FACULTAD DE INGENIERIA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Recuperación de Información y Recomendaciones en la Web 2020

Integrantes:

Nombre y apellido	C.I	Contacto
Rafael Castelo	5.222.324-2	rafaelcastelo222@gmail.com
Manuel Rodriguez	4.853.751-8	manu.rodriguezz@hotmail.com
Diego Wins	4.649.535-4	diegow93@gmail.com

Octubre 2020

Presentación de problema a abordar

Twitter es una fuente inagotable de contenidos, generado por millones de usuarios que interactúan decenas de veces por día. Con tanta información, las posibilidades de extracción y análisis de datos son casi infinitas, lo que lo hace propicio para un proyecto como éste. Además, la propia plataforma ha implementado una API muy completa para poder acceder a toda la información disponible. Si bien existen distintos niveles de acceso a la misma, incluso el más básico brinda enormes posibilidades de recuperación de información.

Tomando esto en cuenta, y considerando qué temáticas podrían ser de interés según la actual coyuntura mundial, se busca proponer el tratamiento de información relevante. Es por eso que, a partir de las recientes elecciones estadounidenses, surgió la idea de ofrecer un análisis de opiniones políticas de los usuarios en base a su preferencia partidaria, lo que podría ser de utilidad para politólogos o sociólogos, por ejemplo. Más específicamente, nos interesa poder descubrir y analizar quiénes son los usuarios más relevantes dentro de cada partido político, para poder entender mejor cómo se originan las diferentes tendencias de opinión dentro de esta red social.

La importancia de un usuario en Twitter puede ser muy relativa, y cada quien podría utilizar distintos parámetros para definirla. En particular, creemos que lo más conveniente es utilizar criterios como cantidad de seguidores y cantidad de interacciones (*likes*, *retweets*, comentarios, etc.) que recibe el usuario por cada *tweet*. Con esta información podríamos encontrar los grandes “nodos” generadores de opinión entre sector político para la red, lo que podría resultar interesante a la hora de analizar la propagación de las ideas debatidas en cada momento.

Además de filtrar según tendencia política, creemos relevante ofrecer un filtrado por temática. De esta forma se puede encontrar qué usuarios han tenido más repercusión por partido en base a un tema en concreto, lo que puede dar indicios de la postura de un sector político ante tal temática. Por ejemplo, podrían buscarse aquellos usuarios más relevantes en referencia al debate de las medidas tomadas en la lucha contra el Covid-19 para cada partido político. El sistema arrojaría a aquellos que han tenido más interacciones en sus *tweets* al respecto, que consideramos como los más relevantes.

En resumen, la idea se centra en disponer de una aplicación web que permita a partir de los *tweets* más relevantes (según los parámetros mencionados) segmentados por filiación política y temática, con el objetivo identificar a los usuarios que tienen más relevancia en ese espectro político para el tema en concreto. Para identificar la filiación política se utiliza la API de Twitter, mientras que para la temática dentro de cada *tweet* se hará un análisis de palabras clave dentro de cada uno de éstos.

Diseño del sistema desarrollado

Teniendo en cuenta las necesidades del proyecto abordado, y en base al análisis de las posibilidades ofrecidas por diversas tecnologías, se implementó una arquitectura compuesta por: una base de datos de *Elasticsearch*[1], un servidor web en *NodeJS*[2], una aplicación web en *ReactJS*[3] que interactúe con el servidor y la API de Twitter que será accedida desde el servidor web. El código implementado puede ser consultado en el siguiente repositorio de github https://github.com/rafael-castelo/twPoliticiansWebir.git (Backend)
https://github.com/manurodriguezz/twitterApp.git (Frontend)

Arquitectura del sistema:

