

PROYECTO FINAL

Santacruz Adriana

Wendy Villegas

ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
BASE DE DATOS MULTIDIMENSIONAL

I. DEFINICION DEL CASO DE ESTUDIO

El proyecto se dirige a la recopilación de información acerca de las temáticas: pulso político, top 10 de twitteros, y femicidios en el mundo. Los cuáles serán tomados de la fuente de Twitter, para luego ser almacenados en la base de datos CouchDB, luego mediante Logstash insertarlos con un mapeo correcto a Elasticsearch y para luego poder manipularlos y visualizarlos mediante Kibana.

II. OBJETIVO GENERAL

Realizar un caso de estudio de las temáticas: pulso político, top 10 de twitteros, y femicidios en el Mundo.

III. FOBJETIVOS ESPECIFICOS

- Recopilar información de las temáticas mediante geolocalización y filtro de palabras.
- Guardar información en una base de datos con formato correcto.
- Generar visualizaciones para la interpretación de la información.

IV. DESCRIPCION DEL EQUIPO DE TRABAJO Y ACTIVIDADES REALIZADAS POR CADA UNO

El equipo de trabajo se encuentra conformado por Adriana Santacruz y Wendy Villegas. Las actividades para la realización del proyecto hemos dividido en dos fases:

- Fase I de Recopilación por Adriana Santacruz
 - Generación de Scripts para la recolección de información en CouchDB de la Fuente Twitter.
 - Ejecución de scripts para la cosecha de información.
 - Creación de índices correctamente mapeados.
 - Creación de una base de datos unificada.
 - Realizar scripts para tomar datos recolectados en CouchDB y pasarlos a Elasticsearch.
- Fase II de Análisis y Visualización por Wendy Villegas
 - Análisis de la información recolectada.
 - Generación de gráficas para la interpretación de cada temática.
 - Creación de un Dashboard para la visualización de las gráficas.

V. CRONOGRAMA DE ACTIVIDADES

Actividades	Semana1 08/01/19	Semana2 15/01/19	Semana3 22/01/19	Semana4 29/01/19
Generación de Scripts				
Recolección de información				
Creación de base unificada				
Creación de Índices				
Análisis de información				
Generación de graficas				
Creación de Dashboard				

VI. RECURSOS Y HERRAMIENTAS UTILIZADAS

- CouchDB: utilizado para el almacenamiento de datos
- Elasticsearch: utilizados para almacenamiento de datos traídos de CouchDB.
- Logstash: utilizado para el envío de datos de CouchDB hacia Elasticsearch.
- Cerebro: permite la visualización de datos de Elasticsearch.
- Kibana: permite la generación de graficas según datos de Elasticsearch.
- Twitter: es nuestra fuente de datos.

VII. ARQUITECTURA DE LA SOLUCION

- Arquitectura que se utilizara dentro de CouchDB

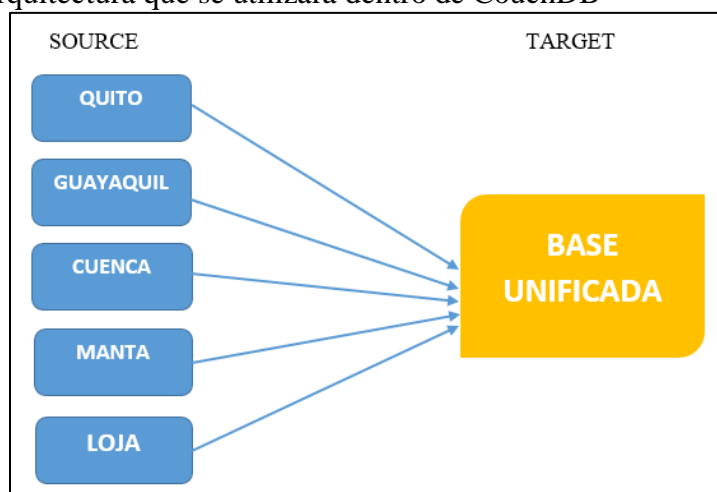


Figure 1 Arquitectura 1

- Arquitectura para la solución de proyecto

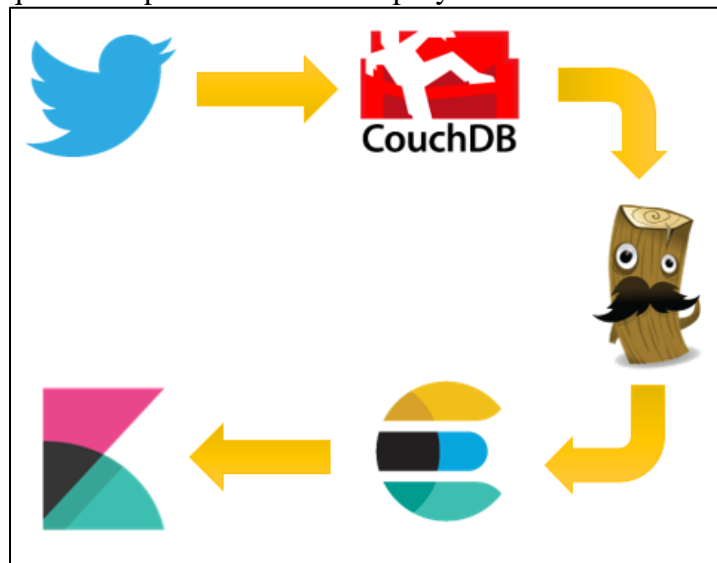


Figure 2 Arquitectura 2

VIII. EXTRACCION DE DATOS

Para la extracción de datos, se han utilizado Scripts de tipo Python, para la recopilación de la fuente de Twitter y estos almacenarlos en CouchDB.

La cosecha será en base a geolocalización y por filtro de palabras.

Estos scripts serán cinco, los cuales constaran de una misma estructura, los únicos que variarán son el nombre de las bases de datos donde se guardaran los datos, y el dato para la geolocalización y las palabras claves para la búsqueda.

A continuación, se mostrará el script para la recolección de datos para la ciudad de Quito:

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

ckey = "unksC6a2wXM5JoZ5cFwqx08F6"
csecret = "EJiq0oTUSQIPNL0pTgJnDfk6IJsjpt7EzeD7sgBG0G143HCWm0"
atoken = "999027538243588098-gAoMQKa1Na0YSvHB3g5Q9558jXCMsGQ"
asecret = "UcQU3N0Sib5MRHbekjYjG1eBjuwaqPjpnKYMfspiQZ6jN"

class listener(StreamListener):

    def on_data(self, data):
        dictTweet = json.loads(data)
        try:
            dictTweet["_id"] = str(dictTweet['id'])
            doc = db.save(dictTweet)
            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())

server = couchdb.Server('http://localhost:5984/')
try:
    db = server.create('quito')
except:
    db = server['quito']

twitterStream.filter(track=["quito","Juan Carlos Holguin","#Creo","Maria Sol Corral","#
twitterStream.filter(locations=[-78.619545,-0.365889,-78.441315,-0.047208])
```

Figure 3 Script Tipo Python

Luego lo ejecutamos y podemos visualizar que se ha almacenado correctamente en CouchDB.

Name	Size	# of Docs	Actions
ambato	461.8 KB	149	
cuenca	2.2 MB	782	
guayaquil	2.2 MB	741	
inmuebles	0.6 MB	1156	
manta	1.8 MB	734	
quito	2.8 MB	1056	

Figure 4 CouchDB

Luego del almacenamiento en CouchDB, procedemos a esos datos pasarlos a Elasticsearch, para ello se crean script de tipo configuración, para ser ejecutados con Logstash. A continuación, se mostrara un ejemplo del archivo.conf para el caso de la ciudad de Quito:

```
input{
  couchdb_changes{
    db=>quito
    host => "127.0.0.1"
    port => 5984
    sequence_path => "seq_files\audits_couchdb_seq"
  }
}
output {
  elasticsearch {
    index => quito
    document_type=> tweets
  }
  stdout {
  }
}
```

Figure 5 Script Tipo Conf

Antes de la ejecución del script tipo conf., debemos levantar los servicios necesarios. Primero para ellos es necesario ejecutar Elasticsearch accedemos a la carpeta con dirección **Elasticsearch/bin**. Ahora abrimos el **CMD**, ejecutamos el comando Elasticsearch

```
C:\BaseMultidim\elasticsearch-6.5.4\bin>elasticsearch
[2019-01-14T23:03:35,330][INFO ][o.e.e.NodeEnvironment ] [node-wendy] using [1] data
le_space [865.7gb], net total_space [930.9gb], types [NTFS]
[2019-01-14T23:04:02,309][INFO ][o.e.e.NodeEnvironment ] [node-wendy] heap size [989.
pointers [true]
[2019-01-14T23:04:02,606][INFO ][o.e.n.Node ] [node-wendy] node name [node
```

Figure 6 Levantar Servicio Elasticsearch

Una vez levantado Elasticsearch, procedemos a levantar Cerebro, que nos permite la visualización de los datos en Elasticsearch. Para ejecutarlo accedemos a la carpeta con dirección **Cerebro/bin**. Ahora abrimos el **CMD**, ejecutamos el comando cerebro.

