

Entrega 2 - Recuperación de Información y Recomendaciones en la Web 2022

Lucas Fontes - 5.395.153-3
Marcelo Marenales - 5.048.023-4
Bruno De Simone - 4.914.555-0
Grupo 10.

25 de noviembre de 2022

Resumen

Segunda entrega de Recuperación de Información y Recomendaciones en la Web donde se expone en trabajo realizado por el equipo extrayendo información de Inmuebles publicados en MercadoLibre para brindarla de forma procesada a los usuarios de la misma con objetivo de analizar históricos de precios y comparar con Inmuebles similares.

Índice

1. Introducción	3
2. Problema	3
3. Enfoque de la solución	3
4. Diseño	3
4.1. Frontend	4
4.2. Backend	4
4.3. API Mercadolibre	4
5. Implementación	5
5.1. Frontend	5
5.2. Backend	6
5.3. Extracción de Datos	8
6. Funcionalidades y uso	9
6.1. PopUp	9
6.2. Historial de precio y Filtrado	10
7. Resultado	11
8. Trabajo Futuro	12
Referencias	12

1. Introducción

Este informe corresponde a la materia Recuperación de Información y Recomendaciones en la Web”, en el se presenta un proyecto enfocado en obtener, procesar y brindar a usuarios información no fácilmente obtenible con el fin de obtener valor agregado de información libre pero no fácilmente accesible en el internet.

2. Problema

Comprar o alquilar una vivienda es una decisión muy importante en la vida de cualquier persona, ya que representa muchas veces un compromiso de varios años e implica un gran porcentaje de su economía personal. Por eso se desea construir una herramienta que facilite el análisis de precios según características de la vivienda, brindado también información sobre el estado del mercado. Se quiere que los usuarios cuando tomen una decisión estén seguros y confiados de la misma a partir de la información objetiva que les aportamos.

El problema hace referencia a la necesidad de los usuarios de conseguir un mejor precio a la hora de comprar o alquilar un inmueble en Mercado Libre, teniendo la posibilidad de acceder a estadísticas que le ayudarán a tomar una mejor decisión. La aplicación tendrá la capacidad de utilizar filtros con el objetivo de buscar inmuebles según lo que el usuario desee y tener gráficas para comparar con el promedio de precios de publicaciones con características similares, así como suscribirse a las publicaciones que desee para poder tener un seguimiento de las mismas. De esta forma el usuario podrá hacer un análisis rápido y de manera sencilla y así elegir el mejor producto que se adecúe a sus necesidades.

Nuestra mayor inspiración fueron publicaciones en redes sociales de varios usuarios repriminando a Mercadolibre la publicidad engañosa de campanas como la de “black friday” donde había casos en los que subían los precios de los productos y agregaban una etiqueta con mensajes como “50 % off” y mintiendo a los consumidores.

3. Enfoque de la solución

La solución propuesta es una extensión del navegador “Chrome” la cual brinda 2 funcionalidades sobre “mercadolibre.com.uy” la web mas usada en cuanto a compraventa de bienes:

- Rápido acceso a las publicaciones de inmuebles que se tienen guardadas, con breve información de cada una. Esto en forma de un PopUp al hacer click en la extensión.
- Panel de filtro e historial de precio dentro de cada publicación de inmuebles.

4. Diseño

La implementación esta claramente separa en 2 secciones, **frontend** que corresponde con todo los visual para el usuario y **backend** encargado de obtener datos de publicaciones y brindarle información al frontend.

