RECUPERACIÓN DE INFORMACIÓN Y RECOMENDACIONES EN LA WEB

CURSO 2018

GRUPO 19

BeneZip

Andrés FULLONI	CI: 4.629.460-7
Damián LANGONE	CI: 4.651.246-3
Facundo PADULA	CI: 4.864.256-3

NOVIEMBRE 2018

Índice

1. Introducción

El presente documento se realiza en el marco del proyecto final del curso de Recuperación de Información y Recomendaciones en la Web año 2018. Se intenta aplicar los conocimientos aprendidos para proponer y ejecutar una posible solución al problema descrito en la sección siguiente.

Para abordar el problema se pretende tener en cuenta distintas técnicas de recolección de datos discutidas en clase como *scraping* o la consulta de datos de terceros expuestos a través de una *API*. Una vez obtenidos los datos también nos resulta de interés poder presentarlos al usuario de una manera que le resulte eficiente e intuitivo para realizar diferentes consultas de la información obtenida.

2. Problema

Hoy en día es cada vez más común el uso de tarjetas de crédito y débito otorgadas por alguna institución financiera o bancaria. Junto a éstas vienen asociados un montón de beneficios y descuentos en distintos rubros que varían según la institución que otorgue la tarjeta.

Dado el amplio abanico de beneficios y descuentos, la exclusividad de éstos y el valor que los clientes ponen sobre los mismos, la decisión de qué tarjeta adquirir no es una decisión para nada directa y sencilla, e incluso en el caso de adquirir más de una surge la problemática de en qué situación conviene usar una o la otra. Ante tal situación proponemos centralizar los beneficios de algunos de los bancos y sellos de tarjetas más importantes del país en una aplicación web, manteniendo el foco en el objetivo y alcance de la materia. Si bien estas instituciones cuentan con sus propias páginas web donde publican sus beneficios, cada una lo hace de manera diferente y no se encuentran centralizados.

Hoy en día existe una aplicación móvil para iOS y Android, la cual resuelve la problemática antes mencionada pero no cuenta con una versión web. Por estos motivos la idea que se plantea es que toda esta información se encuentre homogeneizada y unificada en un único lugar de forma que el usuario no deba ingresar a las distintas páginas web por separado para encontrar la respuesta a su consulta.

3. Enfoque de la solución

La solución propuesta para abordar el problema descrito en el punto anterior fue desarrollar un sistema que, mediante la utilización de los mecanismos de recuperación de información de la web, reúna los beneficios brindados por las distintas instituciones y bancos, y se los presente al usuario de forma centralizada y amigable. El sistema no tiene como único objetivo reunir los diferentes beneficios existentes con las distintas tarjetas en un sólo lugar, sino que también busca otorgarle al usuario diferentes herramientas para que pueda realizar una búsqueda personalizada.

En un principio se pensó un enfoque de solución basado en una arquitectura cliente-servidor, donde el servidor consumiera los datos *scrapeados* de los sitios web de las instituciones y publicara servicios para que el cliente utilice. Esta solución luego fue descartada ya que se decidió utilizar el motor de búsqueda Elasticsearch [1], el cual permite realizar consultas a documentos indexados de una forma sencilla por lo que no se creyó necesaria la utilización de una base de datos ni de un servidor de backend.

Por lo tanto, la solución finalmente desarrollada consiste en una aplicación web compuesta por un componente de frontend, el cual maneja las acciones realizadas por el usuario e interactúa con el servidor de Elasticsearch. Aspectos como las tecnologías utilizadas, el diseño y la implementación del sistema serán detallados en las siguientes secciones.

