|  |
| --- |
| ESCUELA POLITÉCNICA NACIONAL |
| INTELIGENCIA DE NEGOCIOS |
| PROYECTO SEGUNDO BIMESTRE |
|  |

Jorge Carrillo

Roberto Toapanta

**Ing. Eliza Mena**

**16/Agosto/2017**

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**CARRERA DE INGENIERIA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**INFORME DE PROYECTO PRIMER BIMESTRE**

**Objetivo General**

Implementar e investigar el funcionamiento de un clasificador de sentimientos utilizando los algoritmos de aprendizaje vistos en clase y los datos recolectados de Twitter para identificar tendencias de opinión en la ciudad de Quito.

**Objetivos específicos:**

* Crear un clasificador de sentimiento en español utilizando datos extraídos de Twitter para minar opinión pública en la ciudad de Quito
* Identificar y seleccionar las herramientas necesarias para procesar y analizar datos en tiempo real provenientes de Twitter

1. **Introducción**

La Inteligencia de Negocios tiene varias aplicaciones, una de ellas es el análisis de grandes cantidades de datos en tiempo real, a lo que se le conoce como Minería de Datos. Hoy en día la minería de datos es una de las principales fuentes de redito económico para los expertos en inteligencia de negocios.

La minería de datos tiene como objetivo explorar un gran volumen de datos mediante el uso de herramientas para el análisis de la información, es por esto que las empresas grandes se han interesado en esta tecnología actual para poder obtener conocimiento de mercado sobre los productos o interés particulares de ciertos clientes y así poder ofertar sus productos en tiempo real.

Para poder realizar minería de datos tenemos que tener un gran almacenamiento, para esto existen bases de datos no relacionales que nos permiten almacenar la información en tiempo real, como son el caso de CouchDB, Cassandra, entre otras. Existen también herramientas que extraen la información de la base de datos para poder realizar un análisis de los datos mediante herramientas de Python como TextBlob. Para poder realizar una presentación de datos podemos utilizar herramientas como elasticsearch.

Se analizará mediante los tweets cuáles son las horas del día en donde la gente se siente más feliz para poder aplicar los conocimientos obtenidos.

Dentro del presente documento se tratará como se desarrolló la solución del problema, pasos a realizar, las herramientas utilizadas, conclusiones, análisis de resultados y material de referencia.

Para poder tener un conocimiento previo para la solución del problema definiremos las herramientas a utilizar y en el desarrollo explicaremos su instalación y su uso.

**CouchDB:** Es un gestor de bases de datos de código abierto No SQL que emplea documentos JSON para guardar datos, realizando consultas mediante un Map Reduce y Http, una de sus principales características es la facilidad de Replicación de Datos. (CouchDb, 2017)

**ElasticSearch:** Es un servidor de búsqueda basado en Lucene, Provee un motor de búsqueda de texto completo con documentos JSON (Wikipedia, 2017)

**PyCharm:** Servicio integrado para ejecutar archivos de Python.

**TextBlob:** Lenguaje escrito en Python para el procesamiento de datos, Es un procesador de lenguaje natural de texto.

**Kibana:** Kibana es una herramienta open-source perteneciente a Elastic, que nos permite visualizar y explorar datos que se encuentran indexados en ElasticSearch, es decir, un plugin de ElasticSearch. (Echarrouti, 2015).

**Logstash:** es una herramienta desarrollada por Elastic y que funciona bajo la JVM de Java. Ésta permite administrar los logs de nuestras aplicaciones, de manera que podemos usarla para recolectar, parsear y guardar los logs para búsquedas posteriores. (García, 2015)

**Sistema Operativo Ubuntu:** Sistema utilizado para el desarrollo de la practica en el cual se instalarán todas las herramientas.

1. **Metodología**

Para el desarrollo del presente proyecto utilizaremos los programas anteriormente mencionados.

