Recuperación de información y recomendaciones en la web

Curso 2020

Informe final - Aplicación BookitUp

Grupo 5

Gabriela Rodríguez - 4.937.370-9
Lucía Rotela - 4.899.525-9
Lia Colombo - 4.817.448-5
Viviana Luongo - 4.644.064-4

Índice

Introducción

Actualmente al querer buscar y comprar un libro, se pueden encontrar muchas opciones disponibles, hay una gran variedad de géneros, autores e incluso idiomas. En caso de que un usuario esté buscando un libro particular, necesita saber dónde puede conseguirlo y a qué precio; en caso de que busque una recomendación, es importante ofrecerle buena variedad de opciones.

Para este trabajo, decidimos enfocarnos en la resolución de este problema, utilizando herramientas adquiridas durante el curso, para crear una aplicación web que permita obtener información de los sitios web de distintas librerías y presentarla al usuario de manera amigable. Se utilizará la técnica conocida como “scraping”, que se explicará a detalle más adelante. Presentaremos el problema a resolver, describiremos la aplicación diseñada y las tecnologías utilizadas para su implementación, dejaremos ejemplos de su funcionalidad y su uso, evaluaremos su funcionamiento y mencionaremos posibles mejoras.

Motivación y descripción del problema

Como mencionamos en la sección anterior, la motivación principal es la búsqueda y recomendación de libros. Si un usuario está buscando un libro, precisa saber dónde se consigue, cuál es la disponibilidad y el precio dependiendo de los distintos locales de venta para poder comparar. En caso de que no esté buscando un libro en particular puede necesitar una recomendación para definir qué libro comprar. También puede compartir su opinión sobre un libro que ya leyó.

El problema a resolver, es simplificar la búsqueda de un libro reuniendo todas las ofertas en un solo sitio y presentar al cliente tanto el precio como el stock disponible por local. También tendremos en cuenta el problema de presentar y permitir realizar reseñas.

Enfoque de la solución

Se propone tener una página disponible para búsquedas con los criterios antes mencionados: autor, género, idioma o nombre del libro.
Se accederá en tiempo real a las páginas de las librerías seleccionadas para obtener la información según estos criterios. Los datos que se espera obtener de las librerías para cada libro son precio y stock, de forma que el usuario tenga certeza de dónde puede conseguir el libro. Se podrá mostrar información de la librería como por ejemplo, dirección del local, teléfono y correo electrónico de contacto.

Se consideraron varias librerías de Uruguay, para la primera versión de la solución se accederá a Grupo Libros[1] y Bookshop[2].

Para la elección de las librerías se tuvo en cuenta que ofrecieran la información del stock disponible, ya que los precios se encuentran en todas. Se evaluó también el tiempo de respuesta al realizar scraping.

Al momento de mostrar los resultados obtenidos en la búsqueda, se agregarán las reseñas correspondientes al ISBN del libro. Las reseñas se componen por: comentario y nombre del usuario.
Al hacer una búsqueda, el usuario también podrá reseñar alguno de los libros listados sin necesidad de *login*.

Herramientas utilizadas

Beautiful Soup

Beautiful Soup es una biblioteca de Python para realizar scraping, permite obtener la información de archivos HTML y XML. Al usarla con un archivo, se obtiene un árbol que se puede recorrer mediante distintas técnicas, como por ejemplo selectores, para sacar los datos de las etiquetas que necesitemos[3]. Con esto se puede obtener información de cada libro.

Django

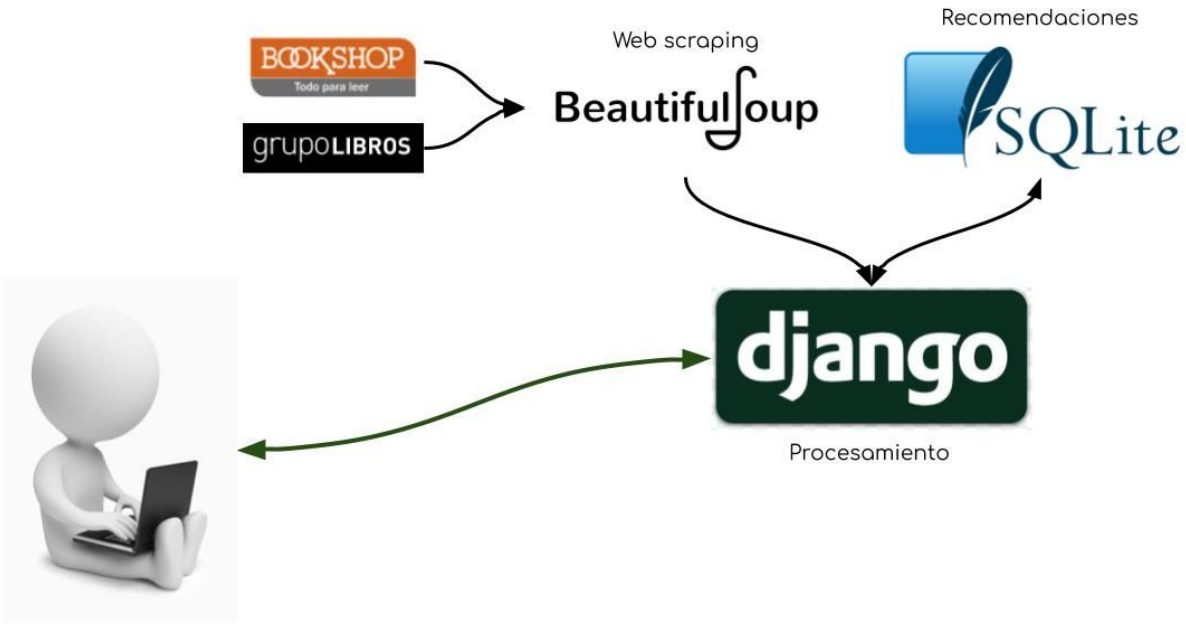
Django es un framework Web para Python que permite diseñar aplicaciones web de forma rápida y limpia, encargándose de crear proyectos con todos los archivos necesarios[4]. Se usará para facilitar la construcción del sitio web.

