

Proyecto FINAL

ESCUELA POLITÉCNICA NACIONAL **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

MAURICIO RODRÍGUEZ
JOSUE GARRIDO
XAVIER JARAMILLO
mauricio.rodriguez01@epn.edu.ec
francisco.jaramillo01@epn.edu.ec
josue.garrido@epn.edu.ec

Antes de instalar cualquier programa se debe tener en cuenta lo siguiente:

- El espacio en disco
- El sistema operativo de 32 o 64 bits
- La Capacidad de RAM
- Y Versión de tu maquina

En esta ocasión procederemos con el entendimiento de como instalar programas dedicados en este caso de programación. Con enfoque a minería de Base de Datos Multinacionales y la debida utilización de herramientas dedicadas al mismo.

I. DEFINICION DEL CASO DE ESTUDIO

El caso de estudio que hemos de realizar con la recopilación de información utilizaremos filtros de palabras para poder subdividir los temas presentados en los ítems, para posteriormente poder analizarlos mediante herramientas antes vistas en el semestre como por ejemplo: KIBANA, GOOGLE CHART, con esta información vamos a visualizar de manera gráfica, sencilla y fácil de entender.

II. OBJETIVOS GENERALES Y ESPECIFICOS

Objetivo General: Identificar los datos concentrados en cada una de las temáticas mediante una interfaz gráfica.

Objetivo Especifico: Analizar la mejor opción para el almacenamiento masivo de datos.

III. DESCRIPCION DEL EQUIPO DE TRABAJO Y ACTIVIDADES DE TRABAJO RESALIZADAS POR CADA UNO.

Nuestro equipo de trabajo siendo analistas en sistemas de la información debemos tener un claro y amplio entendimiento del uso de datos ya que estos están inmersos en diferentes campos ya sean profesionales o no.

Actividades:

Josué Garrido:

- o Indagación de base de datos SQL
- o Selección de la base de datos SQL
- o Instalación de la base de datos SQL
- o Realización de la minería de datos en las plataformas twitter y Facebook

- o Recopilación de datos
- o Creación del dashboard en kibana
- o Presentación de datos

Xavier Jaramillo:

- o Indagación de base de datos No SQL
- o Elección de base de datos No SQL
- o Descarga de los softwares (couchdb y mongo db)
- o Instalación
- o Realización de la minería de datos en couchdb
- o Creación del dashboard en kibana
- o Presentación de datos

Mauricio Rodríguez:

- o Instalación de base de datos No SQL (couchdb y mongo db)
- o Realización de la minería de datos en Mongodb
- o Recopilación de datos
- o Creación del dashboard en kibana
- o Presentación de datos

IV. CRONOGRAMA DE ACTIVIDADES



V. RECURSOS Y HERRAMIENTAS USADAS

NO SQL

Couch DB

Procedemos a instalar como primera instancia a Couchdb ya que con esta herramienta podremos visualizar los datos obtenidos de los siguientes scripts de Python los cuales describiremos mas adelante para ello podemos dirigirnos al siguiente link de descarga.

<https://couchdb.apache.org/>

damos click en la sección de descargar Couch db

Seamless multi-master sync, that scales from **Big Data** to **Mobile**, with an **Intuitive** HTTP/JSON API and designed for **Reliability**.

DOWNLOAD
2.3.1 (2019-03-12)



Una vez descargado lo corremos y lo instalamos con las respectivas guías de instalación una vez instalado esto procedemos a irnos a una terminal y digitamos cmd

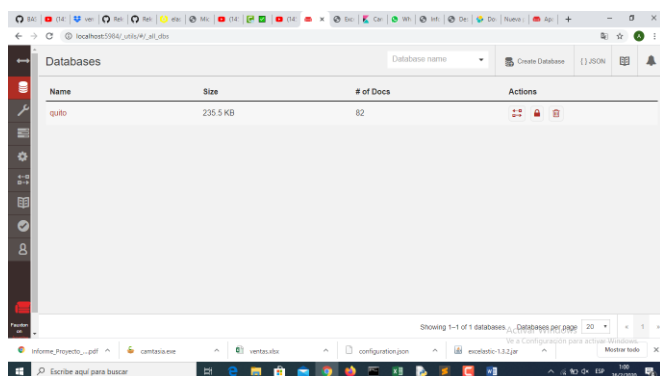
Para luego poner el siguiente código el cual nos ayudara a instalar de manera definitiva a CouchDB

```
C:\Users\HP>pip install couchdb
Collecting couchdb
  Downloading CouchDB-1.2-py2.py3-none-any.whl (67 kB)
    |#####| 67 kB 357 kB/s
Installing collected packages: couchdb
Successfully installed couchdb-1.2
```

C:\Users\HP>

Una vez instalado couchdb procedemos a instalar las siguientes herramientas.

Verificamos la correcta instalación yendo a un navegador para introducir localhost:5984/_utils



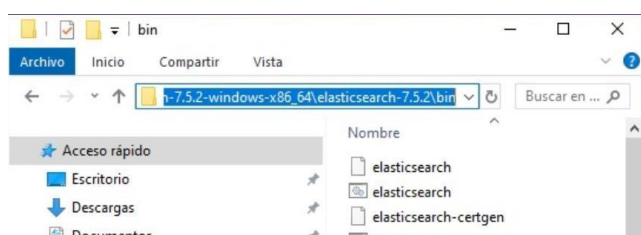
Elasticsearch

Esta es una de las herramientas mas utilizadas en este tipo de proyectos ya que con esta podemos realizar la minería de los datos para luego transformarlos a algún tipo de documento para su respectiva verificación para el usuario final en otras palabras que este sea mas entendible y prácticamente sea fácil de manejar todos los datos descargados.

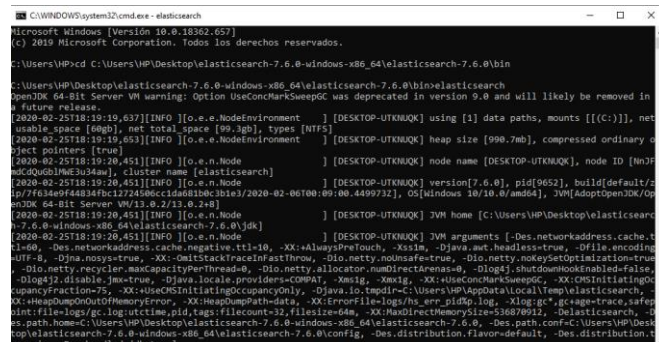
Como primer paso procedemos a descargarnos elasticsearch en el sitio oficial o en el link luego descrito

<https://www.elastic.co/es/downloads/elasticsearch>

una vez en el área de descarga procedemos a bajarnos el .zip de elasticsearch una vez descargado nos dirigimos a la carpeta previamente descomprimida para entrar en la carpeta bin del mismo.



