

Utilizando Elasticsearch

Luis Antonio Manjarrez Torres

Contenido

Contenido.....	2
Creación de un Índice.....	3
Insertando Datos.....	5
Realizar búsquedas sobre el índice	7
Búsqueda 1.....	7
Búsqueda 2.....	8
Creando una vista	10
Problemas presentados.....	15

Creación de un Índice

Un índice de es una colección de datos. Estos son almacenados en formato JSON. Cada documento correlaciona un conjunto de claves (nombres de campos o propiedades) con sus valores correspondientes (textos, números, Booleanos, fechas, variedades de valores, geolocalizaciones u otros tipos de datos).

Para crear un índice, necesitamos del método PUT y el nombre que le daremos el índice, dentro del método escribiremos la configuración y parámetros que necesitamos para el índice, en este caso fue una configuración sencilla que nos permita hacer consultas por los diferentes parámetros que tenemos.

```
DELETE /employee-sps
PUT /employee-sps
{
  "settings": {
    "index.number_of_shards": 1,
    "index.number_of_replicas": 1
  },
  "mappings": {
    "doc": {
      "properties": {
        "guid": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "isActive": {"type": "boolean"},
        "payment": {"type": "double"},
        "picture": {"type": "text"},
        "age": {"type": "double"},
        "eyeColor": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        }
      }
    },
    "name": {
      "properties": {
        "first": {
          "type": "text"
        },
        "last": {
          "type": "text"
        }
      }
    },
    "company": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "email": {"type": "text"},
    "phone": {"type": "text"},
    "address": {"type": "text"},
    "about": {"type": "text"},
    "registered": {"type": "text"},
    "latitude": {"type": "double"},
    "longitude": {"type": "double"},
  }
}
```

```
"tags": {
    "type": "text",
    "fields": {
        "keyword": {
            "type": "keyword",
            "ignore_above": 256
        }
    }
},
"range": {
    "type": "integer",
    "fields": {
        "keyword": {
            "type": "keyword",
            "ignore_above": 256
        }
    }
},
"friends": {
    "properties": {
        "id": {
            "type": "double"
        },
        "name": {
            "type": "text"
        }
    }
},
"greeting": {"type": "text"},
"favoriteFruit": {
    "type": "text",
    "fields": {
        "keyword": {
            "type": "keyword",
            "ignore_above": 256
        }
    }
}
}
```

```
#! Deprecation: 'y' year should be replaced with 'u'. Use 'y' for year-of-era.; 'Z' time zone offset
/id fails when parsing 'Z' for Zulu timezone. Consider using 'X'. Prefix your date format with '8'
to use the new specifier.
```

#! Deprecation: [types removal] The parameter `include_type_name` should be explicitly specified in create index requests to prepare for 7.0. In 7.0 `include_type_name` will default to 'false', and requests are expected to omit the type name in mapping definitions.

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "employee-sps"
}
```

Insertando Datos

Bulk es un api masivo que nos permite indexar o borrar mucha información en una sola llamada, las acciones se especifican en el cuerpo con un formato ndjson, este formato permite enviar información múltiple donde cada salto de línea representa un nuevo objeto json.