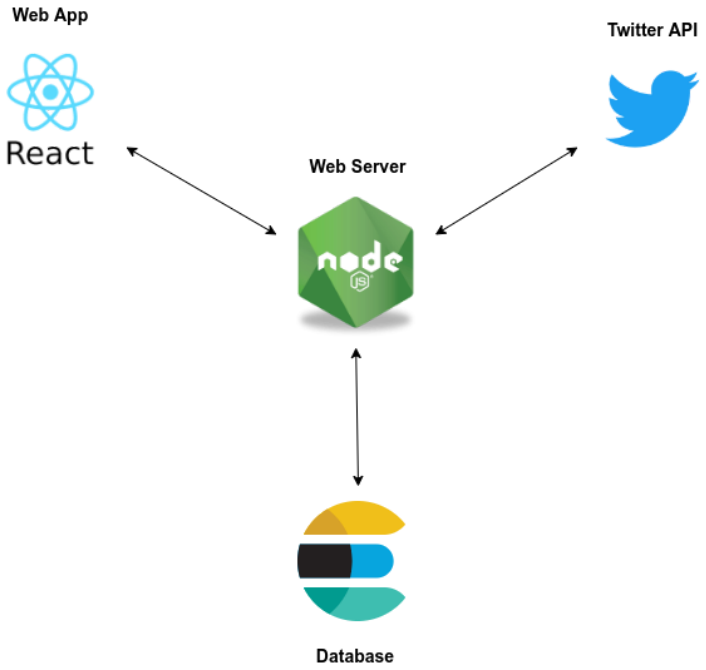


Fig 1. Diagrama de arquitectura del sistema

NodeJS* y *ReactJS: La elección de estas herramientas viene dada por sus buenas capacidades a la hora de crear aplicaciones web basadas en API Rest, además de ser ya conocidas y manejadas por los integrantes del equipo, lo que permitió destinar más tiempo a desarrollo en contraposición al aprendizaje de nuevas tecnologías.

Elasticsearch: En cuanto a la elección de *Elasticsearch*, si bien el equipo tenía poco conocimiento y alguna breve experiencia utilizándola, entendimos que era altamente recomendable para el almacenamiento y procesamiento de grandes volúmenes de datos. Además, ofrece un lenguaje de consulta muy extenso y posibilidad de realizar búsquedas por

[1] Elasticsearch web: *https://www.elastic.co/*
[2] NodeJS web: *https://nodejs.org/en/*
[3] ReactJS web: *https://reactjs.org/*

palabras clave dentro de los documentos a la hora de efectuar las consultas, lo cual era muy importante para nosotros. Por estos motivos consideramos útil decantarnos por esta opción, aunque implicase el aprendizaje de una nueva tecnología.

API de twitter: Para recolectar la información con la que se trabajará, se utiliza la API provista por twitter para desarrolladores[4] [1]. Esta provee los N tweets más recientes, que coinciden con una consulta determinada,por ejemplo, permite obtener los 1000 tweets más recientes, que están relacionados con el término “Trump”.

Manejo de la información: Para poder ofrecer las funcionalidades de la aplicación como ser búsqueda por tags o por tópicos. La información debió ser recolectada y procesada de la siguiente manera:

1- Se recolectaron los tweets más recientes, que responden a las querys “Democrats” y “Republicans”. Obteniendo estos atributos para cada tweet

```
id: tweet.id,
author_id: tweet.author_id,
text: tweet.text,
retweet_count: tweet.public_metrics.retweet_count,
reply_count: tweet.public_metrics.reply_count,
like_count: tweet.public_metrics.like_count,
quote_count: tweet.public_metrics.quote_count,
```

2- Posteriormente cada conjunto de tweets obtenidos fue almacenados en dos index distintos dentro de Elasticsearch, uno para cada query/tópico (“Republicans” y “Democrats”).

Entonces, esto es lo que sucede a la hora de filtrar según partido político y tag:

Desde el frontend se selecciona un partido y una consulta, el partido seleccionado indica sobre que index de Elasticsearch se debe buscar y la consulta indica el string que debe *matchear* el texto de un tweet para ser incluido en el resultado.

Una vez se tiene el conjunto de tweets relacionados con una consulta dentro de un partido, se debe elegir el usuario más relevante. Para esta tarea se modeló un puntaje de relevancia que toma en cuenta la cantidad de likes, retweets, citas y respuestas de un tweet y se implementó un vector de pesos que pondera las citas, retweets y respuestas por sobre los likes.