Una vez luego de copiar la dirección de la carpeta bin lo que realizamos es abrir un cmd para luego correr elasticsearch

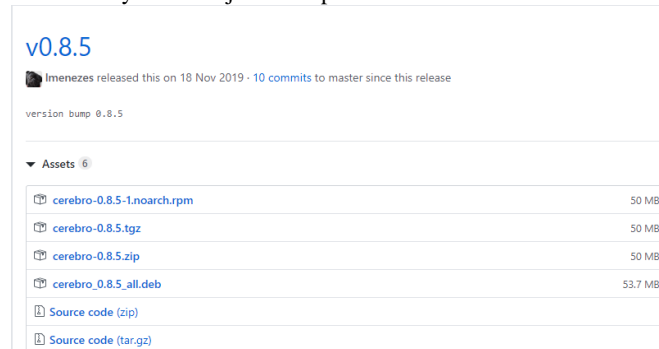


Cerebro

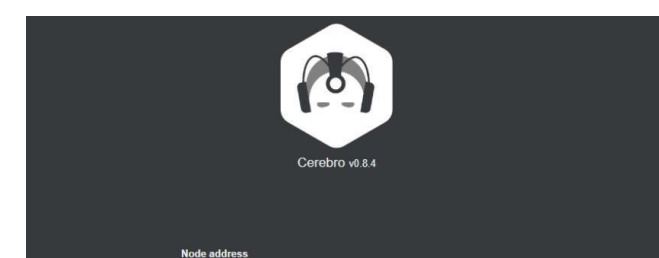
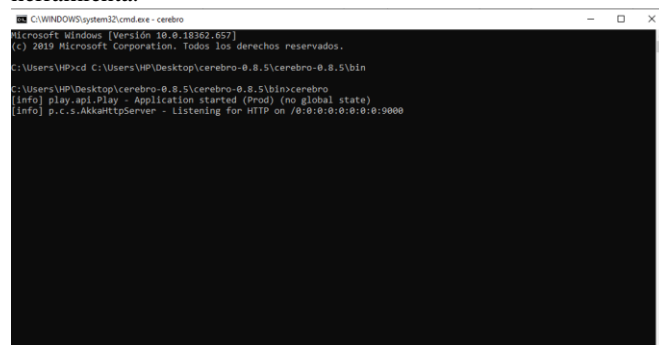
De la misma manera como elasticsearch lo que realizaremos es la descarga pertinente del mismo para ello nos ubicamos en la siguiente dirección o link

<https://github.com/lmenezes/cerebro/releases>

nos dirigimos al link y nos lleva a un repositorio el cual nos ubicamos en el respectivo link de descarga de cerebro damos clic y se nos bajara en .zip

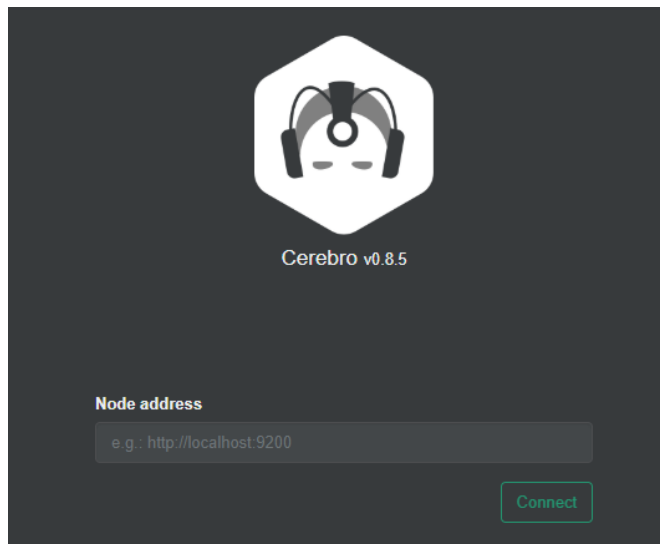


Luego de esto nos dirigimos a la carpeta de bin copiamos la dirección y con la ayuda de un cmd corremos a esta herramienta.

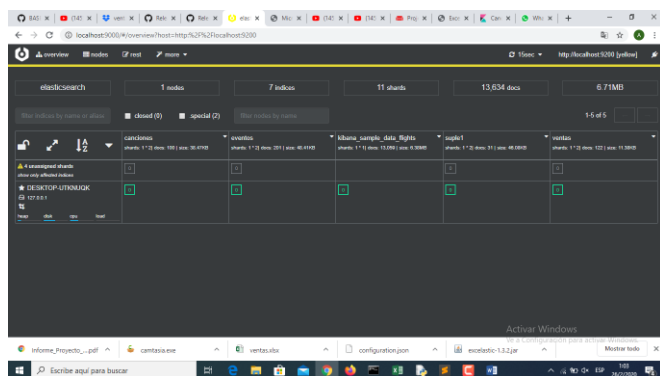


Una vez en la pagina principal de cerebro corremos el siguiente código que nos llevara a elasticsearch

Base de Datos Multidimensionales



Aquí digitamos <http://localhost:9200> para que nos lleve a la pagina principal de elasticsearch.



Una vez instalada esta herramienta procedemos a instalar la ultima que es la siguiente.

Kibana

Como en las herramientas anteriores nos dirigimos al sitio oficial de descarga en el siguiente link.

<https://www.elastic.co/es/downloads/kibana>

Want it hosted? Deploy on Elastic Cloud. [Get Started >](#)

Version: 7.6.0

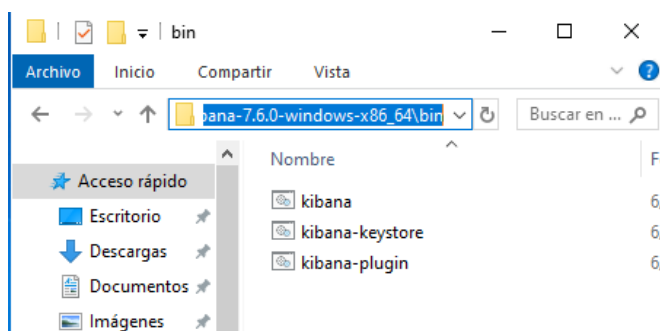
Release date: February 11, 2020

License: [Elastic License](#)

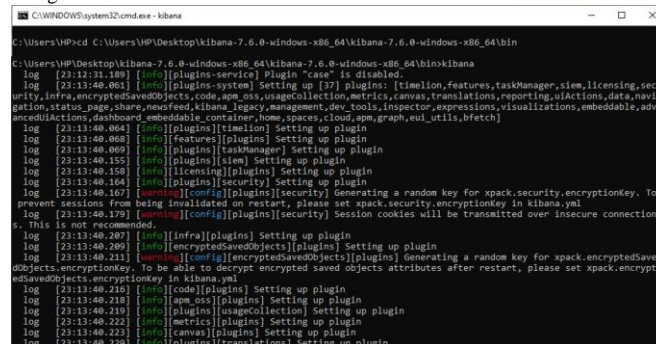
Downloads: [WINDOWS](#) sha asc [MAC](#) sha asc
[LINUX 64-BIT](#) sha asc [RPM 64-BIT](#) sha asc
[DEB 64-BIT](#) sha asc