Antes de comenzar con el desarrollo fue necesario definir el alcance del sistema, ya que con el tiempo disponible resultaría imposible integrar los beneficios de todas las tarjetas emitidas en nuestro país. Para esto se realizó una investigación de los sitios de las distintas instituciones. Al ver que estas instituciones no brindaban una API para poder obtener la información de los beneficios, se decidió emplear técnicas de *scraping*. Las instituciones elegidas fueron:

- **OCA [5]**
- **Scotiabank [6]**
- **Banco República [7]**

4. Diseño

A continuación se presenta un diagrama de la solución propuesta:

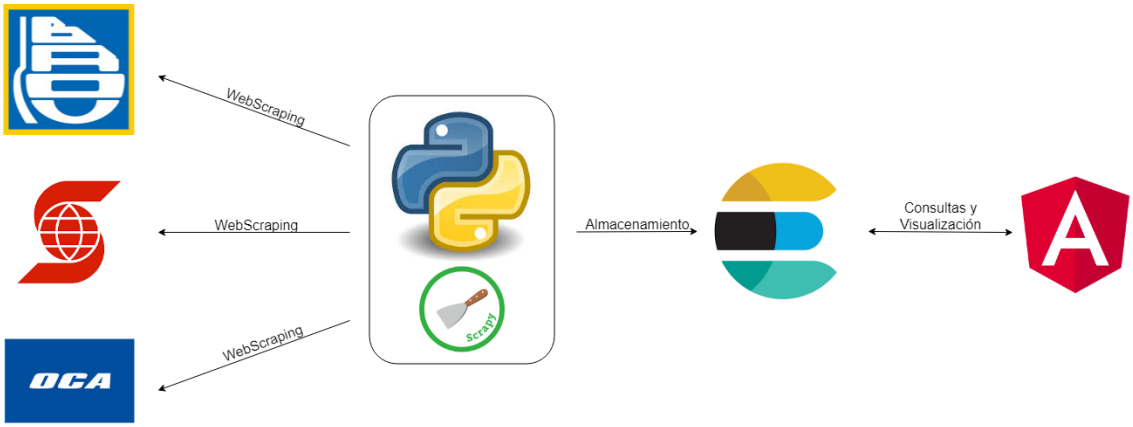

Figura 1: Diagrama de la solución

Como se aprecia en la imagen anterior el proceso comienza *scrapeando* las distintas páginas web. Esto se realiza utilizando el framework Scrapy [2] de Python [4]. Debido a la heterogeneidad de presentación de los beneficios de las distintas instituciones se debieron crear distintas *spiders*[1] y adaptarlas a la estructura de cada página web.

Una vez obtenida la información del sitio web, se estructura en un JSON el cual contiene los datos relevantes del beneficio y se envía al servidor de Elasticsearch utilizando su interfaz RESTful. Una vez almacenados en Elasticsearch, los datos pueden ser eficazmente consultados (también mediante su interfaz RESTful) por la aplicación de Angular que presenta los datos al usuario final de una manera centralizada y amigable, además de ofrecerle distintos filtros y opciones de búsqueda.

[1] Las *spiders* son clases que definen cómo un cierto sitio web será *scrapeado*, incluyendo cómo se realizará el *crawling* (seguimiento de links) y cómo extraer datos estructurados del sitio.

5. Implementación

Para lograr desarrollar el sistema se debió dividir el problema en varias partes, las cuales fueron abordadas individualmente. Dichas etapas se detallan a continuación.

5.1. Análisis de los sitios web de las instituciones que emiten tarjetas

La primer etapa consistió en realizar un análisis del código fuente de los sitios web de las instituciones elegidas con el fin de encontrar las etiquetas HTML que contenían la información relevante para el problema en cuestión. Para esto se utilizó el plugin de Chrome SelectorGadget [8]. Esta herramienta permite, simplemente haciendo clic sobre un elemento de la página, obtener información de estilo relevante del mismo, como el nombre de la clase y la etiqueta HTML. Utilizando esta herramienta se reconoció la información más relevante así como también las referencias a las imágenes, necesarias para su utilización en la aplicación web.

Un problema que se reconoció en esta etapa, fue que los sitios no presentan toda la información relevante del beneficio en un sólo elemento HTML; si no que muchas veces presentan una pequeña introducción y en otra vista presentan una descripción más detallada.