SQLite

SQLite es la base de datos que viene configurada por defecto con Django. Si bien no se recomienda para proyectos grandes por no ser muy escalable[5], en este caso se considera que aplica, dado que solo se va a almacenar las reseñas de usuarios y el código ISBN del libro reseñado.

Diseño e implementación

El sistema será un sitio web implementado con el *framework* Django. Dentro del mismo, habrá un módulo en Python que se encargará de la búsqueda de libros. También realizará la tarea de obtener la información en tiempo real de los sitios web ya mencionados. Para obtener esta información se usará la técnica de *web scraping*, en particular usando la herramienta Beautiful Soup. Para almacenar las puntuaciones de usuarios, se contará con una base de datos SQLite.



En la base de datos quedará almacenada la reseña asociada a cada libro, identificado por su código ISBN. La reseña es un modelo que contiene el nombre del usuario que la crea y el texto correspondiente. El resto de la información se obtendrá en tiempo real al momento de cada búsqueda, esto permitirá traer siempre información actualizada sobre el stock disponible y precio. En particular, la información que se trae de cada libro es: nombre, autor, precio, género, código ISBN, imagen (para mostrar en el sitio), idioma, y si hay disponibilidad. Esta información es luego combinada con la reseña, para mostrar toda la información al usuario. Todos estos datos se almacenan en diccionarios Python (análogos a objetos JSON).

Criterios de búsqueda

- Nombre
    - Representa el título del libro a ser buscado, se puede dejar en blanco si lo que se quiere es conseguir una recomendación por autor o género.
- Autor
    - Es el nombre completo o parcial del autor del libro que se está buscando.
- Género
    - Para definir este criterio de búsqueda se analizaron los definidos en Bookshop y GrupoLibros, se creó una única lista que combina los criterios de ambos sitios. GrupoLibros maneja género y subgénero, cosa que no sucede en Bookshop. Por lo tanto en la implementación se utiliza un enumerado de géneros en común, cuyo identificador luego se mapea en cada proceso de scraping al nombre del género o subgénero correspondiente.
- Idioma
    - Se pueden buscar libros en inglés y/o en español.

Scraping

Esta técnica se utiliza para extraer información de sitios web. Usualmente se simula la navegación de un ser humano en la web accediendo al contenido html y así a los datos que se muestran[6].
Para este caso particular, teniendo definidas las URLs de los sitios que se van a utilizar para aplicar scraping, primero se hace la búsqueda de acuerdo a los criterios seleccionados. En el caso de nombre y autor, se usa directamente el buscador de los sitios agregando la información en la URL.
Para la búsqueda por género existen diferencias, en el sitio de GrupoLibros se puede usar el buscador pero en Bookshop no es posible, se tienen páginas para cada género.