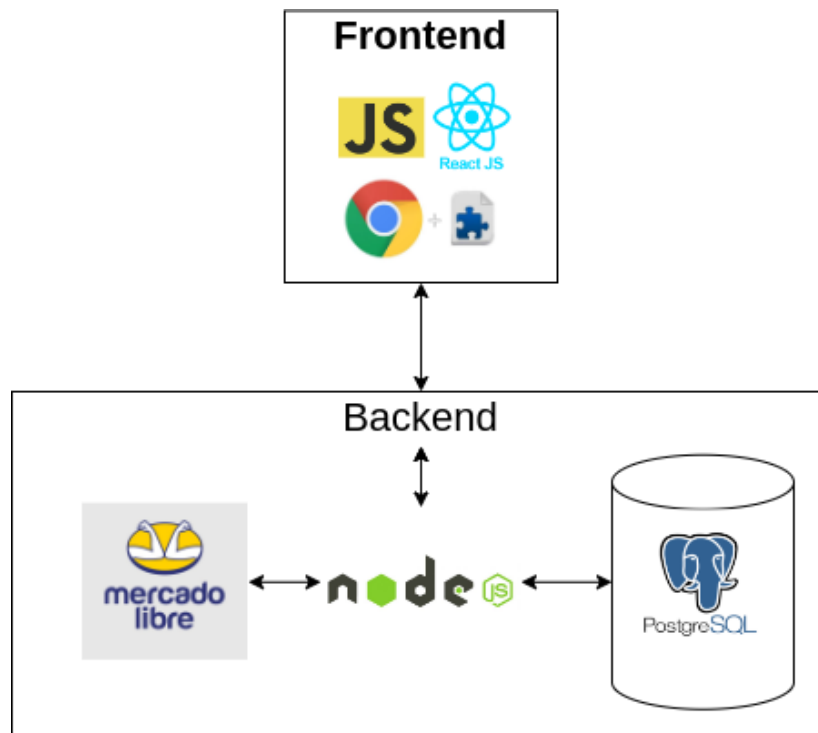


Figura 1: Diseño.

4.1. Frontend

El frontend encapsula, el contenido visual desarrollado en ReactJS [1] como lo son el popup y la visualización y filtro de precios de publicaciones. Javascript para modificar el dominio de ciertas paginas de mercadolibre.com.uy, para detectar acciones de usuarios como agregar publicaciones a favoritos. Debido a que la extensión corren dentro del runtime de Chrome las mismas no tienen acceso a objetos que cualquier web tendría, para solucionar esto Chrome provee una API [2] para almacenar información y comunicarse con las paginas webs que se están visitando.

4.2. Backend

El backend encapsula, un servidor desarrollado en NodeJS [3], el mismo se comunica con la API de MercadoLibre, procesa la información y almacena en una Base de Datos PostgreSQL.

4.3. API MercadoLibre

La API de mercadolibre [4] es una que se brinda principalmente para el uso de vendedores, facilita por ejemplo la publicación y obtención de publicaciones para un usuario (debe ingresar las credenciales de el mismo para obtener esta información). También ofrece, aunque con ciertas limitaciones, la obtención de las publicaciones generales de algunas categorías , por

eso, se detalla mas adelante. Se utiliza un endpoint que permite obtener publicaciones por ubicación geográfica.