**Desarrollo:**

**Pasos Previos**

**Instalar CouchDB**

* Para la instalación de CouchDB procedemos a escribir los siguientes comandos:
  + Actualizamos Ubuntu mediante el siguiente comando.
    - **sudo apt-get update**
  + Instalamos nuestra base de datos No SQL
    - **sudo apt-get install couchdb –y**
* Verificamos la instalación de nuestra base de datos mediante la siguiente dirección:
  + **localhost:5984/\_utils**
* Creamos una base de datos, la creación de la base de datos se la puede realizar mediante comandos por el terminal de Ubuntu o mediante la interfaz de CouchDB.
  + Creación mediante comandos:

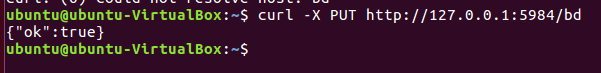


Ilustración 1 Instalación de CouchBD

* + Creación por medio de la interfaz:
    - Damos clic sobre la opción de new Database

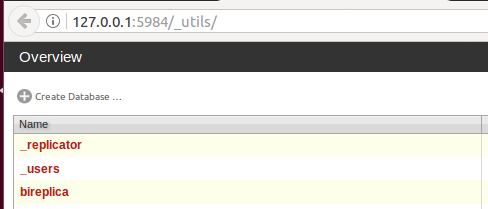


Ilustración 2 Creación de la base de datos CouchDb

* + - Agregamos el nombre de nuestra base de datos a crear en minúsculas y sin espacios caso contrario no se creará nuestra base de datos.

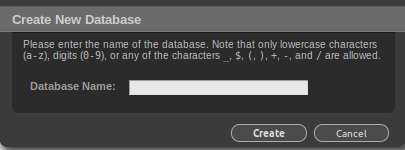


Ilustración 3 Ingreso del Nombre de la DB

**Instalación de ElasticSearch**

* Procedemos a situarnos sobre el repositorio de ElasticSearch 1.2 para no tener conflictos al momento de extraer nuestros tweets a la base, mediante el siguiente comando.



Ilustración 4 Instalación de ElasticSearch

* Actualizamos los paquetes necesarios



Ilustración 5 Actualización de ElasticSearch

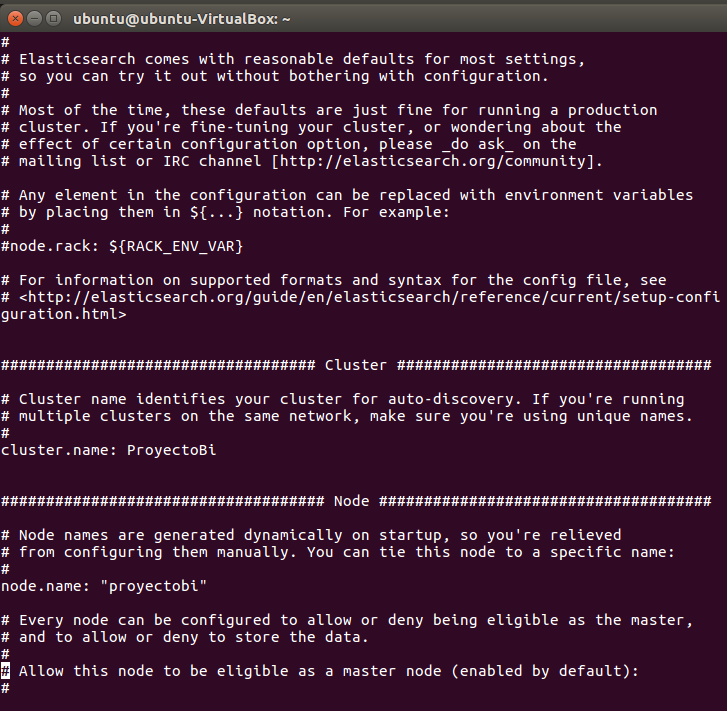
* Instalamos nuestro servidor de Elastic Search



Ilustración 6 Instalación de ElasticSearch

* Configuramos el servicio en el archivo elasticsearch.yml ubicado en la carpeta /etc/elasticsearch/