Con dichos atributos en cuenta, se selecciona el tweet que tenga el mayor score de relevancia, y por último a partir de este se obtiene el usuario autor del mismo siendo este último el resultado de la consulta.

---

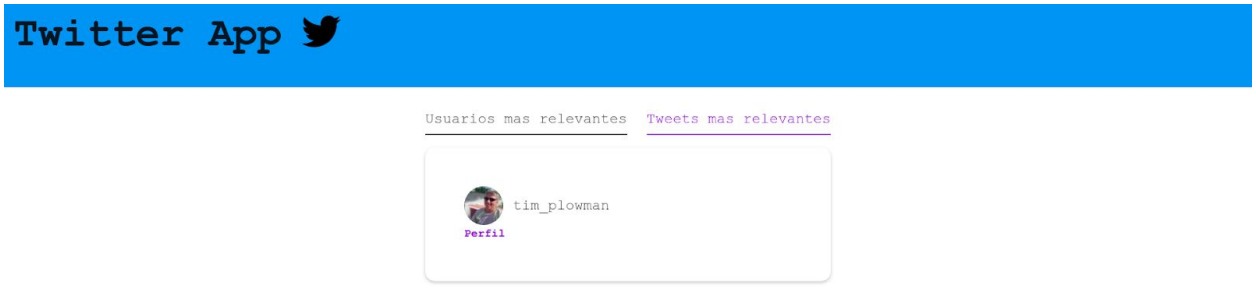
[4] Documentación oficial:
https://developer.twitter.com/en/docs/twitter-api/tweets/search/api-reference/

Funcionalidades y uso

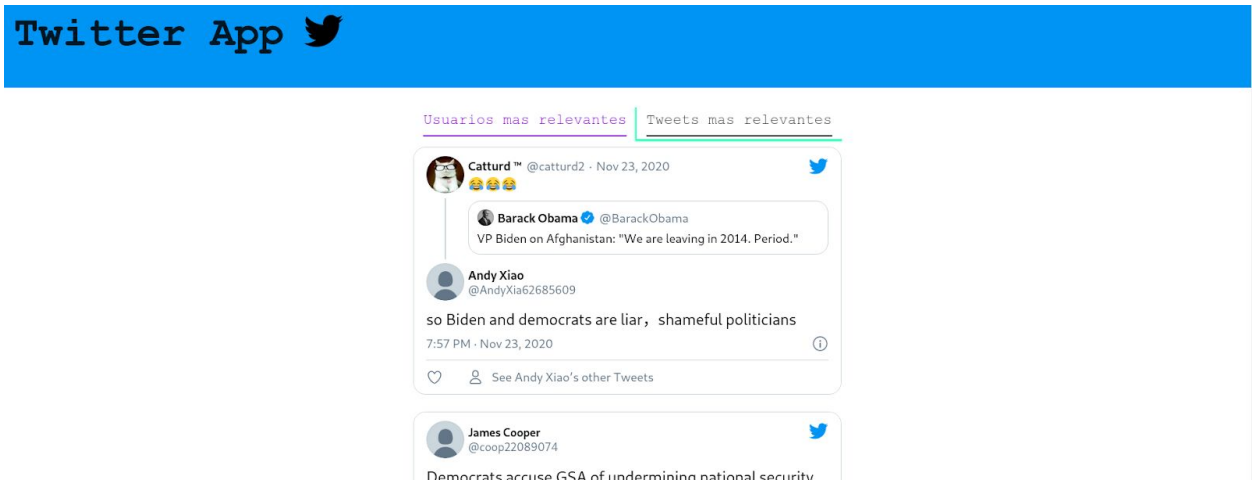
La aplicación cuenta con un menú inicial compuesto por botones y una espacio para introducción de texto. Los botones permiten seleccionar el partido político para el cual se desea realizar la búsqueda, eligiendo siempre uno y sólo uno en cada caso. Una vez seleccionado el partido político, se podrá introducir una temática a buscar dentro del cuadro de texto. El sistema buscará dentro de los *tweets* identificados con el partido político seleccionado aquellos que mejor se aproximen al tema de interés.