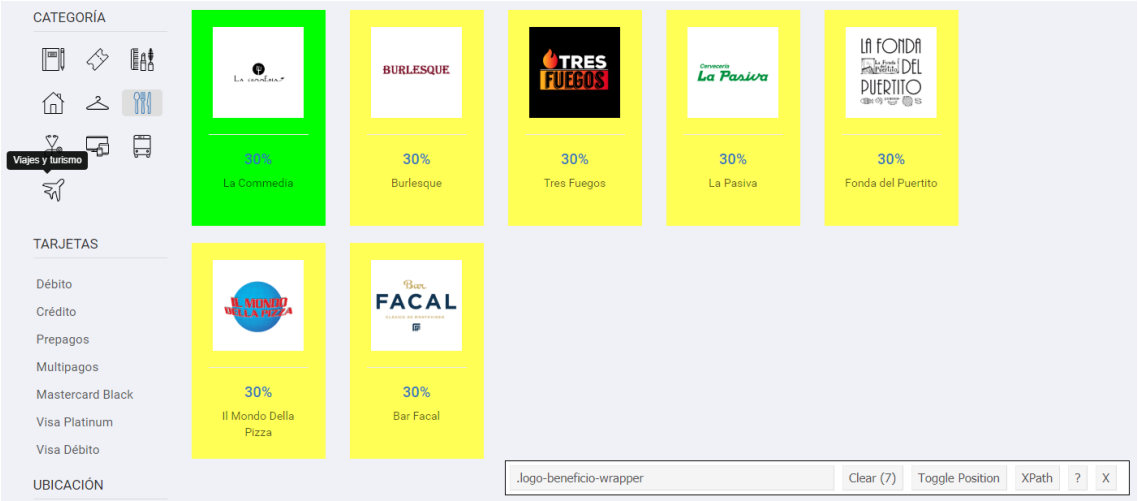

Figura 2: Utilización de SelectorGadget en sitio web del BROU

5.2. Extracción de información de los beneficios

Una vez identificada la información relevante se generan las *spiders* correspondientes para obtenerla. Se realizó una *spider* para cada institución ya que la estructura de los sitios es diferente y por lo tanto se tuvieron que desarrollar funciones de *parsing* distintas. Como se mencionó anteriormente, la herramienta utilizada para esta tarea fue Scrapy. El procedimiento en general fue el siguiente:

- Primero se definieron las urls de los sitios a *scrapear*, por ejemplo "https://beneficios.brou.com.uy/".
- Definir la función de *parsing*. Se obtuvo la información recorriendo el contenido de la página mediante la utilización del lenguaje XPATH [10].
- Para los componentes que tenían un link a otra página con más información se realizó el mismo procedimiento llamando a una función de parsing auxiliar.

5.3. Procesamiento de depuración de los datos obtenidos

Dado que los datos crudos obtenidos muchas veces contenían información innecesaria o presentaban un formato no adecuado, fue necesario cierto procesamiento para asegurar calidad y homogeneidad en los datos persistidos. Esto se realizó de forma automática sobre los datos *scrapeados* y persistidos en archivos de texto. A modo de ejemplo, necesitamos depurar los datos obtenidos del sitio web del BROU. Aquí tenemos por un lado el título del beneficio, que corresponde con el nombre del negocio que lo ofrece, y por otro lado al comienzo de la descripción se encuentra nuevamente el nombre del negocio. Para este caso en particular, el procesamiento realizado consistió en *parsear* la descripción y quitar el nombre del local.