Package Managers: Install with [yum, dnf, or zypper](#)
Install with [apt-get](#)
Install with [homebrew](#)

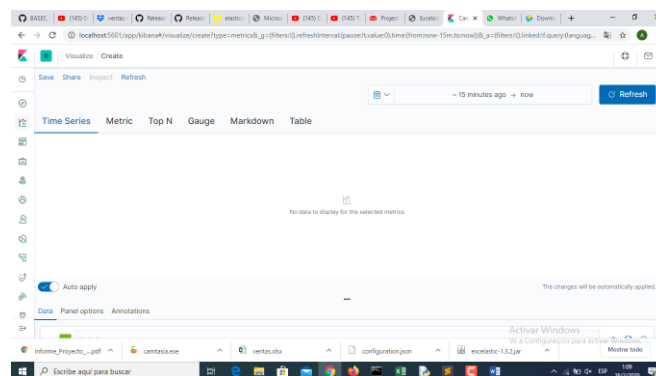
Una vez descargado el .zip los descomprimos y nos dirigimos a la carpeta bin para mandar a correr esta herramienta.



Con la ayuda de un cmd logramos levantar a Kibana como se muestra en lo siguiente.



Como podemos observar ya hemos logrado levantar a Kibana para poder verificar esto nos dirigimos a un navegador y digitamos localhost:5601



SQL

Herramientas necesarias:

- Java SE Development Kit (última versión)
- Visual Studio 2012
- Elasticsearch 7.0

<https://www.elastic.co/es/downloads/past-releases/elasticsearch-7-0-0>

- Cerebro

<https://github.com/lmenezes/cerebro/releases>

- Kibana 7.0

<https://www.elastic.co/es/downloads/past-releases/kibana-7-0-0>

- Logstash 7.0

<https://www.elastic.co/es/downloads/past-releases/logstash-7-0-0>

- JDBC para SQL SERVER (verificar versión según versión sql)

<https://www.microsoft.com/es-es/download/details.aspx?id=11774>

Paso 1)

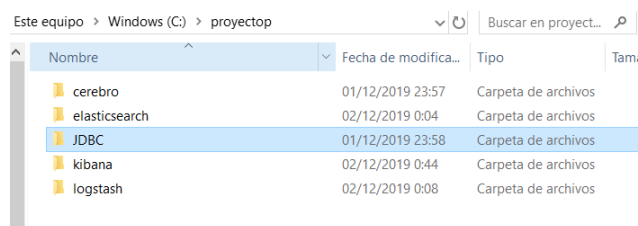
Descargamos todas las herramientas necesarias estas las podemos descargar desde los siguientes enlaces.

cerebro-0.8.4.zip	05/11/2019 14:39	Archivo WinRAR ZIP
elasticsearch-7.0.0-windows-x86_64.zip	01/12/2019 23:50	Archivo WinRAR ZIP
elasticsearch-7.4.2-windows-x86_64.zip	05/11/2019 14:46	Archivo WinRAR ZIP
kibana-7.0.0-windows-x86_64.zip	01/12/2019 23:29	Archivo WinRAR ZIP
kibana-7.4.2-windows-x86_64.zip	05/11/2019 8:54	Archivo WinRAR ZIP
logstash-7.4.2.zip	05/11/2019 14:41	Archivo WinRAR ZIP
sqljdbc.4.2.8112.200_enu.exe	08/11/2019 11:59	Aplicación

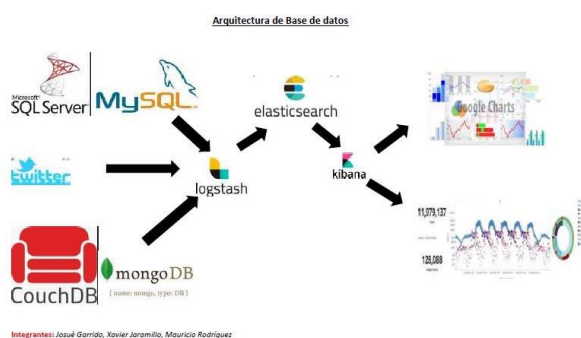
Base de Datos Multidimensionales

Paso 2)

Se van a descargar en unos zip los cuales deberán ser descomprimidos dentro de una carpeta en la raíz incluido en **sqljdbc** que por defecto estará en un “.exe”, tendríamos algo así:



VI. ARQUITECTURA DE LA SOLUCION



VII. EXTRACCION DE DATOS NO SQL

Para esto utilizamos la cosecha.py y utilizamos los diferentes filtros a continuación los mostramos

Código fuente de cosecha.py

```
import couchdb #Libreria de CouchDB (requiere ser instalada primero)
from tweepy import Stream #tweepy es la librería que trae tweets desde la API de Twitter (requiere ser instalada primero)
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import json #Libreria para manejar archivos JSON

#####Credenciales de la cuenta de Twitter#####
#Poner aqui las credenciales de su cuenta privada, caso contrario la API bloqueara esta cuenta de ejemplo
ckey = "HvCw23VwYgWnVe5krvHKuZGO"
csecret = "KUwAEDDqogw0xHr1LQiaECD4q42gpFRPkUS0NLHZm51Y9gMRF"
atoken = "115946548-x5JL2LFNp26y2t0WIBvQE7GirkGeNZUaYLUM1GGL"
asecret = "v12Q4nw9112ATj4iyFu2Vj12IEIG6OGRIr3wBnBPxBAE"
#####

class listener(StreamListener):
    def on_data(self, data):
        dictTweet = json.loads(data)
        try:
            dictTweet['_id'] = str(dictTweet['_id'])
            #Antes de guardar el documento puedes realizar parseo, limpieza y cierto analisis o filtrado de datos previo
            #a guardar en documento en la base de datos
            doc = db.save(dictTweet) #Aqui se guarda el tweet en la base de couchDB
            print ("Guardado " + "> " + dictTweet['_id'])
        except:
            print ("Documento ya existe")
        pass
        return True

    def on_error(self, status):
        print (status)

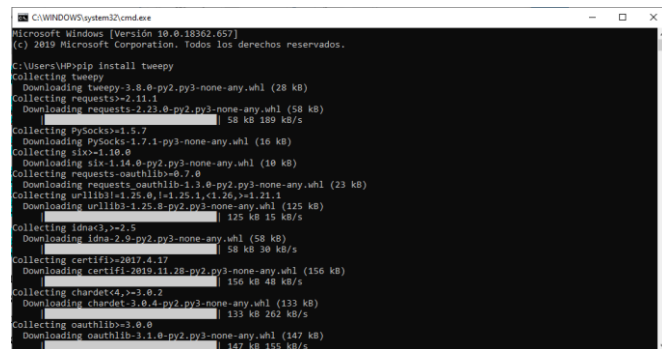
auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)
twitterStream = Stream(auth, listener())

#Setear la URL del servidor de couchDB
server = couchdb.Server('http://localhost:5984/')
try:
    #Si no existe la Base de datos la crea
    db = server.create('cancionesfamosas')
except:
    #Caso contrario solo conectarse a la base existente
    db = server['cancionesfamosas']

#Aqui se define el bounding box con los limites geograficos donde recolectar los tweets
twitterStream.filter(track=["position 1","reggaeton","mundo","canciones","bad bunny","rosalia"])
#twitterStream.filter(locations=[-78.586922,-0.395161,-78.274155,0.021973])
```

