

Recuperacion de Informacion y Recomendaciones en la Web

ENTREGA FINAL INFORME OBLIGATORIO 2020

GRUPO 12

DIEGO OLIVEIRA 4.608.041-6

GUILLERMO EIJO 4.866.107-6

DOCENTE: LIBERTAD TANSINI



UNIVERSIDAD DE LA REPÚBLICA | FACULTAD DE INGENIERÍA

CONTENIDO

Introducción	2
Problema y Motivación	2
Enfoque de la solución	3
Diseño e implementación	4
Recuperación Información de la web	4
Elasticsearch	4
Frontend	4
Funcionalidades y uso	5
Evaluación y resultados	5
Conclusiones	5
Trabajo Futuro	5
Referencias	5

1. Introducción

En el presente documento se aborda la aplicación de las técnicas vistas en el curso “Recuperación de Información y Recomendaciones en la Web” del año 2020 aplicadas a un problema de interés general.

El informe describe una primera aproximación a la solución planteada por el grupo para el problema de la búsqueda y filtrado de automóviles a la venta en la plataforma mercado libre.

2. Problema y Motivación

En los últimos tiempos, cada vez más personas realizan la compra de un vehículo a través de diferentes plataformas virtuales. Esto es debido a la facilidad y comodidad de realizar la compra sin ir personalmente a distintos lugares de venta de automóviles para evaluar las posibles alternativas.

Sin embargo, no todo son beneficios para los usuarios. Esto se debe a que en la web o en las diferentes plataformas virtuales hay demasiada información. Esta información, además, puede resultar difícil de filtrar para acceder a la información de mayor interés para el usuario.

Por último, otro aspecto a tener en cuenta es la facilidad con la que un vendedor con malas intenciones puede hacerse pasar por un vendedor confiable.

Es por esto que entendemos que sería de utilidad la implementación de una aplicación web que pueda ofrecer a los usuarios, la solución a los problemas planteados, permitiendo realizar búsquedas de automóviles utilizando filtros para recuperar únicamente resultados de interés.

3. Enfoque de la solución

El enfoque que se plantea en esta solución, es presentar al usuario una aplicación mediante la cual pueda consultar distintos vehículos en base a una lista de filtros deseados y otros criterios de utilidad.

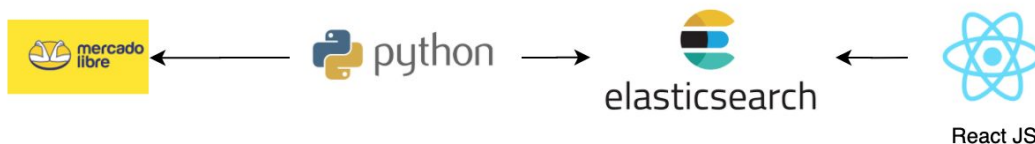
Existen infinidad de lugares en los cuales se publican vehículos para la venta. Como primera aproximación, nos enfocaremos en centralizar la información de un listado de publicaciones en la plataforma Mercado Libre.

El usuario contará con la posibilidad de buscar vehículos utilizando los siguientes criterios:

- Año.
- Marca.
- Modelo.
- Precio/Moneda
- Tipo de combustible.
- Cantidad de puertas.
- Kilometraje.
- Condición (Nuevo, Usado).

4. Diseño e implementación

El diseño de la aplicación se ha dividido en distintos componentes, cada uno de los cuales se encarga de un problema en particular. A continuación se presenta un esquema general de la solución y una breve descripción de cada uno de los componentes:



4.1. Recuperación Información de la web

Se desarrolla un módulo Python para recuperar información sobre vehículos de Mercado Libre haciendo uso de la API para desarrolladores[1] expuesta por Mercado Libre. Luego de obtenidos esos datos se guardarán en Elasticsearch de forma apropiada, en formato JSON, para su consumo desde un frontend.

4.2. Elasticsearch

Elasticsearch [2] es un motor de búsqueda distribuido, RESTful[3], open source para todo tipo de datos desde texto, numérico y hasta geoespacial. Permite búsquedas rápidas y escalables. En este caso lo usaremos para guardar los datos obtenidos de Mercado Libre para ser consumidos desde un frontend React JS[4]. Elasticsearch permite hacer búsquedas específicas con distintos parámetros y hasta corrección de errores en los campos.

