PRUEBA DE CÓDIGO

Dada una API externa (unsplash.com) a la que llamaremos para traer una colección de datos, crear un servicio web con un parámetro que actuará de filtrado sobre los datos obtenidos de dicha API.

La API externa pertenece a la empresa ***UNSPLASH***, el usuario es *vivelibre* y la contraseña *v1v3l1br3*. Se necesita crear un token de acceso y llamar al siguiente *endpoint*:

https://unsplash.com/documentation#list-collections

List collections

Get a single page from the list of all collections.

```
GET /collections
```

Parameters

param	Description
page	Page number to retrieve. (Optional; default: 1)
per_page	Number of items per page. (Optional; default: 10)

Response

```
200 OK
Link: <https://api.unsplash.com/collections?page=8>; rel="last", <https://api.unsplash.co
X-Ratelimit-Limit: 1000
X-Ratelimit-Remaining: 999
```


```
[
  {
    "id": 296,
    "title": "I like a man with a beard.",
    "description": "Yeah even Santa...",
    "published_at": "2016-01-27T18:47:13-05:00",
    "last_collected_at": "2016-06-02T13:10:03-04:00",
    "updated_at": "2016-07-10T11:00:01-05:00",
    "total_photos": 12,
    "private": false,
    "share_key": "312d188df257b957f8b86d2ce20e4766",
    "cover_photo": {
      "id": "C-mxLOk6ANs",
      "width": 5616,
```

Se requiere crear un servicio montado sobre springboot 2 y JAVA 11 que recupere la colección de la API externa y se filtren los datos obtenidos en función del parámetro *filter.* Este campo tiene que ser de tipo *string* y ser capaz de hacer un *like* sobre los campos *id, title, description* y *cover_photo.id.* El resultado debe ser presentado siguiendo el formato descrito a continuación:

Filtrar colección externa de datos

Recuperar una lista con todos los datos filtrados:

```
GET /collection/all
```

Parámetros

param	Descripción
`filter`	Filtro aplicado. (Optional; default: null)

Response

```
200 OK

[
  {
    "id": 296,
    "title": "I like a man with a beard.",
    "description": "Yeah even Santa...",
    "cover_photo_id": "C-mxL0k6ANs",
  },
  // ... más colecciones ...
]
```

El servicio web tiene que ser presentado para su prueba en una imagen de Docker subida a Dockerhub.

Por otra parte, el código debe estar subido a un repositorio de Github; por lo que el resultado constará de dos únicas URLs:

- Github con el código
- Dockerhub con la imagen del servicio

Se valorarán los siguientes aspectos:

- Arquitectura → Uso de patrón CLEAN
- Programación → Funciones propias de Java 11
- Integración con Docker