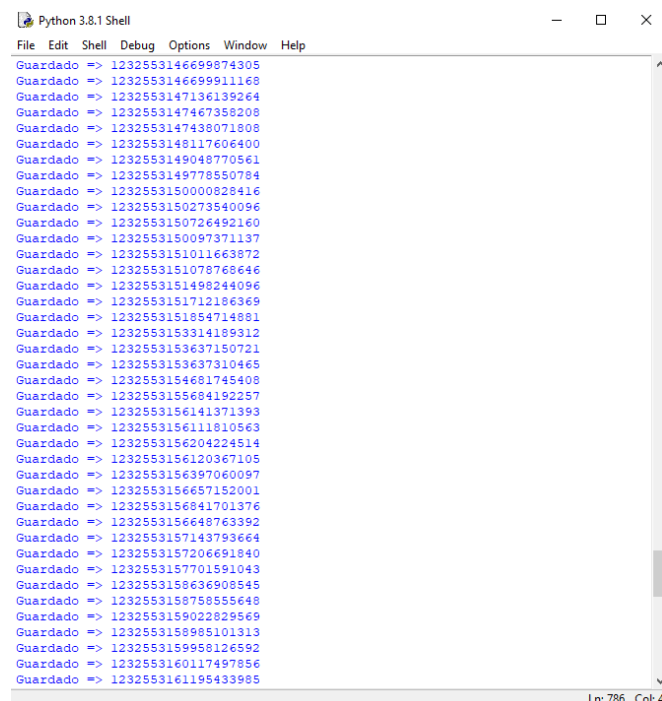
Como podemos darnos cuenta en este archivo de Python se nos hace necesario instalar algunas de las siguientes

herramientas para ello nos dirigimos a la terminal y damos en cmd

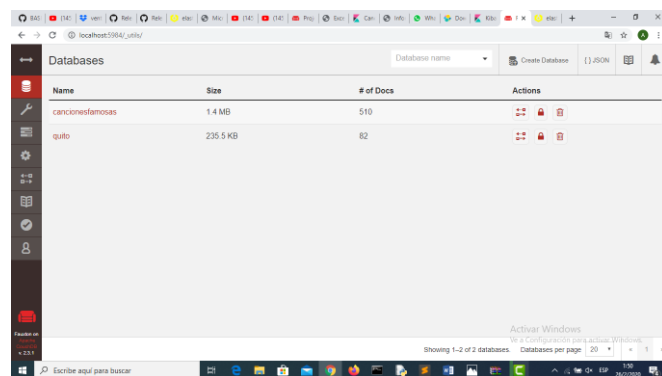


Instalamos una extensión de Python el cual es tweepy

Una vez instalada esta herramienta procedemos a modificar el archivo y una vez que lo corrijamos podemos correr el archivo y se vera algo así.



Luego de esto podemos verificar yéndonos al terminal de Couchdb



Como podemos observar tenemos archivos en .json los cuales con ayuda de herramientas como excelastic podemos transformar dichos archivos a .xlsx los cuales se los puede cargar directamente a elasticsearch de la siguiente manera.

Base de Datos Multidimensionales

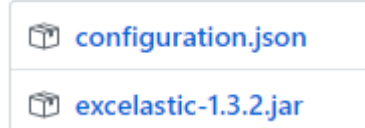
Nos dirigimos al siguiente link.

<https://github.com/codingchili/excelastic/releases>

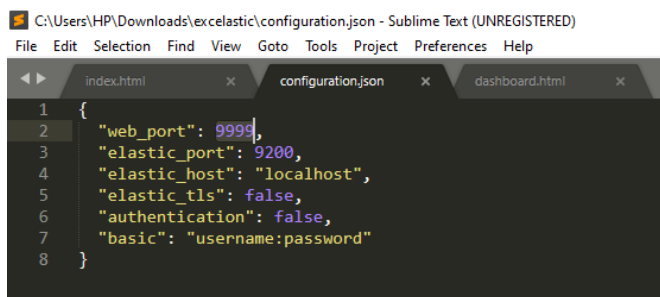
luego de ello escogemos la opción de



para luego descargar los siguientes archivos.



Una vez descargados nos vamos a la carpeta en donde fueron descargados y en el archivo configuration.json lo editamos de la siguiente manera.

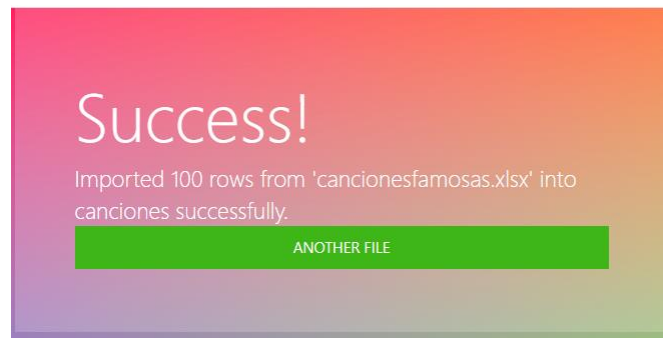


En dicho archivo pondremos un puerto el cual no estemos utilizando en este caso lo pusimos 9999

Y nos dirigió de manera automática a esta pantalla principal

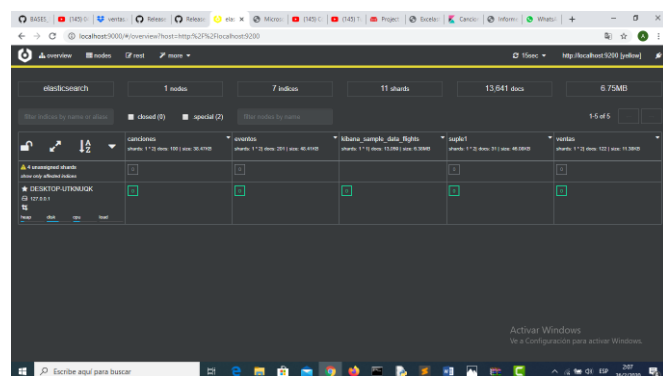


Una vez aquí ponemos el nombre de la base de datos que queramos y damos clic en UPLOAD para cargar nuestro archivo de Excel



Una vez que todo salga correctamente nos saldrá un recuadro como el anterior mostrado.