5.4. Carga de los datos al servidor de Elasticsearch

Una vez realizado el procesamiento se procedió a enviar los datos al servidor de Elasticsearch. Para ello, se creó un script en Python para los datos de cada institución, el cual se encargaba de crear el JSON correspondiente con los atributos relevantes de cada beneficio. Los atributos seleccionados fueron: “category”, que indica la categoría del beneficio, por ejemplo "Vestimenta"; "store", que indica la empresa sobre la cual se aplica el beneficio; "description", que contiene la descripción del beneficio; “url”, que contiene la url con la cual se accede al beneficio en el sitio web de la institución

que lo ofrece; e “image”, que contiene la dirección donde se encuentra almacenada la imagen. Una vez realizado esto, el mismo script se encargaba de abrir una conexión con el servidor de Elasticsearch y enviar los JSON anteriormente creados mediante el método POST de HTTP para su persistencia.


```
{
    "_index": "scotiabank",
    "_type": "benefits",
    "_id": "RGnhCWcBYR3JKgCoAOfU",
    "_score": 1,
    "_source": {
        "category": "Circuito Gourmet",
        "store": "Café Martínez",
        "description": "De lunes a viernes 15% de ahorro con Tarjetas de Crédito y Débito Soctiabank y 25% Con Tarjetas Gold",
        "uri": "/Tarjetas/Promociones/Restaurantes/cafe-martinez",
        "image": "//scotiabankfiles.azureedge.net/scotiabank-uruguay/img/campanas/tiles/cafe_martinez.jpg"
    }
},
```


Figura 3: Datos persistidos en Elasticsearch

5.5. Desarrollo de aplicación de FrontEnd

La solución cuenta con una aplicación SPA (Single Page Application) desarrollada en Angular 6 [3] como plataforma de FrontEnd para la presentación y visualización de la información. La misma cuenta con servicios que contienen métodos para realizar las consultas necesarias (como búsquedas o filtrado) al motor de Elasticsearch. En los servicios se crea la consulta correspondiente a los filtros y/o búsqueda seleccionados y se envía la misma mediante el método HTTP correspondiente hacia el servidor de Elasticsearch para obtener los resultados. Luego, se procede a presentar los mismos en el componente adecuado de Angular para su correcta visualización.

En el servicio de consultas de la aplicación se debió solucionar el problema de la heterogeneidad de los datos extraídos. Este problema surge dado que las instituciones utilizan nombres distintos para categoría de beneficios semánticamente similares. Un ejemplo de esto se puede ver en los sitios de Scotiabank y BROU. En el primero, se utiliza la categoría “Circuito Gourmet” para agrupar los beneficios relacionados con descuentos en locales de comida; sin embargo, en el sitio de BROU estos beneficios se encuentran bajo el nombre de "Gastronomía". Para solucionar este problema sin modificar los datos en Elasticsearch se decidió realizar consultas que abarquen, mediante disyunción, los distintos nombres que referencian al mismo tipo de categoría.

6. Funcionalidades y uso

En esta sección se detallarán y explicarán las diferentes funcionalidades que presenta la aplicación y se darán pautas para correcto uso de la misma.

Al ingresar al sitio web de BeneZip se muestran todas las opciones de beneficios disponibles en el momento de manera que el usuario pueda tener una referencia de los datos correspondientes a los beneficios.