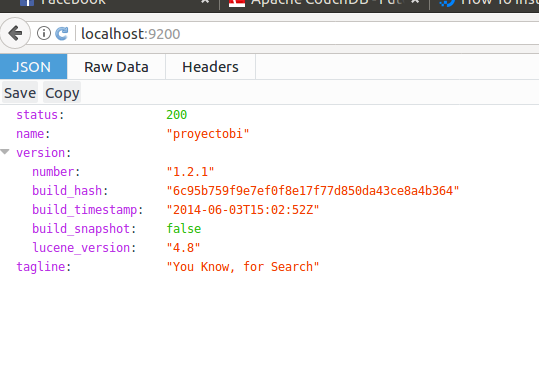


Definimos en el archivo el nombre del clúster, y del nodo.

* Reiniciamos el servicio.



* Accedemos a la siguiente dirección para comprobar que nuestro servicio fue instalado de forma correcta.

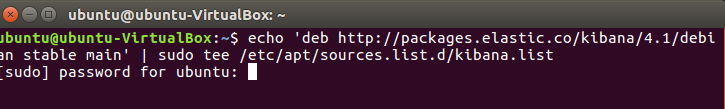


**Instalación de Kibana**

Para poder visualizar los datos de forma gráfica procedemos a instalar Kibana que se para nuestra versión de ElasticSearch.

Nuestra versión de ElasticSearch la versión de Kibana es la 3.

* Nos situamos sobre el repositorio de Kibana 3 para descargar nuestro paquete.

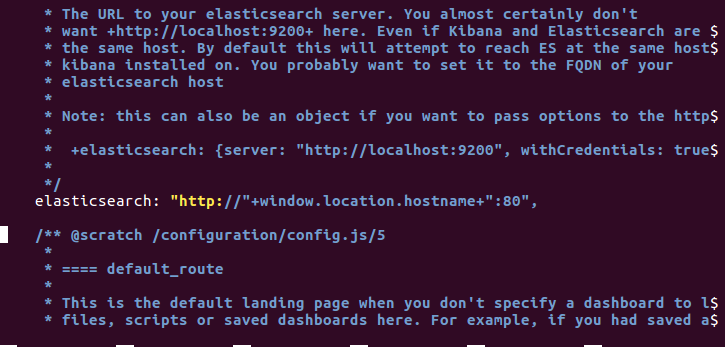


* Descomprimimos el paquete en el lugar que deseamos instalar.





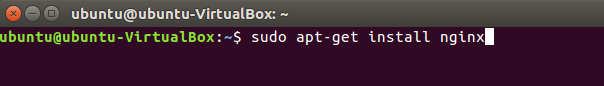
* Configuramos nuestro archivo para poder acceder a la dirección web.