```
C:\BaseMultidim\cerebro-0.8.1\bin>cerebro
[info] play.api.Play - Application started (Prod)
[info] p.c.s.AkkaHttpServer - Listening for HTTP on /0:0:0:0:0:0:0:0:9000
```

Figure 7 Levantar Servicio Cerebro

Ahora en el navegador insertamos: **localhost:9000**, luego insertamos **localhost:9200**.

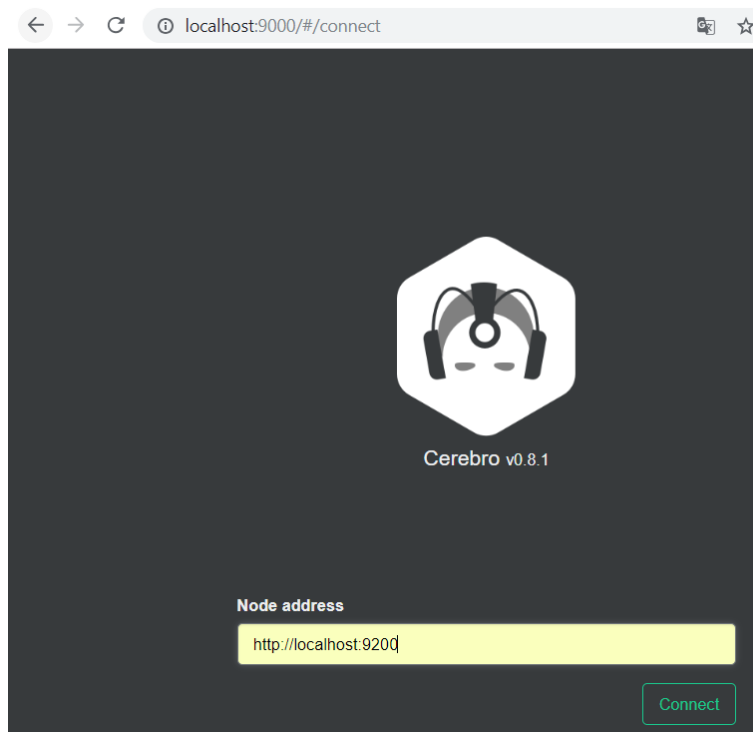


Figure 8 Ingreso a Cerebro

Dentro podemos visualizar todos los índices creados en Elasticsearch, para crear un índice correctamente mapeado, damos clic en **more, create index**.

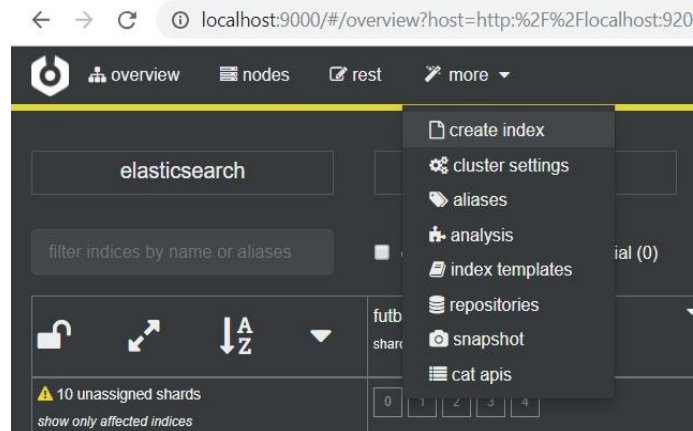
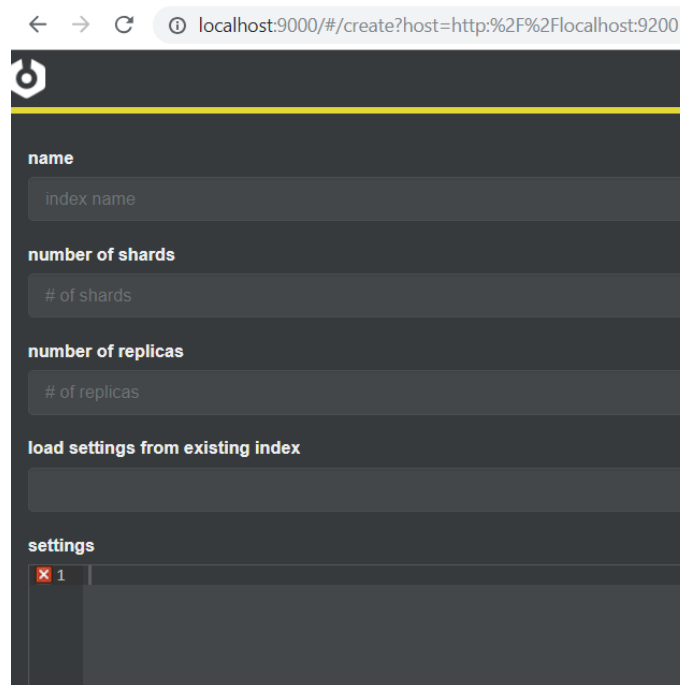


Figure 9 Creación Índice 1

Luego agregamos un nombre, numero de réplicas, entre otras, en la parte de settings, ubicaremos la estructura de mapeo que se utilizará para los datos que se guarden dentro de ese índice.



← → ↻ localhost:9000/#/create?host=http:%2F%2Flocalhost:9200

name

index name

number of shards

of shards

number of replicas

of replicas

load settings from existing index

settings

name	value
1	

Figure 10 Creación Índice 2

La estructura que se usara para el mapeo será la de Twitter, se tomara los datos de fechas y geolocalización correctamente para luego ser manipulados. La siguiente imagen muestra la estructura de mapeo que se utilizó:

```
{
  "mappings": {
    "tweets": {
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "@version": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "coordinates": {
          "properties": {
            "coordinates": {
              "type": "geo_point"
            },
            "type": {
              "type": "text"
            }
          }
        },
        "created_at": {
          "type": "date",
          "format": "EE MMM d HH:mm:ss Z yyyy||dd/MM/yyyy||dd-MM"
        },
        "entities": {
          "properties": {
            "urls": {
              "properties": {
                "display_url": {
                  "type": "text",

```

Figure 11 Estructura de Mapeo

Una vez ya creado el índice procedemos a ejecutar el archivo de configuración, que permite pasar datos de CouchDB a Elasticsearch. Se debe tomar en cuenta, que estos archivos se deben encontrar en la carpeta de **Logstash/bin**, luego en el **CMD**, se ejecuta la línea **Logstash -f archivo.conf**.