```
POST /employee-sps2/doc/_bulk
{"index":{"_index":"employee-sps","_type":"doc"}}
{"about":{"Amet incididunt voluptate in eu dui sint ullamco pariatu irure culpa. Amet mollit dolor excepteur nisi ex sit ullamco. Lorem veniam voluptate non est do
. Est id enim incididunt dolore aliqua incididunt excepteur ipsum sint esse commodo sit. In occaecat occaecat fugiat in mollit ut sint velit reprehenderit tempor
aliquip reprehenderit. Et exercitation sit elit pariatu veniam officia amet quis occaecat. Voluptate adipisicing dolor cillum anim enim sunt Lorem velit ex magna
non enim.", "address":"268 Bergen Avenue, Sattley, Rhode Island, 5425", "age":27, "company":"IMAGINART", "email":"logan.blackburn@imaginart.me", "eyeColor":"green",
, "favoriteFruit":"apple", "friends":[{"id":0, "name":"Loraine Foreman"}, {"id":1, "name":"Elizabeth Burnett"}, {"id":2, "name":"Consuelo Patel"}], "greeting":"Hello,
Logan! You have 8 unread messages.", "guid":"33f177aa-da4a-45cf-9071-9a48c803f744", "isActive":false, "latitude":"76.532393", "longitude":"-44.824277", "name":{"first":
"Logan", "last":"Blackburn"}, "payment":"51722.69", "phone":"+1 (995) 460-3467", "picture":"http://v.placeholder.it/32x32", "range":[0,1,2,3,4,5,6,7,8,9], "registered"
:"Sunday, June 28, 2015 8:48 AM", "tags":["elit", "nisi", "eu", "aute", "deserunt"]}}
{"index":{"_index":"employee-sps","_type":"doc"}}
{"about":{"Veniam laborum labore eu sit. Minim est anim reprehenderit mollit non proident. Adipisicing mollit adipisicing qui aliqua do labore cillum commodo aliqua
id. Cillum aute voluptate occaecat reprehenderit nulla exercitation excepteur ad quis occaecat proident dolor aliqua. Officia dolor aute deserunt nostrud anim
cillum nisi ipsum laborum amet non velit exercitation in. Do fugiat voluptate id officia non adipisicing consequat ex amet consequat.", "address":"379 Coyle Street
, Remington, Connecticut, 6597", "age":21, "company":"PARLEYNET", "email":"castillo.knapp@parley.net.co.uk", "eyeColor":"brown", "favoriteFruit":"strawberry", "friends"
:[{"id":0, "name":"Rachel Daniels"}, {"id":1, "name":"Ester Sweet"}, {"id":2, "name":"Stefanie Mathis"}], "greeting":"Hello, Castillo! You have 9 unread messages."
, "guid":"4905d7e1-224c-41fd-bec8-fc9350383dd0", "isActive":true, "latitude":"42.833405", "longitude":"-6.80161", "name":{"first":"Castillo", "last":"Knapp", "payment"
:"85014.44", "phone":"+1 (829) 418-2414", "picture":"http://v.placeholder.it/32x32", "range":[0,1,2,3,4,5,6,7,8,9], "registered":"Friday, April 3, 2015 7:13 PM", "tags"
:["qui", "labore", "commodo", "nihil", "magna"]}}
{"index":{"_index":"employee-sps","_type":"doc"}}
{"about":{"Adipisicing cupidatat culpa enim cillum cupidatat aute. Irure sunt elit consequat in nulla veniam adipisicing nisi ad dolore ipsum nostrud pariatu elit.
Quis aliquip nulla eiusmod eu nisi tempor laborum do cupidatat velit mollit. Culpa minim aliquip incididunt commodo esse labore ad cillum et labore cillum nulla
incididunt reprehenderit. Ex exercitation et et dolore aute ex veniam aliqua nisi amet adipisicing mollit sint exercitation.", "address":"755 Freeman Street, Roy,
Pennsylvania, 2532", "age":21, "company":"SPLINX", "email":"sweet.burt@splinx.biz", "eyeColor":"green", "favoriteFruit":"apple", "friends":[{"id":0, "name":"Holland
Stephens"}, {"id":1, "name":"Bettie Morales"}, {"id":2, "name":"Morgan Clay"}], "greeting":"Hello, Sweet! You have 5 unread messages.", "guid":"d1225723-5dcd-4522-8669
-417274e6a988", "isActive":false, "latitude":"-39.309262", "longitude":"84.096944", "name":{"first":"Sweet", "last":"Burt"}, "payment":"67455.08", "phone":"+1 (807) 417
-2783", "picture":"http://v.placeholder.it/32x32", "range":[0,1,2,3,4,5,6,7,8,9], "registered":"Wednesday, July 30, 2014 12:40 PM", "tags":["labore", "nulla", "cillum",
"ad", "elit"]}}
{"index":{"_index":"employee-sps","_type":"doc"}}
{"about":{"Dolor consequat elit commodo magna dui sit officia eu qui non esse proident. In ex culpa dolore velit irure non non elit dolore. Aliquip officia sint
fugiat magna et non veniam veniam aliquip non ea veniam. Nulla Lorem officia magna dui labore veniam mollit culpa consectetur dui velit occaecat dolor pariatu
", "address":"530 Stillwell Avenue, Strykersville, New Jersey, 601", "age":26, "company":"OULU", "email":"lydia.gentry@oulu.org", "eyeColor":"green", "favoriteFruit"
:"banana", "friends":[{"id":0, "name":"Margaret Turner"}, {"id":1, "name":"Mai Abbott"}, {"id":2, "name":"Ofelia Davis"}], "greeting":"Hello, Lydia! You have 8 unread
messages.", "guid":"a30caf44-2d95-41a5-b03f-bfeb5b450bc1", "isActive":true, "latitude":"-89.167329", "longitude":"-147.674081", "name":{"first":"Lydia", "last"
:"Gentry"}, "payment":"51104.66", "phone":"+1 (911) 600-3341", "picture":"http://v.placeholder.it/32x32", "range":[0,1,2,3,4,5,6,7,8,9], "registered":"Tuesday, December
22, 2015 9:53 PM", "tags":["ipsum", "pariatu", "eiusmod", "dolore", "ad"]}}
```

