

Elaborado por: Alejandro Viveros Sánchez
Fecha de Entrega: 30 septiembre 2020
Título: Elasticsearch

Contenido

Objetivo:.....	2
Introducción:	2
Las ventajas de Elastic	2
Inconvenientes.....	2
Conceptos sobre Elastic	2
Elaboración de la practica:	2
Consultas:	5
Desarrollando las gráficas:	9
Errores:	12
Ejecución de la practica:.....	13
Conclusiones:	13

Objetivo:

Identificar las habilidades de autoaprendizaje y resolución de problemas, a través de la exploración de una tecnología relativamente nueva llamada “Elasticsearch”.

Introducción:

Elasticsearch es un motor de búsqueda, basado en Lucene, búsquedas con alto rendimiento, y basado en índices invertidos

Es una forma más eficiente de hacer búsquedas, toma una palabra y la tokeniza, cada una de las palabras a nuestras palabras, y así haya más rápido. Hasta se puede obtener la posición de la palabra en el texto.

Las ventajas de Elastic:

- Velocidad de búsqueda.
- Facilidad para aplicar algoritmos de relevancia
- Analizadores de texto

Inconvenientes:

- La velocidad de indexación (solución analizar el texto a meter)
- Búsquedas más costosos (depende lo que busquemos).

ElasticSearch cuenta con varias herramientas: Kibana, ElasticSearch, y Logstack

Conceptos sobre Elastic:

- Settings: Es donde se configuran las características específicas.
- Mapping Es la definición sobre los modelos de los datos en un índice. Cada campo debe de tener un tipo, propiedad y analizadores.
- Analizadores de texto: Es la forma en que se realizan las transformaciones del contenido de los diferentes campos para permitir funcionalidades durante la búsqueda. Manera de hacer un match entre lo solicitado y lo que se tiene.

Elaboración de la practica:

Creación del índice:

```
History Settings Help 201 - Created 658 ms
1 #Si no hay indice lo crea
2 POST indice_prueba/anime
3 {
4   "rating": 4.6,
5   "description": "Amano Yukiteru, un estudiante de 2\u00ba a\u00f1o de secundaria, es un chico solitario e introvertido que gasta todo su tiempo escribiendo un diario en su celular desde el punto de vista de un espectador. Su \u00fanico amigo es un producto de su imaginaci\u00f3n, Deus Ex Machina, el Dios del Tiempo y el Espacio. Un d\u00eda, este dios le da al diario de Yukiteru la habilidad de ver lo que suceder\u00eda en los siguientes 90 d\u00edas en el futuro y enrola a Yukiteru en un torneo mortal donde deber\u00eda luchar por su vida contra los poseedores de otros diarios similares."
6   "img": "/uploads/animes/covers/657.jpg",
7   "genre": ["Acci\u00f3n", "Misterio", "Psicol\u00f3gico", "Shounen", "Sobrenatural", "Suspense"],
8   "type": "Anime",
9   "web_state": "Finalizado",
10  "votes": 2222,
11  "name": "Mirai Nikki",
12  "id": "657",
13  "slug": "mirai-nikki-2011",
14  "episodes_num": 27,
15  "animeRel": [{"name": "Mirai Nikki Redial", "type": " (Secuela)", "slug": "mirai-nikkiredial"}]
16 }
```

```
1 #! Deprecation: [types removal]
   Specifying types in document index requests is deprecated, use the typeless endpoints instead ({index}/_doc/{id}, {_index}/_doc, or {_index}/_create/{id}).
2 {
3   "_index" : "indice_prueba",
4   "_type" : "anime",
5   "_id" : "Ft1A53QBuhK6lo85rjE1",
6   "_version" : 1,
7   "result" : "created",
8   "_shards" : {
9     "total" : 2,
10    "successful" : 1,
11    "failed" : 0
12  },
13  "_seq_no" : 0,
14  "_primary_term" : 1
15 }
```

Figura 1: Creaci\u00f3n de \u00cdndice en Elasticsearch.

