# Practica

## Crear un índice con el nombre log\_consultas

### Paso 1:

En la consola puse este código para crear el índice requerido:

**curl -XPOST "http://localhost:9200/log\_consultas/\_doc" -H 'Content-Type: application/json' -d '**

**{**

**"@timestamp": "2010-05-15T22:00:54",**

**"estado\_consulta": "consumo",**

**"servicio": "consulta",**

**"administrador": "Juan Carlos",**

**"consultas\_requeridas": 52**

**}**

Donde -XPOST es un comando curl q indica el tipo de solicitud HTTP que se esta haciendo , estamos enviando una solicitud POST al servidor elasticsearch, en este caso mandamos este comando porque necesitamos agregar datos JSON al índice log\_consultas.

### Paso 2:

Para recuperar el mapping del índice anterior mandé este curl:

El comando GET se usa para obtener datos específicos.

**curl -XGET "http://localhost:9200/log\_consultas/\_mapping"**

### Paso 3:

Para crear un template tome de de base el mapping que había sacado anteriormente y con el comnado XPUT que sirve para crear o actualizar un template

**curl -XPUT "http://localhost:9200/\_template/template\_log\_consultas" -H 'Content-Type: application/json' -d '**

**{**

**"index\_patterns": ["log\_consultas\*"],**

**"mappings": {**

**"properties": {**

**"@timestamp": {**

**"type": "date"**

**},**

**"estado\_consulta": {**

**"type": "keyword"**

**},**

**"servicio": {**

**"type": "keyword"**

**},**

**"administrador": {**

**"type": "keyword"**

**},**

**"consultas\_requeridas": {**

**"type": "integer"**

**}**

**// Agrega aquí otros campos y tipos de datos según el mapping obtenido**

**}**

**}**

**}'**

### Paso 4:

Intente subir el archivo desde mi escritorio, pero me marcaba este error:

{

"error": {

"root\_cause": [

{

"type": "parse\_exception",

"reason": "request body is required"

}

],

"type": "parse\_exception",

"reason": "request body is required"

},

"status": 400

}

Revise el archivo y tenia correctas las estructuras ,por lo siguiente me fui por subir los datos, subi únicamente 20 registros :

curl -XPOST "http://localhost:9200/log\_consultas/\_doc/" -H 'Content-Type: application/json' -d '

{

"@timestamp":"2010-05-15T19:03:27","estado\_consulta":"informativo","servicio":"modificacion","administrador":"Juan Carlos","consultas\_requeridas":15

}

'

## Realizar busquedas sobre el indice:

### Búsqueda 1.- Obtener el numero de registros con estado\_consulta igual a erro y consumo

--bool permite realizar una busqueda que debe coincidir con uno de los terminos especificados dentro del campo que estamos evaluando

--should dentro de bool permite especificar multiples condiciones

curl -XPOST "http://localhost:9200/log\_consultas/\_count" -H 'Content-Type: application/json' -d '

{

"query": {

"bool": {

"should": [

{ "term": { "estado\_consulta": "error" }},

{ "term": { "estado\_consulta": "consumo" }}

]

}

}

}'

### Búsqueda 2.-Obtener el numero de registros realizados por el administrador Juan Lara

-- term nos ayuda a buscar exactamente la cadena que estamos poniendo como valor.

curl -XPOST "http://localhost:9200/log\_consultas/\_search" -H 'Content-Type: application/json' -d '

{

"query": {

"term": {

"administrador.keyword": "Juan Lara"

}

}

}'

### Búsqueda 3.-Obtener el numero de registros con estado consulta igual a informativo y servicio igual a borrado

curl -XPOST "http://localhost:9200/log\_consultas/\_count" -H 'Content-Type: application/json' -d '

{

"query": {

"bool": {

"should": [

{ "term": { "estado\_consulta": "informativo" }},

{ "term": { "servicio": "borrado" }}

]

}

}

}'

### Búsqueda 4.-Obtener la suma de los valores en consultas realizadas con estado\_consulta igual a error

----primero ponemos el filtro que necesitamos y posteriormente se realiza la suma con el comando aggs, solamente se suma el universo que tenemos como resultado del filtro

Aquí el campo se llama consultas requeridas no consultas realizadas, igual deje ambas consultas.

curl -XPOST "http://localhost:9200/log\_consultas/\_search" -H 'Content-Type: application/json' -d '

{

"query": {

"match": {

"estado\_consulta": "error"

}

},

"aggs": {

"total\_consultas\_error": {

"sum": {

"field": "consultas\_requeridas"

}

}

}

}'

----------------------

curl -XPOST "http://localhost:9200/log\_consultas/\_search" -H 'Content-Type: application/json' -d '

{

"query": {

"match": {

"estado\_consulta": "consumo"

}

},

"aggs": {

"total\_consultas\_error": {

"sum": {

"field": "consultas\_requeridas"

}

}

}

}'

## Realizar un tablero

### 1.- vista de heat map

#### Paso 1. Entre a visualize\_library y entras a crear una nueva, ingresas a esta opción de heat map y eliges tu indice

