

WEBIR 2020 - Proyecto final
Facultad de Ingeniería - UdelaR

WHouses



Estudiantes:

Lucía Doldán 4.373.409-8

Eugenia García 5.040.846-4

Federico Silveira 4.716.537-2

Docente: Libertad Tansini

Introducción	3
Problema	3
Enfoque de la solución	3
Diseño e implementación	4
Arquitectura	4
Componentes	5
Componente backend	5
Componente frontend	5
Base de datos	5
Herramientas	5
Node.js	5
Puppeteer	5
Angular	5
MongoDB	6
Web scraping	6
Funcionalidades y uso	7
Conclusiones	8
Trabajo a futuro	9
Referencias	10

Introducción

Actualmente el mercado inmobiliario tiene un gran flujo de usuarios a través de la web, donde los anuncios de los inmuebles son promocionados en las páginas de las distintas inmobiliarias, pero cada una de ellas muestra la información en un formato específico y no todas las publicaciones se encuentran en todas las páginas.

Particularmente en Uruguay, la mayor cantidad de anuncios de propiedades se encuentran publicados en MercadoLibre, InfoCasas, y el Gallito, habiendo otras que son utilizadas pero no se encuentran tan popularizadas como las mencionadas.

Hoy por hoy, no existe ninguna aplicación que centralice todas las publicaciones de alquiler o venta de inmuebles y creemos que sería muy útil para muchas personas.

Problema

Se abordará el problema de centralizar la información dispersa en distintas páginas web de inmobiliarias, de forma de poder mejorar la experiencia del usuario al realizar una búsqueda, ofreciendo de esta forma un prototipo que provea todas las publicaciones disponibles en una sola aplicación.

De esta manera, el usuario podrá visualizar la información en base a un único formato de los datos y sin perderse de publicaciones que podrían encontrarse en otras páginas. Además la mayoría de los sitios de este estilo poseen filtros básicos para que el usuario pueda obtener las publicaciones según sus preferencias por lo que sería bueno que nuestra aplicación también cuente con los mismos.

Enfoque de la solución

Se implementó una aplicación que se encarga de proporcionar un listado de todos los apartamentos que se encuentran disponibles para alquilar en la ciudad de Montevideo.

El usuario podrá aplicar filtros para obtener distintos resultados de búsqueda según sus intereses.

Los filtros que se ofrecen son los siguientes:

- Precio
- Barrio
- Metraje
- Cantidad de habitaciones
- Cantidad de baños

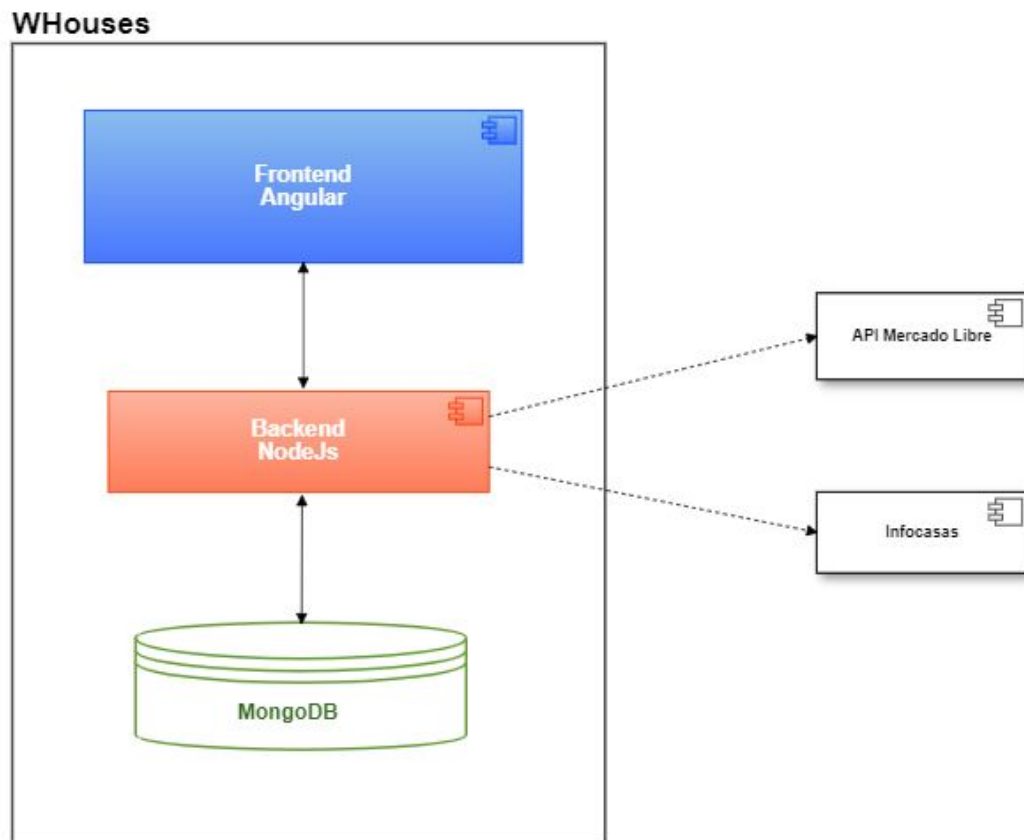
La aplicación tomará los datos de la API de MercadoLibre mediante el protocolo REST y también de InfoCasas realizando web scraping.

