



Recuperación de Información y Recomendaciones en la Web

Obligatorio 2018

InstaStalker

Integrantes grupo 17:

Daniel Martinez

Ernesto Fernandez

Javier Stábile

Leticia Errandonea

Maximiliano Bakkalian

Introducción	3
Problema y motivación	3
Enfoque de la solución	4
Diseño e Implementación	4
Arquitectura	7
Funcionalidades	8
Evaluaciones y Resultados	12
Conclusión	13
Trabajos a futuro	14
Referencias	15

Introducción

Hoy más que nunca las redes sociales están presentes en el día a día de los jóvenes y no tan jóvenes. Dentro de las principales redes sociales se destaca la red social Instagram. En esta red se comparten diariamente miles de fotos y videos pudiendo etiquetar el lugar donde fueron tomadas. Esa información es de mucho valor y no es explotada adecuadamente, para poder utilizarla y darle un valor agregado se creó InstaStalker.

InstaStalker es una aplicación web que a través de técnicas de recuperación de información y reconocimiento de imágenes recupera las fotos de miles de usuarios y les brinda una capa de procesamiento, dando como resultado la posibilidad de ver en un mapa geográfico las fotos subidas por usuarios.

Estas fotos son procesadas permitiendo reconocer objetos en ellas, con los cuales se pueden realizar búsquedas filtrando por estos objetos.

Este procesamiento extra permite brindar la posibilidad de saber qué está pasando en la ciudad o en un barrio específico, si hay algún evento o conocer el día a día de los usuarios, además, abre un abanico de muchas posibles utilidades.

Problema y motivación

Actualmente las personas están generando tanta información como nunca antes en la historia, esta información tiene un gran valor y puede ser muy beneficioso utilizarla, ya sea para comodidad de las personas o para fines de negocios.

Por un lado tenemos que en las personas se despierta la necesidad o la curiosidad de saber qué está sucediendo o qué está haciendo la gente en determinadas zonas de la ciudad. Por otro lado, tenemos que Instagram es la aplicación de moda entre los jóvenes, éstos comparten su vida cotidiana mediante fotos y videos donde muchas veces adjuntan datos de ubicación, es muy común que en cada salida se publiquen varias fotos, etiquetando ubicaciones y personas. Un ejemplo típico de uso de esta aplicación es tomar fotografías en la rambla, en boliches, en restaurantes o en cualquier tipo de salidas, eventos y reuniones con amigos.

InstaStalker debe ser capaz de obtener todas esas imágenes y videos compartidos y luego lograr ubicarlos en un mapa. Posteriormente debe ser capaz de procesar las imágenes para obtener información y poder etiquetarlas según su contenido. Este contenido puede ser mascotas, paisajes o comida entre otras.

Con esas etiquetas se debe ofrecer una búsqueda distinta a la que ofrece la misma aplicación. Pudiendo filtrar los resultados según las etiquetas y la localización.

Todo esto sumado a que se debe implementar una interfaz rápida y amigable. Para lograr la máxima rapidez en la búsqueda de tantos datos, para ello utilizaremos herramientas con ese propósito.

El objetivo es recolectar imágenes y videos que un usuario común no tendría acceso presentarlas de una forma diferente, útil y más fácil que la que la red social proporciona.

Enfoque de la solución

Desde el punto de vista del usuario, la aplicación debe consistir en un mapa en el cual se presentan de alguna manera imágenes capturadas por la comunidad de Instagram en la ubicación donde fueron tomadas, el usuario debe poder acercar y alejar el mapa mostrándole distinta cantidad de imágenes según el zoom elegido, esto para no saturar el mapa de datos.

A su vez, puede seleccionar filtros sugeridos o ingresar él mismo un filtro para reducir la cantidad de resultados devueltos.

Desde el punto de vista de la implementación, la aplicación debe recolectar y guardar datos de la red social. A través de alguna herramienta de reconocimiento de imágenes debe procesar las imágenes obtenidas y generar etiquetas.

Para poder realizar búsquedas rápidamente se indexan las etiquetas y las imágenes con una herramienta que sirva para búsquedas rápidas. Luego se deben poder ofrecer búsquedas de imágenes a través de etiquetas o localizaciones para poder ser consumidos por el frontend.

Diseño e Implementación

El sistema cuenta con:

- Interfaz gráfica web del usuario, desarrollada en Vue.
- Servidor backend, desarrollado en Node.js.
- Base de datos propia, en MongoDB.
- Elasticsearch : este servidor de búsqueda nos permitió indexar y acceder a los tags que se obtienen del procesamiento de imágenes de manera rápida y eficiente.

La solución consta de varias etapas:

- Recolección de datos
- Procesamiento de los datos
- Indexación de los datos
- Presentación de los datos

La etapa de **Recolección de los datos** como su nombre lo indica tiene como objetivo obtener las imágenes y videos publicados por los usuarios en la red social. En un principio se pretendió obtener las imágenes por medio de una API proporcionada por Instagram, luego de varias pruebas e investigación se determinó que ésta estaba deprecada y no se podía usar para el fin requerido. Por lo tanto, se decidió por utilizar una técnica de scraping. Como la mayoría de las imágenes no son públicas, para obtenerlas se requiere tener una cuenta y seguir al usuario. Para eso se creó la cuenta ok.webir, desde el Backend se implementó un proceso el cual realiza un login de la cuenta en la versión de pc de Instagram y se le envían solicitudes para seguir a todos los usuarios que la red social sugiere, de esta forma se obtiene el permiso de recibir imágenes de estos usuarios desde la pantalla principal. De estas imágenes se puede extraer algunos datos del usuario que publicó la foto así como los tags y ubicación que él mismo proporcionó. Esta información junto con las imágenes es almacenada en una base de datos no relacional MongoDB.