```
{
  "took" : 47,
  "errors" : false,
  "items" : [
    {
      "index" : {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "8qQwkHQBxpSWr7U7aL7a",
        "_version" : 1,
        "result" : "created",
        "_shards" : {
          "total" : 2,
          "successful" : 2,
          "failed" : 0
        },
        "_seq_no" : 0,
        "_primary_term" : 1,
        "status" : 201
      }
    },
    {
      "index" : {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "86QwkHQBxpSWr7U7aL7a",
        "_version" : 1,
        "result" : "created",
        "_shards" : {
          "total" : 2,
          "successful" : 2,
          "failed" : 0
        },
        "_seq_no" : 1,
        "_primary_term" : 1,
        "status" : 201
      }
    }
  ],
}
```

Realizar búsquedas sobre el índice

Búsqueda 1

Busca todos los empleados que superen los 30 años.

Para buscar información indexada ocupamos el método GET seguido del índice donde buscaremos finalizando con la palabra `_search` ya que esta devuelve las coincidencias con la consulta generada en la solicitud. Para generar una consulta, necesitamos adaptar nuestra consulta a un formato json.

En este caso empezamos definiendo nuestra consulta con la palabra “query” y dentro del objeto definimos la búsqueda en este caso definimos el objeto “range” y dentro definimos el campo por el cual buscaremos, dentro de este, “lte” y “gte” nos permiten definir los límites de la consulta, “gte” define desde que valor iniciaremos y “lte” en que valor concluiremos el rango de la búsqueda, al solo poner “gte” en este caso la consulta es interpretada como cualquier valor mayor a 30.

```
GET /employee-sps/doc/_search
{
  "query": {
    "range": {
      "age": {
        "gte": 30
      }
    }
  }
}
```

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 23,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "9qQwkHQBxpSwr7U7aL7a",
        "_score" : 1.0
      },
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "KQwkHQBxpSwr7U7aL7a",
        "_score" : 1.0
      },
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "aQwkHQBxpSwr7U7aL7a",
        "_score" : 1.0
      }
    ]
  }
}
```

Búsqueda 2

Busca el top los 5 empleados menores o iguales a 30 años que tienen un sueldo mayor a los 50 mil pesos.

Esta consulta es un poco similar a la anterior solo que en este caso nos pide filtrar por dos campos, para hacer esto los rangos los definimos dentro del objeto "must", lo cual fuerza a que se cumplan las condiciones establecidas, después de declarar la consulta ordenamos por los parámetros "age" y "payment" y solo pedimos los primeros 5 registros

```
GET /employee-sps/doc/_search
{
  "query": {
    "bool": {
      "must": [
        { "range": { "age": { "gte": 0, "lt": 30 } } },
        { "range": { "payment": { "gte": 50000 } } }
      ]
    }
  },
  "sort": [
    {
      "age": { "order": "asc" },
      "payment": { "order": "desc" }
    }
  ],
  "size": 5
}
```



```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 41,
    "max_score" : null,
    "hits" : [
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "C6QwkHQBxpSWr7U7aL_b",
        "_score" : null,
        "sort" : [
          20.0,
          79123.54
        ]
      },
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "96QwkHQBxpSWr7U7aL7a",
        "_score" : null,
        "sort" : [
          20.0,
          59213.35
        ]
      },
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "EqQwkHQBxpSWr7U7aL_b",
        "_score" : null,
        "sort" : [
          20.0,
          57144.61
        ]
      },
      {
        "_index" : "employee-sps",
        "_type" : "doc",
        "_id" : "86QwkHQBxpSWr7U7aL7a",
        "_score" : null,
        "sort" : [
          21.0,
          85014.44
        ]
      }
    ]
  }
}
```

Creando una vista

Para crear la vista previamente creamos un patrón de índice, después de su creación nos dirigimos al apartado Visualize, en ese apartado escogeremos el tipo de grafico que se necesite, en este caso escogimos un grafico de tipo Pie, después escogeremos el patrón de índice.

The image shows the Kibana Visualize interface. At the top, there's a sidebar with navigation options: Discover, Visualize (selected), Dashboard, Timelion, Canvas, and Maps. The main area displays a message: "Looks like you don't have any visualizations. Let's create some!" with a "Create a visualization" button. Below this, a "New Visualization" dialog is open, titled "New Visualization" with a close button (X). The dialog has a search bar labeled "Filter" and a grid of visualization types: Area, Controls, Coordinate Map, Data Table, Gauge, Goal, Heat Map, Horizontal Bar, Line, Markdown, Metric, Pie, Region Map, Tag Cloud, Timelion, and Vega. To the right of the grid, it says "Select a visualization type" and "Start creating your visualization by selecting a type for that visualization." Below the dialog, there are two sections: "From a New Search, Select Index" and "Or, From a Saved Search". The "From a New Search, Select Index" section has a search bar labeled "Filter..." and shows one result: "employee-sps*". The "Or, From a Saved Search" section has a search bar labeled "Saved Searches Filter..." and shows no results, with a "Manage saved searches" button.

Visualize

Search... 0-0 of 0

Looks like you don't have any visualizations. Let's create some!

+ Create a visualization

0 items selected 0-0 of 0

New Visualization

Filter

Select a visualization type

Start creating your visualization by selecting a type for that visualization.