```
Microsoft Windows [Versión 10.0.17134.523]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\BaseMultidim\logstash-6.5.4\bin>logstash -f twitter_logstash.conf
```

Figure 12 Ejecutar Script Tipo Conf

Este proceso se realizará para cada uno de los scripts de las cinco ciudades escogidas. Ahora para la creación de una base de datos unificada que contemple la información de todas las bases de datos creadas. Debemos dirigirnos a la opción de réplicas dentro de la interfaz gráfica de CouchDB.

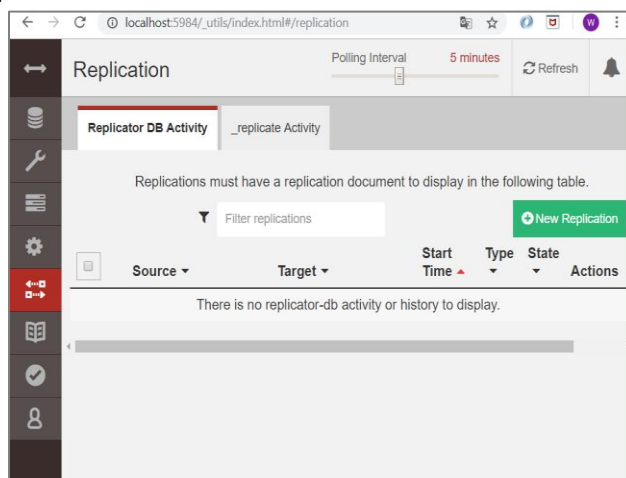


Figure 13 Replicas en CouchDB

Dentro damos clic en la opción **new replication**, nos aparecerá campos para añadir referente a la réplica.

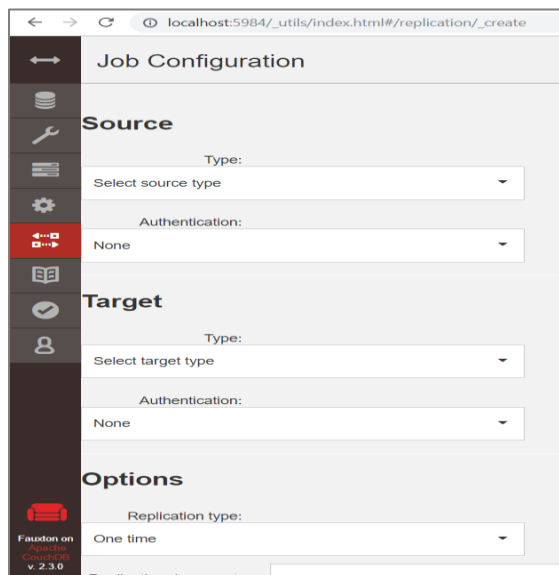


Figure 14 Creación de Replica 1

En la parte de Source, se deberá poner de tipo **base de datos local**, y seleccionamos la base llamada **quito**. En la parte de target es nuestro destino, o sea donde se copiarán todos los datos, elegimos el tipo **base local existente**, y seleccionamos la base de datos llamada **unificada**, esta base con anterioridad debe ser creada, cabe recalcar que esta base se encuentra vacía.

Las opciones que nos ofrece en autenticación, se usara en el caso de que hayamos registrado un usuario y contraseña a CouchDB, caso contrario podemos dejarlo en **None**, mientras en la opción de tipo de réplicas, tenemos dos opciones: **One Time** que permite replicar los datos solo una vez y la opción **Continuos** que nos permite que las réplicas se realicen automáticamente si se detecta q se ha ingresado un dato en la base de origen.

Figure 15 Creación de Replica 2

Una vez creada la réplica, debemos esperar a que se complete el proceso y podemos visualizar que todos los datos, del origen han sido copiados en la nueva base unificada.

Source	Target	Start Time	Type	State	Actions
http://localhost:5984/quito	http://localhost:5984/unificada	Jan 30th, 5:33 pm	One time	Completed	[Icons]

Figure 16 Creación de Replica 3

quito	2.8 MB	1056	[Icons]
unificada	2.8 MB	1056	[Icons]

Figure 17 Comprobación de la Replica

Este proceso se debe realizar con todas las bases de datos de las cinco ciudades. Con el fin que al unir la información de estas ciudades podamos contar con información para la realización del top 10 de Twitteros en esas cinco ciudades de Ecuador.

Para el caso de la última temática, que este caso nosotros hemos elegido el tema de Femicidio en el Mundo. Se realizará el mismo proceso anterior de recolección de datos de Twitter mediante Logstash enviarlos a Elasticsearch. Para ello se creó un archivo de configuración:

```
input {
  twitter {
    consumer_key => "m4Fq2Pr4yHn1YLLg6nmPYxXYZ"
    consumer_secret => "0WC0Z1D9sT4aMiBY5xDrRjFIQqT3KbU8oSaNEsFEkKPHZCAsE4"
    oauth_token => "999027411613356032-NvGF9YveYjVjQq4sf61x5IbFDe0KBej"
    oauth_token_secret => "ZHTEn2rxBoKLikFa57Ksm3Hs4jYtwimhgYcsq8TtAXVXv"