La etapa de **Procesamiento de los datos** consiste obtener las imágenes que aún no fueron procesadas y se encuentran guardadas en la base de datos, para luego procesarlas con la herramienta de reconocimiento de imágenes Google Vision. Para utilizar los servicios de Google Vision se creó una cuenta de Google y se utilizaron los 300 dólares gratuitos que presta el servicio como “trial”. Para ejecutar el proceso se crea un “job” en el Backend el cual busca imágenes aun no procesadas, las envía a la Api de Google Vision y obtiene el conjunto de etiquetas reconocidas. Una vez obtenidas las etiquetas, se actualiza la entrada de la imagen en la base de datos agregando el conjunto de etiquetas. Por último se dispara la ejecución del proceso que realiza la siguiente etapa.

En la etapa de **Indexación de los datos** se utiliza la herramienta elasticsearch. Elasticsearch permite almacenar documentos en formato JSON y realizar búsquedas muy rápidas sobre los valores de estos. Con los datos almacenados en mongoDB se crean JSON con los siguientes campos:

```
{
  storyId: #####;
  location: #####;
  tags: [##,##,##];
}
```

Luego se realiza un POST de estos documentos a la API de elasticsearch y quedan disponibles para una consulta en el caso que se requiera filtrar por tags, location o storyId.

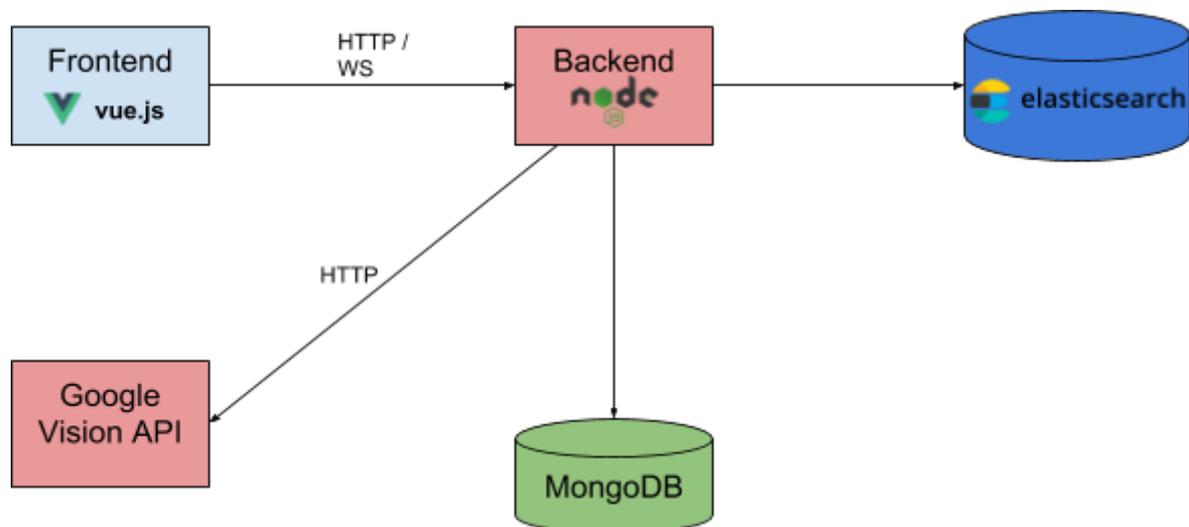
La etapa de **Presentación de los datos** se realizó integrando el mapa de Google Maps, mostrando los distintos tags referentes a las imágenes, esparcidos por los

distintos puntos del mapa. Si se clickea uno de los tags, se muestra la o las fotos vinculadas a dicho tag que supuestamente se tomó en dicha ubicación (es un supuesto ya que la ubicación se lee desde la entrada del usuario). También, cuánto menos zoom se haga en el mapa menos tags se pueden visualizar, mientras que cuando se va acercando el zoom, se van mostrando más tags. Esto último es para no contaminar el mapa cuando una zona es muy pequeña debido al zoom aplicado. Además, para adaptarse aún más a la preferencia del usuario, se cuenta con un buscador por tag o ubicación, por ejemplo posibles tags serían: perro, fiesta, comida, etc. o una ubicación podría ser: Parque Rodó, Facultad de Ingeniería, etc.

Arquitectura

El sistema tiene una arquitectura cliente-servidor, donde el cliente está basado en Javascript y HTML utilizando el framework de Javascript Vue JS. Se cuenta con una base de datos MongoDB para almacenar los datos de las imágenes de las historias de instagram recuperadas incluyendo su ubicación, tags y su imagen .El sistema cuenta además con un servidor ElasticSearch para poder acceder de forma rápida a los datos. El módulo del servidor backend es el encargado de:

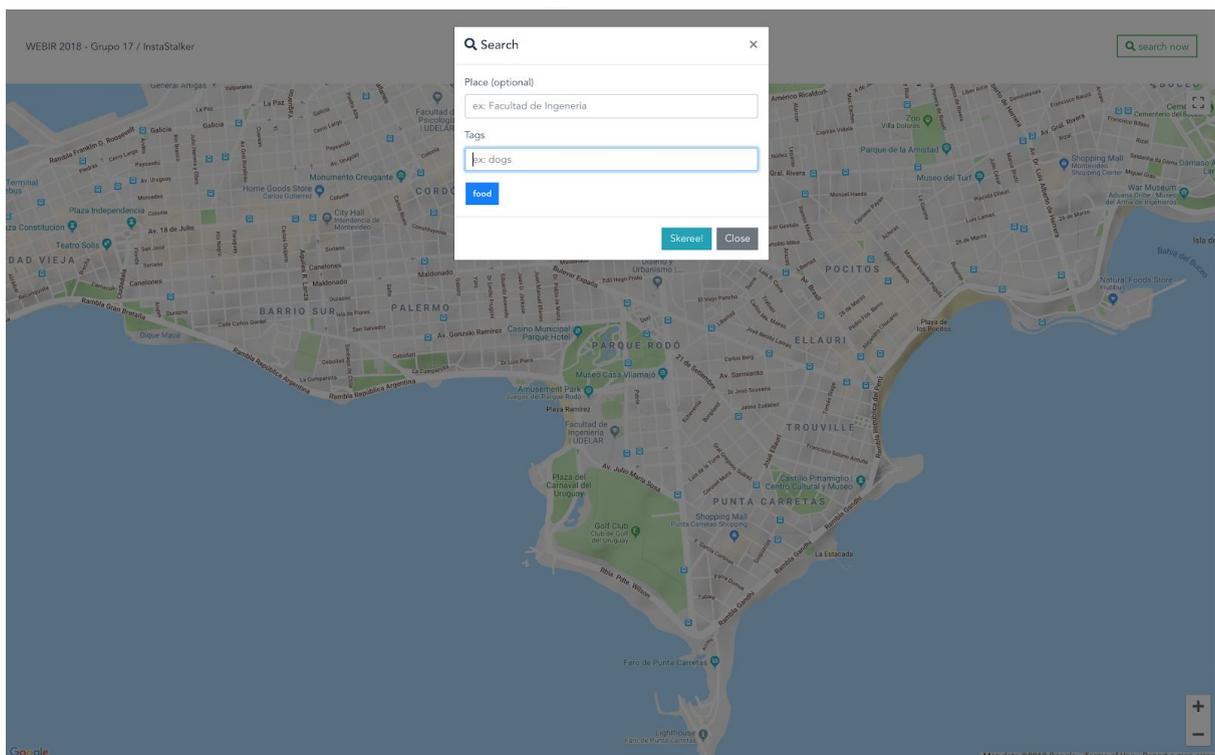
1. Realizar el web scraping para obtener las historias de la página de instagram
2. Guardar las historias en la base de datos
3. Procesar las imágenes con la API de Google Cloud Visión
4. Hacer de intermediario entre el usuario final y ElasticSearch para realizar las consultas por tags.



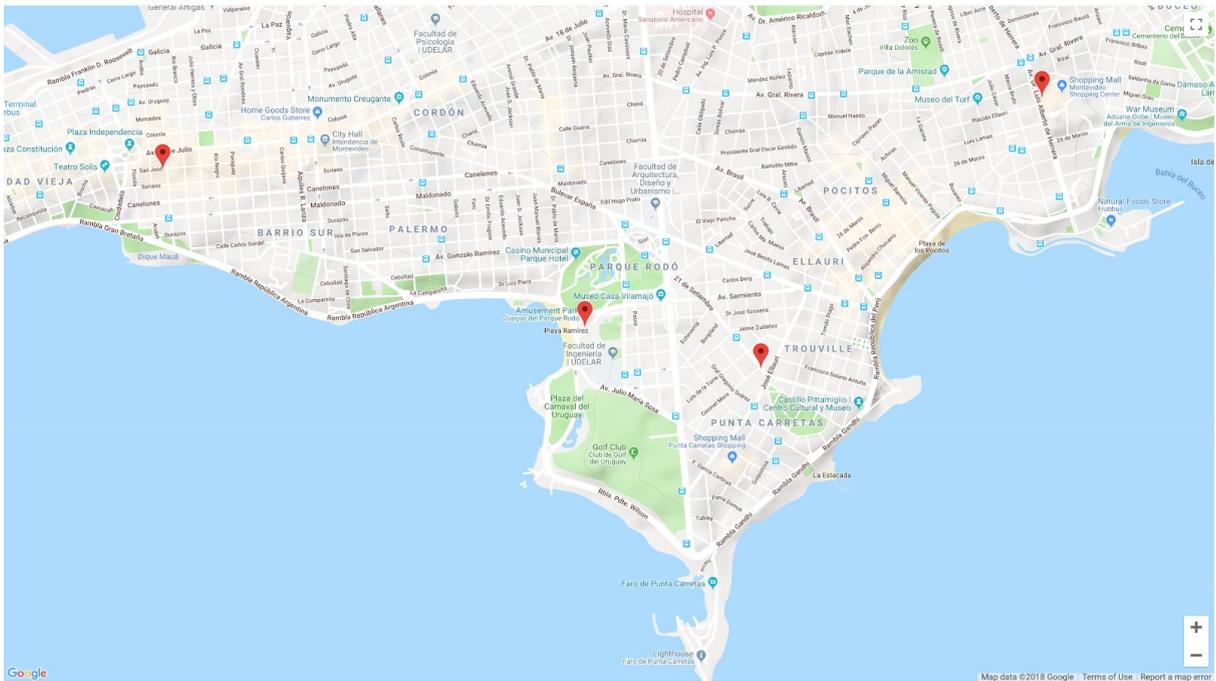
Funcionalidades

- Ver distintos tags en el mapa: Al ingresar a la aplicación, se visualizan distintos tags colocados en las ubicaciones correspondientes a las historias asociadas a cada tag.
- Clickear un tag para ver imágenes asociadas: Se puede hacer click sobre un tag para ver las historias asociadas a dicho tag.
- Búsqueda por tags o ubicación: Es posible filtrar las historias a visualizar mediante un campo de texto.
 - Búsqueda por tag