A screenshot of a web page

Description automatically generated

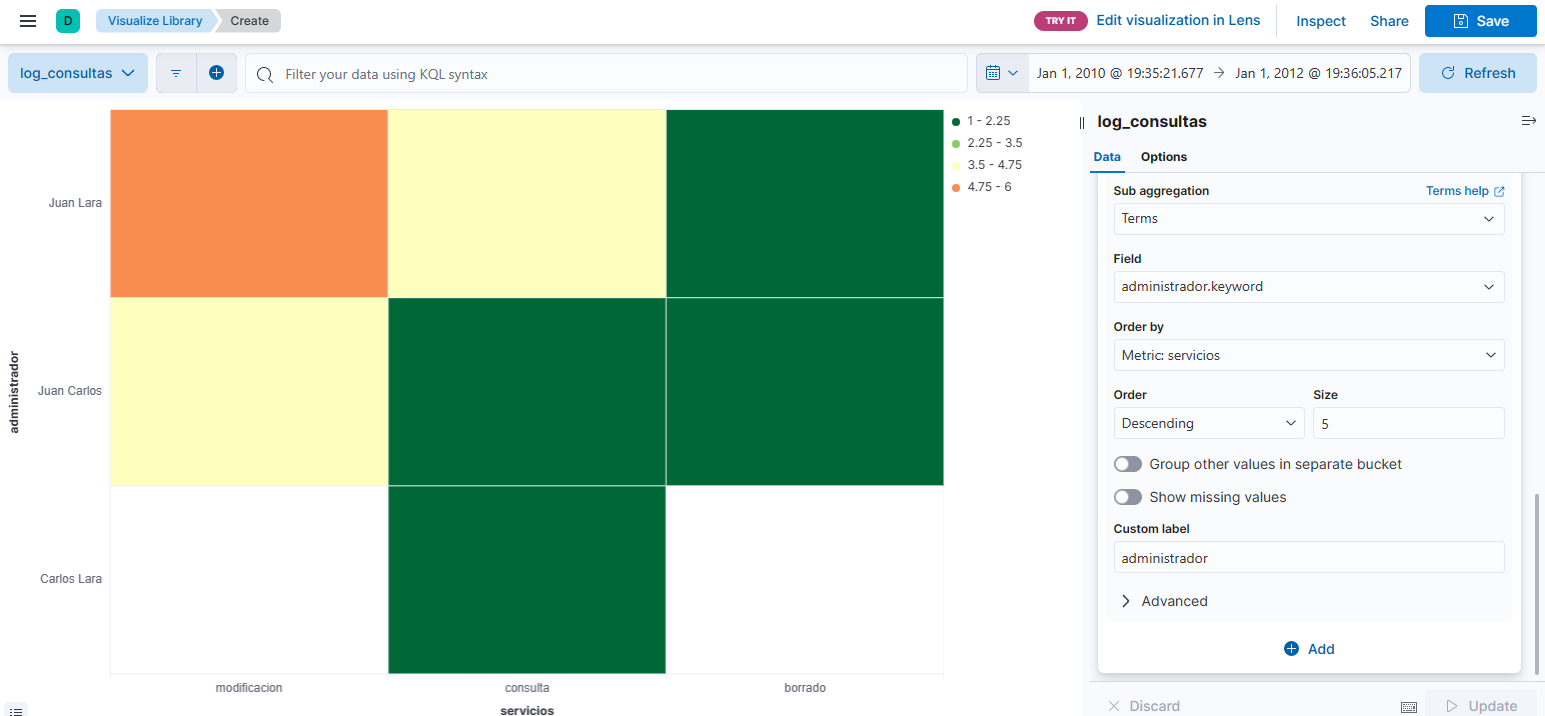
#### Paso 2. Configuras tu vista

A screenshot of a computer

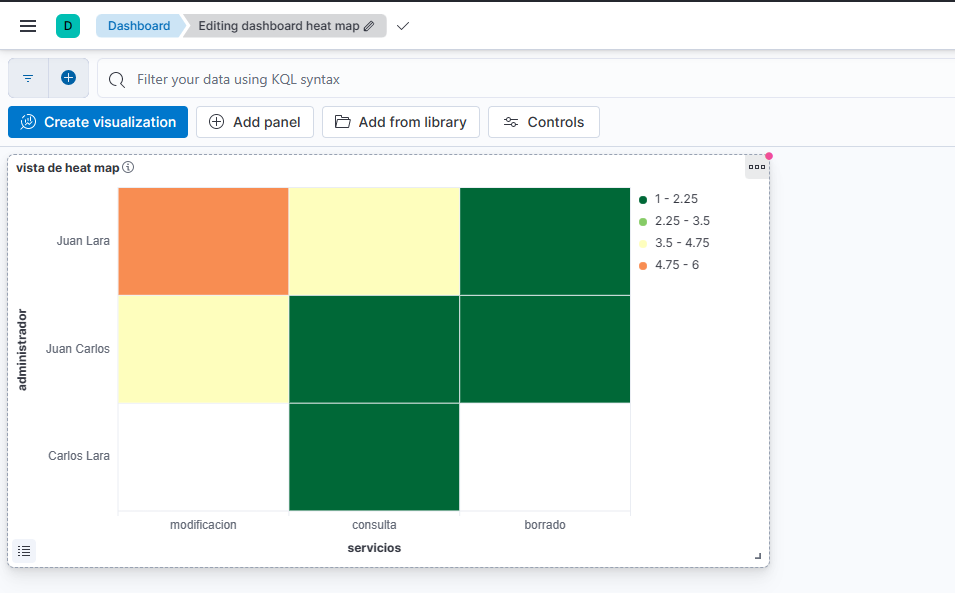
Description automatically generated

A screenshot of a computer

Description automatically generated



#### Paso 3. Guardas y creas tu dasboard



### 2.- vista de barras

#### Paso 1.

Entre a visualize\_library y entras a crear una nueva, ingresas a esta opción

A screenshot of a computer

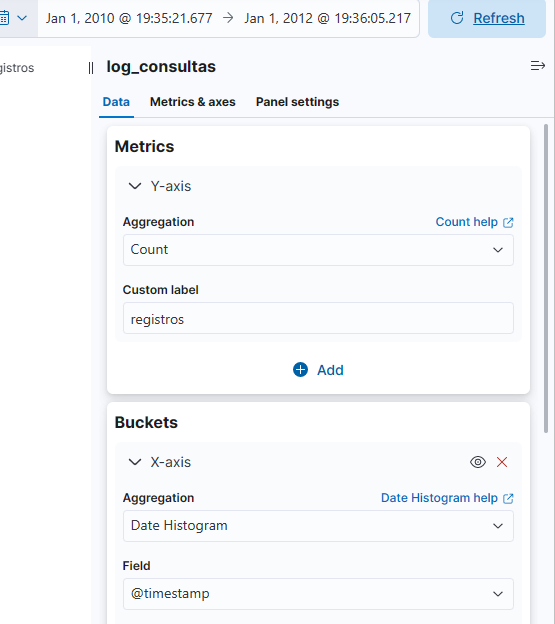
Description automatically generated

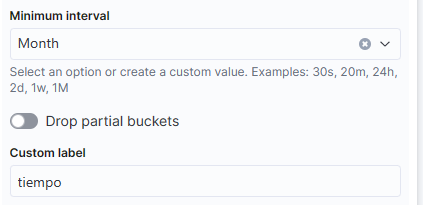
#### Paso 2.Eliges la opción de barras

A white rectangular sign with black text

Description automatically generated