Area Controls Coordinate Map Data Table

Gauge Goal Heat Map Horizontal Bar

Line Markdown Metric Pie

Region Map Tag Cloud Timelion Vega

From a New Search, Select Index

Filter... 1 of 1

Name ▲

employee-sps*

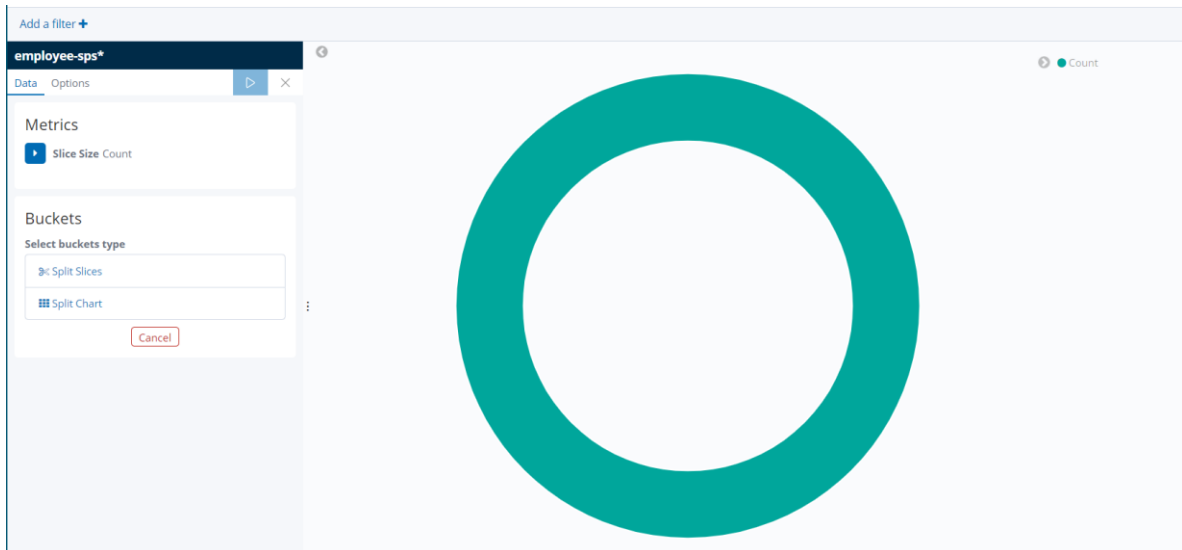
Or, From a Saved Search