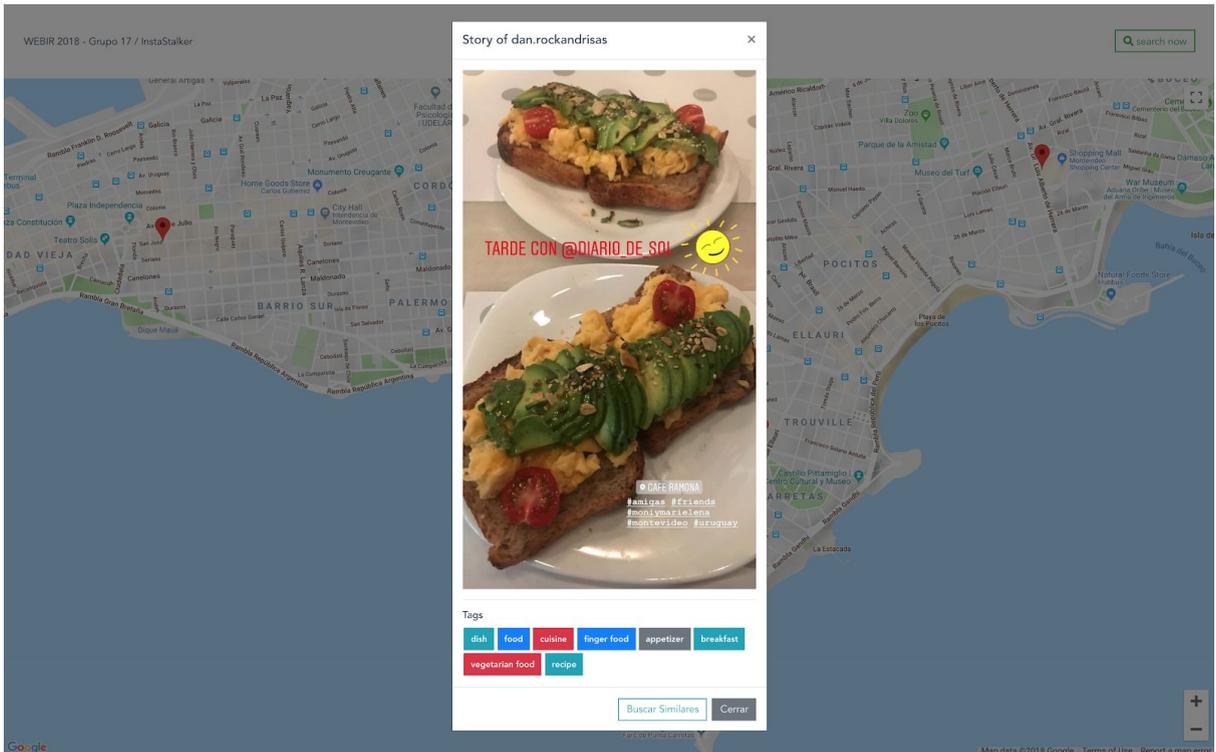
Buscamos todas las historias con el tag “food”.



Luego, vemos que ubica las historias con dicho tag con un pin en el mapa.