#### Paso 3.Eliges tu índice , posteriormente agregas configuras tu visualización yo agregue solamente de 2010 a 2012 ya que los registros traen fecha de 2010 y la visualización de tiempo la puse por mes



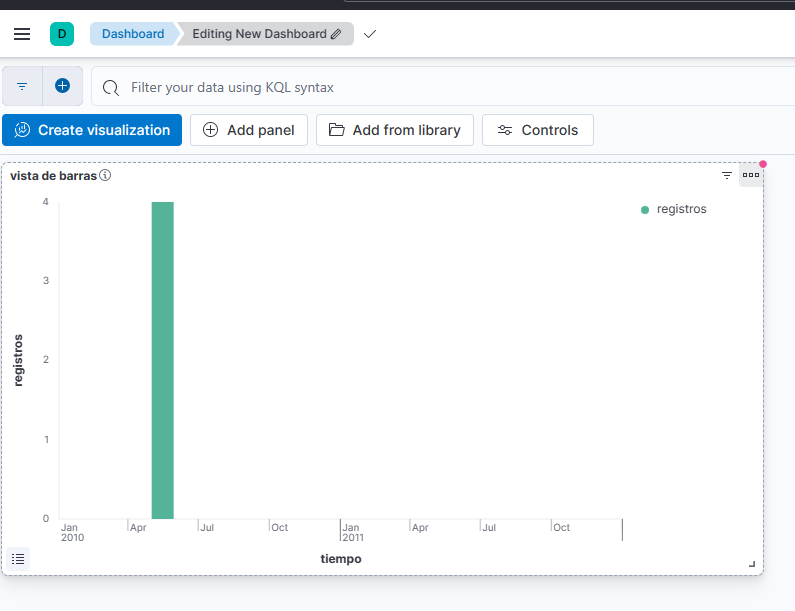


Paso 4.Configuras tu filtro

A person pointing at a line

Description automatically generated with medium confidence

#### Paso 4.Por último lo guardas y creas un dasboard.



## Realizar un dasboard para visualizar las vistas

En el paso anterior desde la vista sale la opción de realizar tu dasboard al guardar tu vista yo le di aceptar y cree uno nuevo , solamente agregue los nombres .

## Crear un repositorio

Creas tu repositorio

A screenshot of a web page

Description automatically generated

Configuras una carpeta donde estará tu repositorio