Los resultados arrojados por la consulta efectuada son desplegados a continuación, permitiendo al usuario ver aquellos *tweets* relevantes a su consulta y, además, el usuario autor del *tweet* más relevante. Para ello se utiliza nuevamente la API de Twitter, a los efectos de poder traer información interesante, como lo puede ser la foto de perfil del usuario, que añadan valor al sistema. Gracias a la información provista por la propia API de Twitter, la aplicación puede redirigir al usuario al perfil de aquella persona que escribió el *tweet* más relevante, así como también ser redirigido a cada uno de los *tweets* relevantes retornados por la consulta.

Por ejemplo, si un usuario buscase el término “Biden” asociado al Partido Demócrata, le sería presentada una interfaz como la que sigue a continuación:



Donde puede verse información básica del perfil del usuario encontrado. Además, pasando a la pestaña de *tweets* más relevantes podría ver lo siguiente:

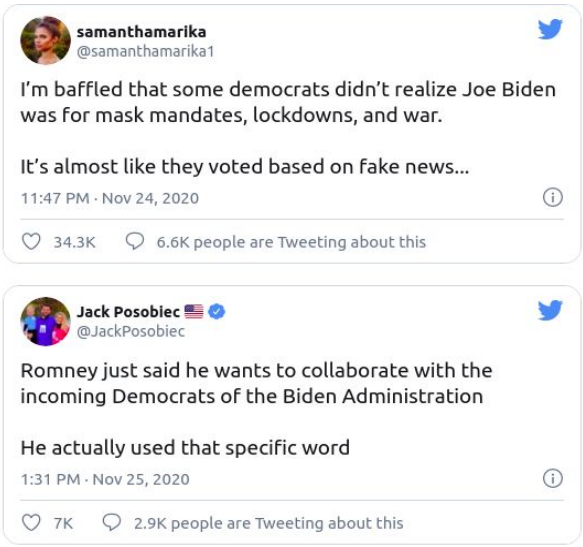


Un panel que ofrece la posibilidad de *scrollear* a través de los *tweets* más relevantes encontrados por el sistema. Como en el caso de los perfiles de usuario, haciendo clic sobre el tweet en cuestión se puede acceder al mismo en la página oficial de Twitter.

Resultados Obtenidos

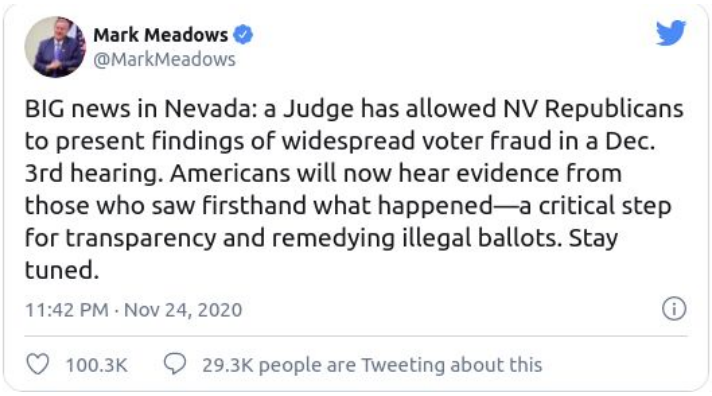
Se hicieron consultas en los dos partidos sobre temas del momento, “Biden”, “Fraud”, “Black Lives Matter”, “President”. Se consiguieron resultados muy interesantes relacionados con las consultas, entre ellos usuarios con muchos seguidores y tweets virales.