* Para poder visualizar Kibana debemos instalar el siguiente paquete NGNIX

**Instalación del Ngnix**

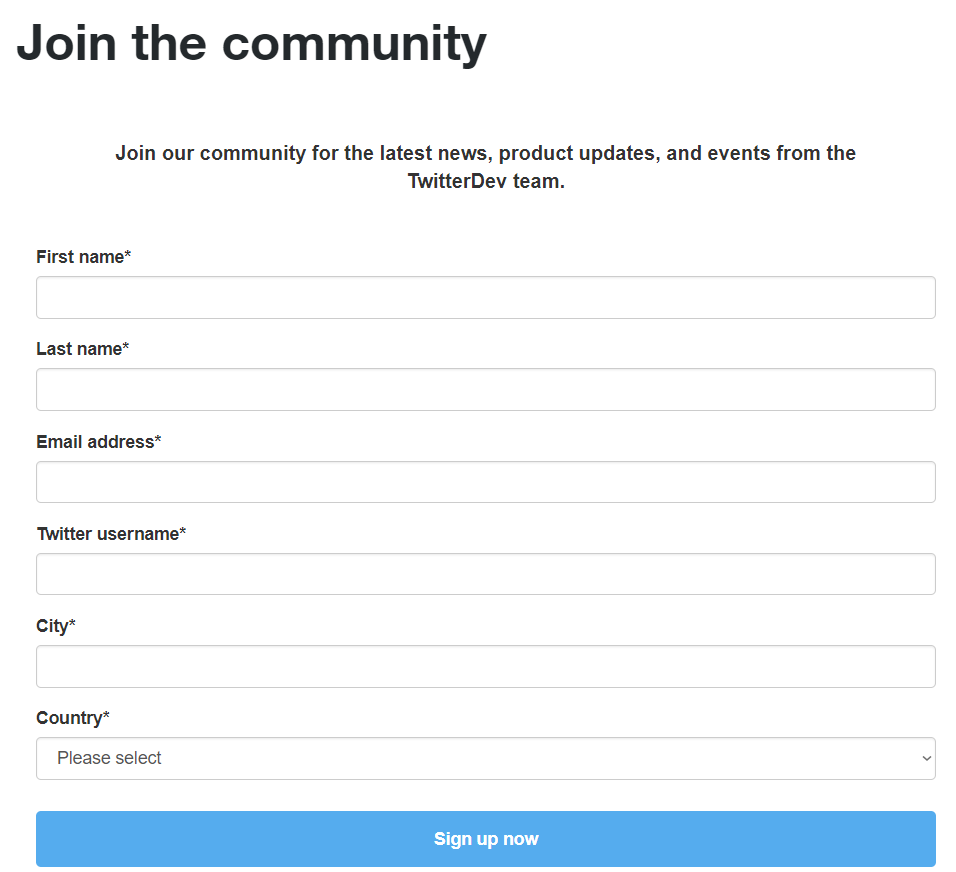
* Instalamos el paquete Ngnix mediante el siguiente comando



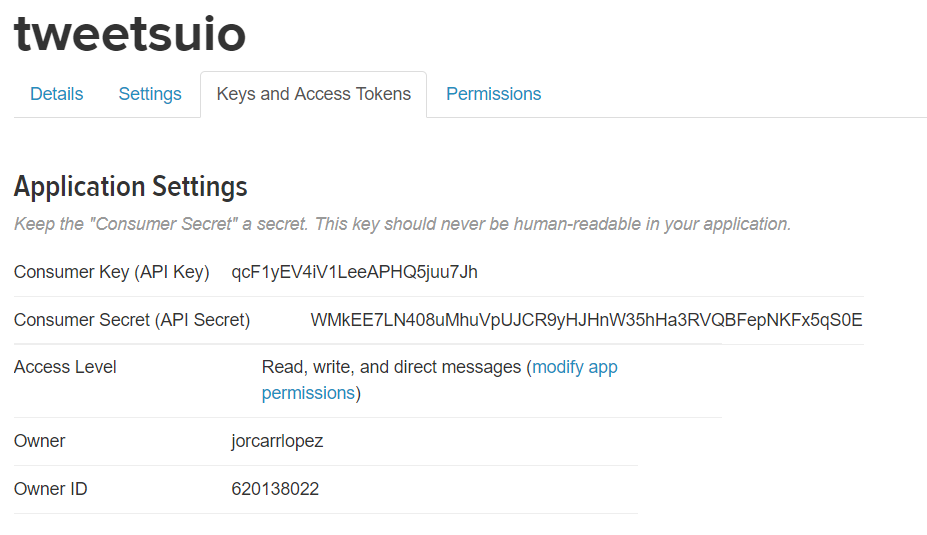
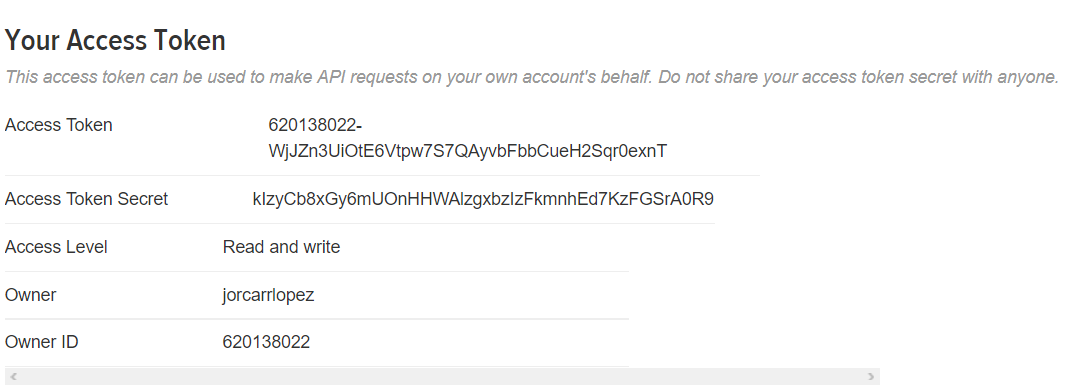
**Adquisición de Datos**