4.3. Frontend

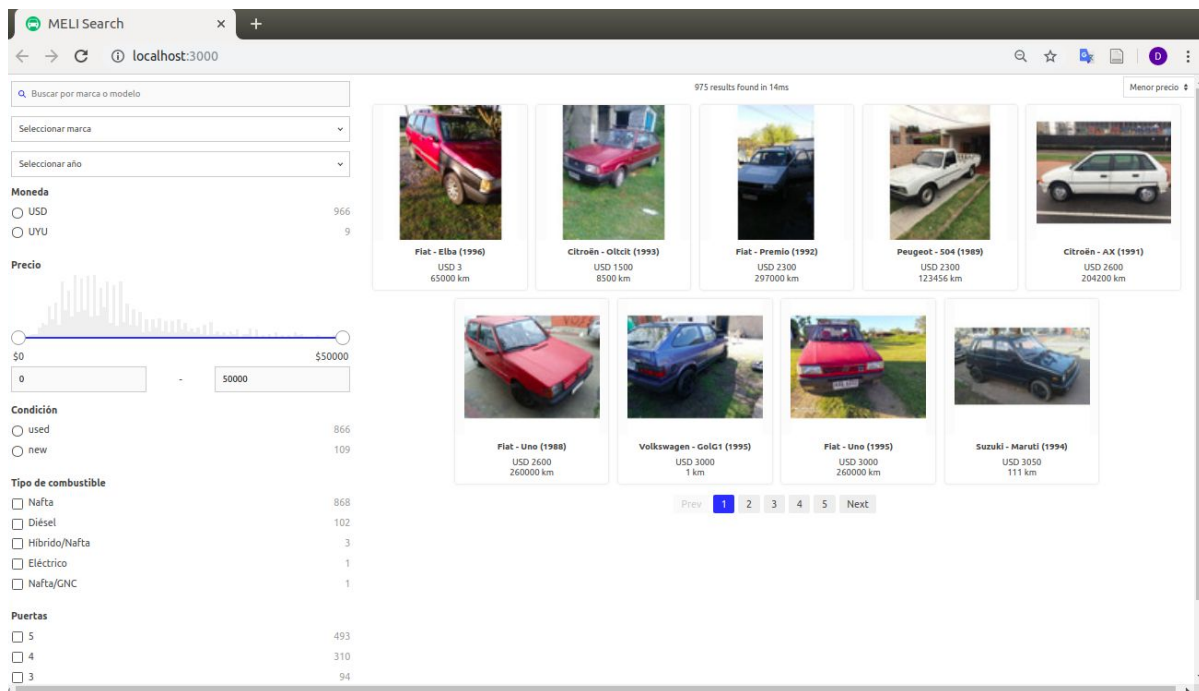
Contaremos con un frontend desarrollado usando React para la búsqueda y visualización de los resultados obtenidos. Esta aplicación permite realizar distintos tipos de búsqueda y se comunicará con nuestro servidor Elasticsearch utilizando consultas HTTP para obtener los datos de los vehículos previamente almacenados.

5. Funcionalidades y uso

El sistema desarrollado tiene dos componentes fundamentales. El primero es un módulo python que se encarga de recuperar la información de Mercado Libre y luego guardarla en el servidor ElasticSearch. Al tratarse de una demo, este módulo ejecuta una única vez dejando disponible la información para su consumo desde la aplicación web.

El segundo componente de nuestra solución es una aplicación web desarrollada en React JS. Esta aplicación recupera la información del servidor de ElasticSearch y brinda la posibilidad de filtrar y ordenar las búsquedas según distintos criterios que están disponibles en la web.

En la siguiente imagen se puede ver la pantalla de inicio de la aplicación web, la cual muestra los primeros resultados (sin filtrar). A su vez, a la izquierda se pueden observar los distintos filtros disponibles.



A partir de allí, el uso de esta aplicación es muy sencillo e intuitivo. Simplemente se debe ir ingresando la información deseada de acuerdo al filtro elegido y la aplicación realizará la búsqueda automáticamente.

Uno de los filtros posibles corresponde a la marca del vehículo que se desea buscar. Las diferentes marcas disponibles pueden seleccionarse de un combobox, el cual informa la cantidad de resultados que se obtendrán.

Una vez seleccionada la marca deseada, la vista se actualiza automáticamente para mostrar los resultados obtenidos como se muestra en la siguiente imagen.