Saved Searches Filter... 0-0 of 0 Manage saved searches

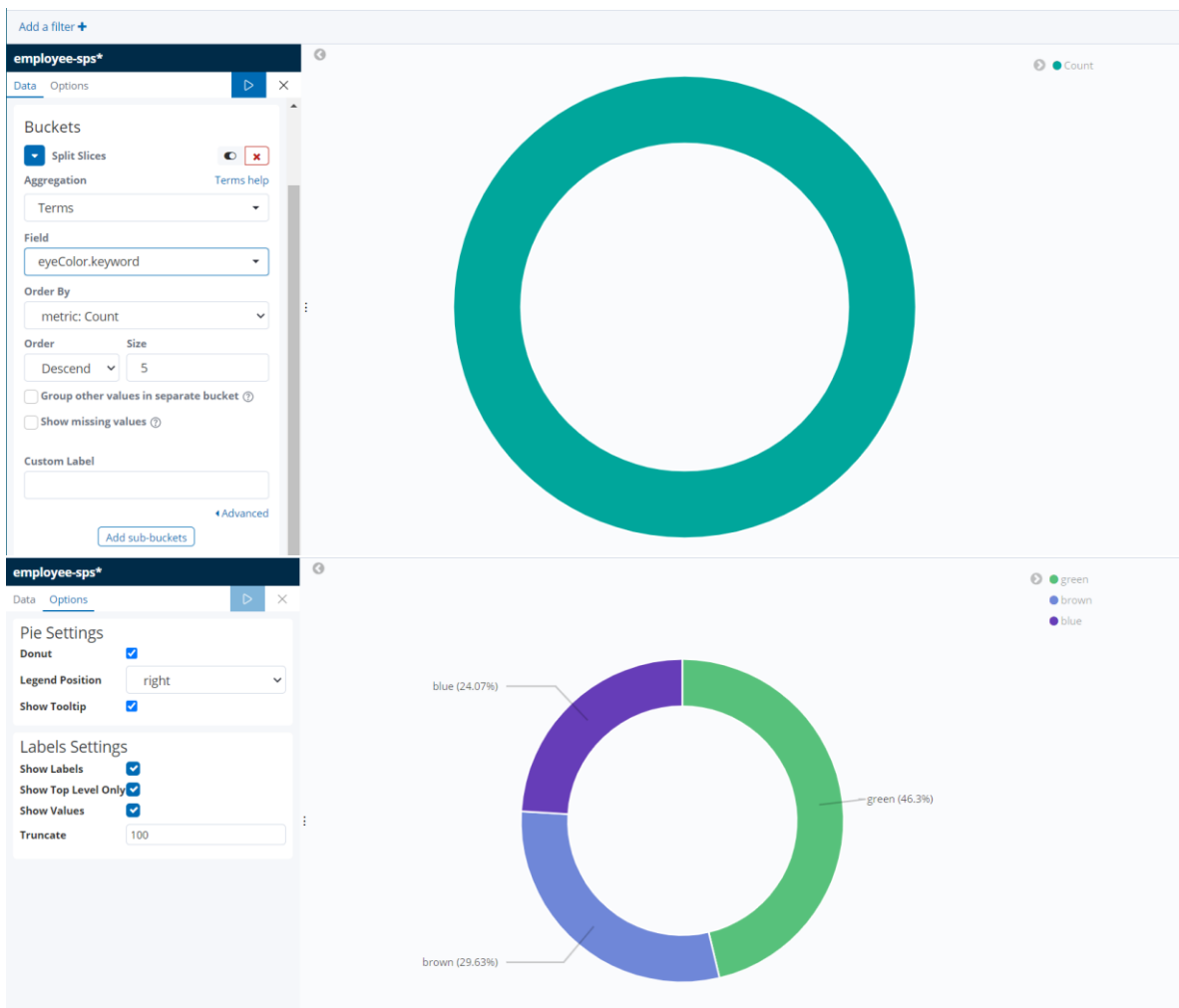
Name ▲

No matching saved searches found.

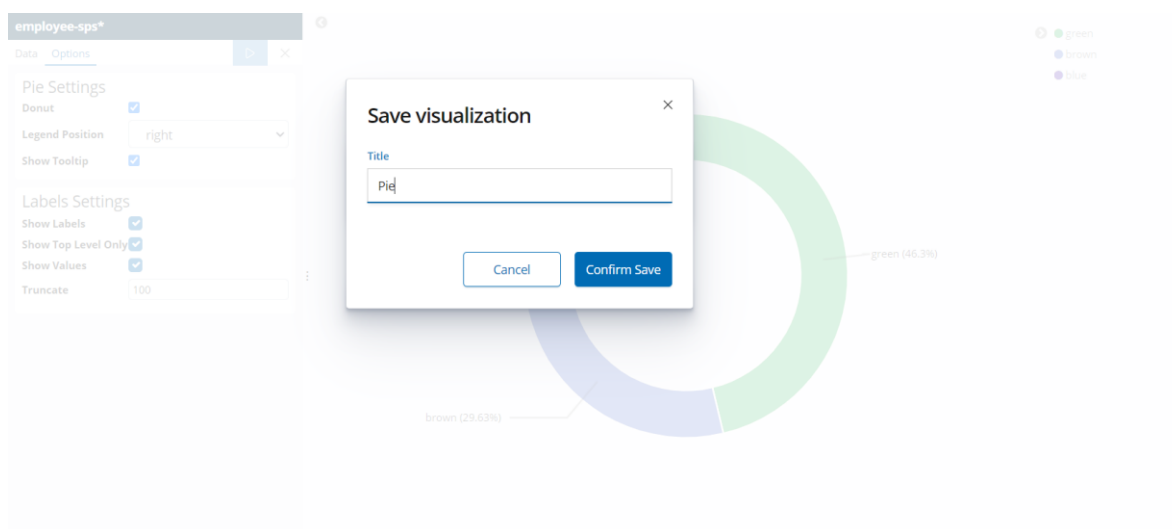
A continuación, veremos que el grafico mostrado, no tiene filtro alguno, para esto nos dirigimos, a la opción Split Slices.



