

Recuperación de Información y Recomendaciones en la Web 2021

Entrega final Grupo 9

Integrantes:

NOMBRE	C.I.
Leandro Paz	4.719.214-7
Diego Pigola	4.716.310-8
Agustina Sierra	4.647.235-6
Sebastian Volti	5.175.914-7

Índice

Introducción	3
Motivación	3
Problema	4
Enfoque de la solución	4
Diseño, implementación y componentes del sistema	5
Frontend	6
Elasticsearch	6
Backend	6
Api Pública	8
Funcionalidades y uso	8
Conclusiones	10
Trabajo Futuro	11
Referencias	12

Introducción

En este documento se presenta un prototipo de sistema que busca dar a conocer a un usuario final información acerca de inmuebles.

Esta información es recuperada desde la web, más precisamente de un sitio que expone sus datos mediante una api pública.

En las siguientes secciones del documento se presentará la arquitectura del sistema con cada uno de sus componentes y las distintas funcionalidades con las cuales cuenta.

Por último se hará una evaluación de posibles trabajos futuros a partir del prototipo y una conclusión final de parte de los estudiantes.

Este trabajo nace bajo el marco del laboratorio de Recuperación de Información y Recomendaciones en la Web en su edición 2021.

Motivación

Es una realidad que hoy en día al buscar un inmueble así sea para alquilar o comprar uno cuenta con un determinado presupuesto para ese fin.

Muchas veces el objetivo de la búsqueda se basa en poder maximizar ciertas características de estos inmuebles. Al enfrentarse a esta situación, no es común encontrar una aplicación en la cual un usuario tenga la posibilidad de colocar el presupuesto estimado a utilizar, y en base al mismo seleccionar características de interés priorizadas tal como habitaciones, metros cuadrados o cantidad de baños, para luego devolver de forma ordenada inmuebles con estas características.

En la actualidad uno puede buscar casas en distintas plataformas filtrando por barrio, por calle, y también incluir algún rango de precios, pero no cuenta con la posibilidad de indicar el presupuesto con el que uno cuenta y en base a ello obtener las distintas propiedades resultantes.

Problema

Dada la situación antes planteada es que surge la necesidad de brindarle al usuario final una interfaz mediante la cual tenga la posibilidad de indicar el presupuesto con el que cuenta, y que en base a ello se realice una búsqueda de las distintas propiedades que cumplen con estas condiciones en una ciudad en particular, permitiendo el ordenamiento de las propiedades de acuerdo a los parámetros indicados por el usuario.

Enfoque de la solución

El sistema que se realizó a efectos de lo que verá el usuario final, es una aplicación web la cual se nutrirá de información previamente recuperada de la web por medio de otro componente.

En el prototipo mostrado en este documento, cada componente se encuentra en el mismo host físico, pero podría pensarse la solución en producción de distintas formas, dado que cada componente es independiente, ganando así performance y escalabilidad.

Diseño, implementación y componentes del sistema

El sistema final construido está formado por los siguientes componentes detallados a continuación:



Frontend

Este componente cumple la función de interfaz para el usuario final, es decir que es el encargado de presentar los datos de manera amigable para el usuario y le permite al mismo hacer uso de los distintos filtros con los cuales cuenta el sistema.

Fue desarrollado en su totalidad con React, conocido framework basado en Javascript, de código abierto y diseñado para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

Su objetivo es ser la interfaz que permite al usuario final interactuar con el sistema.

Elasticsearch

Elasticsearch es un motor de búsqueda y analítica de RESTful distribuido que se basa en Lucene el cual es capaz de abordar un número creciente de casos de uso.

En sí mismo se puede definir a Elasticsearch como un base de datos NoSQL orientada a documentos JSON, los cuales pueden ser consultados, creados, actualizados o borrados mediante API Rest.

Este componente primeramente es la base de datos en la cual se almacenaron los datos recabados de la web.

Elasticsearch fue elegido en segundo lugar por su gran capacidad de procesar datos y almacenarlos en una base de datos no relacional, que cuenta con un lenguaje propio para realizar búsquedas y permite exponer la información a través de una api rest.

En tercer lugar Elasticsearch cuenta con una gran variedad de funcionalidades a la hora de recuperar los datos guardados en índices, una de ellas es utilizada en este prototipo, la cual permite obtener los datos de forma ordenada según uno o varios criterios, permitiendo así trasladar la responsabilidad del ordenamiento de los datos al motor de búsqueda optimizando los tiempos en el backend.

Otra característica que se podría considerar de interés al momento de poner en producción este prototipo es su gran flexibilidad a la hora de alojar los datos, elasticsearch no solo cuenta con configuraciones master-slave para sus posibles nodos sino que da la posibilidad de duplicar índices de forma óptima logrando así tener siempre los datos “cerca” del usuario.

Se dividen los índices en shards y esto puede replicarse o distribuirse de la forma más adecuada para el sistema.