5. Implementación

5.1. Frontend

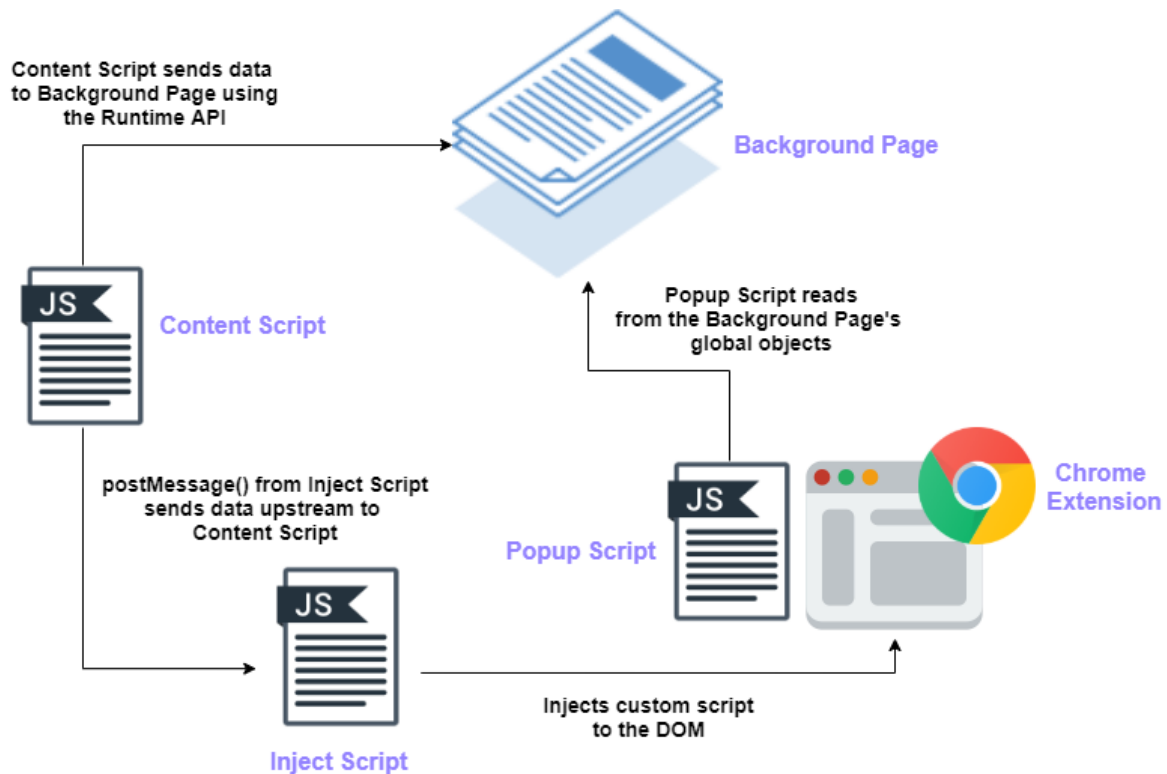


Figura 2: Frontend.

Para la configuración de extensiones de Chrome, el navegador ofrece ejecutar código de varias formas, en las cuales no se va a profundizar, en este caso se utilizan 3.

- **background.js** es un archivo que oficia de serviceworker. Él tiene acceso a la API de chrome, la pagina web no lo tiene, y además puede ser llamado desde la web a través de una cola de mensajes.
- **content.js** es el archivo que se encarga de realizar la inyección de Javascript/Html al dominio de la Web que se está visitando. El es utilizado para inyectar el panel de historial de precio y además para rastrear las acciones que realiza el usuario mediante EventListeners.

- **popup.js** como su nombre lo indica representa el código que se ejecuta al abrir el PopUp de la extensión y que en nuestro caso muestra los inmuebles guardados por el usuario.

El flujo de información que se utiliza es el siguiente, el content encargado de inyectar código, también es el que realiza los pedidos HTTP al backend solicitando la información necesaria sobre inmuebles. Además al detectar acciones que son de interés para la aplicación como agregar a favoritos un inmueble, el content.js despachara un mensaje al background.js avisando de tal acción y el background.js guardará esa información en un lugar (API Chrome) donde luego el PopUp pueda acceder a la misma.

Al abrir el PopUp se obtendrán de la API de chrome, los IDs de las publicaciones guardadas y se consultará directamente a la API de mercadolibre por la información de las mismas.

5.2. Backend

La estructura de Backend, la cual utiliza la tecnología NodeJS como se menciona anteriormente, está constituida por 2 'endpoints' y varios jobs encargados de obtener y mantener la información actualizada:

Ambos endpoints brindan información únicamente, no permiten actualizar/insertar información en la base de datos.

Uno de los endpoints brinda 3 tipos de información: - Información sobre el promedio general de precios del mercado uruguayo de un tipo de inmueble que está incluido en el conjunto: Casa, Alquiler, y que tiene una característica incluida en el conjunto: Alquiler, Venta.

- Información sobre el promedio de todos los inmuebles que estén en Venta/Alquiler, sin importar que sean Casas o Apartamentos. Este promedio de datos es consumido por el servidor de frontend para mostrar la información relacionada a un determinado inmueble que el usuario esté viendo en la aplicación de mercado libre.

- Información sobre el promedio del inmueble que el usuario está viendo actualmente en la aplicación de mercado libre en caso de que el mismo esté almacenado en la base de datos del servidor de backend.