    keywords => ["#NiUnaMenos", "#niunamenos", "#niunamas", "#NiUnaMas",
    "Femicidio"]
    full_tweet => true
  }
}
filter{
}
output {
  elasticsearch {
    index => opcional
    document_type=> tweets
  }
}
```

Figure 18 Archivo de Configuración

IX. ANALISIS DE LA INFORMACION

Para la temática del pulso político en cinco ciudades se han escogido: Quito, Guayaquil, Cuenca, Ambato y Manta. En estas ciudades mediante filtro de palabras y con coordenadas para geolocalización se ha realizado la cosecha de tweets. Para ello se ha determinado que para el análisis de datos se realizara la creación de graficas de barras verticales, en donde se especifiquen los candidatos a la alcaldía de dichas ciudades con sus respectivos porcentajes de aceptación, definiendo así cual podrían ser los resultados del ganador en las próximas elecciones para alcalde en cada ciudad.

Para la temática del top diez de twitteros en cinco ciudades del país, se han elegido las mismas ciudades anteriores y se las recopilo en una base unificada, con la cual mediante una tabla de posiciones se podrá visualizar cuales son los nombres de los usuarios que más tweets genera en el país.

Mientras que para la temática última de Femicidio en el Mundo, se ha recolectado información acerca de las tendencias actuales que han surgió a de la misma, que es **#NiUnaMenos**, **#NiUnaMas**, **#VivasNosQueremos**, entre otras tendencias. Para poder visualizar dicha información se optó por conocer cuáles son los porcentajes según cada país en el que más se realicen tweets, mediante la gráfica de pastel, y para conocer los 3 primeros países en donde más se habla de ello, el filtro de palabras.

X. VISUALIZACION DE LA INFORMACION

Para la visualización de los datos cosechados de Twitter, usaremos la herramienta Kibana.

Primero para levantar el servicio debemos dirigirnos a la carpeta **Kibana/bin**, luego en el **CMD**, ejecutamos el comando **Kibana**.

Una vez levantado el servicio, en el navegador agregamos **localhost:5601**, luego debemos crear índices para poder realizar las gráficas.

Nos dirigimos a la parte de **Management**, elegimos la opción **Index Patterns**.

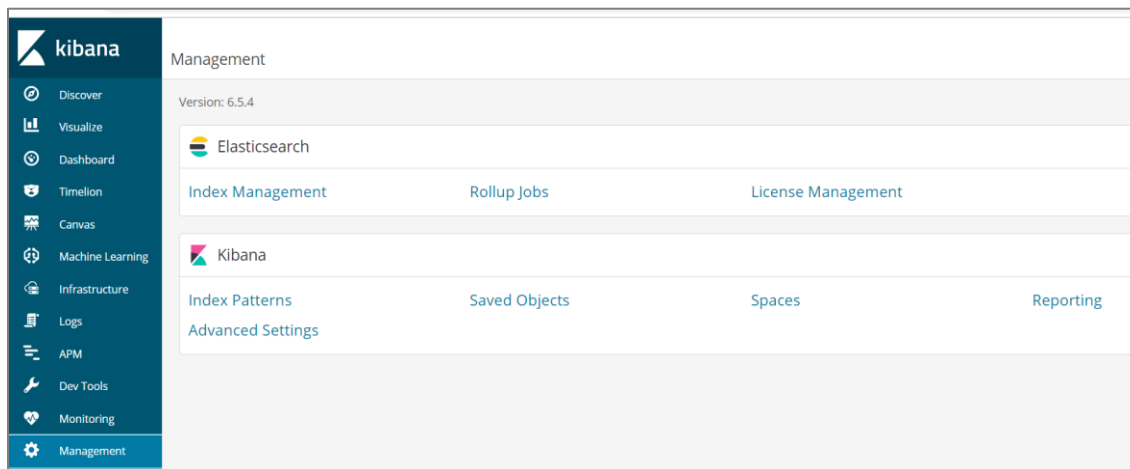


Figure 19 Crear índice Kibana 1

En la primera parte, se pedirá el nombre de la base de datos de Elasticsearch, y **siguiente**.

The image shows the 'Create index pattern' form in Kibana. At the top, it says 'Create index pattern' and 'Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.' There is a toggle switch for 'Include system indices' which is currently turned off. Below this, it says 'Step 1 of 2: Define index pattern'. There is a text input field for 'Index pattern' with the placeholder text 'index-name-*'. Below the input field, there is a note: 'You can use a * as a wildcard in your index pattern. You can't use spaces or the characters \, /, ?, *, <, >, |.' To the right of the note is a button labeled '> Next step'. Below the note, it says 'Your index pattern can match any of your 2 indices, below.' There are two text input fields, one with 'inmuebles' and one with 'quito'. At the bottom, there is a 'Rows per page: 10' dropdown menu.