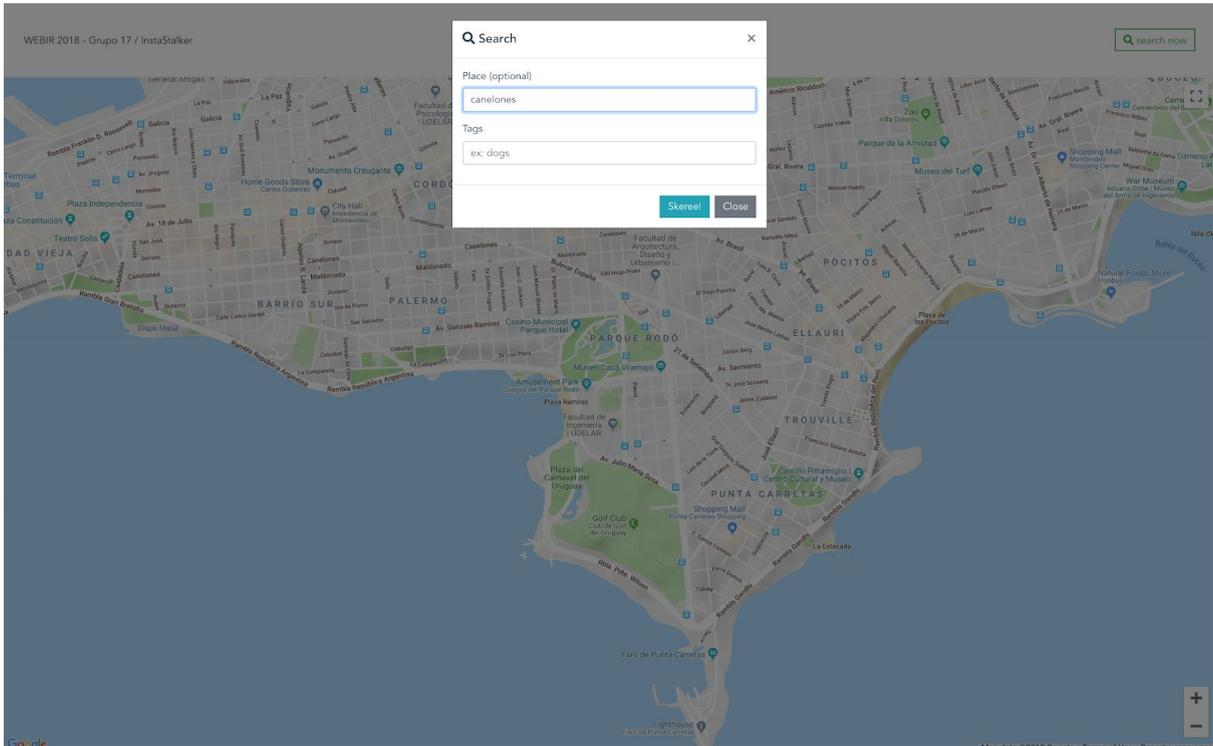


Al hacer click en un pin, se despliega un modal en el que vemos la historia, el nombre del usuario y todas las tags detectadas por GV.