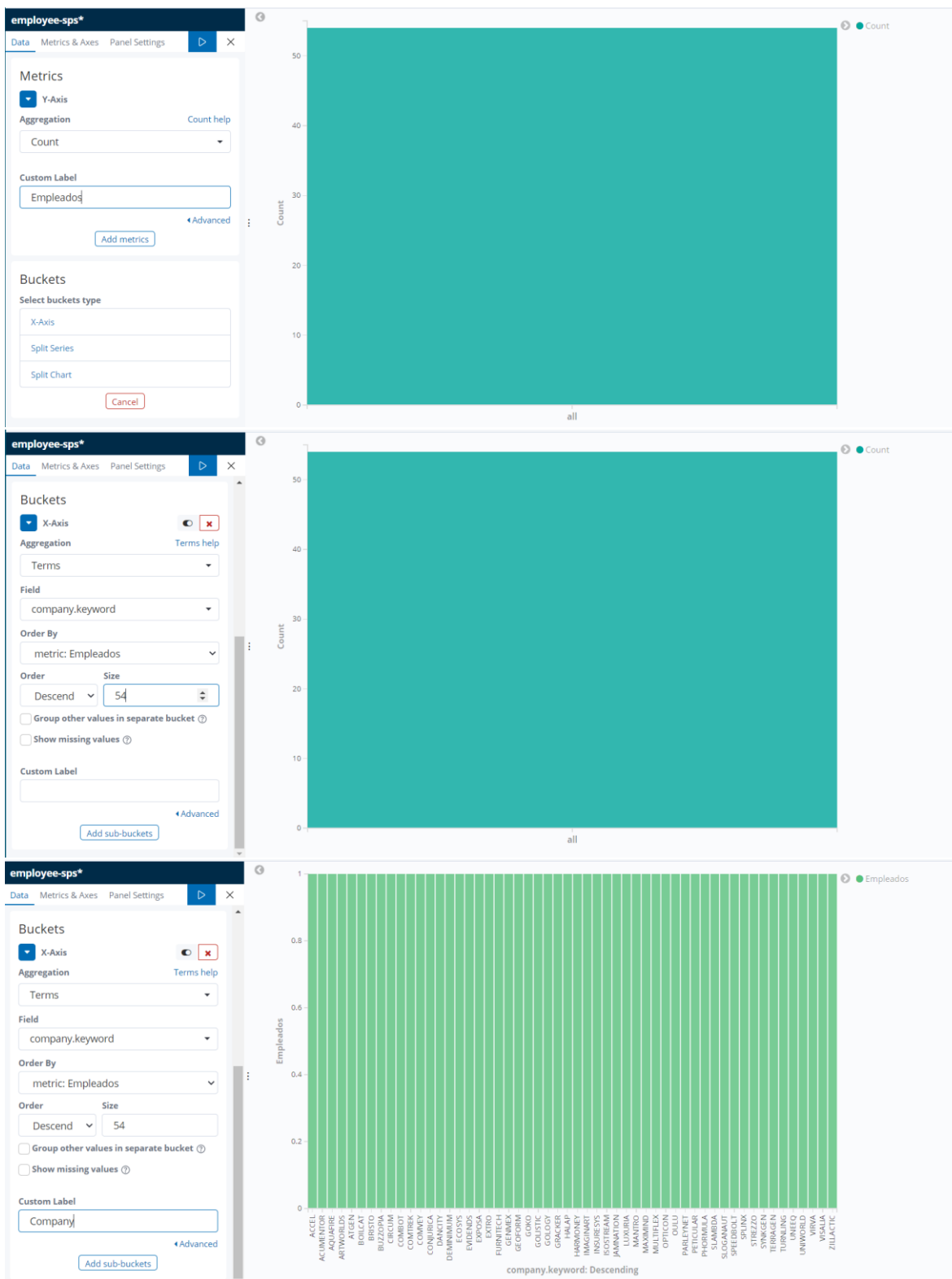
En este apartado en la parte de agregación escogemos la opción de terms, una vez hecho esto solo basta con escoger el termino por el cual se va a filtrar.