Figure 20 Crear índice Kibana 2

En la segunda parte seleccionamos el tipo de tiempo de los filtros y **crear índice**.

Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations. Include system indices

Step 2 of 2: Configure settings
You've defined **quito** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

The Time Filter will use this field to filter your data by time. You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

[> Show advanced options](#)

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

Figure 21 Crear índice Kibana 3

Una vez creado el índice se podrá visualizar los datos del índice creado. Este proceso se deberá crear por base de datos que se quiera generar gráfica con sus datos.

★ quito*

Time Filter field name: created_at

This page lists every field in the **quito*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API.

Fields (747) Scripted fields (0) Source filters (0)

Q Filter All field types

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		•	•	
@version	string		•		
@version.keyword	string		•	•	
_id	string		•	•	
_index	string		•	•	
_score	number				
_source	_source				
_type	string		•	•	
coordinates.coordinates	geo_point		•	•	

Figure 22 Crear índice Kibana 4

Si el índice se correctamente creado en la parte de **Discover**, deberá aparecer los datos que se encuentran dentro de Elasticsearch.

kibana

1,481 hits

> Search... (e.g. status:200 AND)

Discover

Add a filter +

quito*

Figure 23 Visualizar datos del índice Kibana

❖ Pulso Político

Una vez creado el índice, en la parte de **Visualización**, ponemos la opción de crear **visualización**. Para este caso seleccionamos Barras Verticales.

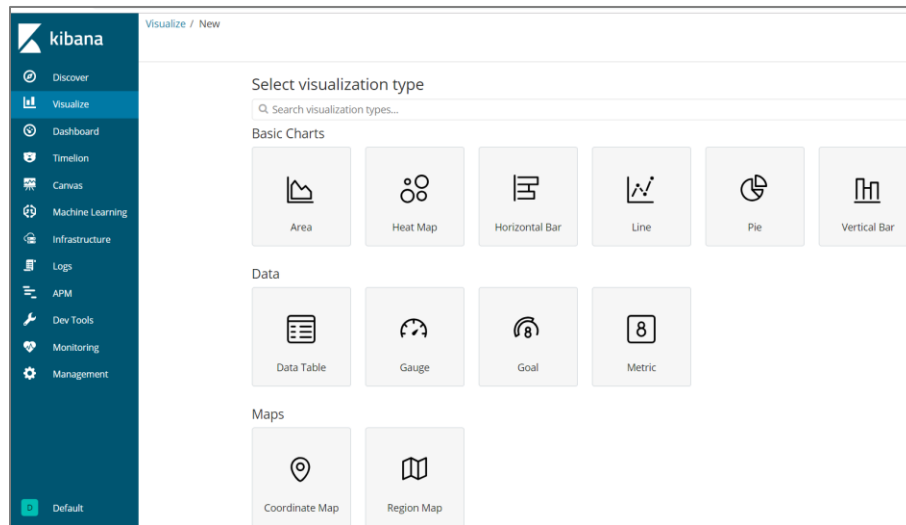


Figure 24 Crear Visualización 1

Luego se selecciona el índice para el cual se quiere crear la gráfica.

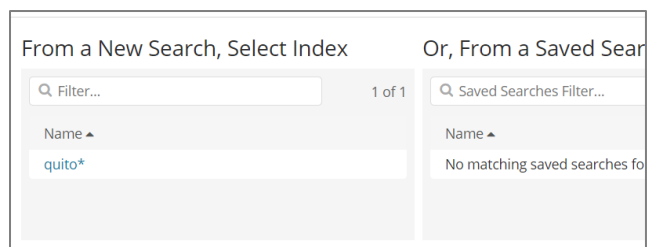


Figure 25 Crear Visualización 2

Se mostrarán todas las opciones de gráficas, nos aparecerá la gráfica por defecto.

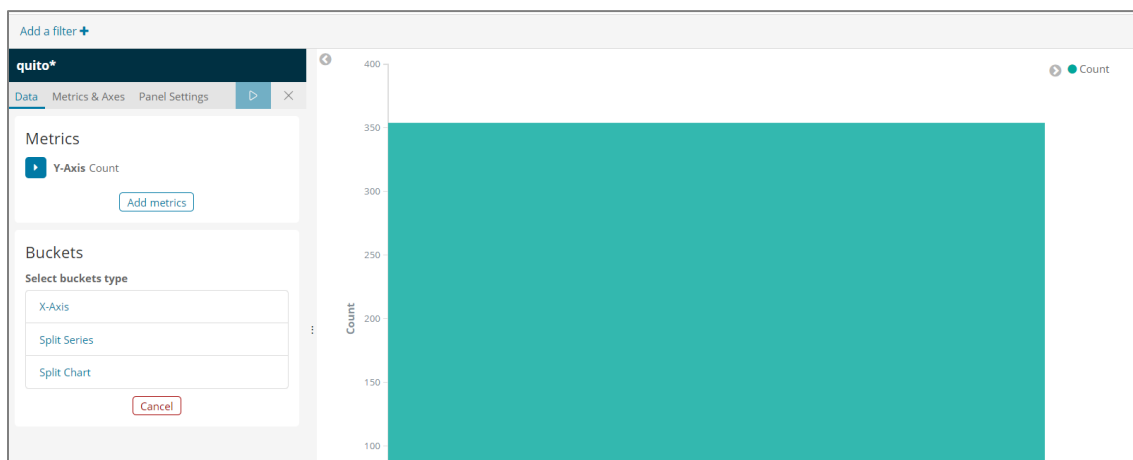


Figure 26 Creación de Grafica en Barras Kibana 1

En la parte de datos seleccionamos **X-Axis**, y seleccionamos **Filtros**, ya que usaremos palabras para saber el porcentaje que apoya a cada candidato a la alcaldía de cada ciudad.

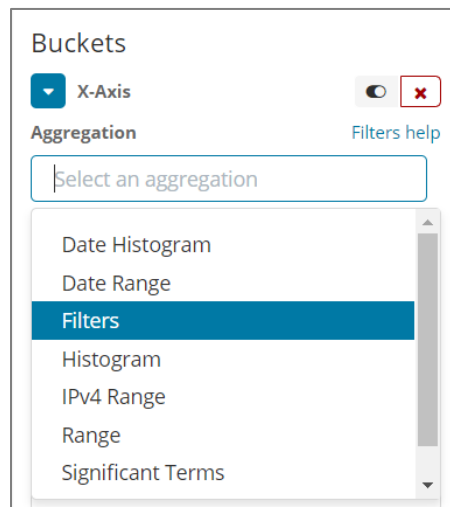


Figure 27 Creación de Grafica en Barras Kibana 2

Luego procedemos a ingresar las palabras claves, puede ser una o varias, y podemos añadir una etiqueta para que aparezca en la gráfica. Podemos añadir varios filtros. En este caso se usaron 6 filtros para cada candidato.

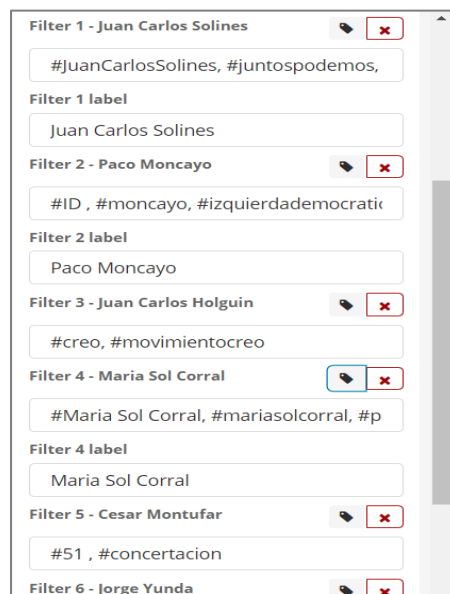


Figure 28 Creación de Grafica en Barras Kibana 3