Para poder realizar el análisis de tweets necesitamos adquirir de datos para esto se procede a ejecutar el script harvester\_uio.py.

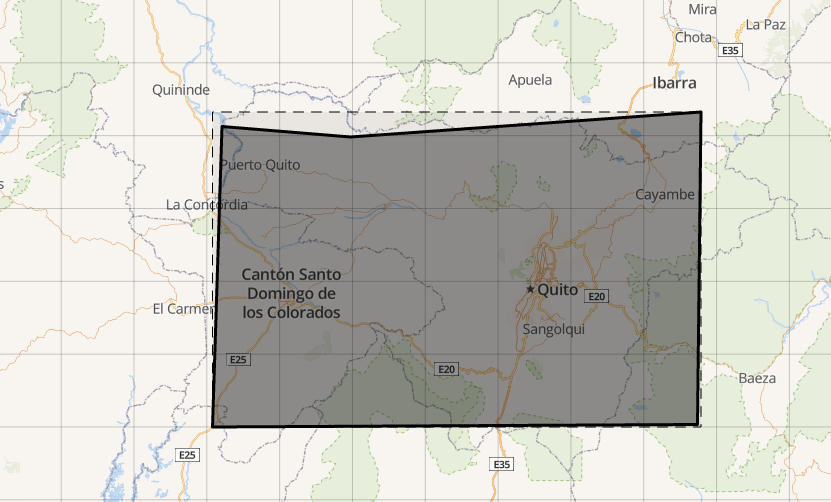
* En el script se necesita ingresar las credenciales del API de Twitter que se puede obtener en la siguiente dirección web: <https://dev.twitter.com/>



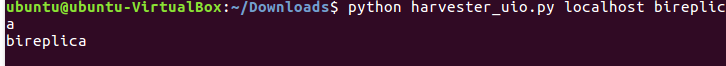
* Creamos una aplicación de twitter y dentro de estas tendremos las credenciales necesarias para poder correr el siguiente script.

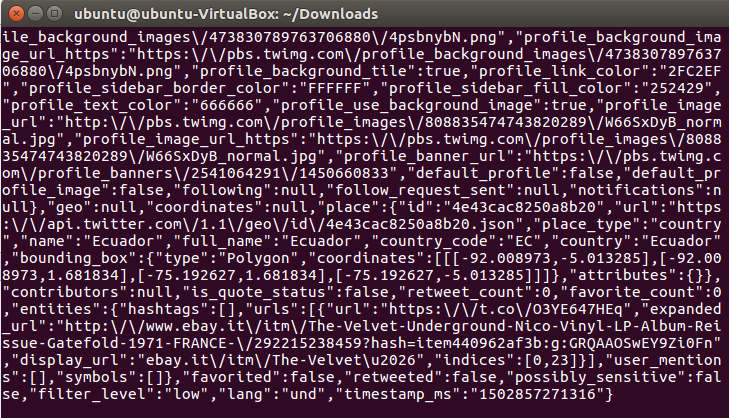


* Seleccionamos las coordenadas de la zona a cosechar los tweets.