Inicialmente se consideró la opción de consumir datos del Gallito Luis pero dado a la escasez de tiempo y a la heterogeneidad de la estructura de la información de las

dos fuentes que se utilizaron se creyó pertinente trabajar únicamente con Mercado Libre e Infocasas.

Diseño e implementación

Arquitectura



Componentes

Componente backend

Se encarga de manejar la lógica de la aplicación. Se comunica con la API de MercadoLibre y realiza la lógica de web scraping así como también la transformación de los datos para ser persistidos en la BD.

Este componente expone una API REST que es consumida por el frontend.

Componente frontend

El frontend está implementado en Angular y está conformado por una única pantalla que muestra los filtros de búsqueda, un botón para disparar la consulta y el listado correspondiente al resultado de dicha búsqueda.

A continuación, se dá muestra de la vista principal de la aplicación:

Base de datos

Se encarga de persistir la información normalizada que fue recabada de las distintas páginas elegidas, con un formato común de tipo JSON.

Para llevar a cabo esto, se utilizó MongoDB, creando esquema de documento para la entidad house, en la que se agregaron los campos principales para poder hacer el filtrado de la información. Se consideró esta base de datos, ya que permite guardar grandes volúmenes de información y generar índices por determinadas propiedades, lo que acelera las consultas sobre los datos.

Herramientas

Node.js

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Puppeteer

Es una librería de Node que se utilizó para extraer los diferentes apartamentos en alquiler en Montevideo desde la página de InfoCasas.

Puppeteer provee una API de alto nivel con la cual se puede controlar Chrome (o Chromium) a través del protocolo DevTools.

Angular

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones

basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

MongoDB

MongoDB es una base de datos no relacional y distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. Ninguna otra ofrece un nivel de productividad de uso tan alto.

Web scraping

Es la técnica que se utiliza para extraer información de páginas web de forma automática. Consiste en leer el código de una página para obtener datos en bruto y transformarlos en datos estructurados que se pueden guardar en bases de datos.