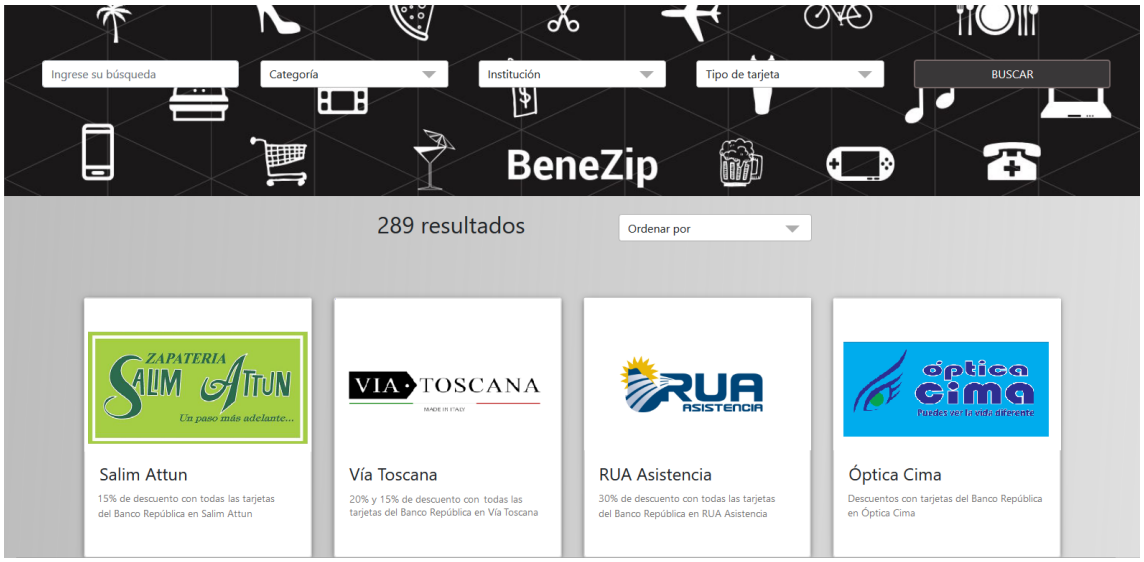

Figura 4: Inicio de la aplicación

A continuación se detallarán las funcionalidades de la aplicación, las cuales se pueden combinar para realizar una búsqueda avanzada.

6.1. Filtrado

En la parte superior de la pantalla se dispone de diferentes dropdowns, junto a un campo de texto y a un botón de “Buscar” para realizar la búsqueda por los filtros especificados. Se puede seleccionar información tanto en un solo dropdown como en múltiples a la vez, así como seleccionar múltiples elementos en cada uno de los mismos de manera de generar un filtrado personalizado. A continuación se detallan los diferentes filtros posibles a utilizar:

- **Categoría:** el usuario puede filtrar por la categoría a la cual pertenece el beneficio. Las categorías disponibles son Vestimenta, Especiales, Entretenimiento,

Gastronomía, Enseñanza, Niños, Hogar, Turismo, Transporte, Tecnología, Estética, Joyerías, Ópticas y Supermercados.

- **Emisor de la tarjeta:** el usuario puede filtrar por la institución emisora de la tarjeta. Como se señalo anteriormente, las instituciones disponibles son Scotiabank, BROU y OCA.
- **Tipo de tarjeta:** los tipos de tarjeta disponibles para filtrar son Crédito y Débito.

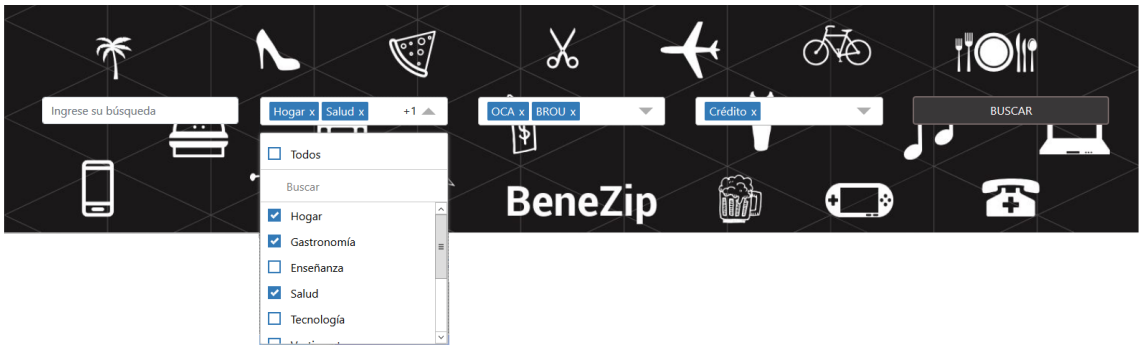

Figura 5: Componente de búsqueda y filtrado

6.2. Buscador

A su vez, en la parte superior junto a los filtros, se dispone de un campo de texto como se menciono anteriormente, mediante el cual el usuario puede realizar una búsqueda libre sobre la información del beneficio. Es decir, el sistema retornará los beneficios que contengan el texto ingresado en su información.