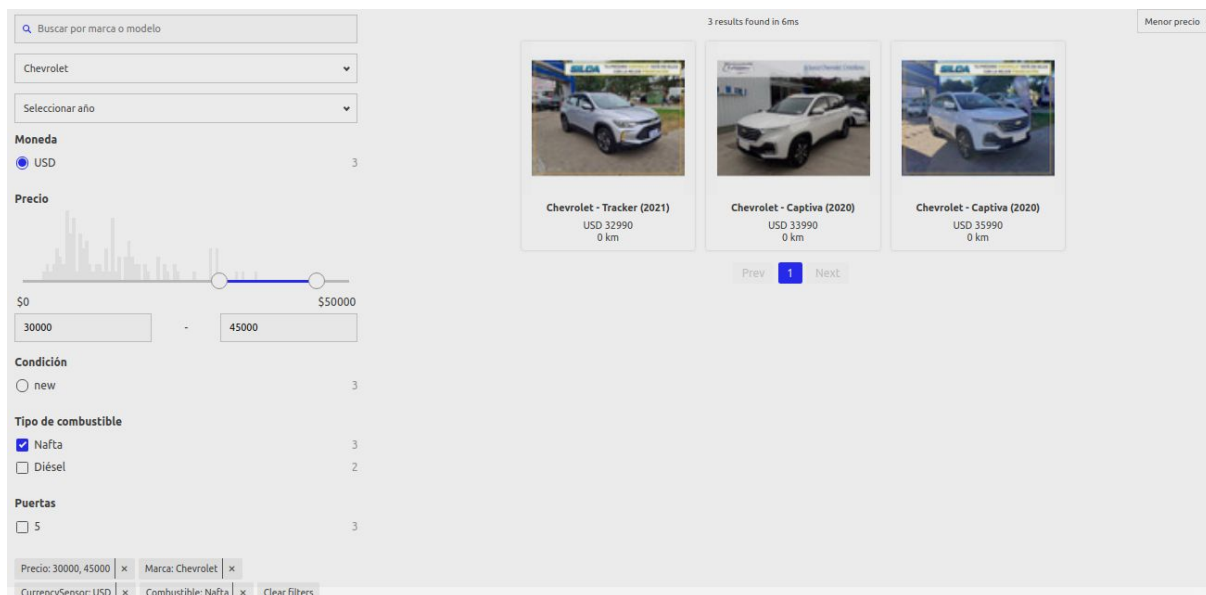
The screenshot displays a car search interface. On the left, there are several filter sections: a search bar with the text 'Buscar por marca o modelo', a dropdown menu for 'Chevrolet', a dropdown for 'Seleccionar año', a 'Moneda' section with 'USD' selected and a count of 129, a 'Precio' section with a price range slider from \$0 to \$50,000 and input boxes for '0' and '50000', a 'Condición' section with 'used' selected and a count of 100, and a 'Tipo de combustible' section with 'Nafta' selected and a count of 118. The main area shows '129 results found in 23ms' and a grid of six car listings. Each listing includes a photo, the car model and year, the price in USD, and the mileage in km. The listings are: Chevrolet - Corsa Classic (1999) for USD 3490 with 159000 km; Chevrolet - Corsa (1994) for USD 3990 with 140000 km; Chevrolet - Corsa Classic (2000) for USD 3990 with 150000 km; Chevrolet - Corsa (2000) for USD 4490 with 90000 km; Chevrolet - Corsa (1999) for USD 4500 with 170000 km; and Chevrolet - Aveo G3 (1999) for USD 4990 with 97000 km. A 'Menor precio' dropdown is visible in the top right corner.

Además de la marca, los filtros disponibles son:

- Modelo.
- Año del vehículo.
- Rango de precio y moneda.
- Condición (nuevo o usado).
- Tipo de combustible.
- Cantidad de puertas.

Todos los filtros disponibles se pueden utilizar en conjunto para obtener una búsqueda precisa de acuerdo a los intereses del usuario.