Con ello nos dirigimos al navegador y verificamos en la pantalla principal de elasticsearch si se nos subió correctamente.

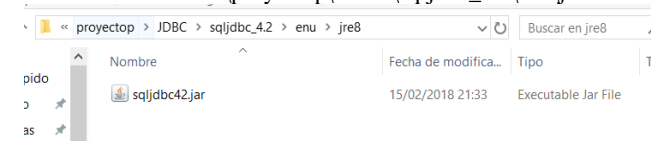


Listo como podemos ver si se nos subió de la mejor manera posible todas las bases de datos y con ello procedemos a ir a Kibana para efectuar las respectivas gráficas y visualización de datos existentes.

SQL

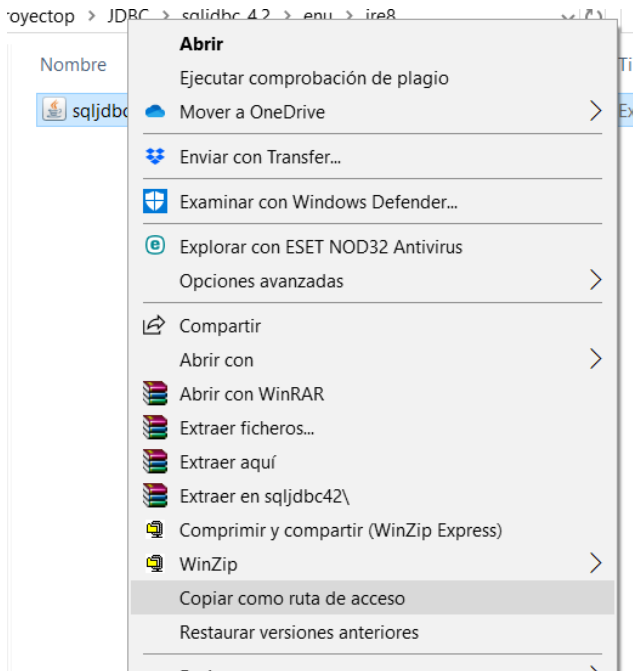
Ahora tenemos listo donde vamos a trabajar, falta importar los datos, para esto vamos a configurar nuestro CLASSPATH en las variables de entorno de Windows, para esto vamos al directorio donde habíamos extraído el JDBC.

En mi caso es esta “C:\proyectop\JDBC\sqljdbc_4.2\enu\jre8”

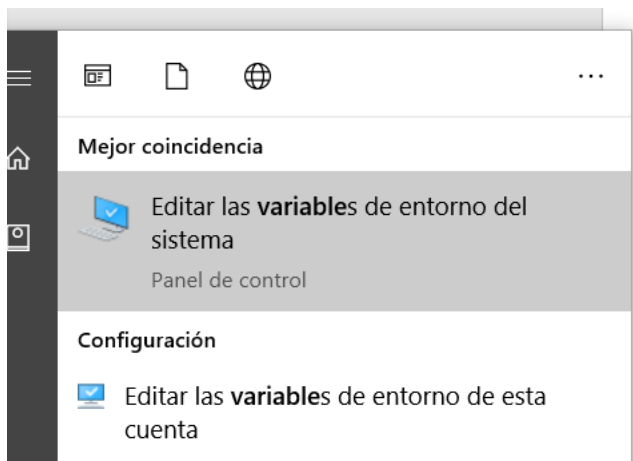


Copiamos como ruta de acceso el “.jar” haciendo **shift+click derecho** sobre el archivo.

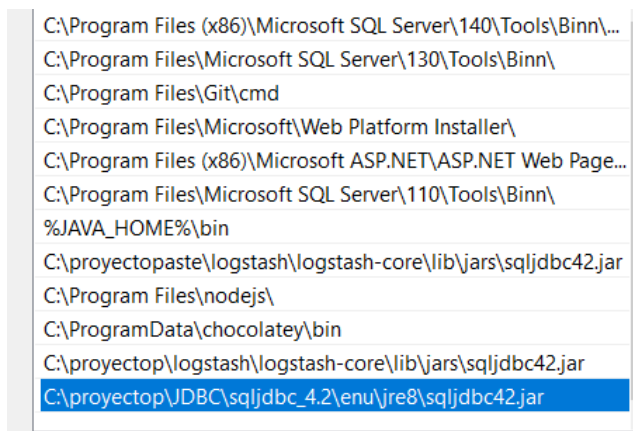
Base de Datos Multidimensionales



Vamos a las variables de entorno de Windows



Y pegamos la ruta copiada anteriormente en la CLASSPATH del sistema

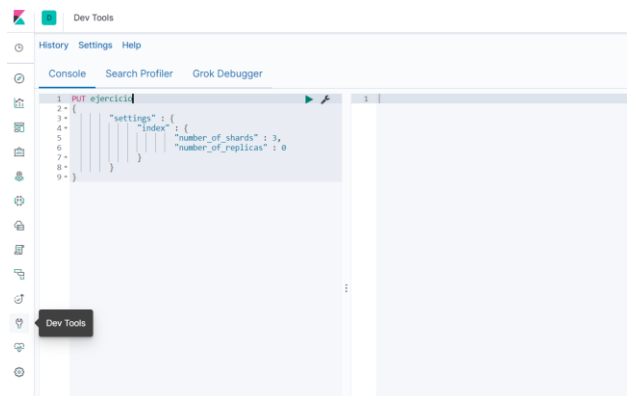


Ahora vamos a crear un índice en Elasticsearch por medio de Kibana para importar los datos.

Ya tenemos corriendo los servicios y si no los tenemos abiertos, corremos de nuevo Elasticsearch, cerebro y Kibana ya que los vamos a utilizar.

Una vez ya abiertos todos los servicios nos dirigimos a Kibana.

Hacemos click en el menú lateral en la herramienta **“Dev Tools”**



Y pegamos el siguiente código

PUT ejercicio

```
{
  "settings": {
    "index": {
      "number_of_shards": 3,
      "number_of_replicas": 0
    }
  }
}
```

En este caso el índice se llamará “ejercicio”. Corremos el código en el símbolo de “play” y después de unos segundos tendremos esto.