Una vez terminado, damos clic en el botón de **Play**, y se visualizará la gráfica según las palabras clave que hemos ingresado.

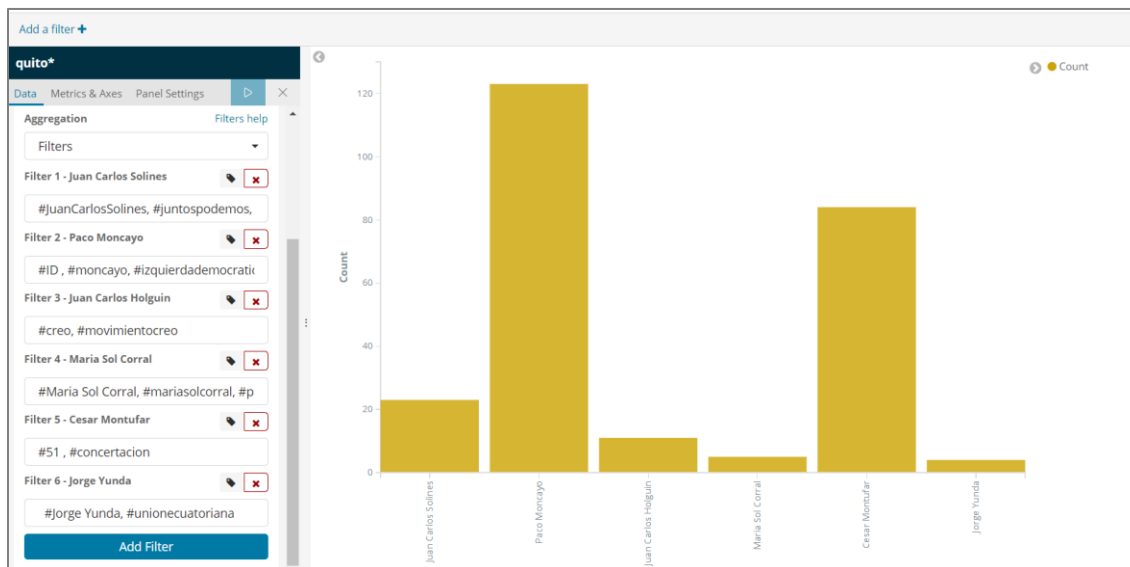


Figure 29 Creación de Grafica en Barras Kibana 4

❖ Top 10 de Twitteros en 5 ciudades del país

En la parte de **Visualización**, ponemos la opción de crear **visualización**. Para este caso seleccionamos **Data table**.

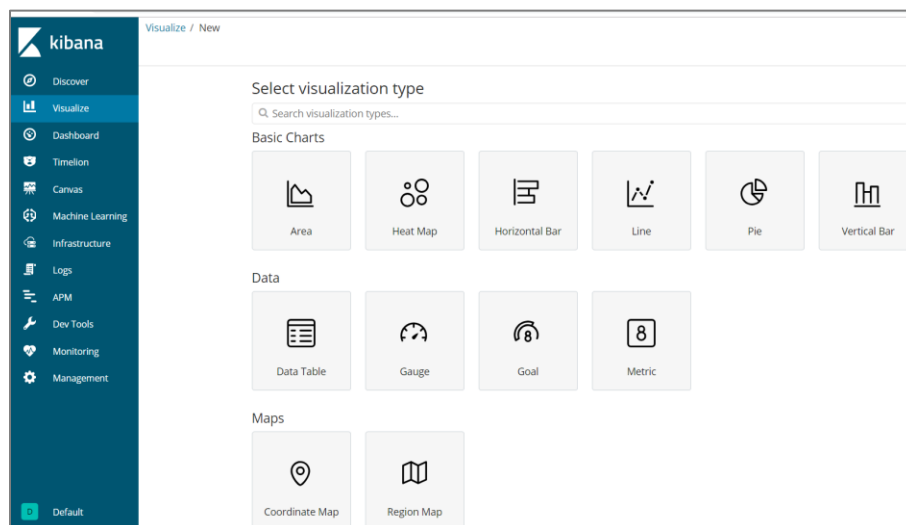


Figure 30 Creación Tabla de Datos Kibana1

Luego se selecciona el índice para el cual se quiere crear la gráfica. En esta temática como elegimos las mismas ciudades anteriores, podemos crear un índice que abarque todas ellas. Dentro de la opción **Managment**, elegimos crear **IndexPattern**. Y agregamos todas las bases de datos de las ciudades separadas con comas (,).

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

☐ Include system indices

Step 1 of 2: Define index pattern

Index pattern

quito,ambato,cuenca,guayaquil,manta

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

> Next step

✓ Success! Your index pattern matches 5 indices.

ambato

cuenca

guayaquil

manta

quito

Rows per page: 10

Figure 31 Creación Tabla de Datos Kibana2

Seleccionamos el índice creado, nos mostrará la tabla por defecto. En la parte de datos seleccionamos **Split Rows**, y seleccionamos **Terms**, ya que atributos de Twitter para conocer el nombre de usuarios que más cantidad de tweets genera.

Split Rows

☒ Split Rows

Aggregation

Terms help

Select an aggregation

Date Range

Filters

Geohash

Histogram

IPv4 Range

Range

Significant Terms

Terms

Figure 32 Creación Tabla de Datos Kibana3

Usaremos el atributo **quoted_status.user.screen_name.keyword**, como la temática nos pide los 10 primeros, en la parte de tamaño añadimos el número 10 y con orden descendiente.

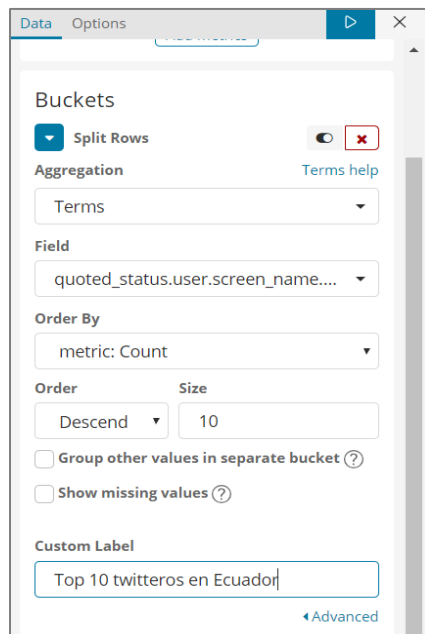


Figure 33 Creación Tabla de Datos Kibana4

Damos clic en **Play**, y nos mostrara la tabla de posiciones.

Top 10 twitteros en Ecuador	Count
thedeptsbelow	92
Miliduarte17	90
periodista5977	6
PadreJosePalmar	4
PaolaPabonC	4
maraton_compol	4
GonzalezACarlos	3
PolitiHacksE	3
elcomerciocom	3
pecesdeciudad85	3

Figure 34 Creación Tabla de Datos Kibana5

❖ Femicidios en el Mundo

Igual que los anteriores nos dirigimos a la parte de **Visualización, Nueva Visualización**. Elegimos la opción de **Tag Cloud**.

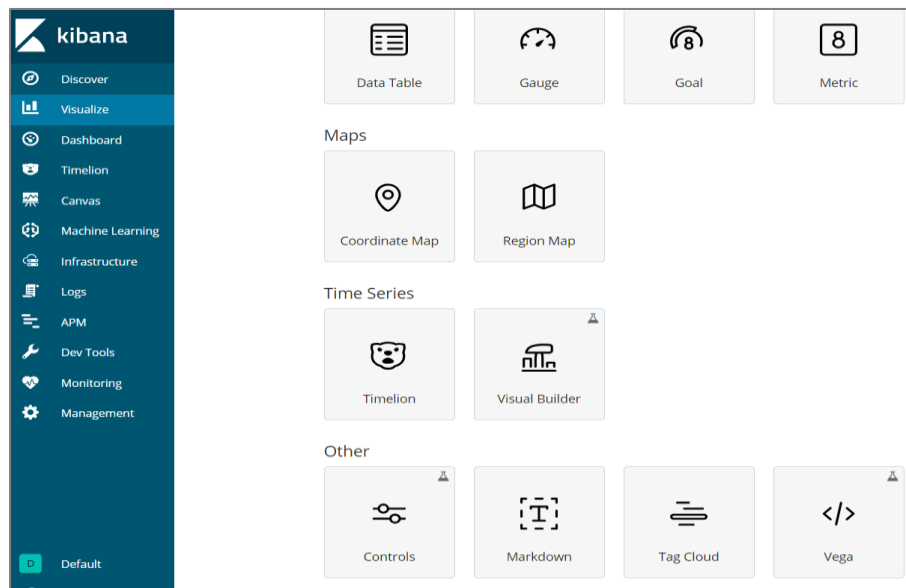


Figure 35 Creación Tag Cloud Kibana1

Seleccionamos el índice creado, nos mostrará la un dashboard por defecto. En la parte de datos seleccionamos **Tag**, y seleccionamos **Terms**, ya que atributos de Twitter para conocer el nombre de la ciudad que más cantidad de tweets genera. El atributo que agregaremos será: **retweet_status.place.country.keyword**.

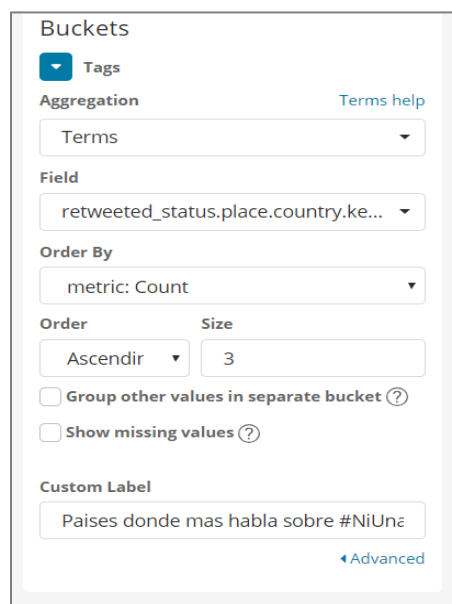


Figure 36 Creación Tag Cloud Kibana2

Damos clic en **Play**, y podemos visualizar el nombre de las ciudades en donde más se tweetea sobre la temática.