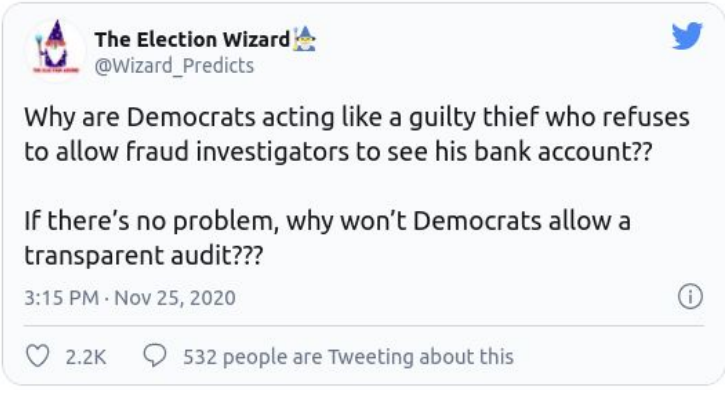
Al consultar por “Biden” en Demócratas (izquierda) y Republicanos (derecha) por igual:



Como se puede observar, ambos tweets tuvieron mucha difusión, y en especial el usuario “charliekirk11”, cuyo tweet se obtuvo al buscar entre republicanos, es famoso en este circulo, y cuenta con casi 2 millones de seguidores, claramente un referente en la temática.

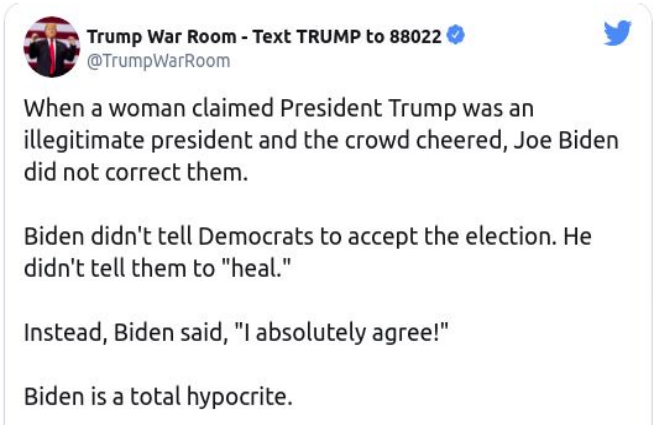
Consultando “Fraud” en ambos partidos:





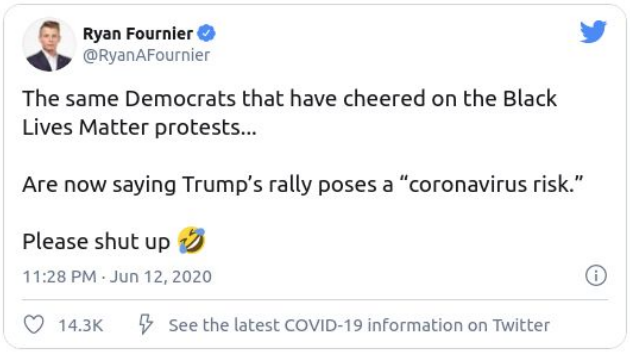
Tanto “MarkMeadows” como “Wizard_Predicts” tienen una gran cantidad de seguidores, lo que lleva a pensar que son usuarios importantes y que pueden tener influencia en la opinión del público con sus tweets.

En la consulta de “President” se encontraron los siguientes resultados:



Podemos ver que en el caso de la consulta dentro del partido demócrata, encontramos un tweet de “TrumpWarRoom”, una cuenta oficial que cuenta con más de un millón de seguidores, mientras que “marklutchman”, resultado de la búsqueda dentro del partido republciano, es un usuario que cuenta con más de 300 mil.

Por último, al consultar por “Black Lives Matter”:



Se puede observar, en el caso de consultar dentro del Partido Demócrata, 2 cuentas oficiales, con una gran cantidad de seguidores, mientras que en la consulta dentro del Partido Republicano, el algoritmo encontró un tweet en respuesta a la cuenta “AOC”. que pertenece a una senadora estadounidense de gran cantidad de seguidores.