De igual manera en cerebro se nos mostrará nuestro índice creado



Bien ya tenemos nuestro índice, debemos configurar Logstash en un archivo para la migración de datos para esto vamos a donde tenemos extraído Logstash y creamos un archivo llamado **“sql.conf”** el cual nos permitirá extraer los datos de la base de datos SQL.

```
input {

  jdbc {

    jdbc_connection_string =>
    "jdbc:sqlserver://uiosrv25:1433;databaseName=ControlProduccion;integratedSecurity=true;"

    jdbc_driver_class =>
    "com.microsoft.sqlserver.jdbc.SQLServerDriver"

    jdbc_driver_library =>
    "C:\proyectop\JDBC\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"

    jdbc_user => "consulta"
```

```
jdbc_password => "consulta2016"
```

```
statement => "select (CONVERT(datetime, fecha, 111) + ' '+  
CONVERT(datetime, HoraStr + ':' + MinutoStr, 108)) as FechaHour ,  
Maquina, Cabezal,
```

```
sum(UndProducidas)*60.0/sum(SegundosDuracion)as  
UnidadesProducidas
```

```
from DatosPlcAdipackV1 where
```

```
OnOff =1 and
```

```
DateStr >= DATEADD(DAY, -2, GETDATE())
```

```
group by Fecha, HoraStr, MinutoStr, Maquina, Cabezal
```

```
order by Fecha, HoraStr, MinutoStr, Maquina, Cabezal"
```

```
}
```

```
}
```

```
output {
```

```
elasticsearch {
```

```
hosts => ["localhost:9200"]
```

```
index => "ejercicio"
```

```
}
```

```
}
```

Abrimos otra ventana de cmd donde tenemos instalados el logstash

```
C:\proyectop\logstash>
```

Y con el siguiente comando importamos los datos en nuestro índice

```
bin\logstash -f sql.conf
```

```
C:\proyectop\logstash>bin\logstash -f sql.conf
```

```
2019-12-05T11:10:59,020 [INFO] [logstash.outputs.elasticsearch][main] Using default mapping template
2019-12-05T11:10:59,054 [WARN] [org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.specialized.RubyArray$OneObject) has been create for key: cluster.uids. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team.
2019-12-05T11:10:59,063 [INFO] [logstash.javapipeline][main] Starting pipeline (:pipeline_id=>"main", :pipeline.workers=>8, :pipeline.batch.size=>125, :pipeline.batch.delay=>50, :pipeline.max_inflight=>1000, :threads=>#{thread:0x3f34fed run})
2019-12-05T11:10:59,222 [INFO] [logstash.outputs.elasticsearch][main] Attempting to install template (:manage_templates=>{"index_patterns">["logstash-*"], "version">60001, "settings">{"index.refresh_interval">"5s", "number_of_shards">1}, "mappings">{"dynamic_templates">[{"message_field">{"path_match">"message", "match_mapping_type">"string", "mapping">{"type">"text", "norms">false}}, {"string_fields">{"match">"*", "match_mapping_type">"string", "mapping">{"type">"text", "norms">false, "fields">{"keyword">{"type">"keyword", "ignore_above">256}}}], "properties">{"@timestamp">{"type">"date"}, "@version">{"type">"keyword"}, "geoip">{"dynamic">true, "properties">{"ip">{"type">"ip"}, "location">{"type">"geo_point"}, "latitude">{"type">"half_float"}, "longitude">{"type">"half_float"}}}}])
2019-12-05T11:11:00,777 [INFO] [logstash.javapipeline][main] Pipeline started (:pipeline_id=>"main")
2019-12-05T11:11:01,051 [INFO] [logstash.agent] Pipelines running (:count=>1, :running_pipelines=>[main], :not_running_pipelines=>[])
2019-12-05T11:11:00,425 [INFO] [logstash.agent] Successfully started Logstash API endpoint (:port=>9600)
2019-12-05T11:11:15,912 [INFO] [logstash.inputs.jdbc][main] (11.330/32s) select (CONVERT(datetime, fecha, 111) + ' '+ CONVERT(datetime, HoraStr + ':' + MinutoStr, 108)) as FechaHour , Maquina, Cabezal, sum(UndProducidas)*60.0/sum(SegundosDuracion)as UnidadesProducidas from DatosPlcAdipackV1 where OnOff =1 and DateStr >= DATEADD(DAY, -2, GETDATE()) group by Fecha, HoraStr, MinutoStr, Maquina, Cabezal order by Fecha, HoraStr, MinutoStr, Maquina, Cabezal
```

Verificamos en Kibana

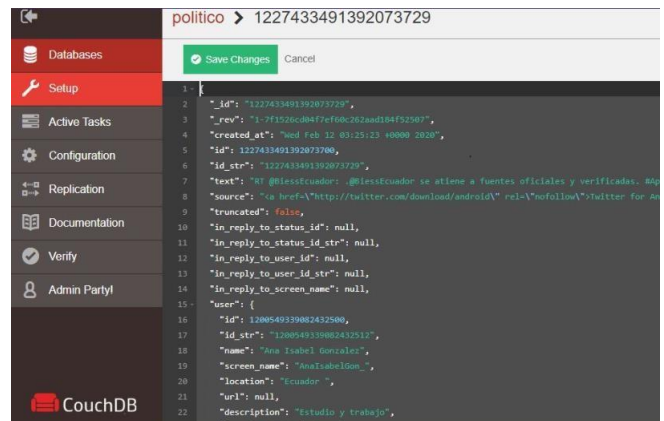
```
1 GET ejercicio/_count
2 {
3   "count": 2970,
4   "total": 3,
5   "successful": 3,
6   "skipped": 0,
7   "failed": 0
8 }
9
10
```

Con el siguiente comando

```
GET cs_users/_count
```

VIII. ANALISIS DE INFORMACION

En los diferentes casos y diferentes ítems los cuales obtuvimos podemos observar que al momento de visualizar y minar los datos de algunos de los temas no era factible ya que en ocasiones el y los filtros ya que palabras clave en los determinados temas existen conflictos al momento de minar



Es fácil darse cuenta de que los datos se nos muestran de tal manera que son difíciles de entender para lo cual verificaremos una a una las bases de datos.

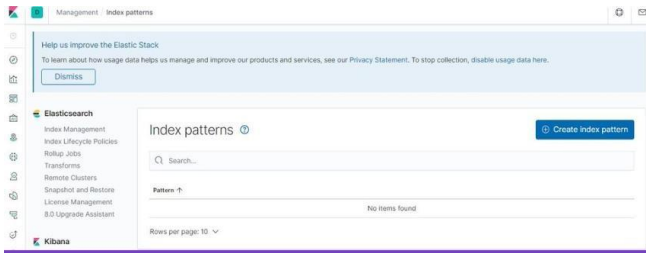
```
{ -
  "eventos": { -
    "alias": { -
      },
    "mappings": { -
      "properties": { -
        "ciudad": { -
          "type": "text",
          "fields": { -
            "keyword": { -
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "fecha": { -
          "type": "date"
        },
        "lugar": { -
          "type": "text",
          "fields": { -
            "keyword": { -
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "nombre": { -
          "type": "text",
          "fields": { -
            "keyword": { -
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "tipo": { -
          "type": "text",
          "fields": { -
            "keyword": { -
              "type": "keyword",
              "ignore_above": 256
            }
          }
        }
      }
    }
  }
}
```

Como podemos darnos cuenta se nos despliega un sin número de información la cual es necesaria de clasificarla con lo cual con la utilización de varias herramientas existentes lograremos que sean visibles de tal manera que sean entendibles para el usuario final.

