id": 9,
      "recent_reviews": [
        {
          "id": 8,
          "texto": "Encantadora y mágica. La combinación entre cocina y emociones es perfecta.",
          "calificacion": 4,
          "fecha": "2025-11-12T18:21:12.969458-06:00",
          "rating": 4.4,
          "libro": 8
        }
      ],
      "author_name": "Laura Esquivel",
      "titulo": "Como agua para chocolate",
      "fecha_publicacion": "1989-09-01",
      "resumen": "Historia romántica donde la cocina y los sentimientos se entrelazan con magia y tradición.",
      "autor": 6
    },
    {
      "id": 2,
      "recent_reviews": [
        {
          "id": 1,
          "texto": "Encantadora y mágica. La combinación entre cocina y emociones es perfecta.",
          "calificacion": 4,
          "fecha": "2025-11-12T18:21:12.969458-06:00",
          "rating": 4.4,
          "libro": 8
        }
      ],
      "author_name": "Laura Esquivel",
      "titulo": "Historia romántica",
      "fecha_publicacion": "2025-11-12T18:21:12.969458-06:00",
      "resumen": "Historia romántica donde la cocina y los sentimientos se entrelazan con magia y tradición.",
      "autor": 6
    }
  ]
}
```

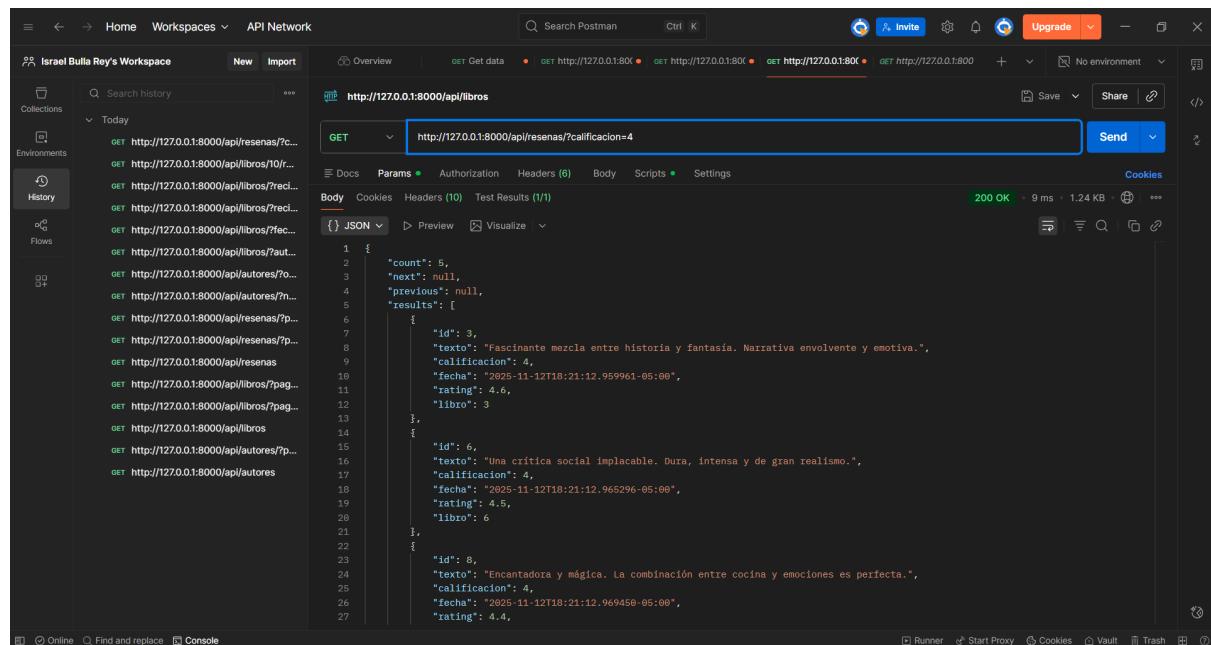
Mostrar el rating_aprox de un libro:



The screenshot shows the Postman interface with a collection named "Israel Bulla Rey's Workspace". A GET request is made to `http://127.0.0.1:8000/api/libros/10/rating_aprox`. The response status is 200 OK, and the body contains the following JSON:

```
{ "rating_aprox": 5.0 }
```

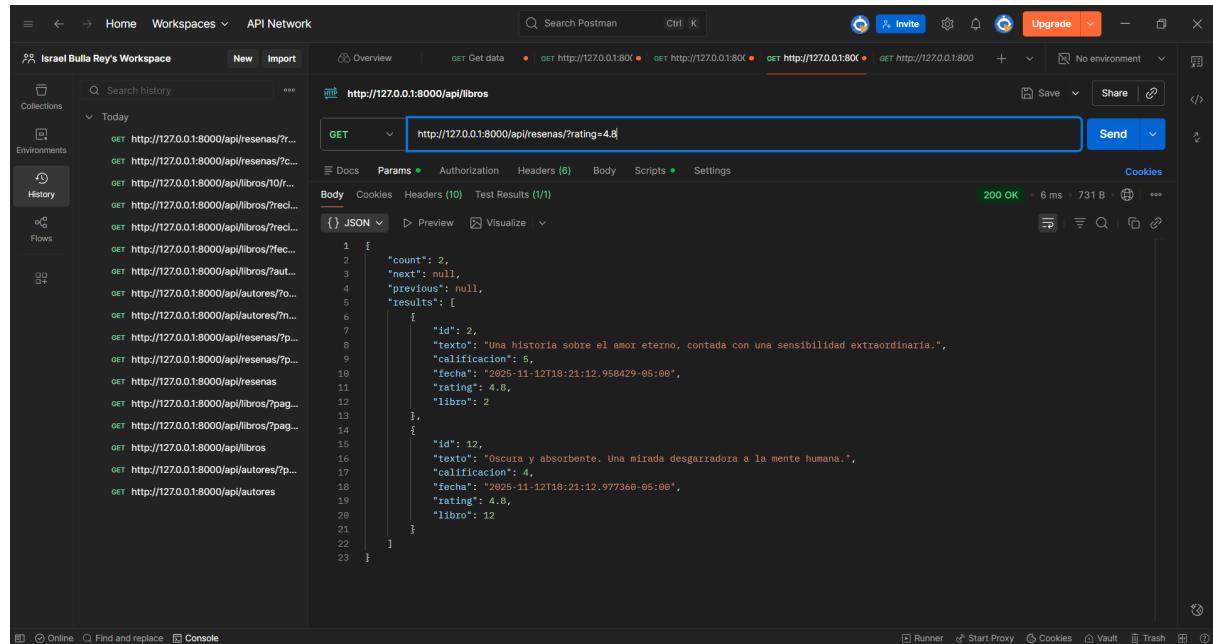
Filtrar las reseñas de libros con calificación 4:



The screenshot shows the Postman interface with a collection named "Israel Bulla Rey's Workspace". A GET request is made to `http://127.0.0.1:8000/api/resenas?calificacion=4`. The response status is 200 OK, and the body contains the following JSON:

```
{ "count": 5, "next": null, "previous": null, "results": [ { "id": 3, "texto": "Fascinante mezcla entre historia y fantas\u00eda. Narrativa envolvente y emotiva.", "calificacion": 4, "fecha": "2025-11-12T18:21:12.959961-05:00", "rating": 4.0, "libro": 3 }, { "id": 6, "texto": "Una cr\u00f3tica social implacable. Dura, intensa y de gran realismo.", "calificacion": 4, "fecha": "2025-11-12T18:21:12.965296-05:00", "rating": 4.5, "libro": 6 }, { "id": 8, "texto": "Encantadora y m\u00e1gica. La combinaci\u00f3n entre cocina y emociones es perfecta.", "calificacion": 4, "fecha": "2025-11-12T18:21:12.969450-05:00", "rating": 4.4, "libro": 8 } ] }
```

Filtrar las reseñas de libros con rating 4.8:



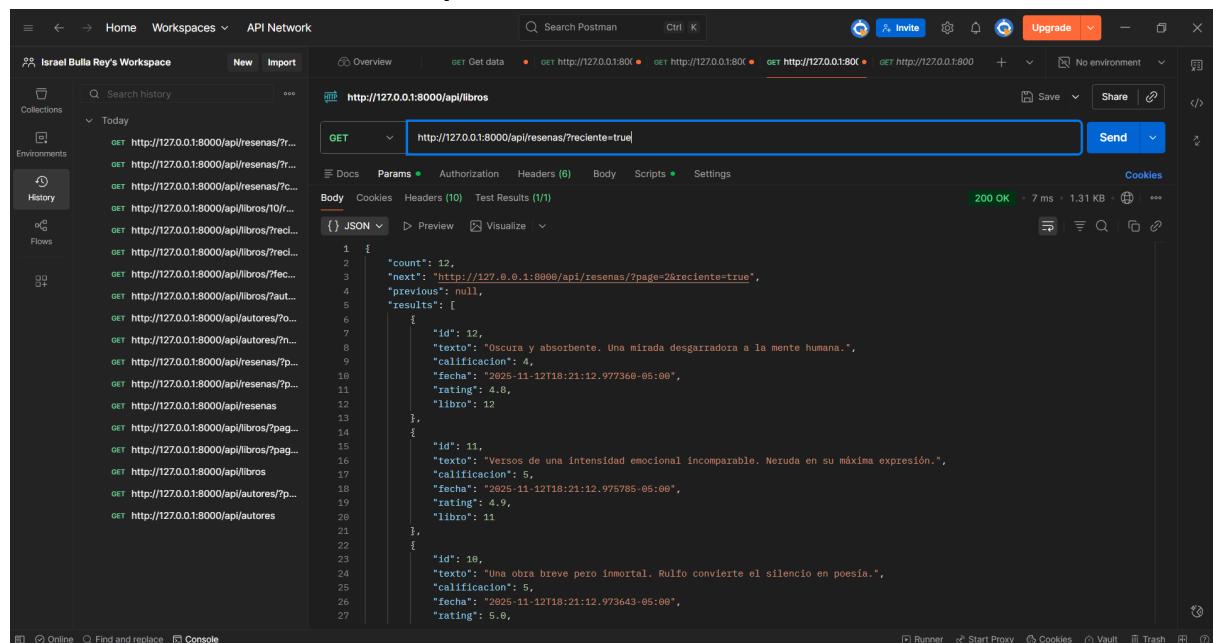