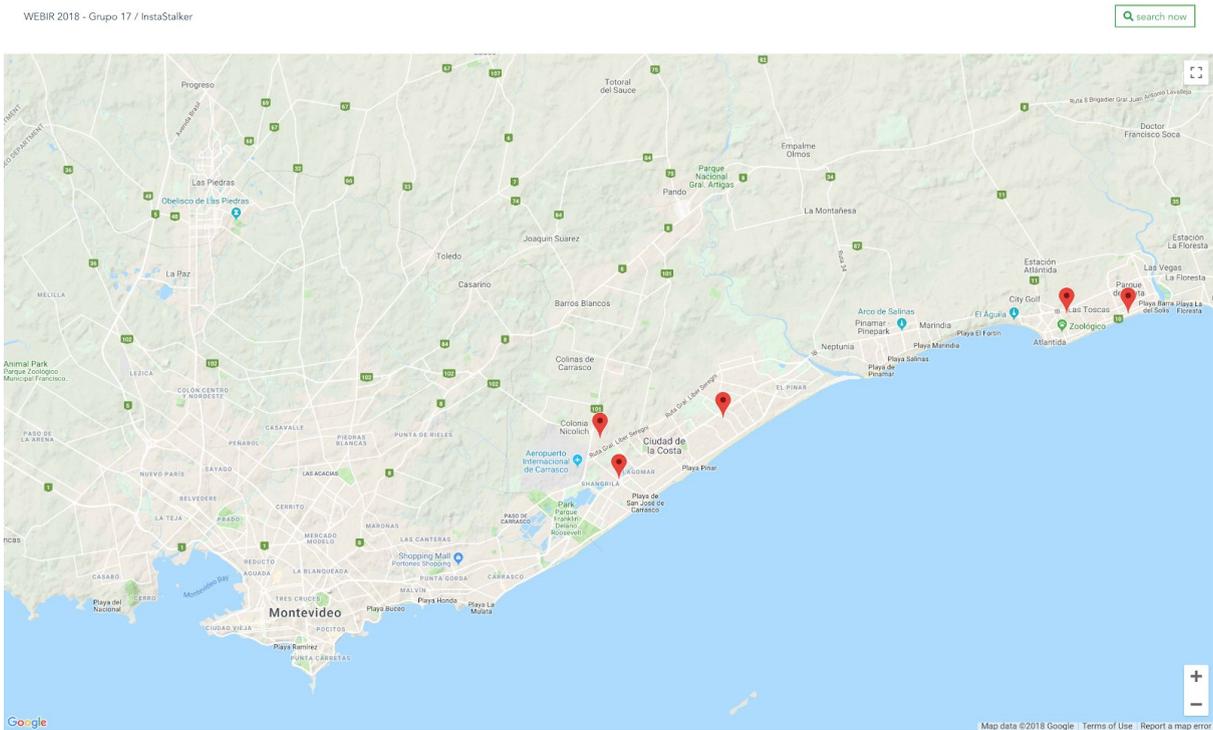


- Búsqueda por ubicación

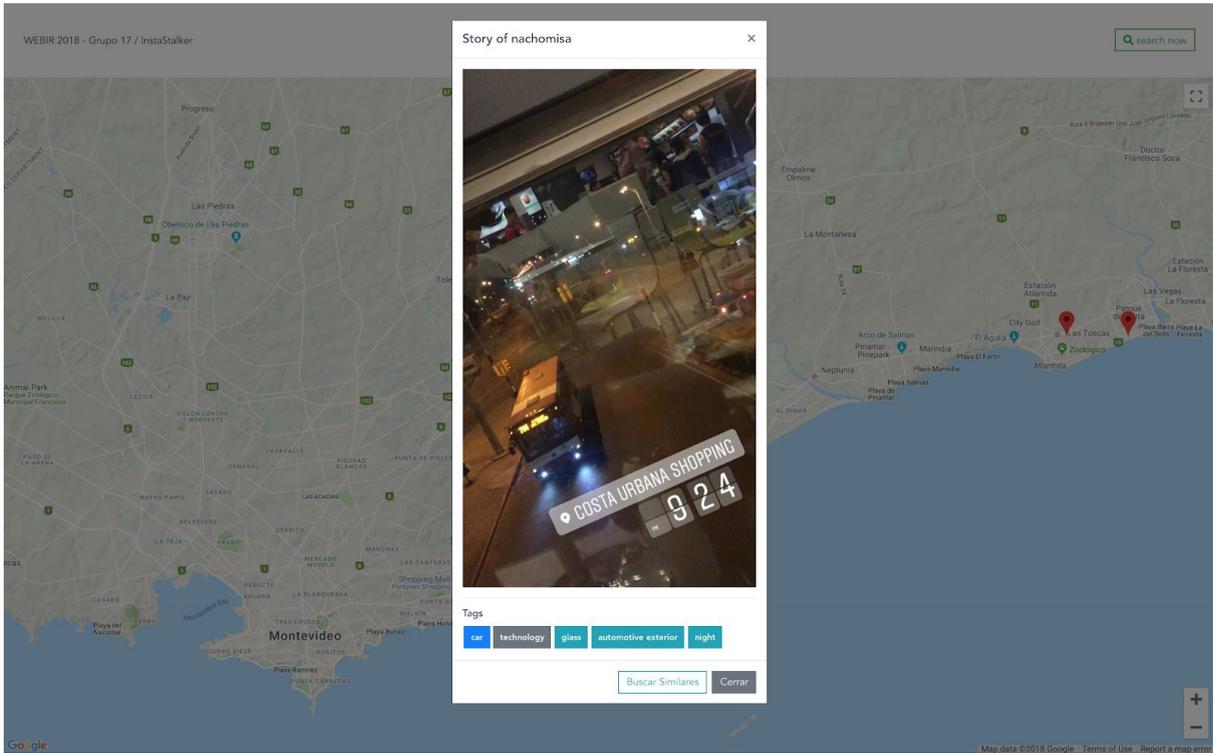
Buscamos todas las historias en la ubicación Canelones.



Luego, vemos que ubica las historias con dicha ubicación con un pin en el mapa.

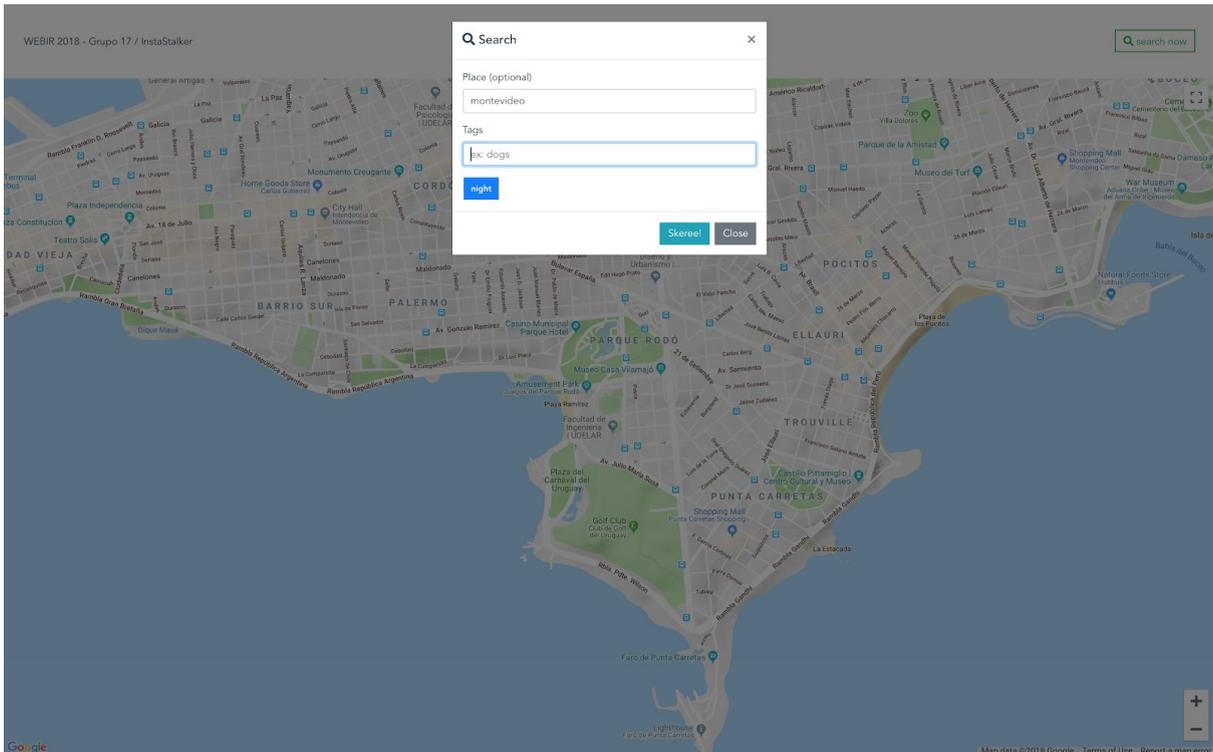


Al hacer click en un pin, se despliega un modal en el que vemos la historia, el nombre del usuario y todas las tags detectadas por GV.