A pesar de haber conseguido buenos resultados en ciertas consultas, se encontró el problema de que elasticsearch no diferencia entre el tweet original y un retweet, por lo que muchas veces el usuario obtenido mostrado en la pestaña de “Usuarios más relevantes” no es exactamente el autor del tweet sino alguien que lo retuiteó. Esto también fue una complicación a la hora de mostrar los tweets, ya que el *id* que se manda es el del tweet original y no el del retweet, por lo que se muestran tweets repetidos.

Conclusiones

El proyecto sirvió al equipo para comprender las posibilidades reales de recuperación de información en la web, así como para su almacenamiento y posterior procesamiento. El hecho de poder trabajar con una API tan completa como la de Twitter permite hilar muy fino en el tipo de datos que se puede obtener, lo cual ayuda a optimizar el espacio necesario para el almacenamiento.

El desafío de aprender una nueva tecnología, como fue el caso de *Elasticsearch*, siempre supone un desafío que conlleva riesgos. Sin embargo, el resultado final fue satisfactorio, pues se consiguieron superar los desafíos encontrados para terminar obteniendo el producto deseado. Resultó muy interesante trabajar con esta base de datos, porque provee herramientas de procesamiento de datos al momento de realizar las consultas que van más allá de las que normalmente se encuentran en las bases de datos tradicionales. Para nuestro caso particular, lo más interesante fue la posibilidad de procesar el contenido de los *tweets* para buscar palabras claves al momento mismo de realizar la consulta a la base de datos, de manera eficiente y aprovechando la indexación que el propio sistema realiza por detrás. Si hubiéramos tenido que implementar ese sistema nosotros mismos, sin dudas no hubiéramos obtenido resultados ni remotamente similares.

A pesar de lo antes expresado, nos hubiera gustado contar con mayores recursos para ver hasta dónde podría llegar un sistema de este tipo. El principal inconveniente que encontramos fue el límite en cuanto a la cantidad de datos que podíamos obtener de Twitter y almacenar en nuestro *hardware*. Esto hace que las posibilidades de búsqueda de términos relevantes queden bastante acotadas. Si tuviéramos la posibilidad de manejar los volúmenes de datos que una red social como Twitter maneja, sería posible obtener datos realmente interesantes, con un nivel de granularidad mucho mayor al que fuimos capaces de obtener en este proyecto.

Trabajo futuro

En línea con lo comentado en la sección correspondiente a las conclusiones, creemos que la principal línea de trabajo futuro está en obtener un mayor volúmen de datos y la correspondiente capacidad de procesarlos. Esto podría abrir la puerta a descubrir nuevas formas de tratar los datos y optimizar el funcionamiento del sistema. Además, haría mucho más interesante el trabajo, dado que podríamos obtener datos más precisos que arrojaran resultados más relevantes.

Por otro lado, nos gustaría poder trabajar más a fondo con *Elasticsearch*, ya que entendemos que recién estamos “rascando la superficie” de las posibilidades ofrecidas. Con mayor tiempo disponible, podríamos estudiar y probar a fondo las infinitas posibilidades de su lenguaje de consulta, ya que este es realmente amplio y complejo, resultando imposible lograr un conocimiento posible en el transcurso de este proyecto.

Otro punto que creemos interesante explorar, es la posibilidad de no restringir el espacio de búsqueda únicamente a Twitter. Si bien es una red social enorme y provee más información de la que podríamos llegar a procesar nunca, el reto de trabajar con otras API también resulta motivante. Sería interesante poder observar las diferencias entre los datos que proveen distintas fuentes para una misma consulta, ya que los perfiles de usuarios de cada plataforma no siempre coinciden en sus rasgos de personalidad, lo cual podría brindar resultados dignos de estudio.

Como detalle final, creemos que también la interfaz gráfica de la aplicación merecería un poco más de trabajo del que el plazo disponible nos permitió dedicarle. Sería interesante lograr un resultado más pulido estéticamente y con más funcionalidades disponibles para los usuarios de la aplicación. Hasta el momento, la aplicación cumple más las condiciones de un prototipo que de un sistema de producción, y por eso nos gustaría apuntar a mejorarlo.