Para crear un \u00cdndice es necesario usar el comando: `POST #nombre_indice/#tipo_indice`. El error de la parte superior es sobre usa los valores predefinidos en ciertos valores.

Corroboramos que se haya insertado nuestro \u00cdndice, por medio del comando: `GET #nombre_inidce/_search`. El resultado del comando anterior ser\u00eda el siguiente:

```
History Settings Help 200 - OK 24 ms
1
2 #Si no hay indice lo crea
3 #1-Al nombre del indice
4 #Ingresamos el archivo JSON
5 POST /anime_data/anime_data
6 {
7   "rating": 4.6,
8   "description": "Amano Yukiteru, un estudiante de 2\u00ba a\u00f1o de secundaria, es un chico solitario e introvertido que gasta todo su tiempo escribiendo un diario en su celular desde el punto de vista de un espectador. Su \u00fanico amigo es un producto de su imaginaci\u00f3n, Deus Ex Machina, el Dios del Tiempo y el Espacio. Un d\u00eda, este dios le da al diario de Yukiteru la habilidad de ver lo que suceder\u00eda en los siguientes 90 d\u00edas en el futuro y enrola a Yukiteru en un torneo mortal donde deber\u00eda luchar por su vida contra los poseedores de otros diarios similares."
9   "img": "/uploads/animes/covers/657.jpg",
10  "genre": ["Acci\u00f3n", "Misterio", "Psicol\u00f3gico", "Shounen", "Sobrenatural", "Suspense"],
11  "type": "Anime",
12  "web_state": "Finalizado",
13  "votes": 2222,
14  "name": "Mirai Nikki",
15  "id": "657",
16  "slug": "mirai-nikki-2011",
17  "episodes_num": 27,
18  "animeRel": [{"name": "Mirai Nikki Redial", "type": " (Secuela)", "slug": "mirai-nikkiredial"}]
19 }
20
21 GET anime_data/_search
22
23
24 {
25   "took": 0,
26   "timed_out": false,
27   "_shards": {
28     "total": 1,
29     "successful": 1,
30     "skipped": 0,
31     "failed": 0
32   },
33   "hits": {
34     "total": {
35       "value": 1,
36       "relation": "eq"
37     },
38     "max_score": 1.0,
39     "hits": [
40       {
41         "_index": "anime_data",
42         "_type": "anime_data",
43         "_id": "Ud233QBuhK6lo85S5tia",
44         "_score": 1.0,
45         "_source": {
46           "rating": 4.6,
47           "description": "Amano Yukiteru, un estudiante de 2\u00ba a\u00f1o de secundaria, es un chico solitario e introvertido que gasta todo su tiempo escribiendo un diario en su celular desde el punto de vista de un espectador. Su \u00fanico amigo es un producto de su imaginaci\u00f3n, Deus Ex Machina, el Dios del Tiempo y el Espacio. Un d\u00eda, este dios le da al diario de Yukiteru la habilidad de ver lo que suceder\u00eda en los siguientes 90 d\u00edas en el futuro y enrola a Yukiteru en un torneo mortal donde deber\u00eda luchar por su vida contra los poseedores de otros diarios similares.",
48           "img": "/uploads/animes/covers/657.jpg",
49           "genre": ["Acci\u00f3n", "Misterio", "Psicol\u00f3gico", "Shounen", "Sobrenatural", "Suspense"],
50           "type": "Anime",
51           "web_state": "Finalizado",
52           "votes": 2222,
53           "name": "Mirai Nikki",
54           "id": "657",
55           "slug": "mirai-nikki-2011",
56           "episodes_num": 27,
57           "animeRel": [{"name": "Mirai Nikki Redial", "type": " (Secuela)", "slug": "mirai-nikkiredial"}]
58         }
59       }
60     ]
61   }
62 }
```

Figura 2: Corroboraci\u00f3n de la creaci\u00f3n del \u00cdndice "anime_data".

