

Practica 2

Índice

Fue complicado y me llevó como 2hrs comprender a un nivel suficiente que era Elasticsearch y como es que se utilizaba dentro de su nube. Ya que en alguna documentación inicial que encontré, mostraban pantallas de otras versiones y no lograba llegar a dichas ventanas. Afortunadamente, encontré un video del equipo de Elasticsearch donde mostraban un recorrido breve y los comandos básicos, lo cual me ayudó a comprender mejor esta tecnología y mi desplazamiento por la aplicación comenzó a mejorar.

Mi código lo logré gracias a la documentación de Elasticsearch, y específicamente en las siguientes pantallas.

Create index API

Creates a new index.

```
PUT /my-index-000001
```

Get mapping API

Retrieves mapping definitions for one or more indices.

```
GET /my-index-000001/_mapping
```

Put index template API

Creates or updates an index template. Index templates define the settings for new indices.

```
PUT /_index_template/template_1
{
  "index_patterns" : ["te*"],
  "priority" : 1,
  "template": {
    "settings" : {
      "number_of_shards" : 2
    }
  }
}
```

Bulk API

Performs multiple indexing or delete operations in a single API call. This reduces overhead and increases speed.

```
POST _bulk
{ "index" : { "_index" : "test", "_id" : "1" } }
{ "field1" : "value1" }
{ "delete" : { "_index" : "test", "_id" : "2" } }
{ "create" : { "_index" : "test", "_id" : "3" } }
{ "field1" : "value3" }
{ "update" : { "_id" : "1", "_index" : "test" } }
{ "doc" : { "field2" : "value2" } }
```

Búsquedas

Las búsquedas las logre, en gran parte al video introductorio que mencioné anteriormente <https://www.elastic.co/es/webinars/getting-started-elasticsearch> y con apoyo de la documentación para secciones específicas como cuando se tenía que sacar el TOP 3.

Paginate search results

```
GET /_search
{
  "from": 5,
  "size": 20,
  "query": {
    "match": {
      "user.id": "kimchy"
    }
  }
}
```

anime_data*

This page lists every field in the **anime_data*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#)

Fields (31) Scripted fields (0) Source filters (0)

All field types ▾

Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	
animeRel.name	string		●		
animeRel.name.keyword	string		●	●	
animeRel.slug	string		●		
animeRel.slug.keyword	string		●	●	
animeRel.type	string		●		

Rows per page: 10 ▾

< 1 2 3 4 >



anime_data*

Data

Options

Metrics

> Tag size Count

Buckets

Tags

Aggregation

Terms

Field

genre.keyword

Order by

Alphabetical

Order

Descending

Size

100

☐

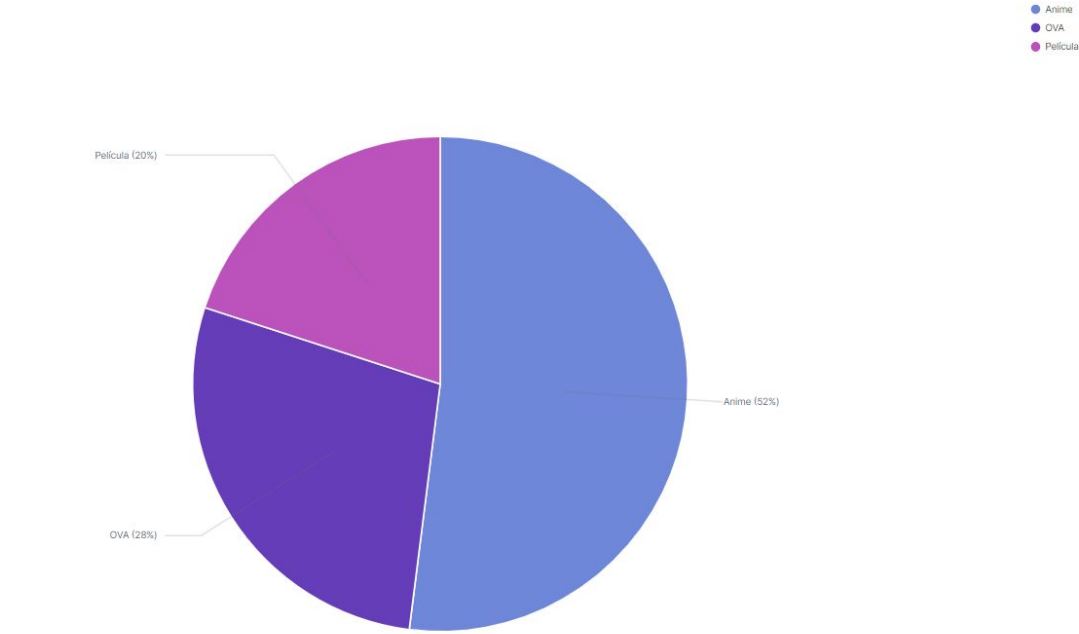
Group other values in separate bucket

☐

Show missing values

Custom label

> Advanced



anime_data*

DataOptions

Metrics

▼ Slice size

Aggregation

Count

Custom label

> Advanced

Buckets

▼ Split slices

Aggregation

Terms

Field

type.keyword

Order by

Metric: Count

Order

Descending

Size

5

☒ Group other values in separate bucket

Label for other bucket

Other

☐ Show missing values

Custom label



anime_data*

DataMetrics & axesPanel settings

Metrics

> Y-axis Count

+ Add

Buckets

☒ X-axis

Aggregation

Filters

Filter 1

type.keyword : "OVA"

+ Add filter

> Advanced

> Split series votes ranges

+ Add



anime_data*

DataMetrics & axesPanel settings

Metrics

> Y-axis Count

+ Add

Buckets

> X-axis filters

▼ Split series

Sub aggregation

Range

Field

votes

≥ 0

→

< 100

≥ 100

→

< 200

≥ 200

→

< +∞

+ Add range

Tablero

Esta parte no siento que me costara tanto trabajo, fue ya muy intuitiva después de estar probando tanto con las visualizaciones.

