Recuperación de Información y Recomendaciones en la Web 2016

Grupo 11

Docente : Libertad Tansini

Integrantes:

Nicolas Fabre

Valentina Franchi

Índice

1.Introducción

Esta aplicación planea brindar una solución a la búsqueda y localización de supermercados en el Uruguay que cumplan con ciertos requisitos que desee el cliente. Estos requisitos pueden ir desde características comunes como la ubicación, cadena de supermercado y el horario hasta servicios más específicos como Banred, Red Brou, estacionamiento, entre otros.

Se eligieron las cadenas de supermercados Devoto, Disco, Kinko y Multiahorro por ser de las cadenas más grandes en el Uruguay y con mayor y más clara información en su página web. A pesar del hecho que dichas páginas web proveen información de cada supermercado y su localización, la idea es proveer toda la información de los mismos junta, donde también se consideró importante realizar un buscador brindando filtros avanzados sobre sus características.

2.Enfoque de la solución

El sistema de obtención de las distintas noticias fue desarrollado utilizando el lenguaje Python y el framework Scrapy. También se utilizó una Base de Datos no relacional (MongoDB), y para el frontend se utilizaron AngularJS , HTML5 y CSS.

La solución que se planteó fue obtener la información de los supermercados mediante web scraping, utilizando la herramienta scrapy, sobre las páginas de Devoto, Disco, Kinko y Multiahorro, para luego procesarla y poder mostrarla en la aplicación, así como también localizarla en un mapa. Los datos obtenidos para cada supermercado fueron:

- dirección,
- barrio,
- teléfono,
- horario,
- coordenadas,
- servicios brindados.

Al obtener esta información se decidió persistir la misma en una base de datos, debido a que los datos de los mismos son rígidos y no cambian

constantemente, reduciendo así el costo en tiempo que implica conectarse con otro servicio externo, así como en ancho de banda.

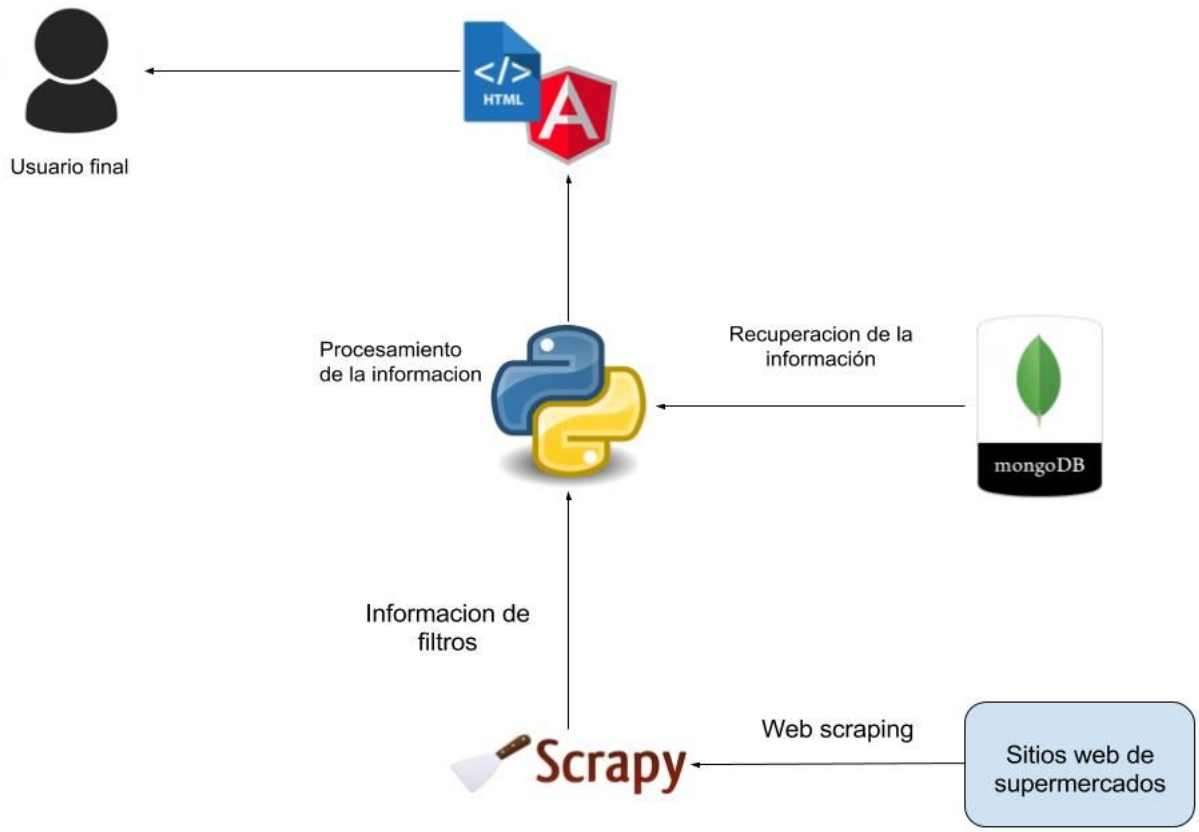
Una vez obtenida toda la información se procedió a procesarla para luego mostrarla en la aplicación, esto implicó entre otras cosas, obtener las coordenadas de algunos supermercados ya que en otros casos fueron obtenidas de los datos de la misma página.