Ahora recuperaremos los Mappings, por medio del siguiente comando: GET #nombre_indice/_mapping. Obteniendo:



The screenshot shows a web browser interface with a console on the left and a JSON response on the right. The console shows a series of commands and their outputs, including a GET request to the Elasticsearch API. The JSON response on the right shows the mapping for the 'anime_data' index, which includes properties like 'name', 'type', 'fields', 'keyword', and 'ignore_above'.

```
1- {
2-   "indice_prueba" : {
3-     "mappings" : {
4-       "properties" : {
5-         "anime1" : {
6-           "properties" : {
7-             "name" : {
8-               "type" : "text",
9-               "fields" : {
10-                "keyword" : {
11-                  "type" : "keyword",
12-                  "ignore_above" : 256
13-                }
14-              }
15-            },
16-            "slug" : {
17-              "type" : "text",
18-              "fields" : {
19-                "keyword" : {
20-                  "type" : "keyword",
21-                  "ignore_above" : 256
22-                }
23-              }
24-            },
25-            "type" : {
26-              "type" : "text",
27-              "fields" : {
28-                "keyword" : {
29-                  "type" : "keyword",
30-                  "ignore_above" : 256
31-                }
32-              }
33-            }
34-          }
35-        }
36-      }
37-    }
38-  }
39- }
```

Figura 3: Observación de los mappings de “anime_data”.

Es una captura de la recuperación parcial de los mappings obtenidos del archivo que se recibe.

Una parte importante a recalcar es que Elastic ayuda a insertarle a ciertos campos un analizador de texto, como lo sería el siguiente ejemplo obtenido:



The image shows a close-up of the JSON mapping for the 'name' field. It is a text field with a keyword sub-field. The keyword sub-field has a type of 'keyword' and an 'ignore_above' value of 256.

```
"name" : {
  "type" : "text",
  "fields" : {
    "keyword" : {
      "type" : "keyword",
      "ignore_above" : 256
    }
  }
}
```

Figura 4: Ejemplo de mapping.

Donde por medio de un keyword, podemos usar estos analizadores. Hablando más afondo de los analizadores utilizados, existen dos:

- Standard: Donde es usado para campos de tipo texto, y se le aplica ciertas operaciones independientes para tokenizarla.
- Keyword: En este caso no se aplica ninguna operación. Pero tiene algunas utilidades.

Corroborando que los datos de mapping tengan sentido con los de los archivos puestos. Ingresamos los archivos del anime.json. Para ello lo ingresamos uno por uno con la API-bulk, para el uso de esta API, es necesario de usar el comando: POST _bulk, y después los archivos que tenemos, siendo la siguiente manera como se ingresa:

```

30 POST_bulk
31 {"index": {"_index": "anime_data"}}
32 {"rating": "4.5", "description": "La
historia se centra en Hotaru, una
chica que se pierde en un bosque en
el que se dice habitan muchos
fantasmas. Ante ella se le aparece
Gin, un joven muchacho que le
embelesa con una primera mirada
pero al cual ella no se atreve ni a
tocar por miedo a que desaparezca
.\n", "img": "/uploads/animes
/covers/745.jpg", "genre": ["Drama"
, "Romance", "Shoujo",
"Sobrenatural"], "type":
"Pel\u00e9cula", "web_state":
"Finalizado", "votes": 1329, "name"
: "Hotarubi no Mori e", "id": "745"
, "slug": "hotarubi-no-mori-e",
"episodes_num": 1, "animeRel": []}
33 {"index": {"_index": "anime_data"}}

```

Figura 5: Ingreso de los datos por medio de la AP_bulk.

Seguidamente se comprobó si efectivamente todos los JSON, se anexaron al respectivo índice, para ello se ejecuto de nuevo el comando: GET anime_data/_search. Se verifico en la siguiente captura:

```

"hits" : {
  "total" : {
    "value" : 51,
    "relation" : "eq"
  },
  ..

```

Figura 6: Verificación de los índices de “anime_data”

Donde se observa que hay 51 archivos en el índice.

Consultas:

Ahora realizando la primer consulta, donde se pide buscar los animes donde sean “Película”. Esta consulta queda de la siguiente manera:

```
POST /anime_data/anime_data/_search
{
  "query": {
    "match": {"type": "Película"}
  },
  "_source": {
    "includes": ["name", "type"]
  }
}
```

Figura 7: Consulta 1.