Base de Datos Multidimensionales

IX. VISUALIZACION DE LA INFORMACION

Como podemos observar lo primero es minar los datos y demás en Kibana para lo cual levantamos lo siguiente



Luego de poner los datos en excelastic procedemos a la verificación de los mismos creando un nuevo INDEX PATTERN

Nos dirigimos al navegador y digitamos localhost:5601 y se nos presenta la página principal y ahí procedemos a realizar todo lo que a continuación se les mostrara.



Primero damos clic en créate index pattern.

Step 1 of 2: Define index pattern

Index pattern

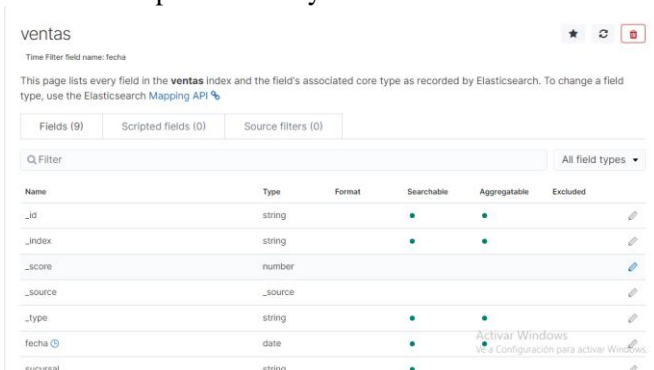
ventas

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

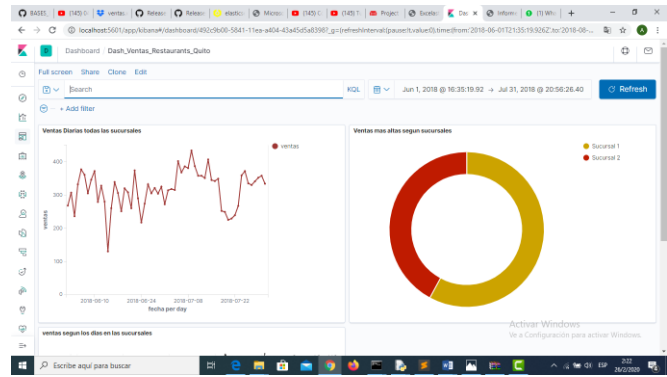
✓ **Success!** Your index pattern matches 1 index.

ventas

Como vemos en este caso se nos muestra la base de datos ventas previamente ya inicializada.

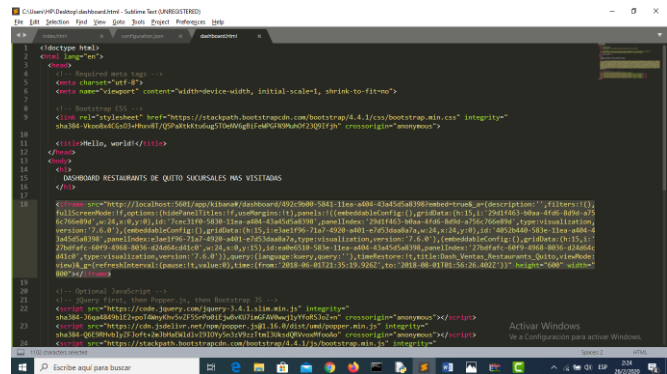


Aquí ya podemos observar todos los datos antes ingresados. Con ello ya podemos realizar las diferentes visualizaciones como a continuación se mostraran.



Y con ello podemos ver que hemos logrado visualizar los datos obtenidos desde twitter.

Una vez entendido estos diferentes pasos procedemos a verificar con base de datos SQL. Y para dar una mejor explicación de los mismos podemos compartir directamente o podemos pegar código html y visualizar en cualquier pagina.

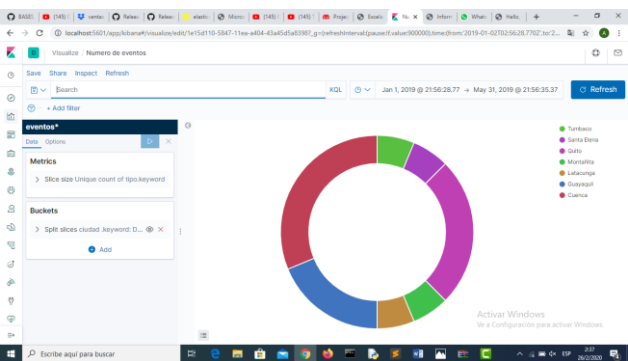
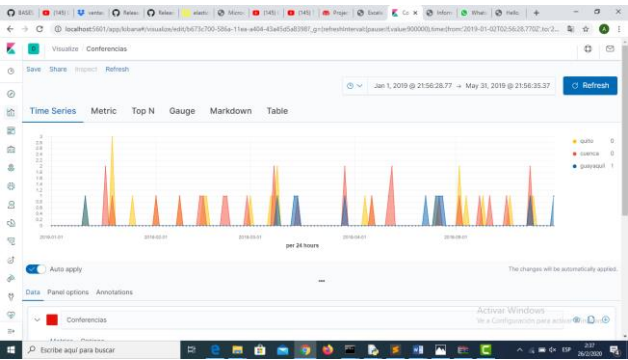
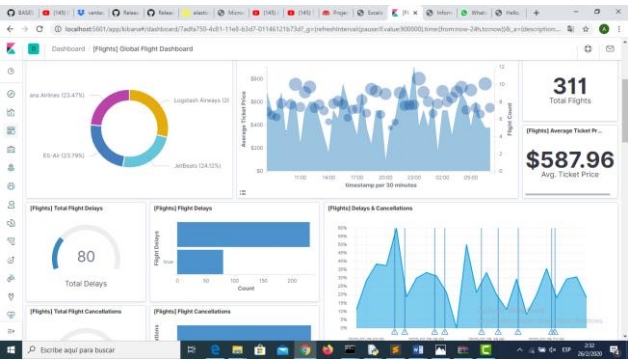
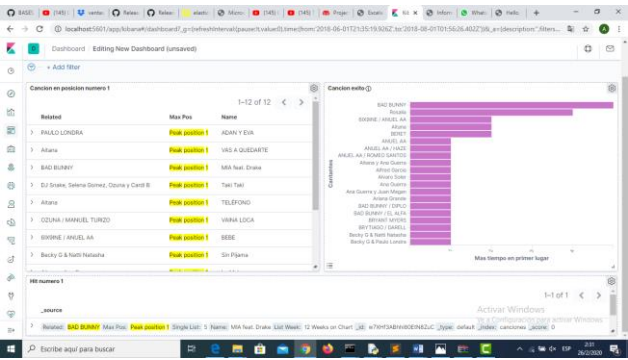
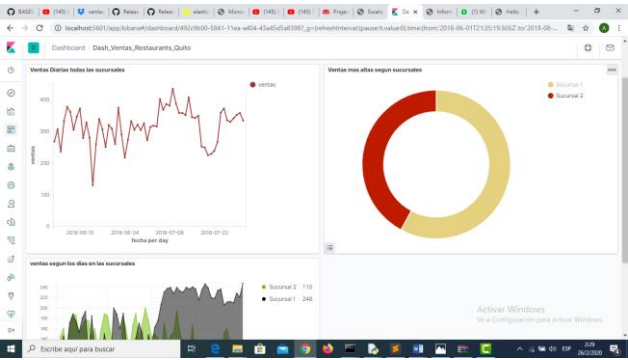


En este caso con la ayuda de sublime text logramos implementar y con la ayuda de bootstrap logramos dar una visualización adecuada para presentación profesional.