Backend

Componente del sistema encargado de consumir la api pública en la cual se exponen las propiedades y enviar esa información a la base de datos antes mencionada.

La información que es almacenada, es previamente procesada, es decir que se almacenan algunos de los datos recibidos de la api (los que creemos relevantes para la aplicación en desarrollo).

Por lo cual, se podría decir que el flujo de procesamiento que reciben estos datos es el siguiente:

- Backend consume api pública recibiendo un objeto con a lo sumo 50 propiedades.

- En base a los datos recibidos deduce si la propiedad está para la venta o alquiler.
- Genera nuevos objetos con los datos que consideramos relevantes almacenar.
- Envía la información a la base de datos.

En esta sección del sistema se desarrolló un script en Python el cual puede ser ejecutado de forma manual o configurando una tarea programada o un cron que ejecute este script diariamente en búsqueda de nuevos datos para popular de manera continua la base de datos.

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Los datos almacenados tienen el siguiente formato:

```
{
  "title": "title",
  "baths": 3,
  "mts": 75,
  "price": 12000,
  "currency": "currency_id",
  "city": "city",
  "neighbourhood": "neighbourhood",
  "mode": "ALQ",
  "rooms": 5,
  "link": "permalink"
}
```

Explicación de los distintos elementos del json almacenado:

- title: String en el cual se almacena el título de la publicación.
- baths: Campo numérico para almacenar cantidad de baños.
- mts: Campo numérico para almacenar mts² de la propiedad.
- price: Campo numérico para almacenar el costo en \$U.
- currency: String con la moneda.
- city: String para almacenar la ciudad en la cual se ubica la propiedad.
- neighbourhood: String para almacenar el barrio en el cual se ubica la propiedad.
- mode: String para identificar si la propiedad se alquila o se vende.
- rooms: Campo numérico que almacena la cantidad de cuartos de la propiedad.
- link: String con un enlace a la publicación.

Todas estas propiedades son almacenadas en elasticsearch indexando por modo-rango de precio. Es decir, se trunca el costo de la propiedad para generar franjas de precios.

Si por ejemplo tenemos una propiedad para alquilar con un costo de 16500, esta será almacenada en el índice **/alq-16000**, aunque en el campo price se mantiene su verdadero precio de 16500.

Api Pública

Api pública en la cual están expuestas las interfaces para nutrir de información al sistema.

En este caso utilizaremos la API pública de Mercado Libre <https://api.mercadolibre.com> .

En la misma podemos obtener datos acerca de múltiples viviendas en distintos barrios de cualquier ciudad indicada, sus características y precios.

Aquí presentamos un ejemplo de la ciudad de Montevideo :

<https://api.mercadolibre.com/sites/MLU/search?category=MLU1459&q=apartamento&city=TUxVQ1BBUmU3Y2Nj>

En el mismo se recupera información acerca de un apartamento ubicado en Parque Rodó. Elegimos este sitio por su popularidad y gran cantidad de datos que contiene, así como también cuenta con varios filtros que serán de utilidad a la hora de desarrollar nuestro trabajo.

Funcionalidades y uso

En esta sección se presenta un flujo normal dentro del sistema con sus distintas funcionalidades.

Inicialmente observamos la siguiente vista:

Prototipo proyecto webir

Búsqueda de inmueble

Presupuesto Moneda Tipo de Operación

Ordenar por:

Precio	#Baños	Metraje	#Habitaciones	Link
1-1 of 1 < >				

Aquí se puede colocar inicialmente el presupuesto, filtrando por el tipo de moneda (pesos uruguayos o dólares) y el tipo de operación (venta o alquiler).

Luego se puede seleccionar en orden las características que interesa priorizar (maximizar), que pueden ser cantidad de baños, habitaciones o metrajes, para luego confirmar la búsqueda.

Aquí tus resultados! ✕

Búsqueda de inmueble
 Presupuesto
 25000

Moneda
 \$

Tipo de Operación
 Alquiler

Ordenar por:

- Cantidad de baños
- Cantidad de habitaciones
- Metraje

CONFIRMAR

Precio	#Baños	Metraje	#Habitaciones	Link
24000\$	2	77mts	3	https://apartamento.mercadolibre.com.uy/MLU-604138098-apartamento-alquiler-cordon-montevideo-imasuy-fref-ims-8143-_JM
24000\$	1	60mts	4	https://casa.mercadolibre.com.uy/MLU-604021921-alquiler-casa-1-dor-2-patios-la-blanqueada-montevideo-e-_JM
24000\$	1	50mts	2	https://apartamento.mercadolibre.com.uy/MLU-604150903-alquiler-apartamento-parque-battle-con-balcon-imasuy-n-ref-ims-8075-_JM

1-3 of 4 < >

Aquí tus resultados! ✕

Búsqueda de inmueble
 Presupuesto
 25000

Moneda
 \$

Tipo de Operación
 Alquiler

Ordenar por:

- Cantidad de habitaciones
- Metraje
- Cantidad de baños

CONFIRMAR

Precio	#Baños	Metraje	#Habitaciones	Link
24000\$	1	60mts	4	https://casa.mercadolibre.com.uy/MLU-604021921-alquiler-casa-1-dor-2-patios-la-blanqueada-montevideo-e-_JM
24000\$	2	77mts	3	https://apartamento.mercadolibre.com.uy/MLU-604138098-apartamento-alquiler-cordon-montevideo-imasuy-fref-ims-8143-_JM
24000\$	1	50mts	2	https://apartamento.mercadolibre.com.uy/MLU-604150903-alquiler-apartamento-parque-battle-con-balcon-imasuy-n-ref-ims-8075-_JM

1-3 of 4 < >

Los filtros se pueden elegir en el orden de prioridades que se desea, y además no es necesario seleccionarlos todos.

Búsqueda de inmueble

Presupuesto
25000

Moneda
\$

Tipo de Operación
Alquiler

Ordenar por:

Cantidad de baños

CANTIDAD DE HABITACIONES

METRAJE

CONFIRMAR

Precio	#Baños	Metraje	#Habitaciones	Link
24000\$	2	77mts	3	https://apartamento.mercadolibre.com.uy/MLU-604138098-apartamento-alquiler-cordon-montevideo-imasuy-f-ref-ims-8143_JM
24000\$	1	50mts	2	https://apartamento.mercadolibre.com.uy/MLU-604150903-alquiler-apartamento-parque-battle-con-balcon-imasuy-n-ref-ims-8075_JM
24000\$	1	50mts	2	https://apartamento.mercadolibre.com.uy/MLU-604099364-alquiler-apartamento-parque-battle-montevideo-imasuy-n-ref-ims-7662_JM

1-3 of 4 < >

La búsqueda para Ventas es análoga.

Conclusiones

Desde el punto de vista educativo este proyecto fue de gran utilidad para afianzar los contenidos impartidos en el curso.

El reto de haber elegido elasticsearch como servidor de búsqueda en lugar de utilizar una base de datos relacional, e implementar una api para exponer los datos fue sumamente enriquecedor para los objetivos del grupo.

Esta fue una gran decisión por parte del equipo, ya que no contábamos con conocimiento de la herramienta. Aun así nos fue de gran utilidad pudiendo obtener un producto de calidad habiendo adquirido conocimiento de cómo funciona la herramienta y habilidades para poder utilizarla de aquí en más.

El gran potencial que encontramos, fue la posibilidad de indexar todos los datos mediante una lógica bastante sencilla en el backend, para que esta información fuese guardada y posteriormente expuesta por elasticsearch. Una vez finalizado esto, el mismo elasticsearch se encarga de exponer esa información de forma sencilla y con un sin fin de posibilidades tanto de ordenamiento como de formato en el cual es devuelta la información a la interfaz web.

En contraparte a lo recién planteado, nos quedó en el debe tanto el hecho de recuperar información de otros sitios en los cuales se expongan propiedades para la venta/alquiler, de forma de tener varias fuentes que nutren la base de datos del sistema. Como el hecho de que al ejecutar todo de forma local, no logramos ejecutar varias instancias de elasticsearch para sacarle el verdadero provecho que se le puede sacar a esta herramienta.

Partiendo de la premisa que es un prototipo estamos más que conformes con el producto final obtenido.

Aun así nos hubiese gustado mejorar varias de las partes que componen al sistema para mejorar el todo y tal vez lograr un trabajo de mayor calidad el cual pueda ser utilizado por personas en todo el mundo.

Trabajo Futuro

Es bien sabido que el sistema planteado es un prototipo funcional el cual se encarga de recuperar información de Mercado Libre, información que se encuentra expuesta mediante una api pública.

Luego esta información es procesada, indexada y almacenada para ser presentada al usuario final de una forma más amena y útil.

Este prototipo tiene un gran potencial como herramienta a ser desarrollada y potenciada ya que pone el foco en los usuarios finales y sus necesidades, sin contar de que no existen otras herramientas las cuales pongan el foco de la búsqueda de propiedades para alquiler en el presupuesto del usuario y las prioridades del mismo.

En caso de ser tomado como base se puede potenciar esta herramienta y exprimirla o bien podría ser un excelente módulo de expansión para herramientas que existen actualmente.

Dentro de las posibilidades de crecimiento futuro sería de suma utilidad agregar más filtros, como pueden ser por ejemplo la selección de barrios, aumentar la cantidad de lugares en los cuales se realiza la búsqueda (en este caso solo se recabo información de propiedades en una ciudad determinada), mejorar la interfaz con el usuario o incluso yendo más a requerimientos no funcionales el hecho de hacer que este producto escale tanto vertical como horizontalmente.

Referencias

ElasticSearch

<https://www.elastic.co>

API Mercado Libre :

https://developers.mercadolibre.com.uy/es_ar/categorias-y-atributos-inmuebles

Python:

<https://es.wikipedia.org/wiki/Python>

React:

<https://es.wikipedia.org/wiki/React>