Y su respectivo resultado es:

```
{
  "took" : 6,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10,
      "relation" : "eq"
    },
    "max_score" : 1.5998683,
    "hits" : [
      {
        "_index" : "anime_data",
        "_type" : "anime_data",
        "_id" : "492e53QBuhK6lo85QDsL",
        "_score" : 1.5998683,
        "_source" : {
          "name" : "Hotarubi no Mori e",
          "type" : "Película"
        }
      }
    ]
  }
}
```

Figura 8: Resultado de consulta 1.

Mientras que en la consulta 2, que dice buscar el top tres de animés cuyo “rating” es entre 4.5 y 5, además de que tenga un número de episodios entre 5 y 12 y ordenados de forma descendente por el campo “votes”. La siguiente consulta se realizó de la siguiente manera:

```
#2.- Consulta dos
POST /anime_data/anime_data/_search
{
  "query": {
    "bool": {
      "filter": [
        { "range": { "rating": { "gte": 4.5, "lte": 5 } } },
        { "range": { "episodes_num": { "gte": 5, "lte": 12 } } }
      ]
    }
  },
  "sort": [
    { "votes": "desc" }
  ],
  "_source": {
    "includes": [ "title", "votes", "rating", "episodes_num" ]
  }
}
```

Figura 9: Consulta 2

Y como resultado se obtuvo:

```
1  #! Deprecation: [types removal] Specifying types in search requests is deprecated.
2  {
3    "took" : 2,
4    "timed_out" : false,
5    "_shards" : {
6      "total" : 1,
7      "successful" : 1,
8      "skipped" : 0,
9      "failed" : 0
10   },
11   "hits" : {
12     "total" : {
13       "value" : 5,
14       "relation" : "eq"
15     },
16     "max_score" : null,
17     "hits" : [
18       {
19         "_index" : "anime_data",
20         "_type" : "anime_data",
21         "_id" : "Ct2e53QBuhK6lo85QDwL",
22         "_score" : null,
23         "_source" : {
24           "episodes_num" : 10,
25           "rating" : "4.6",
26           "votes" : 347
27         },
28         "sort" : [
29           347
30         ]
31       },

```

Figura 10: Resultado de la consulta 2.

Mientras que en el despliegue de las gráficas se realizó de la siguiente manera.

Primeramente, desplegamos el menú de la parte superior izquierda, y nos dirigimos a la parte de hasta abajo donde dice “Management” y presionamos la opción de “Stack Management”. De ahí se desplegará la siguiente ventanas:

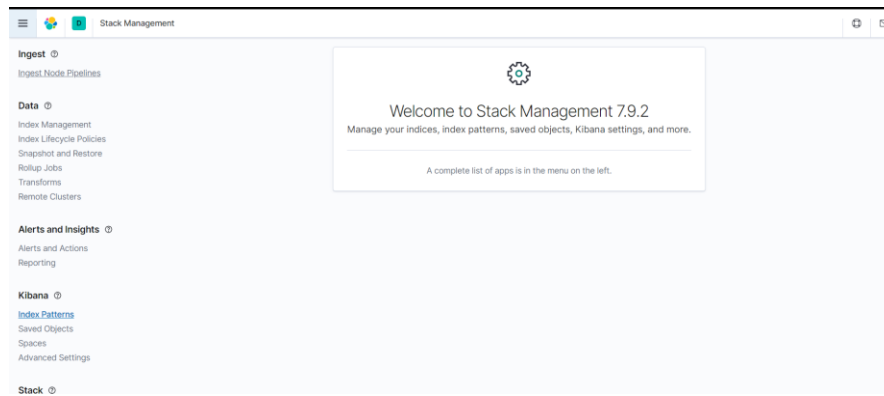


Figura 11: Navegación en Elastic

Ahí nos dirigimos al menú del lado izquierdo, y en apartado de “Kibana”, seleccionaremos la opción de “Index Patterns”, para así poder crear el índice para visualizar los datos introducido en “Elasticsearch”, Los pasos para la creación de este índice se muestran en las siguientes capturas:

Step 1 of 2: Define index pattern

Index pattern name

[Next step >](#)

Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, *, <, >, | are not allowed.