Funcionalidades y uso

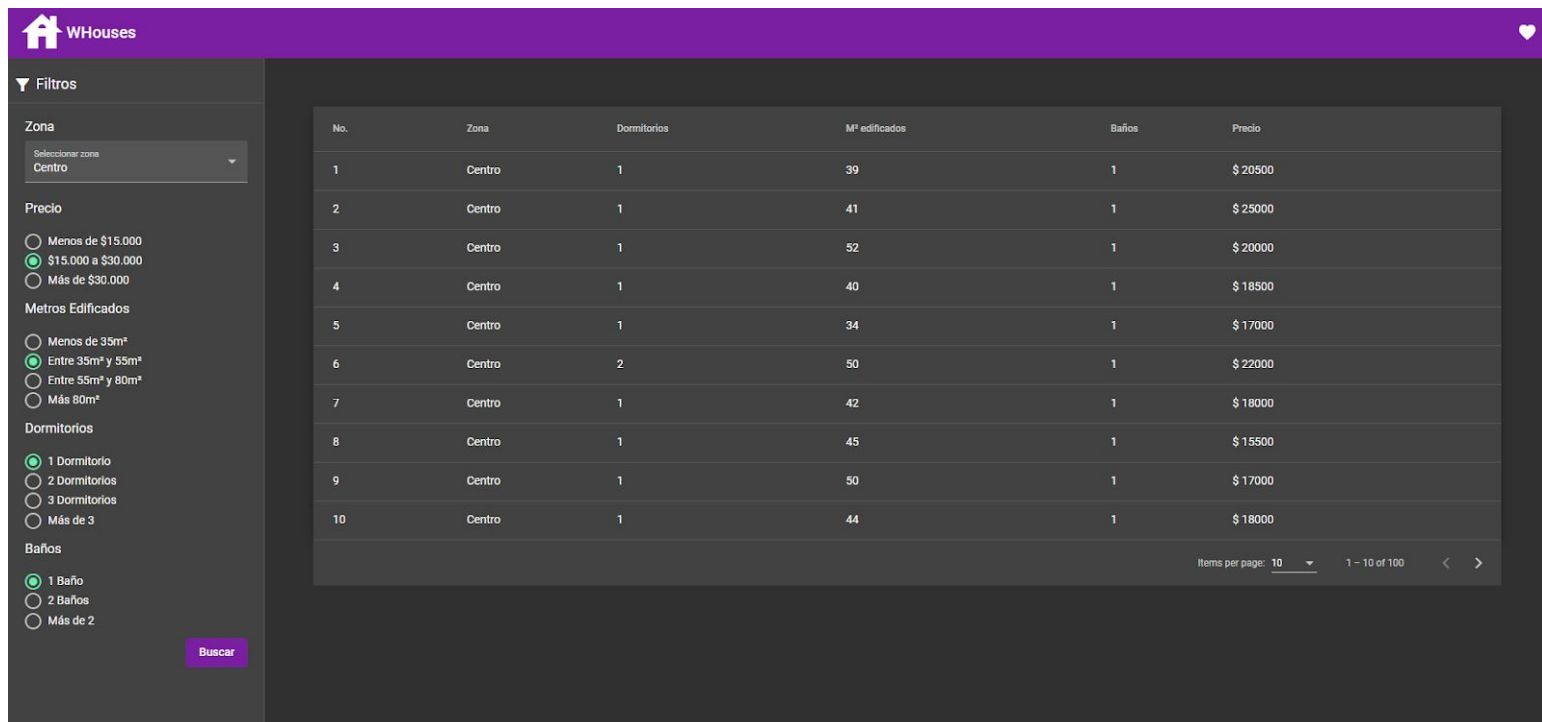
La aplicación cuenta con una interfaz que provee una sección para realizar filtros y una sección para mostrar los resultados de las búsquedas de apartamentos que se encuentran para alquilar en Montevideo.

Una vez se ingresa al sitio, el usuario puede comenzar a realizar la búsqueda de su futuro apartamento acotando las opciones al ingresar los filtros que más se adecuen a sus necesidades.

Al comenzar la búsqueda se realiza una consulta a la base de datos MongoDB que busca entre todos los resultados disponibles en Mercado Libre y en InfoCasas que cumplen con los filtros de la búsqueda.

Una vez que se tenga la respuesta, los datos son mostrados en la aplicación en forma de tabla.

A continuación se muestra la pantalla principal de la aplicación:



The screenshot displays the WHouses application interface. On the left, there is a sidebar with filter options under the heading 'Filtros'. The filters include:

- Zona:** A dropdown menu currently set to 'Centro'.
- Precio:** Radio buttons for 'Menos de \$15.000', '\$15.000 a \$30.000' (selected), and 'Más de \$30.000'.
- Metros Edificados:** Radio buttons for 'Menos de 35m²', 'Entre 35m² y 55m²' (selected), 'Entre 55m² y 80m²', and 'Más 80m²'.
- Dormitorios:** Radio buttons for '1 Dormitorio' (selected), '2 Dormitorios', '3 Dormitorios', and 'Más de 3'.
- Baños:** Radio buttons for '1 Baño' (selected), '2 Baños', and 'Más de 2'.