Cabe destacar que la información que brinda este endpoint es dentro de un lapso de tiempo, las posibilidades son: Promedios en las últimas 2 semanas, o los últimos 6 meses.

El otro endpoint brinda información sobre el promedio de inmuebles en un lapso de tiempo y que a su vez cumplan con las condiciones de filtrado adjuntas a continuación:

Background Jobs: El backend cuenta con los siguientes jobs que se ejecutan diariamente o semanalmente (Cron jobs):

- Obtener información de inmuebles de la API de mercado libre: Este job está descrito a continuación ya que se explica a mayor detalle su implementación. (Ejecutado 1 vez al día para agregar nuevos inmuebles/ actualizar inmuebles existentes). Para cada inmueble se guardan los datos necesarios para el filtrado del mismo según sus propiedades, su identificación adentro de la base de datos de mercado libre, el tipo y si está publicado a la venta o alquiler.



Figura 3: Frontend.

- Crear una instancia de 'precios' a partir de inmuebles existentes en la base de datos: Diariamente, una vez que se actualizan los inmuebles, se almacena un precio para la fecha del día actual de ese inmueble. Estos precios están clasificados como 'Precio Diario' Y se usan para tener un historial de precios.
- Crear semanalmente una instancia de precios, los cuales tienen un tipo 'Precio Semanal' a partir de los precios diarios existentes para esa semana: Una vez a la semana, se ejecuta este job. (Esto es una optimización, ya que cuando el usuario pide el promedio de precios los últimos 6 meses, no se realiza el promedio sobre los precios diarios, si no que semanales).
- Crear diariamente un objeto (record) que contenga el promedio de precios de los siguientes inmuebles con las siguientes características: Casas en Venta, Casas en Alquiler, Apartamentos en venta, Apartamentos en Alquiler. (Esto se podría realizar por medio de una query cuando el servidor de frontend pide los datos al backend, pero dada la cantidad de registros que se guardan, y todo el proceso que hay que hacer para obtenerlos, se decidió que iba a estar guardada esa información en la base de datos diariamente).
- Crear semanalmente una objeto (record) que contenga la misma información anterior para un inmueble, pero difiriendo en que en vez de guardar el promedio de precios del mercado para un día, guarda para una semana. Esto se utiliza para el promedio de

precios del mercado los últimos 6 meses.

- Por último, para limpiar información que no será utilizada, se cuenta con jobs de limpieza de datos, que se ejecutan a diario y semanalmente (depende que se desee limpiar).

5.3. Extracción de Datos

Mercadolibre permite buscar según varios filtros, pero no permite extraer más de 4000 publicaciones para cada respuesta.

Teniendo en cuenta que existen aproximadamente 160.000 publicaciones de inmuebles de las categorías que nos interesan, primero se planteó realizar pedidos por departamento y en caso de Montevideo y Maldonado donde había mucha mayor cantidad de inmuebles dividir por Barrios. El resultado fue que habían muchas publicaciones que quedaban por fuera y que además no garantizaba traer todas las publicaciones ya que habían varios barrios con más de 4000 publicaciones.

La solución encontrada a esto fue el un endpoint que permite obtener todas las publicaciones que se encuentran dentro de 2 pares de latitudes y longitudes. Para ello diseñamos un algoritmo que empieza pidiendo por todas las publicaciones que se encuentren dentro de Uruguay y en caso de que haya más de 4000 divide el cuadrado en 4 cuadrantes, esto se hace recursivamente para cada cuadrado hasta llegar a aquellos que tengan menos de 4000 inmuebles. De esta forma logramos obtener todos los inmuebles.