Los filtros implementados fueron la búsqueda por barrio, cadena de supermercado, horario, si desea que tenga estacionamiento y todo tipo de servicios, permitiendo cualquier combinación entre ellos.

3.Diseño e implementación

El diseño del sistema corresponde a un proyecto Python, implementado con el framework Scrapy , el cual tiene como principales componentes a los Ítems y Spiders. Los Ítems son los contenedores de la información resultante de la extracción. Es un DSL mediante el cual se define la estructura de los datos a extraer. Los Spiders son clases, piezas de código y módulos escritos con el fin de descargar el código fuente de la URL objetivo, y recorrer la información del sitio web descargado, para luego depositarlo en los Ítems. Decidimos persistir los datos mencionados en una base de datos NoSQL, ya que persistimos en formato JSON, y las operaciones involucradas son get de un super (se requiere de la información completa del objeto) y get de todos los objetos super. Luego, la web accede a los JSON obtenidos como resultado del procesamiento de los datos obtenidos, para luego presentar la información en el mapa y como filtros, para hacerla visible al usuario.

3.1.Arquitectura del sistema



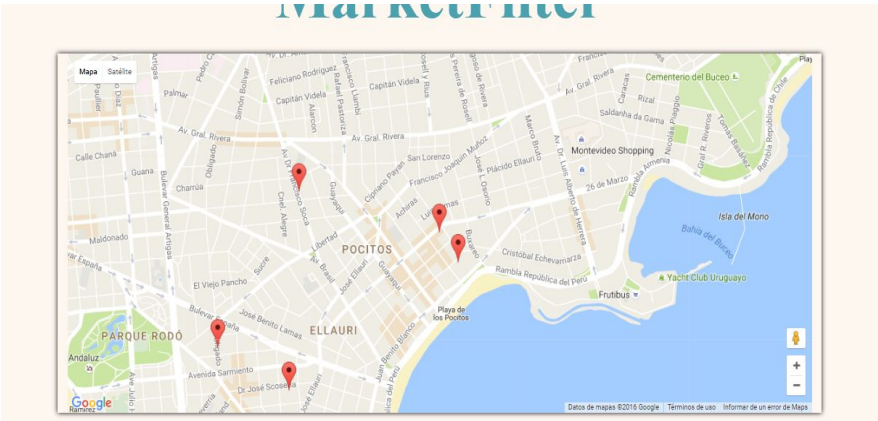
3.2 Tecnologías utilizadas

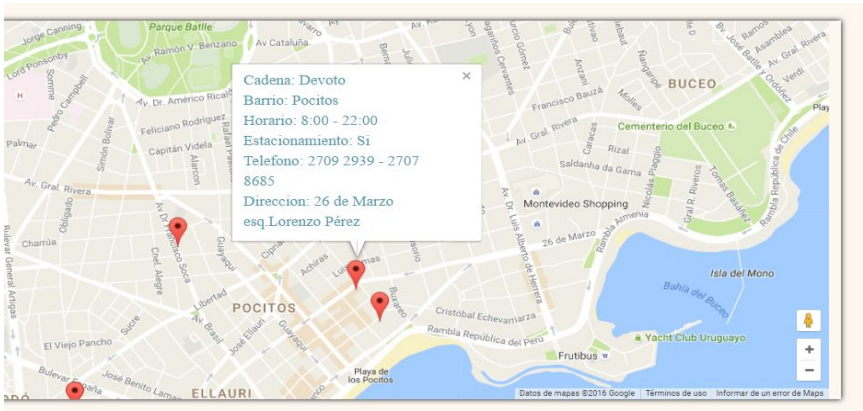
- Se utilizaron para el desarrollo de la lógica del programa el lenguaje python y la herramienta de Scrapy.
- Para la persistencia se utilizó mongoDB.
- En cuanto a Frontend se utilizaron las tecnologías HTML 5, CSS y AngularJS.

4.Pruebas

4.1 Prueba 1

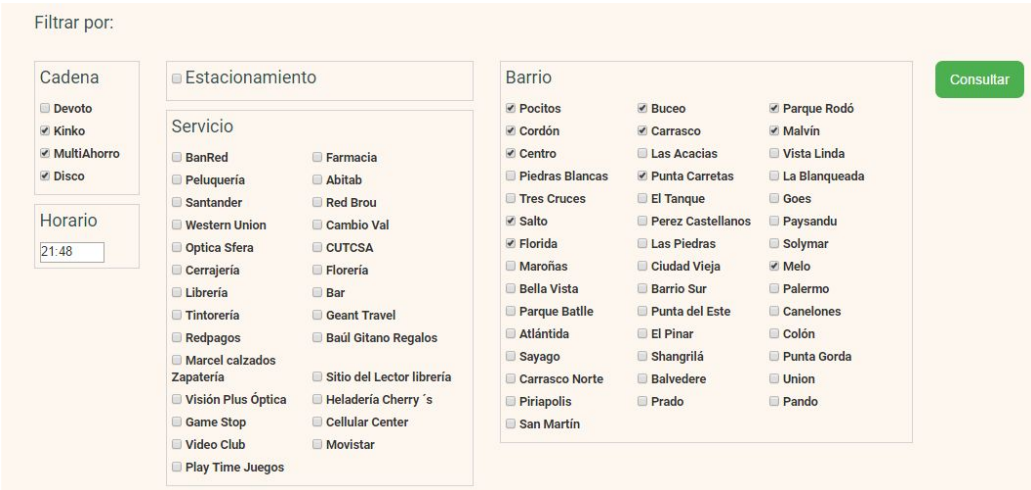
Se toma como precondición que exista la información de algún supermercado en la base de datos del sistema. Al seleccionar cualquier filtro ya sea de cadena de supermercado, horario, estacionamiento o servicio y seleccionando un solo barrio, el mapa se centra sobre el barrio seleccionado mostrando aquellos supermercados que cumplan además del barrio elegido, con el resto de los filtros.