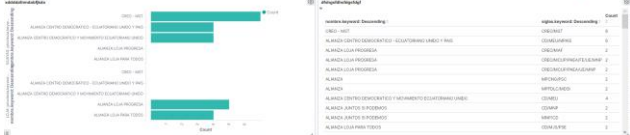
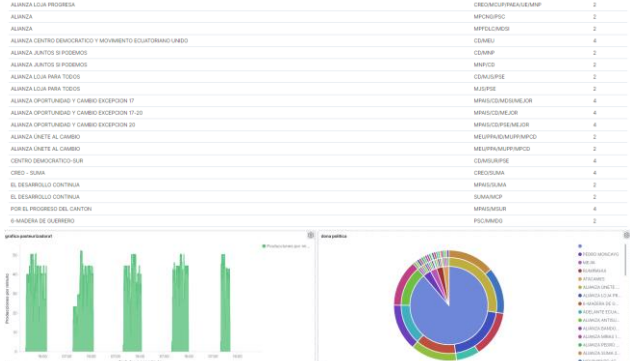
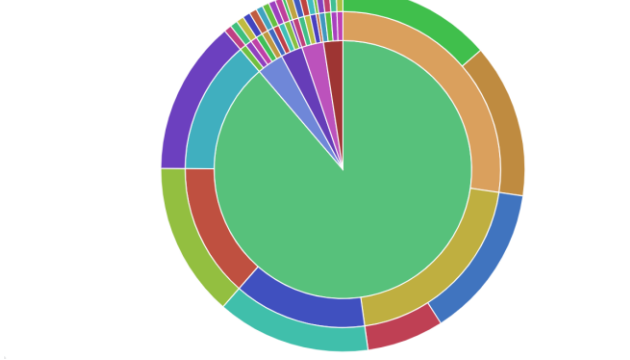
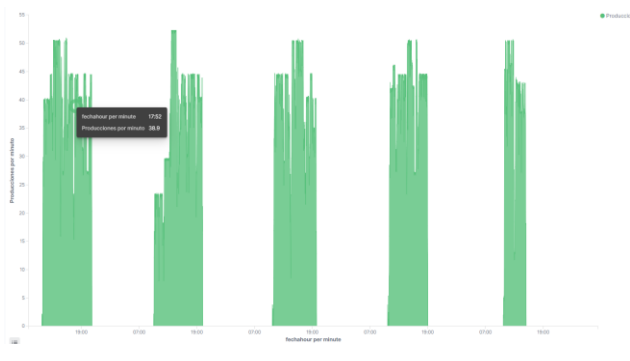
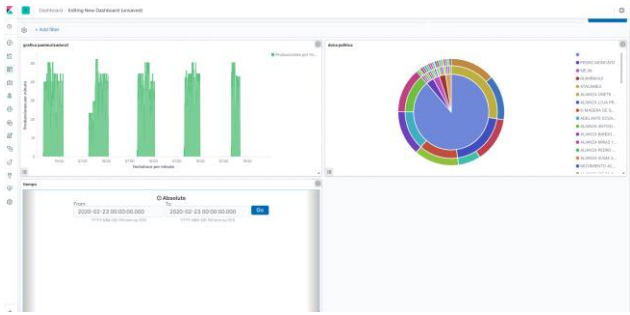


X. RESULTADOS OBTENIDOS

BASE DE NO SQL



BASES DE DATOS SQL



XI. CONCLUSIONES Y RECOMENDACIONES

Como principales conclusiones podemos decir que la extracción de datos es verdaderamente fácil ya que con los api de twitter y facebook podemos realizarlos de mejor manera y rápidamente y con ella la manipulación de los datos con kibana es verdaderamente eficiente.

Recomendamos primero transformar los datos extraídos con elasticsearch a un archivo de Excel para poder visualizarlos de mejor manera.

Es una manera muy eficiente para poder representar gráficamente los datos que tenemos en una base de datos sql u no sql, ya que podemos utilizarlas en aplicaciones web y ver las gráficas en cualquier lugar mediante estas.

Es un conocimiento adquirido muy importante ya que nos sirve para manejar datos estadísticos de empresas y filtrar datos significativos y precisos de las mismas.

XII. DESAFIOS Y PROBLEMAS ENCONTRADOS

- *Desafío mas significativo fue la extracción de datos en los cuales el filtro conlleve conflictos de escritura*
- *Uno de los problemas fue con la base de datos mongodb ya que los apis de twitter nos establece problemas tales como bloqueos y demás.*
- *Otro problema encontrado fue con el driver de sql no le reconocía al configurar en el conf del Logstash por lo que se tuvo que añadir en 2 rutas a las variables de entorno.*

XIII. LINKS DE GITHUB DEL PROYECTO

https://github.com/JosueGarrido/ProyectoFinal_Base_De_Datos_Multidimensionales_DataLake/upload/master

XIV. BIBLIOGRAFIA

[1] *"Cómo instalar MongoDB en Windows, Linux y Mac", Platzi, 2020. [Online]. Available: <https://platzi.com/blog/como-instalar-mongodb-en-window-linux-y-mac/>. [Accessed: 12- Feb- 2020].*

[2] *"Primeros pasos con CouchDB - Adictos al trabajo", Adictos al trabajo, 2020. [Online]. Available: <https://www.adictosaltrabajo.com/2018/11/27/primeros-pasos-con-couchdb/>. [Accessed: 12- Feb- 2020].*

[3] *"{{metadataController.pageTitle}}", Subscription. packtpub.com, 2020. [Online]. Available: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781786465580/12/ch12lvl1sec144/installing-and-using-cerebro. [Accessed: 13- Feb- 2020].*

[4] *"How to copy SQL Server data to Elasticsearch using logstash", codeshare, 2020. [Online]. Available: <https://codeshare.co.uk/blog/how-to-copy-sql-server-data-to-elasticsearch-using-logstash/> [Accessed: 12- Feb- 2020].*