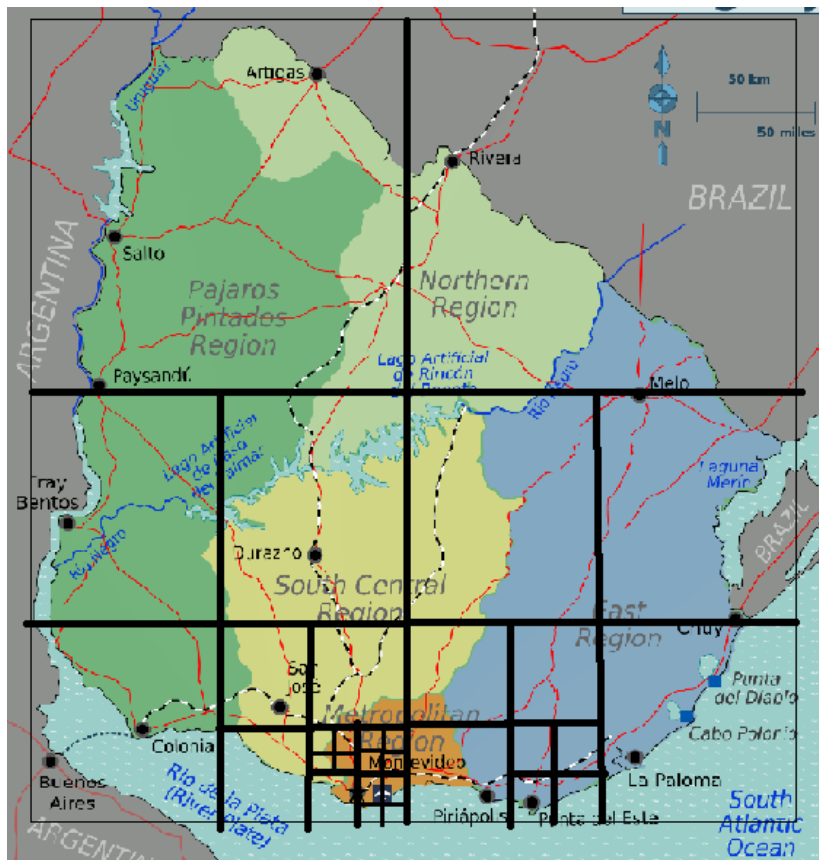


Figura 4: Ejemplo de resultado de algoritmo.

6. Funcionalidades y uso

6.1. PopUp

Es donde se muestra la información de los inmuebles que se tienen guardados, se muestra el título, tipo de publicación, ubicación, precio, duración de la publicación (ya que pueden vencer) y una foto de portada de la publicación.



Figura 5: PopUp.

6.2. Historial de precio y Filtrado

Se provee un panel dentro de cada publicación con el histórico de precios que se tiene guardado en la base de datos, además se ofrecen datos sobre el histórico de mercado para el tipo de transacción (venta/alquiler) para dar contexto de la tendencia del mercado y precios para el tipo de inmueble y tipo de transacción que es para dar información mas precisa.

Además se precargan los filtros con los valores de la publicación que se está visitando para darle la opción al usuario de ver los precios de publicaciones de similar característica. Esos filtros son modificables y puede buscar lo que le apetezca.

Se pueden visualizar los valores diarios o semanales de la publicación, muy similar a la visualización de los valores de las acciones en google.