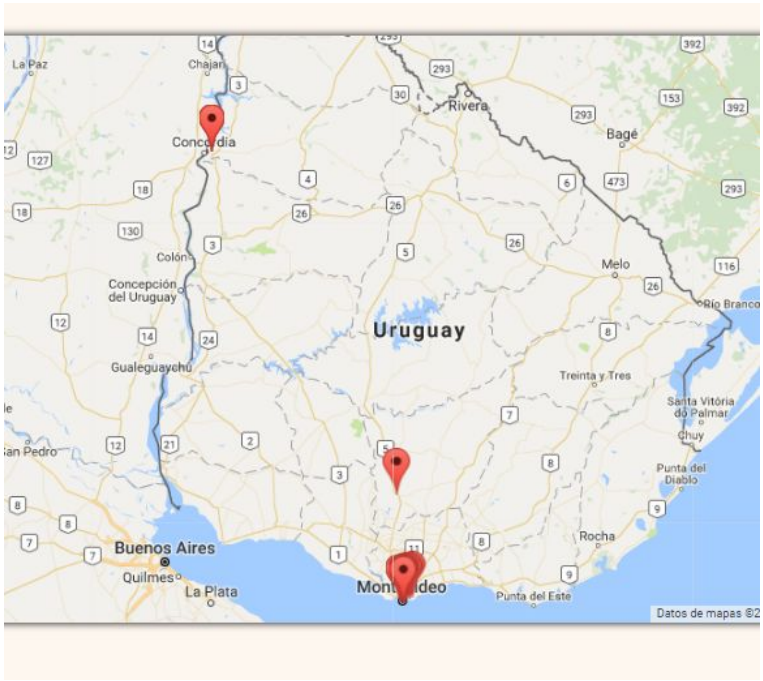


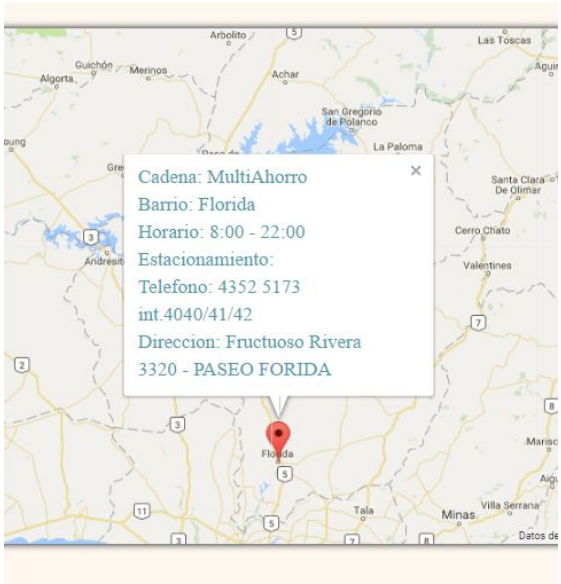


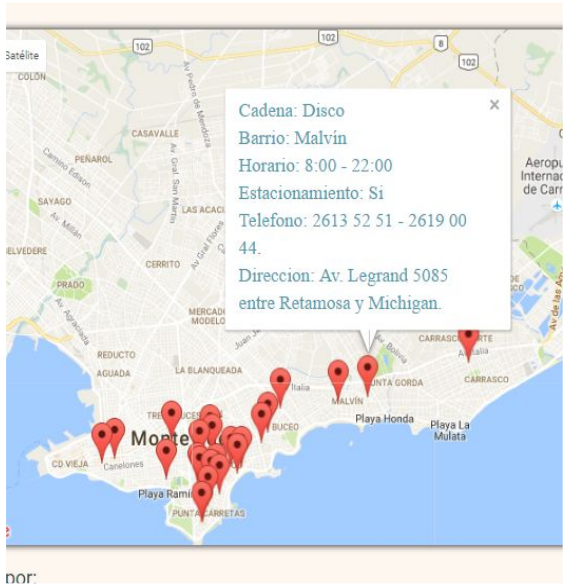
4.2 Prueba 2

Se toma como precondición que exista la información de algún supermercado en la base de datos del sistema. Al seleccionar varios filtros y hacer click en el botón “Consultar”, se muestran en el mapa solo los supermercados que cumplan con los mismos, siendo cualquiera esa combinación de filtros. A su vez clickeando sobre los markers en el mapa se podrá ver toda la información de ese supermercado, y de esta forma también se podrá comprobar que los mismos cumplan con los filtros seleccionados.