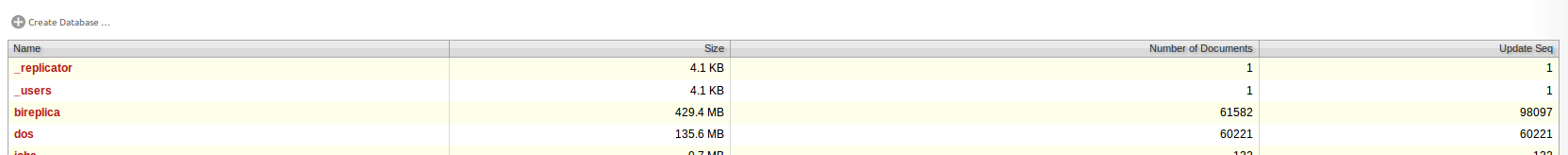


* Conectamos a la base de CouchDB donde se almacenarán los tweets recolectados en la zona de Quito.
* Ejecutamos nuestro Script desde la terminal de Ubuntu mediante el siguiente comando, si todo fue registrado de forma correcta se procederá a extraer los tweets de las coordenadas registradas con anterioridad.

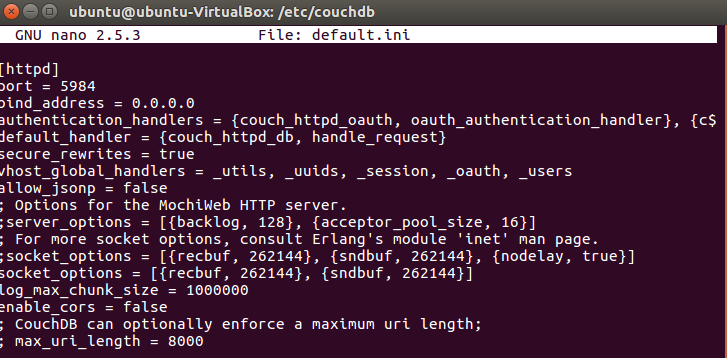




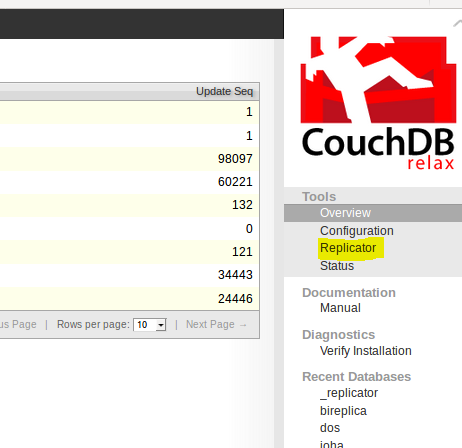
* Vista de los tweets cosechados en la base de datos



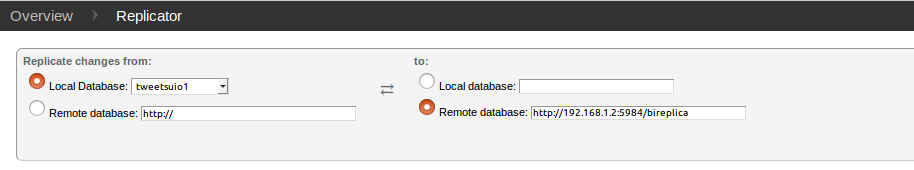
* + **Replicación de la Base de Datos**
    - En el proyecto para poder tener una mayor cantidad de datos a analizar se cosecharon datos en dos computadoras diferentes, luego se procedió a realizar la replicación.
    - Para realizar la replicación se configura el archivo defaul.ini que se encuentra en la carpeta etc/couchdb para escribiendo 0.0.0.0 en la línea bind\_address.



* + - Una vez configuro el archivo la replicación se puede realizar de dos maneras
      * Replicación por mediante la interfaz de CouchDB
        + Se accede en la parte inferior derecha de la pantalla y de clic sobre la opción de replicación



* + - * + Se ingresa la base que de desea replicar a la dirección Ip con el nombre de la base a guardar los tweets replicados.



* + - * Replicación mediante comandos.
        + Para replicación mediante comandos se ingresa el siguiente comando con el usuario y la contraseña donde se almacenarán los tweets.