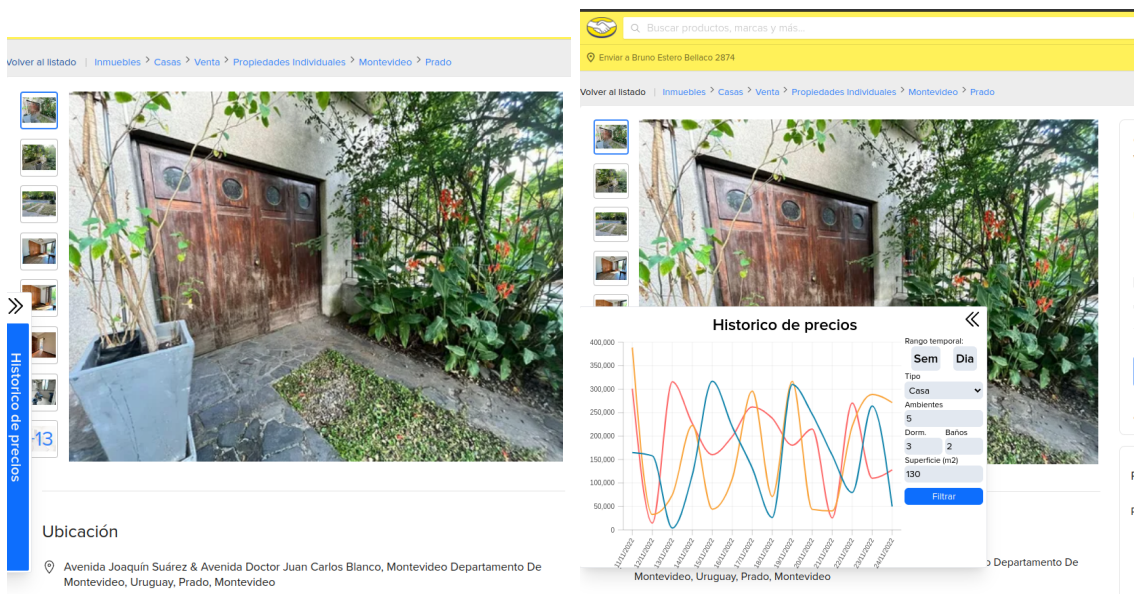


Figura 6: Filtro

7. Resultado

Se logró el objetivo principal del proyecto que consistió en obtener la información de Mercadolibre, procesarla, almacenarla y brindársela a un usuario de forma darle información que no tiene a simple vista, de calidad y útil.

Se logró obtener toda la información que se quería de Mercadolibre, superando documentación no muy clara y utilizando la API enfocada a los vendedores para beneficio de los compradores.

Consideramos que la aplicación que construimos tiene mucho potencial y que nuestra implementación es básica con respecto al abanico de posibilidades que ofrece la información extraída. Evidentemente, debido a que la característica principal de la aplicación es el manejo de historial de precio, con el paso del tiempo la información que obtenemos adquiere valor ya que permite realizar análisis con mayor profundidad pero además es información que no se puede obtener en otro lugar ya que mercadolibre no ofrece la posibilidad de ver precios pasados.

Además del procesado de información, se comenzó a realizar manejo de autenticación pensando la posibilidad de brindarle información mas personalizada al usuario y no depender de Chrome para almacenar la información del mismo, sin embargo no se logro terminar de implementar. De igual forma consideramos que no es un tópico relevante al curso esta implementación técnica.

8. Trabajo Futuro

Existen muchas formas de las cuales se podría ampliar las funcionalidades que ofrece la aplicación para brindar una experiencia sin igual, se procede a listar algunas:

- **Fluctuación reciente de precio de viviendas guardadas**, ya que sabemos cuales son los inmuebles que se tienen guardados para cada usuario y además para todos los inmuebles se guarda el historial se podría en el PopUp mostrar si hubo alguna variación reciente.
- **Filtrado más completo**, actualmente el panel de filtrado permite filtrar por valores fijos. Se podría definir una sintaxis de forma de realizar consultas con “OR” o con rangos.
- **Generación de reporte**, se podrían generar reportes con gráficas y tendencias del mercado y de todas las publicaciones que sigue un usuario. La información necesaria ya se esta guardando, faltaría implementar el procesamiento.
- **Análisis Geográfico**, debido a la implementación necesaria también se obtuvo información sobre la densidad geográfica de los inmuebles publicados, se podrían realizar reportes o estudios respecto a la misma. Incluso se podría realizar una visualización con zonas de calor.
- **Manejo de publicaciones guardadas desde PopUp**, hoy en día depende de si se guardan publicaciones dentro de la aplicación de Mercadolibre, a forma de mejor experiencia manejar de forma separada en el PopUp las guardadas.

Referencias

- [1] Meta, *ReactJS*, 2022. [Online]. Available: <https://reactjs.org/>
- [2] Google, *Chrome API*, 2022. [Online]. Available: <https://developer.chrome.com/docs/extensions/mv3>
- [3] R. L. Dahl, *NodeJS*, 2022. [Online]. Available: <https://nodejs.org/>
- [4] Mercadolibre, *API Mercadolibre*, 2022. [Online]. Available: https://developers.mercadolibre.com.uy/es_ar/api-docs-es