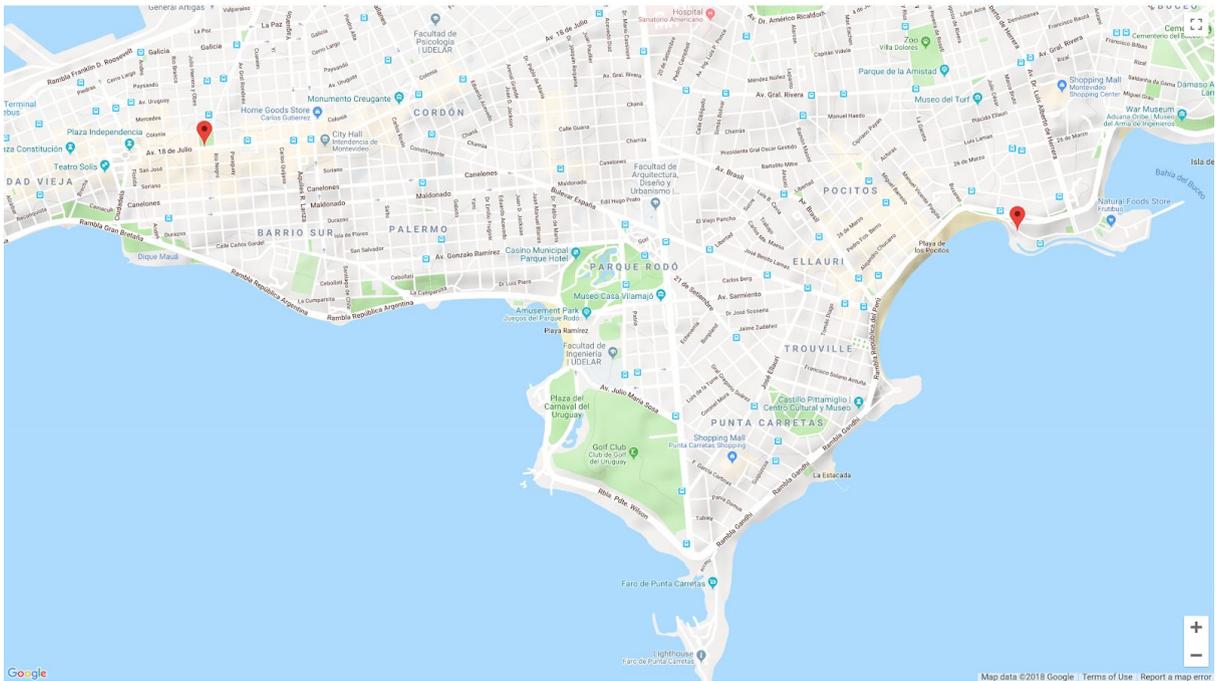


- Búsqueda por tag y ubicación

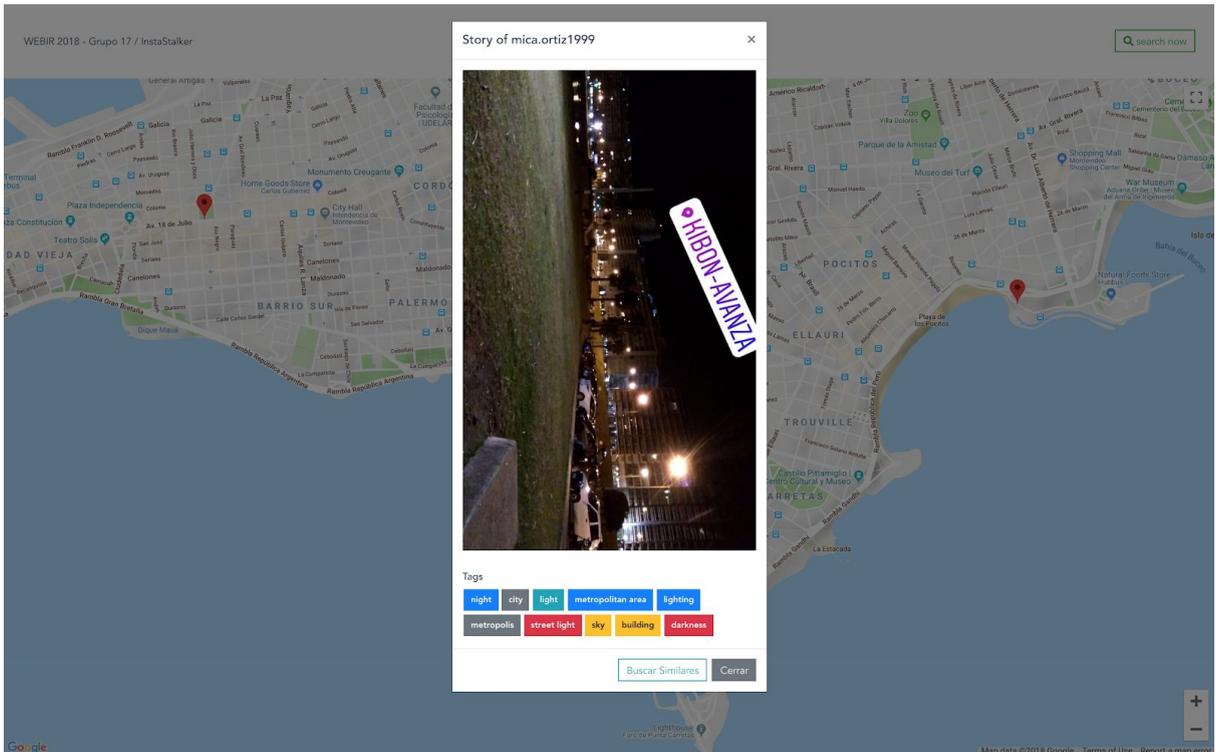
Buscamos todas las historias en la ubicación Montevideo con el tag “night”.