A purple 'Buscar' button is located at the bottom of the filter sidebar. The main area on the right shows a table of search results with the following columns: No., Zona, Dormitorios, MP edificados, Baños, and Precio. The table contains 10 rows of data:

No.	Zona	Dormitorios	MP edificados	Baños	Precio
1	Centro	1	39	1	\$ 20500
2	Centro	1	41	1	\$ 25000
3	Centro	1	52	1	\$ 20000
4	Centro	1	40	1	\$ 18500
5	Centro	1	34	1	\$ 17000
6	Centro	2	50	1	\$ 22000
7	Centro	1	42	1	\$ 18000
8	Centro	1	45	1	\$ 15500
9	Centro	1	50	1	\$ 17000
10	Centro	1	44	1	\$ 18000

At the bottom right of the table, there is a pagination control showing 'Items per page: 10' and '1 - 10 of 100'.

Conclusiones

Durante la construcción de la aplicación nos enfrentamos a algunas dificultades a la hora de juntar los datos provenientes de ambas fuentes en un mismo formato. Los datos extraídos de MercadoLibre estaban en un formato que se encontraba debidamente documentado en el sitio de su API por lo que no tuvimos mayores dificultades. Sin embargo al obtener los datos de la página InfoCasas encontramos algunas dificultades como por ejemplo: valores faltantes con un formato distinto al deseado. Algunos de estos decidimos ignorarlos y a otros modificarlos y guardarlos. En esto podemos observar una dificultad a tener en cuenta a la hora de hacer aplicaciones que toman datos de distintos sitios y mediante distintas herramientas.

Más allá de lo expresado en el párrafo anterior, obtuvimos una primera versión de la aplicación funcional, amigable y simple de utilizar, que permite al usuario filtrar y visualizar una lista de inmuebles, para poder obtener resultados según su interés. Es por esto que logramos los principales objetivos que nos planteamos al comenzar a desarrollar.

También concluimos que las técnicas de recuperación de información de información vistas en el curso son muy útiles para poder centralizar información de la web que está dispersa en varios sitios. También puede utilizarse para obtener información de un sitio y mostrarla de manera más amigable o permitiendo al usuario filtrar la misma según sus intereses.

Trabajo a futuro

Si bien la aplicación desarrollada cumple con los objetivos propuestos y resuelve el problema planteado, consideramos que se podría mejorar y optimizar en muchos aspectos para lograr darle un mayor valor al usuario. Algunas de estas mejoras se detallan a continuación:

- *Obtener información de más sitios.* Si bien elegimos los dos sitios más utilizados por la gente para buscar alquileres de inmuebles, sabemos que existen una cantidad de otros sitios, de los cuales se podría extraer información para mostrar al usuario. Cuantos más de estos se incluyan, el usuario podrá obtener un mayor número de resultados que concuerden con sus preferencias. Algunos de los que se podrían incluir son: Gallito Luis, información de grupos de Facebook, sitios web de inmobiliarias, entre otros.
- *Actualizar la base de datos más frecuentemente.* Como la aplicación desarrollada es un prototipo, la misma reinicia la base de datos y toma los datos de ambos sitios al comenzar su ejecución. Algo deseable si se pretende usar es que, periódicamente (1 vez al día al menos) se actualice la base de datos para así poder mostrar al usuario información actualizada.
- *Actualizar los datos eficientemente.* Actualmente, cada vez que inicia la aplicación se eliminan todos los datos y se obtienen todos los datos actualizados. Al escalar la aplicación y trabajar con muchos sitios, se debería encontrar una manera más eficiente de actualizar los datos.
- *Mejorar los filtros y el ordenamiento.* En la aplicación desarrollada se permite al usuario filtrar las publicaciones según 5 filtros. Una mejora para la misma sería agregar más filtros, para que su búsqueda se adapte más a sus necesidades. A su vez, sería bueno permitir al usuario definir de qué manera quiere ordenar los resultados obtenidos.
- *Referencias a la publicación original e imágenes.* Creemos que sería bueno poder agregar un link a la publicación original para que el usuario pueda navegar fácilmente a la misma, por ejemplo, para hacerle una pregunta al dueño. A su vez, otra mejora que queda pendiente es extraer imágenes de cada inmueble y mostrarlas si el usuario lo desea.

Referencias

- https://developers.mercadolibre.com.uy/es_ar/guia-para-inmuebles
- <https://www.infocasas.com.uy/>
- <https://nodejs.dev/>
- <https://www.mongodb.com/es>
- <https://pptr.dev/>