Algo a destacar de esta funcionalidad es la posibilidad de visualizar posibles resultados a medida que se escribe en el campo de texto. Esto es posible gracias a la funcionalidad “Query-Time Search-As-You-Type” de Elasticsearch. Esto no solo permite al usuario recibir sus resultados de búsqueda en menos tiempo, sino que también lo guía hacia los resultados que realmente existen en la aplicación.

6.3. Ordenamiento

La aplicación también presenta la funcionalidad de ordenamiento de los elementos. Para ello, sobre los resultados obtenidos en el filtrado, se dispone de un dropdown para el ordenamiento el cual solamente permite seleccionar un campo a la vez. Los diferentes tipos de criterios de ordenamiento son:

- **Orden alfabético:** se pueden ordenar los resultados por orden alfabético tanto de manera ascendente como descendente de acuerdo a su título.
- **Porcentaje de descuento:** se pueden ordenar los resultados por porcentaje de descuento ya sea de manera ascendente o descendente. En el caso de beneficios que no cuentan con porcentaje como lo son por ejemplo beneficios de 2x1, estos se listan al final de los resultados cuando se ordena por porcentaje de descuento.

7. Evaluación y resultados

A lo largo del proyecto se realizaron varias pruebas de funcionalidad y de coherencia de datos para verificar el correcto funcionamiento de la misma. Se cree necesario destacar la utilización de la herramienta Postman [9] para verificar la correctitud de los datos cargados en Elasticsearch. Para esto, luego de que algún dato era cargado, se procedía a formular varias consultas utilizando Postman. Luego, se procedía a analizar manualmente la respuesta obtenida comparando la misma con los datos extraídos mediante Scrapy. Todas las pruebas realizadas arrojaron un resultado positivo, y sirvieron para detectar y solucionar rápidamente diferentes errores.

Luego de realizadas todas las pruebas pertinentes, se puede afirmar que se cumplió exitosamente con el alcance del proyecto en el período de tiempo de la asignatura, logrando desarrollar una aplicación web que reúne beneficios de tarjetas de distintos sitios web en un único lugar, que además ofrece al usuario distintas opciones de búsqueda y filtrado con tiempos de respuestas razonables.

Como se explicó anteriormente, ninguna de las instituciones provee ningún tipo de API por lo que la información se recolectó utilizando técnicas de *scraping*. Esto resulta en mucha información que tiene formato de “texto libre”, lo cual se traduce en que ciertos atributos de los beneficios no pudieron ser extraídos y solo quedaron disponibles mediante la búsqueda de texto libre.

También debemos resaltar que si bien se lograron recolectar todos los beneficios de las instituciones que entraron en el alcance, quedaron fuera ciertos bancos que ofrecen tarjetas con beneficios tales como Itaú y Santander. Esto implica que la aplicación es un buen prototipo del cual partir, pero no ofrece toda la información requerida para ser puesta en producción.

Con respecto a los dos puntos anteriormente mencionados, se considera que tanto la incorporación de los beneficios de otras instituciones como de otra fuente de datos (APIs) son completamente compatibles con el producto desarrollado.

8. Conclusiones

Luego de realizar el presente proyecto podemos afirmar que adquirimos valioso conocimiento durante la realización del mismo, tanto de las herramientas utilizadas como de la metodología aplicada.