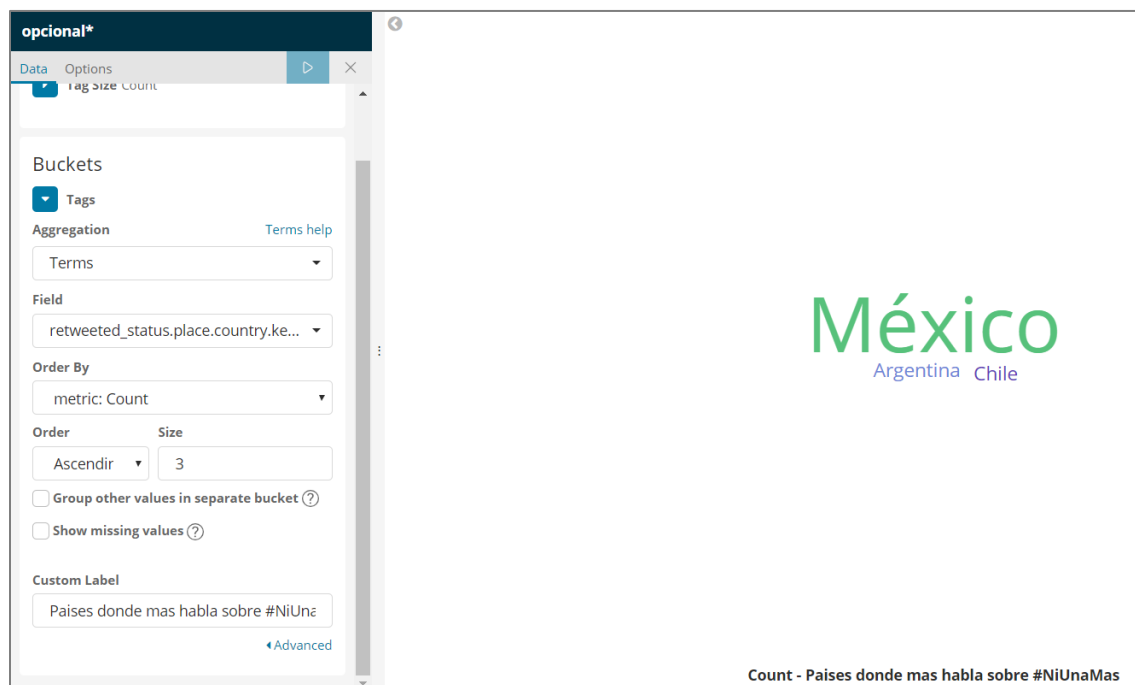


Figure 37 Creación Tag Cloud Kibana3

Ahora para conocer el porcentaje de cada país que tweetea sobre la temática generamos una gráfica de tipo pastel.

En donde, seleccionamos **Split Slices**, en la parte de datos y mediante filtros ingresamos las palabras clave para conocer el resultado, en este caso se seleccionaron los países de Ecuador, Perú, Colombia, Argentina, Mexico, Chile y España.

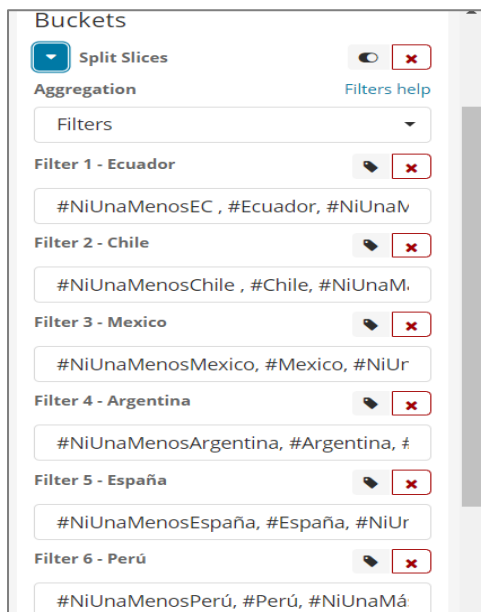


Figure 38 Creación Patel Kibana1

Hacemos que se genere la gráfica dando clic en **Play**, y observamos los resultados.

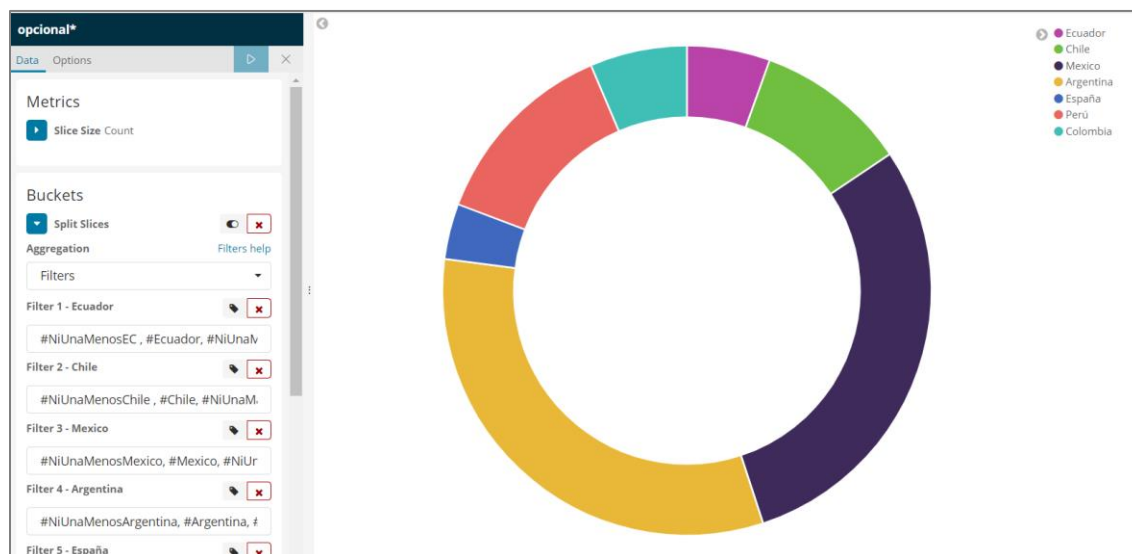


Figure 39 Creación Pastel Kibana2

Una vez creadas las visualizaciones podemos guardarlas, damos clic en **save**, solo agregamos un nombre a la visualización y confirmamos para guardar.

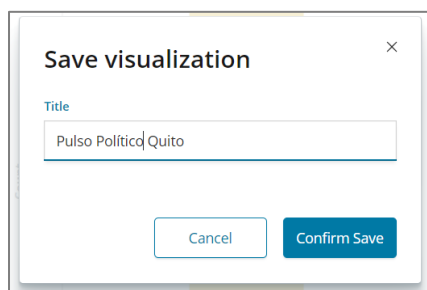


Figure 40 Guardar Grafica Kibana

XI. RESULTADOS OBTENIDOS

Pulso Político Quito:

Entre los alcaldes más sonados para el posible mandado en la Alcaldía de la capital del Ecuador Quito, el alcance Paco Moncayo es uno de los apoyados, con más aceptación de los ciudadanos y con más probabilidades de ganar, mientras que otro candidato que también cuenta con apoyo de una buena parte de la ciudad es Cesar Montufar. Ellos son los que representan mayor apoyo en la ciudad de Quito.

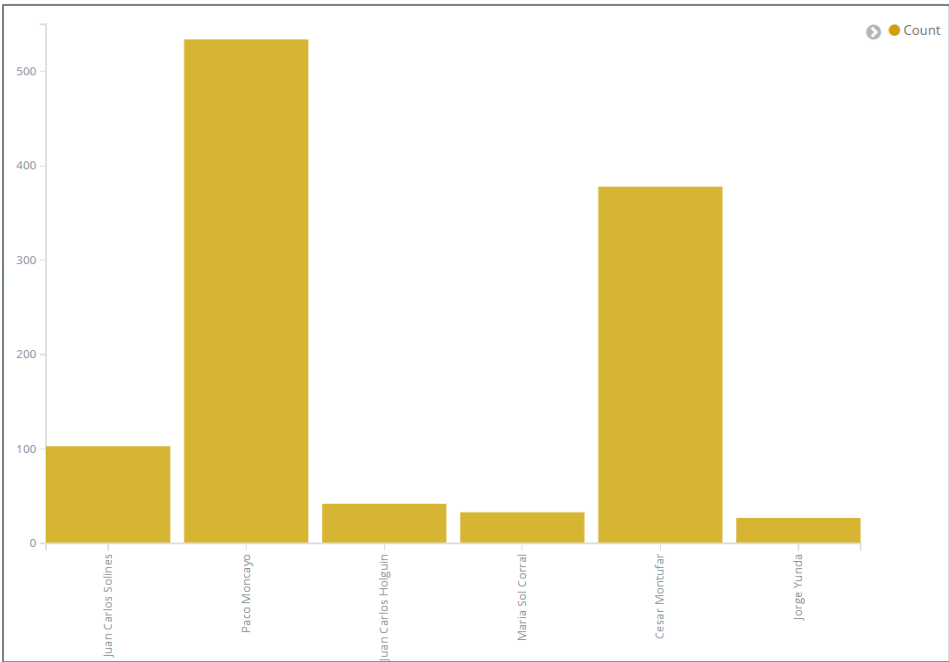


Figure 41 Pulso Político Quito

Pulso Político Guayaquil:

Entre los candidatos para la Alcaldía de Guayaquil, se encuentran: Jimmy Jairala, Paco Buendia y Cynthia Viteri como los candidatos con más apoyo de la ciudadanía. Posicionando así a Jimmy Jairala como uno de los candidatos con más probabilidades de ganar en las próximas elecciones, mientras que los demás candidatos presentan muy poco apoyo con respecto al anterior.

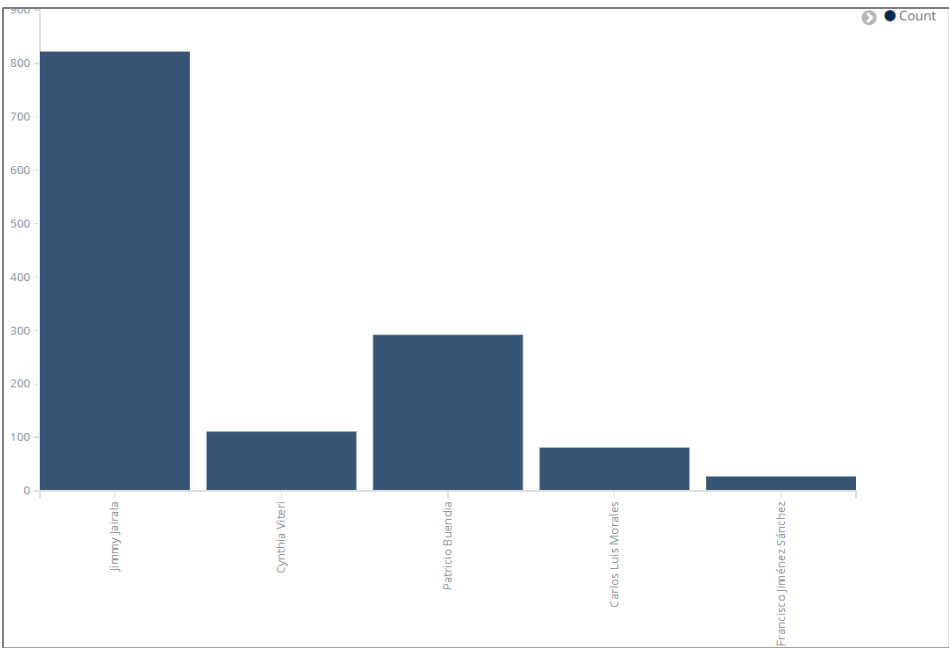


Figure 42Pulso Político Guayaquil

Pulso Político Cuenca:

En la graficas de barras horizontales, se puede observar que las tendencias de voto a la alcaldía de Cuenca probablemente, se encuentran dirigidas con más aceptación con el candidato Marcelo Cabrera, el candidato Pedro Palacios es otro que se encuentra compitiendo con su rival muy a la par. Ellos son los dos candidatos que se reporta más probabilidades de que ganen en el puesto a la Alcaldía de la ciudad.

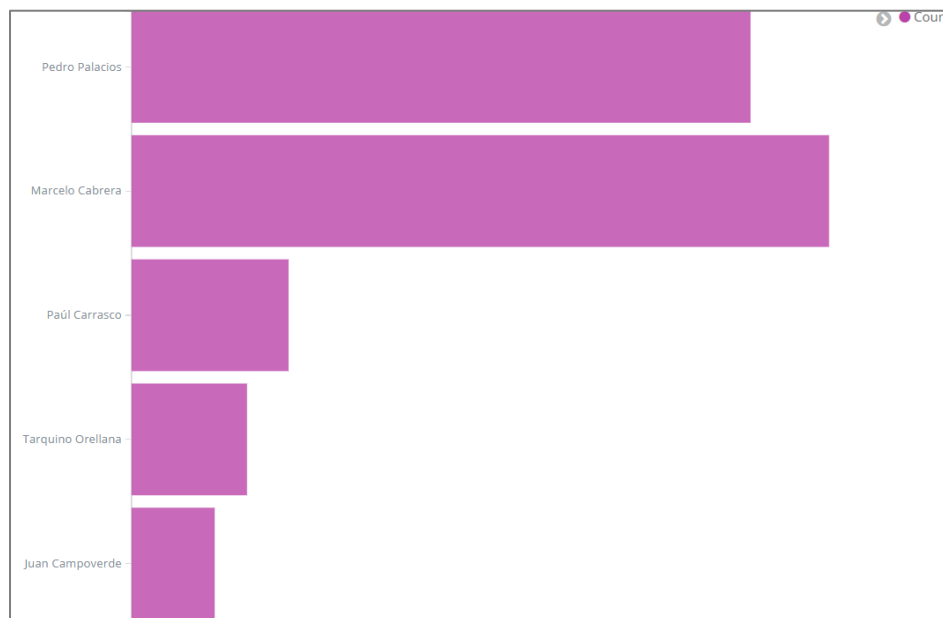


Figure 43Pulso Político Cuenca

Pulso Político Manta:

En la visualización de nuestros datos, llegamos a la conclusión de que el candidato Víctor Chiriboga es el que cuenta con más apoyo de la ciudadanía de la Cuidad de Manta, es el candidato con más probabilidades de ganar en estas elecciones 2019. Mientras que los demás cuentan con un porcentaje de apoyo demasiado bajo, que no podrían competir directamente con Víctor Chiriboga.

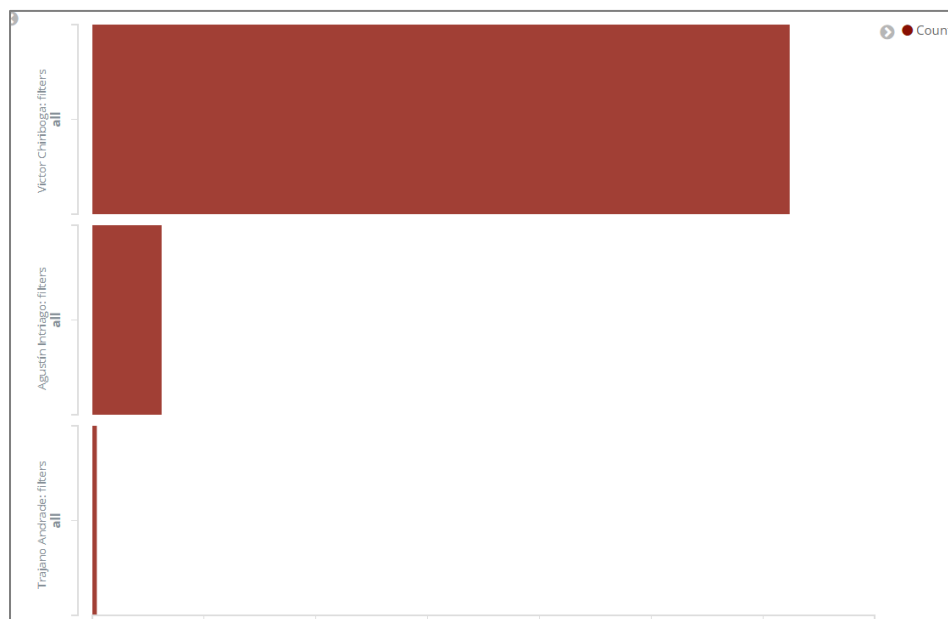


Figure 44Pulso Político Manta

Pulso Político Ambato:

Los candidatos para alcaldía de Ambato, se encuentran entre los más propensos a ganar: Javier Altamirano, Fernando Naranjo y Luis Amoroso. De ellos Fernando Naranjo es el que presenta mayor porcentaje de apoyo de la ciudadanía, le sigue Javier Altamirano y por último Luis Amoroso. Dado como ganador a Fernando Naranjo como alcalde de Ambato en las próximas elecciones 2019.

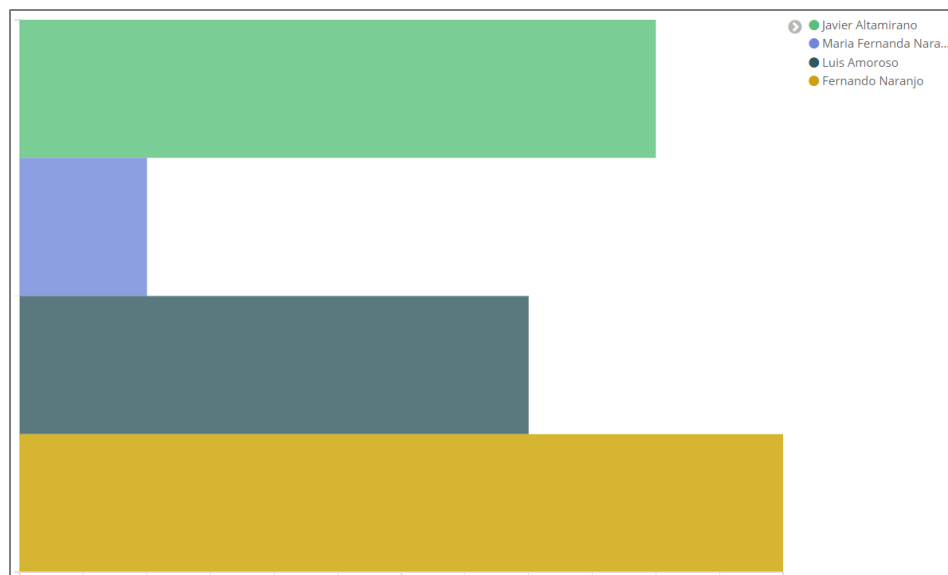


Figure 45Pulso Político Ambato

Top 10 de Twitteros en Ecuador:

En la tabla de posiciones se puede observar que existen dos personas con alrededor de 90 veces que tuitean información en Ecuador, tomando como base las ciudades de Quito, Guayaquil, Cuenca, Ambato y Manta.

Top 10 twitters en Ecuador ↕	Count ↕
thedeptsbelow	92
Miliduarte17	90
periodista5977	6
PadreJosePalmar	4
PaolaPabonC	4
maraton_compol	4
GonzalezACarlos	3
PolitiHacksE	3
elcomerciocom	3
pecesdecuidad85	3

Figure 46Top 10 Twitteros

Femicidios en el Mundo:

Para esta temática según las gráficas obtenidas, se puede concluir que existe un gran porcentaje en Latinoamérica, que habla sobre un alto a la violencia hacia la mujer, en los últimos tiempos se han dado muchas noticias sobre ataques a mujeres, por ello a gran escala mujeres de todo el mundo se pronuncian y exigen de esa manera un alto a esta situación.

Entre los países que más se encuentran liderando este movimiento de Alto al Femicidio, se encuentran México como principal y le siguen Argentina y Chile correspondientemente.

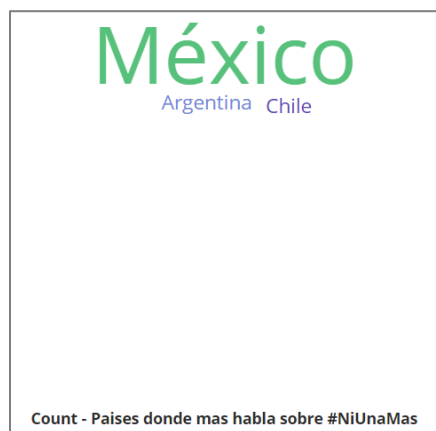


Figure 47 Femicidio en el Mundo1

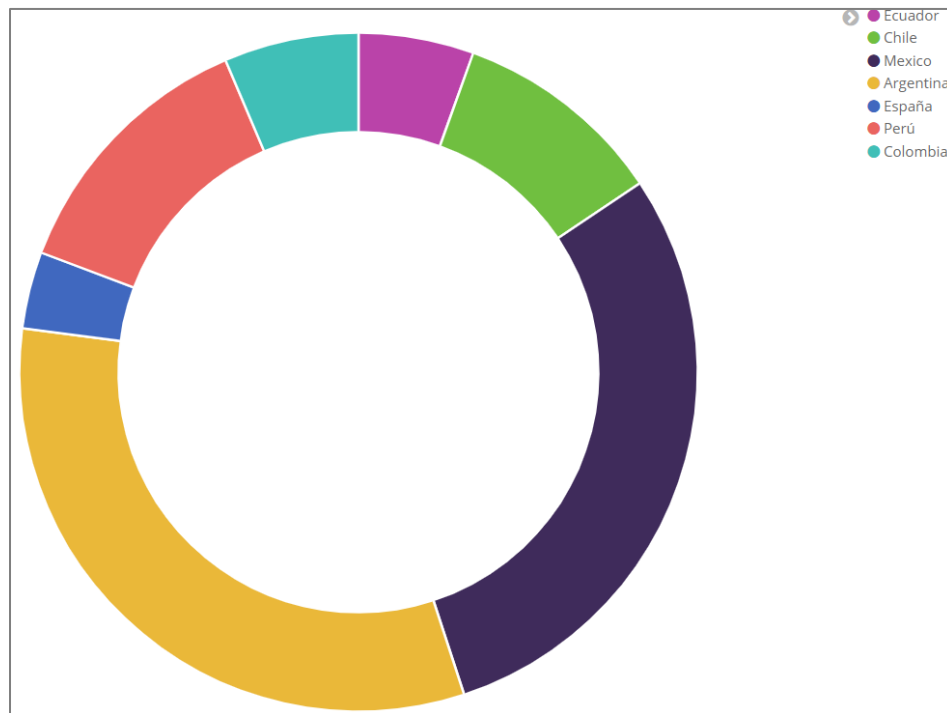


Figure 48 Femicidio en el Mundo2

XII. CONCLUSIONES Y RECOMENDACIONES

- Gracias a la herramienta Kibana, podemos analizar los datos recolectados mediante gráficas, y para mejorar la forma de presentarlos, permite la generación de dashboard.
- Para dar cumplimiento de nuestros objetivos se utilizaron varias herramientas que son consideradas para la recolección, almacenamiento y presentación de los datos, estas herramientas son: CouchDB, Elasticsearch, Logstash, Cerebro, Kibana, Twitter.
- Es recomendable diseñar una arquitectura de recopilación de información donde se identifique el destino y el final que tendrán los datos de esta manera será más fácil su implementación
- Hay varias formas de extraer datos desde Twitter y guardarlos en CouchDB pero la más recomendable es realizarlo mediante scripts de Python que con las correctas credenciales la recolección será fácil y rápida

XIII. DESAFIOS Y PROBLEMAS ENCONTRADOS

Uno de los desafíos encontrados en el proyecto, fue en la etapa de generar gráficas para la visualización del pulso político, ya que no sabíamos cómo realizar el conteo de tweets según el candidato a la alcaldía de cada ciudad elegido. La solución que encontramos fue la generación de filtros, mediante estos logramos insertar palabras claves para que dependiendo de estas se realice el conteo.