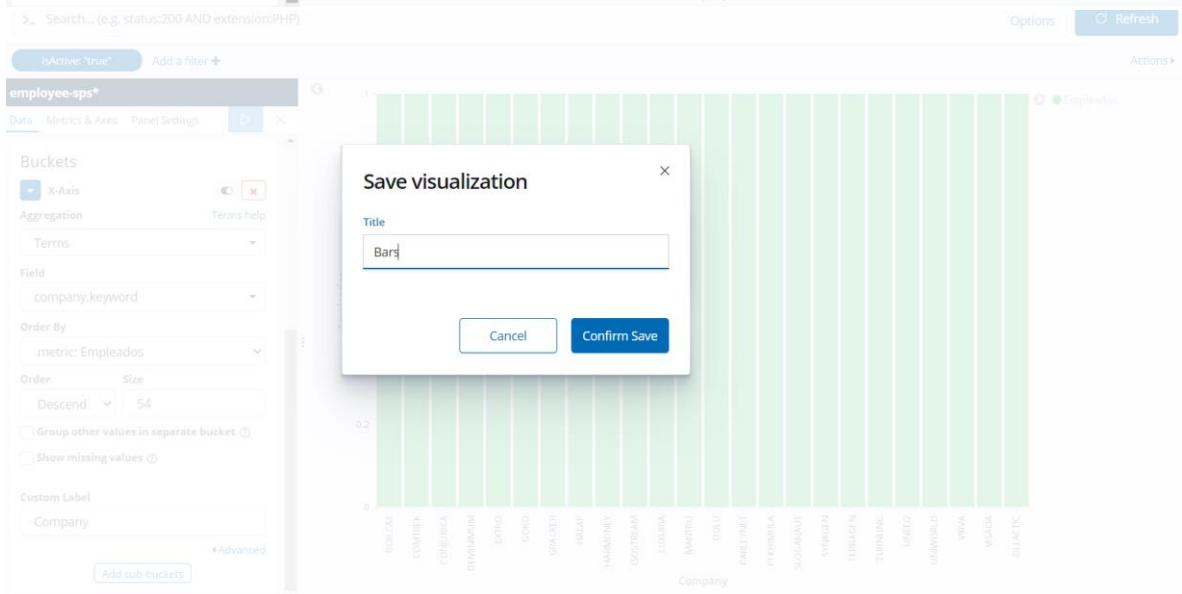
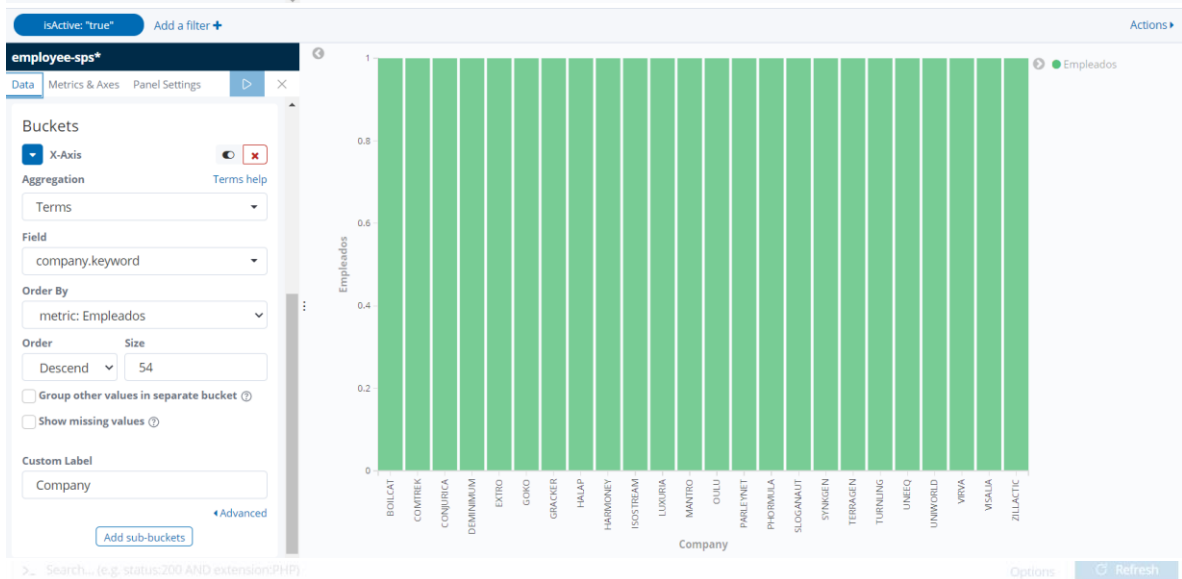
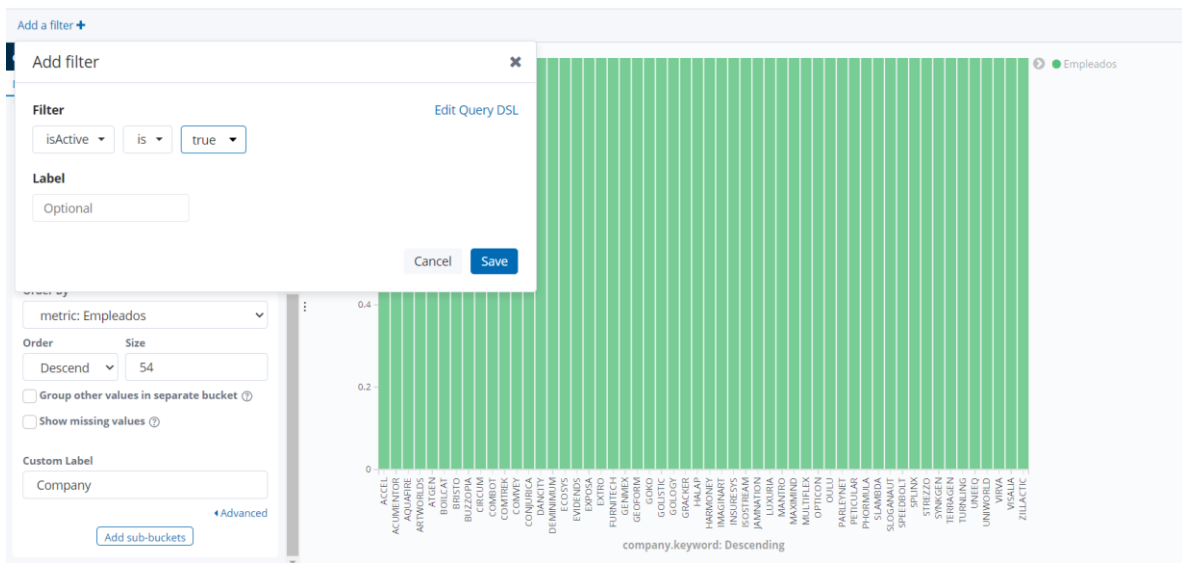


Guardamos el grafico creado.