En cuanto a las herramientas utilizadas podemos destacar el potencial de las mismas y la gran facilidad de aprendizaje, ya que a pesar de contar con un margen de tiempo acotado, se logró incorporar el conocimiento necesario para cumplir con los objetivos planteados al comienzo. A modo de ejemplo, la utilización de Elasticsearch resultó una experiencia positiva que nos permitió conocer una tecnología distinta a las que estamos acostumbrados, tanto como motor de búsqueda como base de datos. Dicha herramienta provee servicios para realizar búsquedas y filtros de los datos almacenados previamente. Esto facilita la búsqueda de datos, en especial si lo comparamos con una búsqueda tradicional en una base de datos relacional. En general, se puede destacar la velocidad de búsqueda y la gran facilidad para escalar el proyecto que presenta Elasticsearch.

Otra observación realizada es que, investigando, se descubrió que el uso de las API es bastante sencillo de implementar cuando están bien documentadas. Lamentablemente, ninguno de los sitios considerados para el proyecto exponían una API, por lo que se tuvo que recurrir a *scraping*. Si bien esta técnica nos permitió cumplir con el objetivo, implicó un poco más de trabajo y personalización a cada sitio en particular (aunque esto último también hubiera aplicado a las API, pues difícilmente el formato de consultas/respuestas hubiera sido el mismo para las distintas instituciones).

Finalmente, y dado lo comentado anteriormente de las herramientas, se puede concluir que realizar este proyecto fue una experiencia enriquecedora ya que la información en la web crece de manera exponencial día a día y creemos que resulta de utilidad conocer y manejar una pequeña parte de las herramientas que contribuyen a recuperar la información en la web con calidad.

9. Trabajo futuro

Una posible mejora a futuro es la incorporación de usuarios y sesión. De esta forma sería posible registrarse en la aplicación y así se podrían implementar otras funcionalidades que aportan valor a la aplicación. Un ejemplo sería permitir que un usuario logueado pueda marcar cierto beneficio como favorito, de forma que al volver a iniciar sesión el usuario pueda identificar los beneficios que destacó anteriormente.

Una vez incorporada la funcionalidad anterior se puede empezar a manejar el concepto de "Popularidad". Esto se implementaría llevando un conteo sobre la cantidad de usuarios que marcan como favoritos los beneficios, siendo los populares aquellos que fueron marcados más veces. Esto también permite agregar una opción de ordenamiento por popularidad.

Siguiendo la misma línea también se puede aprovechar los beneficios marcados como favoritos para implementar algún sistema de recomendación que utilice algoritmos de aprendizaje automático para brindar recomendaciones basadas en la actividad de los usuarios con los cuales hay favoritos en común.

Por otro lado, anteriormente se mencionó que, debido a la utilización de *scraping* y el formato de los sitios web utilizados, ciertas características de los beneficios no quedaron modelados como atributos en Elasticsearch, sino que quedaron dentro de un atributo de texto expresado en lenguaje natural. Esto implica realizar búsquedas que no son tan eficientes como podrían ser si se tuvieran bien definidos dichos atributos.

Como posible mejora a esta situación proponemos utilizar técnicas de procesamiento de lenguaje natural para analizar las descripciones de los beneficios y poder extraer atributos en común de los beneficios para luego agruparlos según éstos. Esto resultaría en búsquedas más eficientes, lo que se traduce en dejar disponible más y mejores filtros para el usuario del sitio.

Por último, consideramos que un trabajo a futuro es adaptar la aplicación para que sea responsive, de forma de utilizarla en otros dispositivos como celulares y que sea accesible por un público más amplio.

Referencias

[1] Elasticsearch, https://www.elastic.co/es/

[2] Scrapy framework, https://scrapy.org/

[3] Angular 6, https://angular.io/

[4] Python, https://www.python.org/

[5] Beneficios OCA, https://www.oca.com.uy/institucional/promociones.aspx

[6] Beneficios Scotiabank, https://www.scotiabank.com.uy/Tarjetas/Promociones/default

[7] Beneficios BROU, https://beneficios.brou.com.uy/

[8] Selector Gadget, https://selectorgadget.com/

[9] Postman, https://www.getpostman.com/

[10] XPATH, https://doc.scrapy.org/en/xpath-tutorial/topics/xpath-tutorial.html