POSTMAN API Network Overview GET Get data GET http://127.0.0.1:8000 GET http://127.0.0.1:8000 GET http://127.0.0.1:8000 GET http://127.0.0.1:8000

GET http://127.0.0.1:8000/api/reseñas/?rating=4.8

200 OK 6 ms 731 B

```
{ "count": 2, "next": null, "previous": null, "results": [ { "id": 2, "texto": "Una historia sobre el amor eterno, contada con una sensibilidad extraordinaria.", "calificación": 5, "fecha": "2025-11-12T18:21:12.958429-05:00", "rating": 4.8, "libro": 2 }, { "id": 12, "texto": "Oscura y absorbente. Una mirada desgarradora a la mente humana.", "calificación": 4, "fecha": "2025-11-12T18:21:12.977360-05:00", "rating": 4.8, "libro": 12 } ] }
```

Mostrar las últimas reseñas de personas:



POSTMAN API Network Overview GET Get data GET http://127.0.0.1:8000 GET http://127.0.0.1:8000 GET http://127.0.0.1:8000

GET http://127.0.0.1:8000/api/reseñas/?reclente=true

200 OK 7 ms 1.31 KB

```
{ "count": 2, "next": "http://127.0.0.1:8000/api/reseñas?page=2&reclente=true", "previous": null, "results": [ { "id": 12, "texto": "Oscura y absorbente. Una mirada desgarradora a la mente humana.", "calificación": 4, "fecha": "2025-11-12T18:21:12.977360-05:00", "rating": 4.8, "libro": 12 }, { "id": 11, "texto": "Versos de una intensidad emocional incomparable. Neruda en su máxima expresión.", "calificación": 5, "fecha": "2025-11-12T18:21:12.975785-05:00", "rating": 4.9, "libro": 11 } ] }
```