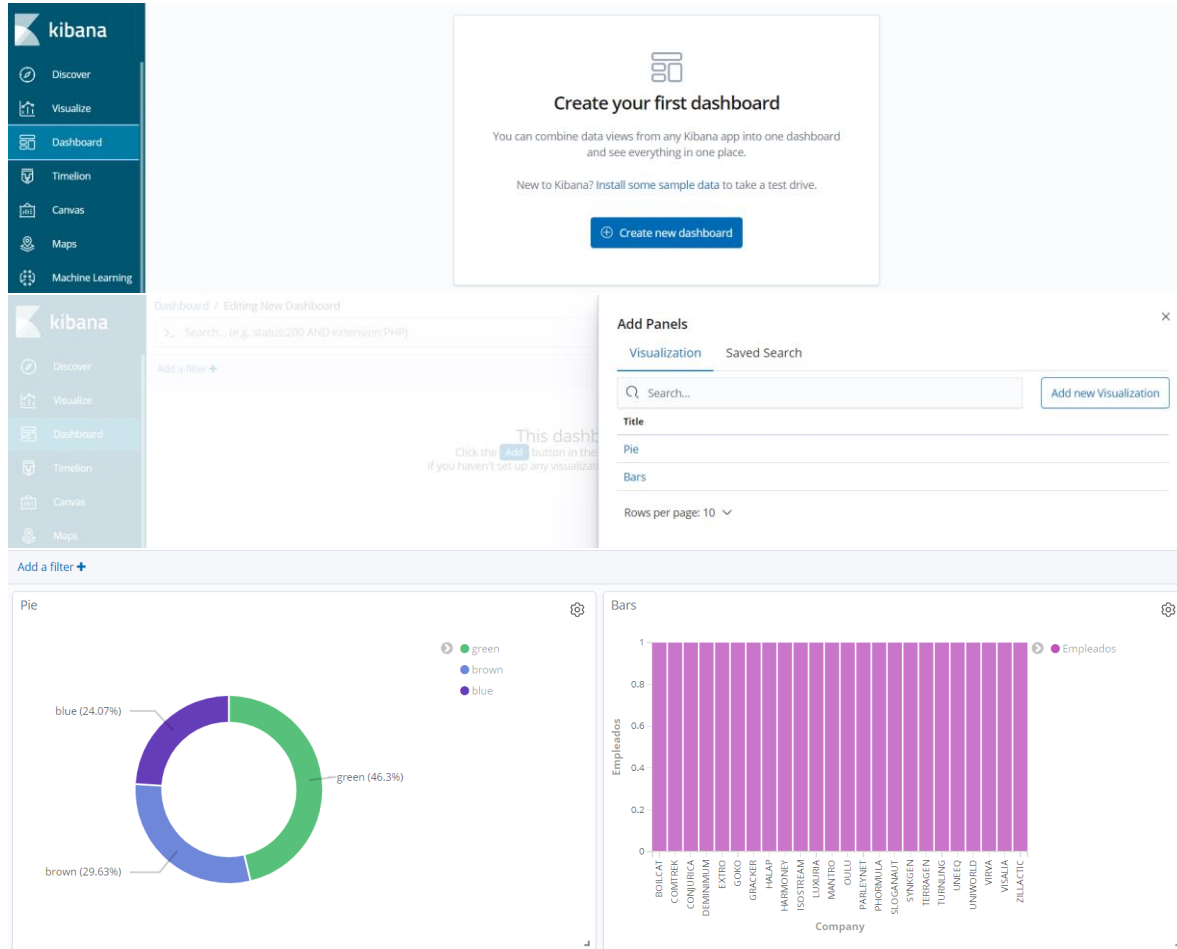


Para el grafio de barras el proceso es el mismo, a diferencia del anterior en este crearemos un filtro para que solo se muestren los elementos activos.





Para crear el dashboard, nos dirigimos a ese apartado y agregamos uno nuevo para finalizar agregamos las visualizaciones hechas anteriormente.



Problemas presentados

El principal problema fue entender cómo funcionaba la tecnología, ya que fue bastante confuso entenderlo, desde crear el índice ya que no fue una parte sencilla de entender, y tuve bastantes problemas en este apartado ya que aunque se creo el índice y las consultas hechas funcionaban, pero cuando empecé en el apartado de visualización no lograba comprender por que no mostraba la información, comencé viendo videos y buscando ejemplos de cómo lo hacían otras personas pero no entendía por qué no funcionaba, leí la documentación que Elasticsearch proporciona pero seguía sin entender. La solución fue que hice el índice sin propiedades ya que de esta manera cuando agregas un registro el api crea las propiedades, y me di cuenta de que mis propiedades estaban mal hechas, volví a estructurar mis propiedades y logre solucionar el problema.

Por otro lado, entender, el cómo hacer las consultas también fue complicado sin embargo la documentación explica bastante bien como generar las consultas por lo que no tuve mucho problema.