Uruguay
Buenos Aires
Quilmes
La Plata
Montevideo
Punta del Este
Rocha
Treinta y Tres
Melo
Rivera
Bagé
Concordia
Colón
Concepción del Uruguay
Gualeguaychú
Datos de mapas ©2


Cadena: MultiAhorro
Barrio: Florida
Horario: 8:00 - 22:00
Estacionamiento:
Telefono: 4352 5173
int.4040/41/42
Direccion: Fructuoso Rivera
3320 - PASEO FORIDA


Cadena: Disco
Barrio: Malvín
Horario: 8:00 - 22:00
Estacionamiento: Si
Telefono: 2613 52 51 - 2619 00 44.
Direccion: Av. Legrand 5085 entre Retamosa y Michigan.

por:

5.Dificultades

- Surgieron varios problemas al unificar datos de distintas fuentes debido a que la calidad de los datos contenidos no es la mejor y el formato no se mantiene en todos los casos. Se encontraron diferencias semánticas en las fuentes elegidas, por ejemplo en el caso de Devoto se maneja el concepto de estacionamiento como un servicio más, y por otro lado el Disco provee información adicional referenciada como Parking.

- También se encontraron muchos datos inconsistentes en la misma fuente así como direcciones mal ingresadas o abreviaciones de calles incorrectas, lo cual dificulto para la obtención de los datos para su Geolocalización.

- Se requiere de un gran trabajo de procesamiento de lenguaje natural para el desarrollo de la aplicación.

- El hecho de que no existan APIs predefinidas por parte de las cadenas importantes de supermercados, dificulta el objetivo de centralizar y normalizar la información. Dado que el resultado del web scraping depende del código HTML de la página web y este puede variar.

6.Trabajo a futuro

- Una de las características que puede mejorar la calidad del producto podría ser agregar más cadenas de supermercados, logrando centralizar la información sobre las mismas, para poder brindarle al usuario más opciones para poder elegir mejor.

- Por otro lado, a su vez como en el punto anterior se podría expandir la aplicación no solo a supermercados sino también a estaciones de servicio, farmacias, etc. La desventaja que podría tener esta seria que no todos tienen página web por lo que sería difícil la obtención de los datos.

- Otro punto que consideramos importante es crear una aplicación móvil, ya que permite el acceso inmediato a la información de manera sencilla. A su vez, otra característica de Geolocalización y filtrado que podría ser de interés para el usuario sería la sugerencia de supermercados que se encuentren cerca de él mediante la obtención de su ubicación.

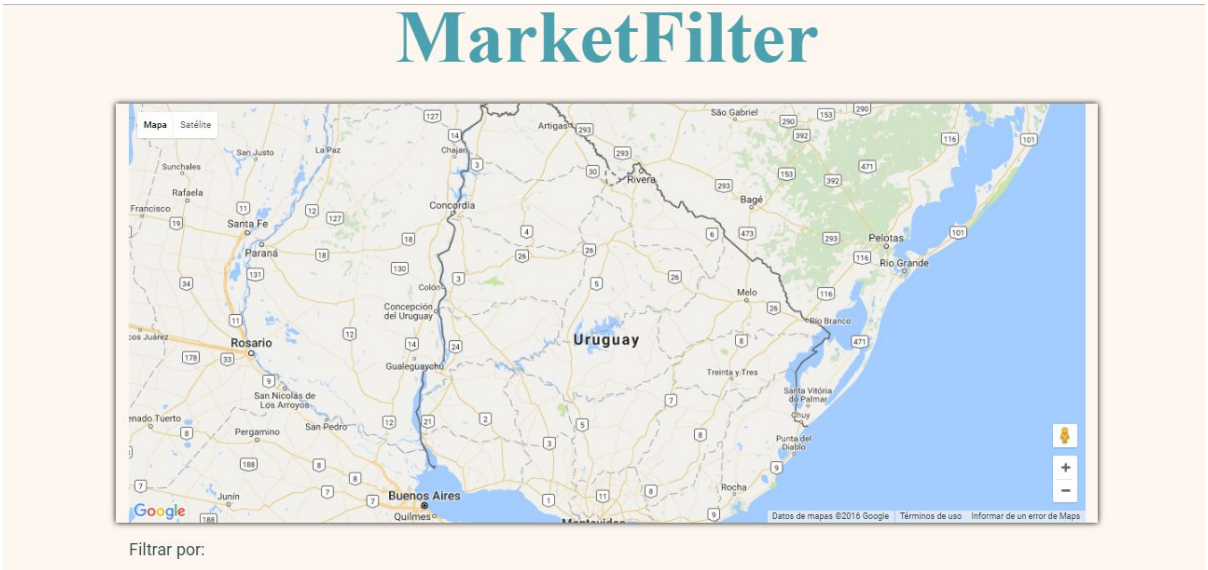
- También la posibilidad de que la web se actualice cada cierto periodo de tiempo,realizando nuevamente el scraping de la web y actualizando los datos de los supermercados, y no que se realice en parte de forma manual como se encuentra actualmente.