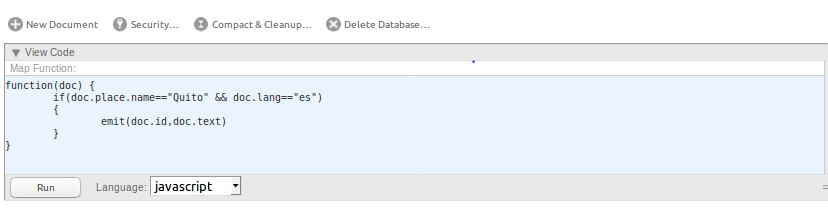
**Pre-Procesamiento**

Para poder analizar los tweets es necesario filtrarlos debido a que dentro de la base se encuentran Tweets que se encuentran fuera de cuidad de Quito, texto en inglés, en Japonés, de igual manera ingresan tweets de otros países.

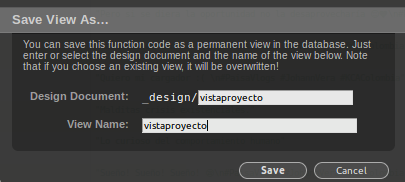
Para esto creamos una vista dentro de la base de datos la cual filtrara los datos que se encuentran dentro de la cuidad de Quito y que tengan texto en español.

**Creación de la Vista**

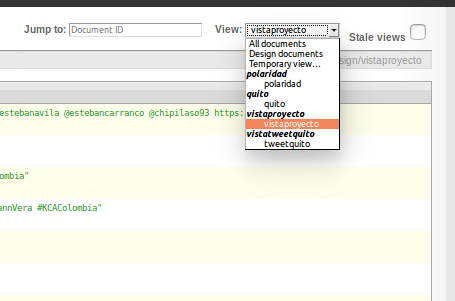
* Para crear nuestra vista nos dirigimos a la base de datos CouchDB seleccionamos el nombre de la base en la cual deseamos crear la vista, nos dirigimos a la opción de View y desplegamos el menú. Seleccionamos Temporal View.
* Escribimos dentro de la función el código de la vista a ejecutar.



* Mandamos a correr la vista, si la vista fue escrita de forma correcta se desplegarán los resultados caso contrario no.



* Se procede a guardar la vista en este caso la llamaremos Vista Proyecto.

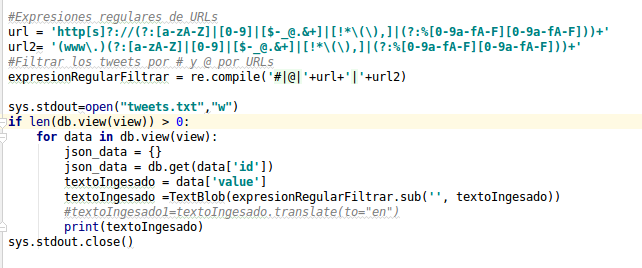


**Procesamiento**

Después de filtrar nuestros datos por la cuidad de Quito y con texto en español, procedemos a extraer el texto de los tweets para luego limpiar los tweets mediante expresiones regulares, en esta fase de procesamiento se realizara mediante un script realizado en Python.

En el texto extraído se limpia las url dentro del tweet, los emoticones simples, caracteres especiales como el @ o #, para que se puedan procesar de forma correcta al momento de realizar el análisis de estos.

Estos tweets se podrán observar en el archivo tweetslimpios.txt.



**Fase de Análisis**

En esta etapa se realiza el análisis y la clasificación de los textos determinando los sentimientos del texto para esto es necesario definir el train el cual se va a definir el Sustantivo, adjetivo, verbo, emoticones y la calificación de la oración para entrenar a nuestro clasificador de palabras.

Se ingresara el train de forma manual con 100 tweets que serán elaborados teniendo en cuenta el sustantivo, adjetivo, verbo, para generar un nuevo archivo. Este nuevo archivo servirá para el ingreso de un nuevo ingreso para nuestro train y así obtener una clasificación mayor al 80%.

El test que nos permitirá evaluar nuestro clasificador será ingresado mediante un archivo.

El clasificador utilizara un script de Python el cual se utiliza funciones como TextBlob y se clasificara mediante el algoritmo de Bayes.