Luego, vemos que ubica las historias con dicha ubicación con un pin en el mapa.



Al hacer click en un pin, se despliega un modal en el que vemos la historia, el nombre del usuario y todas las tags detectadas por GV.



Evaluaciones y Resultados

Para evaluar la performance de la aplicación se tuvieron en cuenta la velocidad de respuesta ante una búsqueda y el espacio en disco que ocupan los datos obtenidos.

También se tuvo en cuenta el apartado de interacción con la interfaz.

En cuanto a la velocidad de respuesta de la aplicación se tiene un resultado muy bueno, ante cualquier búsqueda, la respuesta fue excelente, presentando todos los resultados consistentes con los filtros inmediatamente.

Luego de seguir a varias personas y obtener sus fotos publicadas se llegó a una cantidad de unos cientos de fotos. El promedio de peso de las fotos fué de 20Mb cada 100 fotos, lo que es una relación de peso-foto muy buena, esto hace viable el almacenar el contenido publicado por los usuarios en nuestra propia base de datos. De lo contrario, no se podría realizar una persistencia de las imágenes y estaríamos sujetos a la persistencia de las imágenes en la red social.

Desde el punto de vista de la interacción, InstaStalker es muy intuitiva de usar, el 90% de la pantalla es el mapa y tan solo un 10% corresponde al cabezal de opciones lo que genera una sensación de sencillez (también se puede utilizar el modo pantalla completa). Asimismo, los filtros son muy fáciles por lo que no requieren práctica para utilizar.

Conclusión

Como primera conclusión, destacamos la oportunidad de trabajar con distintas prácticas (scrapping), así como tecnologías (Elasticsearch) y servicios externos que de otro modo no hubiera ocurrido. Al trabajar en esta herramienta se pudo dimensionar el potencial que tiene el área de recuperación de datos en la web y poner en práctica conocimientos adquiridos en el curso, como ser la búsqueda de palabras en un índice.

Podemos decir que se logró implementar un sistema funcional, el cual logra centralizar y recuperar la información de las historias de la aplicación instagram en un solo sitio. Es posible buscar y filtrar información según la ubicación o palabras claves que se especificaron y poder visualizarlas en el mapa, sobre todo de forma fácil y sencilla para los usuarios. Por lo tanto podemos afirmar que nuestra aplicación cumple los requerimientos básicos definidos por el equipo para el alcance del proyecto

Por otro lado, pudimos comprobar la cantidad de información que se puede recabar de internet con datos que la gente deja públicos. En nuestro caso, sin tener la expertise ni haberle dedicado un tiempo considerable, pudimos obtener imágenes (con contenido de personas, animales, inmuebles, autos) con sus ubicaciones, de las historias que los usuarios suben a instagram frecuentemente. Esta información estando expuesta, en manos de gente malintencionada puede ser peligrosa.

Un problema con el que nos topamos, es que la red Instagram, elimina la metadata de las imágenes, por lo tanto no podíamos conocer la ubicación a partir de esta información. Por lo tanto, no todas las historias son útiles en esta aplicación, sino sólo aquellas en las que el usuario haya asignado una ubicación manualmente. Esto es una limitante importante, ya que gran parte de las historias no contienen ubicación, lo que significa una gran pérdida de información que podría ser interesante. Esto no es una limitante si fuera una funcionalidad oficial de la red, ya que podrían contar con la metadata de las fotos.

Trabajos a futuro

Una posible mejora, y poco costosa de implementar, al proyecto es que sea una aplicación en donde cada usuario que la va a usar se loguee con su usuario de instagram, en vez de que sea un usuario (ok.webir) transparente para el usuario final y que sigue a muchas personas al azar. Con esto el proyecto se transformaría en una nueva funcionalidad para cada usuario de instagram, permitiendo ver historias de acuerdo a los tags que resulten interesantes, y no solamente por el usuario que la sube. Con esta funcionalidad el proyecto es más realista ya que no consta simplemente de ver historias de personas que tal vez conozca o no, si no que de los que sigue cada usuario y le podrían llegar a interesar.

Otra funcionalidad que se podría agregar sería la de realizar una puntuación de las ubicaciones según la información obtenida en las fotos. Si en una ubicación se toman muchas fotos de fiestas, ésta podría puntuarse como una ubicación donde las personas se divierten, o si en una ubicación se toman fotos de parejas, esa podría ser catalogada como romántica. Con esas puntuaciones y catalogaciones se le puede sugerir al usuario distintos establecimientos o lugares para visitar. Esto abre la posibilidad de vender una recomendación o una publicidad para distintos establecimientos.

Una funcionalidad nueva a desarrollar que utiliza la ubicación del usuario es la de sugerir intentar simular una foto tomada por otro usuario cuando pasa por el mismo lugar. A esta foto se la procesaría y se le daría una calificación de acuerdo a la exactitud de la copia. De esta forma un usuario que pasa por la rambla por ejemplo, puede intentar copiar una selfie hecha por otro usuario en ese mismo lugar. Según la calificación obtenida se lo ubica en un ranking diario, semanal y mensual de la mejor foto copiada.

Referencias

- <https://www.instagram.com>
- <https://www.elastic.co>
- <https://cloud.google.com/vision/>
- <https://nodejs.org/es/>
- <https://vuejs.org/>
- <https://www.mongodb.com/es>