Un ejemplo de búsqueda avanzada podría ser la siguiente. Automóviles marca Chevrolet cuyo tipo de combustible sea nafta y su precio en dólares se encuentre entre \$30.000 y \$50.000. En este caso el resultado sería el siguiente:



The screenshot displays a web application interface for searching cars. On the left, there is a sidebar with various filters: a search bar, a brand dropdown set to 'Chevrolet', a year dropdown, a currency selector set to 'USD', a price range slider from \$0 to \$50,000 (with input boxes for 30,000 and 45,000), a condition selector set to 'new', a fuel type selector with 'Nafta' checked, and a door count selector set to '5'. On the right, three search results are shown as cards, each with a car image, title, price, and mileage. The results are: 'Chevrolet - Tracker (2021)' for USD 32,990 (0 km), 'Chevrolet - Captiva (2020)' for USD 33,990 (0 km), and 'Chevrolet - Captiva (2020)' for USD 35,990 (0 km). A 'Menor precio' button is visible in the top right corner. At the bottom of the sidebar, there are filter tags for 'Precio: 30000, 45000', 'Marca: Chevrolet', 'CurrencySensor: USD', and 'Combustible: Nafta', along with a 'Clear filters' button.

Por otro lado, la aplicación permite la posibilidad de ordenar los resultados según su precio en orden ascendente o descendente.

Finalmente una vez que el usuario selecciona un vehículo, haciendo click sobre él, se abre la página de la publicación elegida en MercadoLibre.

6. Evaluación y resultados

Al tratarse de una demo, la cantidad de información almacenada en el servidor ElasticSearch no es tanta como para poder realizar mediciones.

Sin embargo nuestra aplicación web muestra la información de cuántos resultados se obtuvieron para cada búsqueda y cuánto tiempo demoró en obtenerlos.

En la siguiente imagen vemos como una búsqueda que filtra autos usados cuyo tipo de combustible sea nafta devuelve 767 resultados en 11 milisegundos.

The screenshot displays a web application for searching cars. On the left, there is a sidebar with various filters: a search bar, dropdowns for brand and year, a currency selector (USD: 758, UYU: 9), a price range slider (0 to \$50,000), and checkboxes for condition (used: 767, new: 101) and fuel type (Gasoline: 767, Diesel: 96, Electric: 1, Hybrid: 1, Gas/GNC: 1). The main area shows 767 results found in 11ms, with a 'Menor precio' button. The results are presented in a grid of car images, each with its model, year, price, and mileage. The visible results include:

Modelo	Año	Precio	Kilómetros
Citroën - Oltcit	1993	USD 1500	8500 km
Fiat - Premio	1992	USD 2300	297000 km
Peugeot - 504	1989	USD 2300	123456 km
Citroën - AX	1991	USD 2600	204200 km
Fiat - Uno	1988	USD 2600	260000 km
Fiat - Uno	1995	USD 3000	260000 km
Suzuki - Maruti	1994	USD 3050	111 km
Volkswagen - Gol	1997	USD 3100	280000 km

A nuestro entender, la performance lograda por el sistema es muy buena.

7. Conclusiones

A nuestro entender, el hecho de centralizar la información (incluso pudiendo integrar nuevas fuentes de datos) y el poder personalizar distintos filtros de búsqueda hace que sea menos tediosa la tarea de buscar un vehículo en distintos sitios web.

Por otro lado, en nuestra opinión, la utilización de Elasticsearch permite tener una aplicación con buena performance en la consulta y procesamiento de grandes cantidades de información. Otro aspecto positivo del uso de Elasticsearch es la flexibilidad que brinda al momento de utilizar filtros en las consultas realizadas y la facilidad en su uso al trabajar con documentos JSON.

8. Trabajo Futuro

Una posibilidad de trabajo a futuro consiste en poder integrar otras fuentes de datos para permitir la visualización de más publicaciones de automóviles y así tener más opciones para que los usuarios puedan elegir.

Otro aspecto interesante podría ser automatizar el módulo en python para que ejecute regularmente, permitiendo así, almacenar casi en tiempo real las nuevas publicaciones realizadas.

Por último, una funcionalidad que podría mejorar la experiencia de uso sería poder realizar búsquedas y filtros que permitan seleccionar diferentes atributos sobre quienes publican los vehículos. Esto con el objetivo de generar confianza a aquellas personas que quieren comprar un vehículo a través de una aplicación web.

9. Referencias

- [1] Mercado Libre Developers API - <https://developers.mercadolibre.com/api-docs/>
- [2] Elasticsearch - <https://www.elastic.co/products/elasticsearch>
- [3] RESTful - https://en.wikipedia.org/wiki/Representational_state_transfer
- [4] React - <https://reactjs.org/>