**Presentación de Datos**

En esta fase de presentación de los datos se necesita la conexión con ElasticSearch para poder extraer nuestros datos que se encuentran presentes en la base CouchBD para esto necesitamos tener instalado el Plugin \_river, head, kopf, estos plugin nos servirán para poder analizar los datos.

Después de haber extraído los tweets e instalar Kibana se procede a realizar el análisis de los datos de forma visual mediante en mapping que tendrá la hora y la ciudad del tweet.

**Análisis de Resultados**

* Podemos analizar que las personas en la ciudad de Quito envían una mayor cantidad de tweets dentro de las 16h00 y 20h00 de lo cual puede ser por el estado de ánimo del trabajo o del día que paso. Esto puede ser interesante para realizar el estudio de mercado y ofertar cierta cantidad de productos.
* Como se puede observar el para una clasificación correcta de los sentimientos efectuados por las personas es necesario entrenar a nuestro clasificador de la mejor manera posible para poder obtener una mayor precisión.
* Al aplicar las vistas se perdieron gran cantidad de datos en nuestro caso se redujeron de 70000 tweets a 30000 tweets en el tiempo de ejecución hasta las 0,21 horas de la fecha 16/08/2017
* En la cosecha de tweet se pudo observar que entraron tweets de los países aledaños con diferentes idiomas, esto es por las coordenas establecidas al momento de ejecutar nuestro script. Para solucionar este problema se desarrolló una vista para recoger los datos que estén solo en español y sean emitidos dentro de la ciudad de Quito.
* La configuración de las herramientas se las desarrollo de forma correcta pero no fue de fácil acceso puesto que son tecnologías se encuentran en investigación y hoy en día son de un reciente surgimiento, es importante mencionar en este punto que para poder conectarnos y extraer los datos a Elastic Search se tuvo que desinstalar la versión 2.2 que se la tenía instalada por que el plugin river es un plugin que solo se desarrolló hasta la versión 1.7 de ElasticSearch.
* Al momento de separar los Sujetos, Adjetivos, verbos de las oraciones en español existe problema para la clasificación ya que el clasificador se encuentra en inglés y no traduce de forma correcta, por esta razón se diseña un nuevo train de forma manual para brindar un entrenamiento adecuado.
* El proyecto se puede encontrar en el siguiente enlace de GitHub <https://github.com/JorgeLuis1395/ProyectoBi>

**Conclusiones**

* Para el desarrollo del proyecto es necesario tener conocimientos previos de programación de Python para poder realizar los scripts de conexión limpieza, extracción de datos.
* La instalación de ciertas herramientas como ElasticSearch, kibana y los plugin de ElasticSearch fueron complejos de instalar ya que existían diferentes versiones y no son compatibles con la versión instalada.
* El análisis de los datos necesita de herramientas que se conectan a bases de datos para realizar búsquedas sobre los datos mapeados, de esta manera se necesita conocer la forma de realizar, la falta de tiempo y de documentación no impidieron realizar de forma correcta esta parte de la práctica.
* Es necesario tener instalado Python 3 para poder procesar los emoticones de los tweets ya que estos son de gran valor para determinar el sentimiento de las personas.
* Tomar en cuenta el espacio utilizado por las herramientas y los datos recolectados. Crear una máquina virtual con al menos 50 GB de espacio libre, puesto que nos da conflictos al momento de almacenar los tweets.
* Para seleccionar herramientas de procesamiento y análisis hubo constantes problemas debido a que no existía compatibilidad entre versiones de programas o la mayoría de programas que realizan Data Manning trabajan en inglés.
* La API de traducción de Google permite traducir cierta cantidad de palabras y para continuar con la traducción de los tweets es necesario pagar para la implementación de aplicación.
* Al diseñar un diccionario de datos de forma manual se para cada uno de los complementos de la oración se pierde precisión en la clasificación de los datos.

**Bibliografía**

* CouchDb. (17 de 05 de 2017). *CouchDb*. Obtenido de https://es.wikipedia.org/wiki/CouchDB
* Wikipedia. (1 de 08 de 2017). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Elasticsearch