☒ Include system and hidden indices

✓ Your index pattern matches 1 source.

anime_data	Index
------------	-------

Figura 12: Creación del índice paso 1.

Step 2 of 2: Configure settings

anime_data*

The indices which match this index pattern don't contain any time fields.

▼ Hide advanced options

Custom index pattern ID

Kibana will provide a unique identifier for each index pattern. If you do not want to use this unique ID, enter a custom one.

[< Back](#) [Create index pattern](#)

Figura 13: Creación del índice paso 2.

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 (30) Scripted fields (0) Source filters (0)					
<input type="text" value="Search"/> 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		•		

Figura 14: Datos del índice en Kibana

Ya creado el índice desplegamos el menú de la parte superior izquierda y en el apartado de “Kibana”, seleccionaremos “visualize”.

Desarrollando las gráficas:

1.- En la visualización de la gráfica de los géneros de anime, se configuro los siguientes parámetros:

anime_data*

Data

Options

Buckets

Tags

Aggregation

Terms

Terms help

Field

genre.keyword

Order by

Metric: Count

Order

Descending

Size

50

☐

Group other values in separate bucket

☐

Show missing values

Custom label

×

Discard

▶

Update

Figura 15: Datos de consulta 1.

Se puso en “size” un numero de 50, para mostrar todos los géneros del índice. Con ello se desplego la siguiente gráfica:

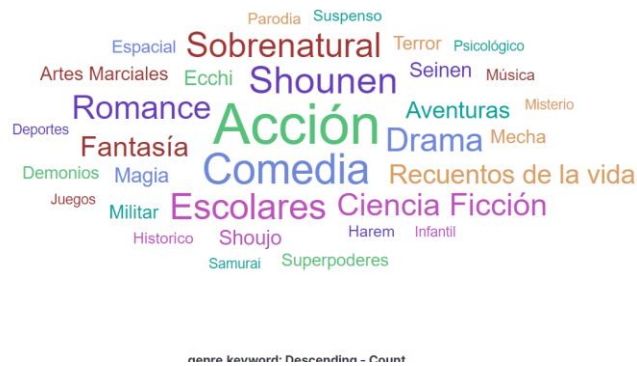


Figura 16: Grafica de consulta 1.

2.- Para crear la gráfica: Donde se observe la categoría de los animes. Para ello se selecciona, la gracias de tipo PIE, y se configura los siguientes parámetros:

anime_data*

Data

Options

Buckets

Split slices

Terms help

Terms

Field

type.keyword

Order by

Metric: Count

Order

Size

Descending

5

☐

Group other values in separate bucket

☐

Show missing values

Custom label

×

Discard

▶

Update

Figura 17: Datos de consulta 2

Con ello entramos al parámetro “type” y decimos que nos cuente los términos que haya en cada uno de los JSON para poder formar la siguiente gráfica:

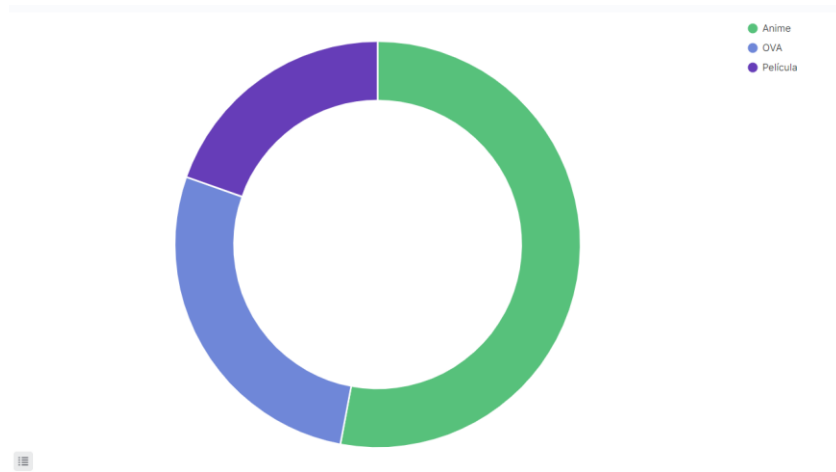


Figura 18: Grafica de consulta 2

3.- Por último, para generar la grafica de barras invertidas, donde se grafique el numero de votos de anime tipo OVA, divididos en rangos: 0-100, 100-200 y mas de 2000. Para ello primero se configuro un filtro con los siguientes datos:

The 'EDIT FILTER' interface shows a filter configuration. The 'Field' is set to 'type.keyword' and the 'Operator' is set to 'is'. The 'Value' is set to 'OVA'. There is a checkbox for 'Create custom label?' which is currently unchecked. At the bottom, there are 'Cancel' and 'Save' buttons.

Figura 19: Filtro de consulta 3.

Con ello nos regresa los animes de tipo OVA. Seguidamente se configuraron los siguientes datos en la parte de “Buckets”:

The 'anime_data*' interface shows the 'Buckets' configuration for the 'votes' field. The 'X-axis' is set to 'votes'. The 'Aggregation' is set to 'Range'. The 'Field' is set to 'votes'. The 'Buckets' are defined by three ranges: 0 to 100, 100 to 200, and 200 to +∞. Each range has a corresponding 'Add range' button. At the bottom, there are 'Discard' and 'Update' buttons.

Figura 20: Datos de consulta 3.

Con ello se obtuvo la siguiente gráfica:

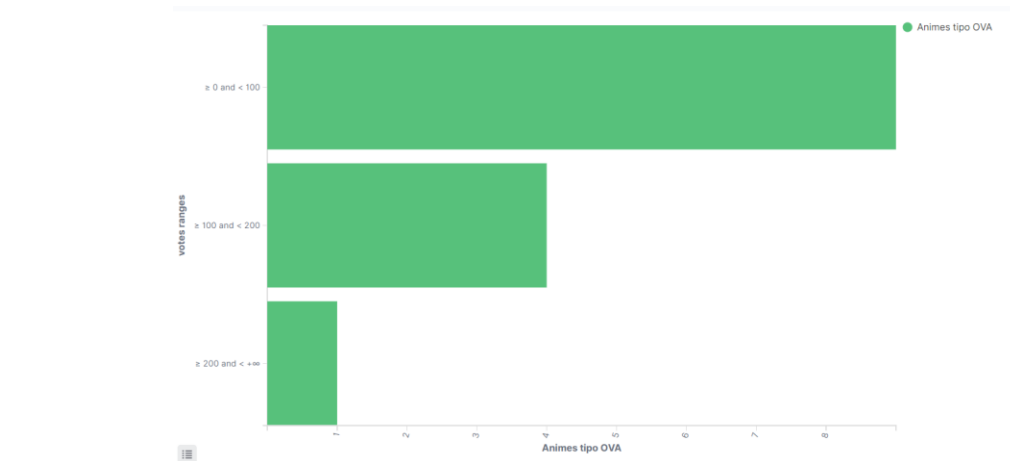


Figura 21: Grafica de consulta 3

Errores:

1.- PUT para crear el índice. Al usar PUT para crear un índice, se encontró que lanzaba un error. El cuál era el siguiente:

```
History Settings Help
1
2
3 #Si no hay indice lo crea
4 #1º-Al nombre del indice
5 #Ingresamos el archivo JSON
6
7 PUT /anime_data
8 {"rating": 4.6,
9  "description": "Amano Yukiteru, un estudiante de 2º de secundaria, es un chico solitario e introvertido que gasta todo su tiempo escribiendo un diario en su celular desde el punto de vista de un espectador. Su único amigo es un producto de su imaginación, Deus Ex Machina, el Dios del Tiempo y el Espacio. Un día, este dios le da al diario de Yukiteru la habilidad de ver lo que sucederá en los siguientes 90 días en el futuro y enrola a Yukiteru en un torneo mortal donde deberá luchar por su vida contra los poseedores de otros diarios similares.",
10 "img": "/uploads/animes/covers/657.jpg",
11 "genre": ["Acción", "Misterio", "Psicológico", "Shounen", "Sobrenatural", "Suspense"],
12 "type": "Anime",
13 "web_state": "Finalizado",
14 "votes": 2222,
15 "name": "Mirai Nikki",
16 "id": "657",
17 "slug": "mirai-nikki-2011",
18 "episodes_num": 27,
19 "animeRel": [{"name": "Mirai Nikki Redial", "type": "(Secuela)", "slug": "mirai-nikkiredial"}]
20 }
21
22 400 - Bad Request 36 ms
1 {
2   "error": {
3     "root_cause": [
4       {
5         "type": "parse_exception",
6         "reason": "unknown key [episodes_num] for create index"
7       },
8     ],
9     "type": "parse_exception",
10    "reason": "unknown key [episodes_num] for create index"
11  },
12  "status": 400
13 }
14 }
```

Figura 22: Error en la creación del índice.

Para solucionar problema anteriormente, se usó el método POST, con ello modificamos los valores que se le mandaba a la API, así se le agregó el tipo, al momento de agregar los archivos JSON.

2.- El intento de haber creado “mapping” después de haber creado el índice. Esto marcaba un error debido a que, una vez creado el índice, no se puede modificar. Para ello se borró el índice y se volvió a crear desde cero.

3.- En el despliegue de las consultas, esto debido a los parámetros que se le indicaban, para ello se checo la documentación y de ahí se obtuvieron los respectivos resultados.

4.- Al momento de mostrar las respectivas graficas debido a que no se le ingresaban de manera correcta los parámetros para que se mostrara lo que se debería de mostrar. Por ejemplo, en la creación de la primera gráfica, solo se visualizaban 5 géneros, para ello se tuvo que cambiar ciertos parámetros.

5.- Acceder a la plataforma, cuando se realizaba la practica se tuvo que realizar de manera local esto con el fin de que no se pudo acceder a la plataforma. Se soluciono solo haciendo uso de Elasticsearch y Kibana, para el sistema operativo Windows.

Ejecución de la practica:

Para la ejecución de la practica se debe de tener instalados Elasticsearch y Kibana. Ya instalados y en ejecución los servicios. Se debe de acceder al localhost:5601.

De ahí se debe ejecutar el archivo que viene el repositorio llamado: "Elasticsearch_Consultas", el contenido de este archivo de ir en la parte de "DevTools", que esta en el índice de la parte superior izquierda en el apartado de "Managemet".

Para la ejecución de las respectivas gráficas, solo se deben de seguir los pasos indicados previamente, con las respectivas configuraciones.

Como nota la carpeta de "Archivo_uso", son los JSON que se indexaron.

Conclusiones:

Una de las principales conclusiones en la realización de la práctica es que este es mi primer acercamiento a la tecnología de Elastic, y uno de mis principales errores fue al intentar modificar un índice ya hecho, siendo que no es posible realizarlo en Elastic.

Otro inconveniente fue en la documentación, la encontré un poco rara, en la parte de la creación del índice, así que tuve que optar por otras fuentes menos oficiales, siendo estas de mucha ayuda. Con esa información y la documentación obtuve resultados mas rápidos, en un tiempo menor a lo que se necesitaba para la práctica.

Encontre la tecnología Elastic de mucho interés y mucho uso, en la parte de ciencia de datos para analizar los datos a trabajar, además de la facilidad de búsqueda en Elastic por medio de lo que son los índices invertidos.

Cuando se usa una nueva tecnología es necesario tomar en cuenta su sintaxis y algunos temas básicos sobre esta tecnología, pero lo que ayuda mucho es la documentación, la experimentación constante con la nueva tecnología y sobre todo nuevos enfoques al intentar resolver problemas por medio de nuevas tecnologías. Hay que siempre innovar los conocimientos en cuanto al uso de estas herramientas y sobre todo nutrirnos de ellas.