7. Conclusiones

Se desarrolló una aplicación web con el fin de brindar a los usuarios información personalizada de supermercados de interés. Ya que se ofrece la posibilidad de consultar por los locales que cumplan con las preferencias y necesidades deseadas.

El producto final es un servicio simple e intuitivo que cumple con los requerimientos iniciales propuestos por el equipo, combinando diferentes tecnologías y técnicas de recuperación, procesamiento y presentación de información de la web.

8. Manual de usuario



Cuando se inicia la aplicación, los datos ya están cargados en la base de datos y no es necesario realizar ningún procedimiento para cargar los mismos. Se mostrará en la pantalla el mapa sin cargar las ubicaciones aun, ya que las mismas son muchas y demoran en cargar. Una vez iniciado ya se podrán utilizar los filtros.

Los filtros posibles son los siguientes:

1. Por barrio
2. Por horario
3. Por cadena de supermercado
4. Si desea que tenga estacionamiento
5. Por servicios

(1) Se utiliza para filtrar los supermercados por el barrio en que se encuentran.
(2) Se filtra por horarios, el usuario ingresa una hora en que desea encontrar supermercados abiertos y se muestran los mismos en el mapa.
(3) Se utiliza para filtrar por la cadena de supermercado, como lo son Devoto, Disco, Multiahorro y kinko.
(4) Se utiliza para obtener aquellos supermercados que tienen estacionamiento.
(5) Se filtra por todo tipo de servicios, por ejemplo si desea que el mismo tenga farmacia, peluquería, abitab, BanRed, Red Brou y otros.

Los filtros no son excluyentes entre sí, es decir, se pueden combinar los mismos para generar una búsqueda más específica.

Una vez que se marquen los filtros deseados por el usuario, seleccionando el botón “Consultar” se mostrarán en el mapa aquellos supermercados que cumplan con aquellos que fueron seleccionados.

9.Referencias

[1] Scrapy 1.0 framework, scrapy.org.
[2] Base de datos No SQL, Mongo DB, www.mongodb.org.
[3] Angular JS, angularjs.org.
[4] Tutorial proyecto en Scrapy, http://doc.scrapy.org/en/latest/intro/tutorial.html.
[5] Items, http://doc.scrapy.org/en/latest/topics/items.html.
[6] Spiders, http://doc.scrapy.org/en/latest/topics/spiders.html.
[7] Pagina web de Devoto, http://www.devoto.com.uy/aindex.aspx
[8] Pagina web del Kinko, http://kinko.com.uy/
[9] Pagina web del Disco, http://www.disco.com.uy/
[10] Pagina web de Multiahorro, http://www.multiahorro.com.uy/