



## Prueba para desarrollador BackEnd - Django 4, Postgres 13

Desarrollar una plataforma que almacena información sobre películas. Para todo el ejercicio se sugiere utilizar djangorestframework.

## Sección 1

1. Generar los servicios para realizar las siguientes operaciones:

```
[get] /movies -- get all movies
[get] /movies/:id -- get one movie
[delete] /movies/:id -- delete one movie
[get] /search?query={query} -- get movies whose titulo, calificacion or pais be like {query}
[put] /movies/:id -- modify one movie
[post] /movies -- create one movie
```

2. Bonus. Desarrollar funcionalidad de filtrado, paginación y ordenamiento. [get] /movies?' + query params -- get filtered movies

## Por ejemplo:

/movies?titulo=rodeo&calificacion=3&pais=colom&page=2&page\_size=10&ordering=titulo Debe retornar todas las peliculas que contengan la palabra rodeo en el título y colom en el país de calificación 3, ordenadas por título y en su segunda página.

3. Queries de resumen

```
[get] /summary -- Obtener un conteo de cuantas películas son de cada país y cuántas de cada calificación por rangos.
[get] /top -- Retornar las 5 películas mejor rankeadas.
```

7. La información de las películas se retornará con la siguiente estructura:

8. El servicio del /summary retornará la siguiente estructura {





9. Se debe conectar a una base de datos Postgres 13 o mayor

## Sección 2

Para todo el ejercicio se sugiere utilizar djangorestframework-gis.

1. Desarrolle un servicio que devuelva un geojson con los puntos geográficos de los paises de las películas que estén en la base de datos. Este geojson también debe incluir los nombres y calificaciones de las películas.

```
el endpoint debe ser:
[get] /geojson
body de respuesta:
 "type": "FeatureCollection",
 "features": [
       "type": "Feature",
       "properties": {
               "titulo": "El rodeo",
               "calificación": 2,
               "país": "Colombia"
               },
               "geometry": {
                       "type": "Point",
                       "coordinates": [
                               -74.06982421875,
                               4.631179340411012
                       ]
               }
       },
```





**Nota**: Tenga en cuenta que no importa que no cumpla con todas las características. Por favor envíe sus resultados hasta donde haya podido llegar.

**Nota 2**: Los servicios deben ser accesibles desde un Browser y no solo desde postman. Evitar el error de "CORS not allowed". Ya que el ejercicio se probará con un frontEnd ya desarrollado.

Nota 3: El código debe entregarse en un repositorio git ya sea de gitlab o de github.