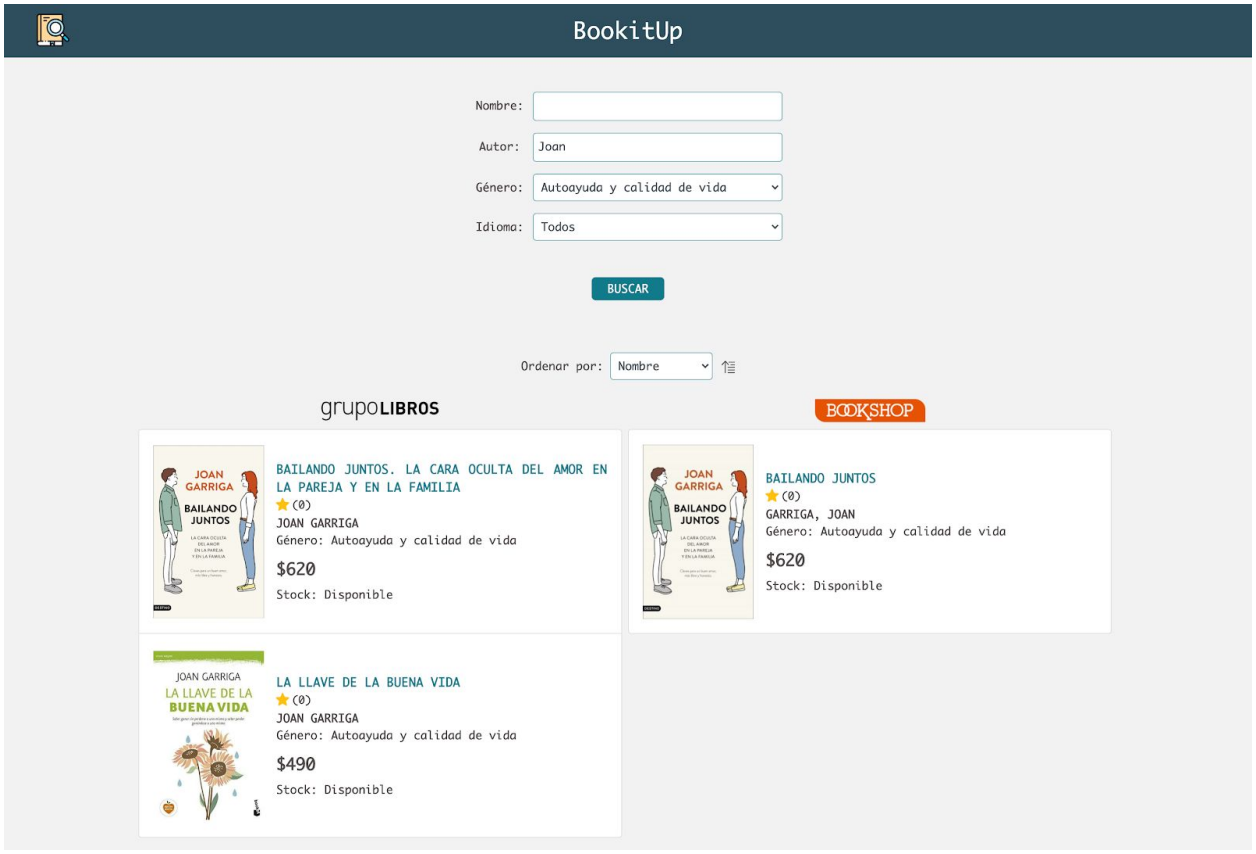
Si bien existe la opción de buscar libros en inglés, hasta el momento GrupoLibros solamente dispone de libros en español, por lo que la búsqueda no retorna resultados de este sitio. Es por esto que cuando se selecciona la opción de idioma inglés, la búsqueda se realiza únicamente en Bookshop. En el sitio de Bookshop no es posible realizar una búsqueda por idioma entonces, necesariamente primero se filtra por los otros criterios y entrando al detalle de cada libro se verifica si es en inglés o no. Es por este motivo que se implementó el buscador de manera que no se pueda buscar únicamente por idioma, de esta forma se evitan problemas de performance.

Otro tema relacionado con la mejora de la performance, es que la búsqueda se limitó la cantidad de libros a 40, el motivo de el número elegido es que Bookshop (que tiene la mayor demora en la consulta) lista de a 40 libros y de esta forma se trae una sóla página, con respecto a GrupoLibros éste pagina de a 12 libros y lo que se hace es ir buscando resultados hasta llegar a 40 para devolver de ambos sitios la misma cantidad.

Funcionalidad y uso

La aplicación web luce como se muestra en la siguiente imagen, los criterios que se pueden aplicar para la búsqueda son nombre del libro, autor, género e idioma. Los once géneros que se tienen en cuenta surgen de una lista que combina los sitios seleccionados (ya que no tienen exactamente los mismos géneros). El idioma puede ser español o inglés, si se busca uno en particular.

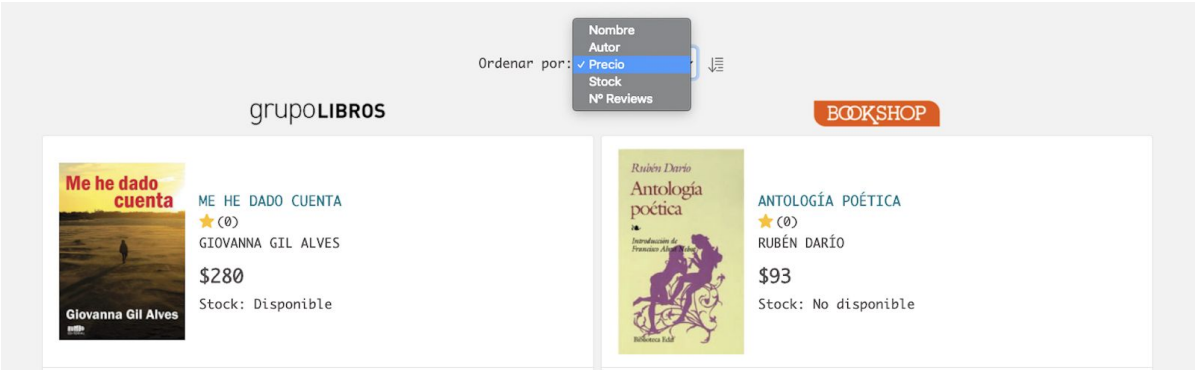
En la siguiente imagen se puede ver un ejemplo de búsqueda por Nombre con los resultados de ambos sitios.



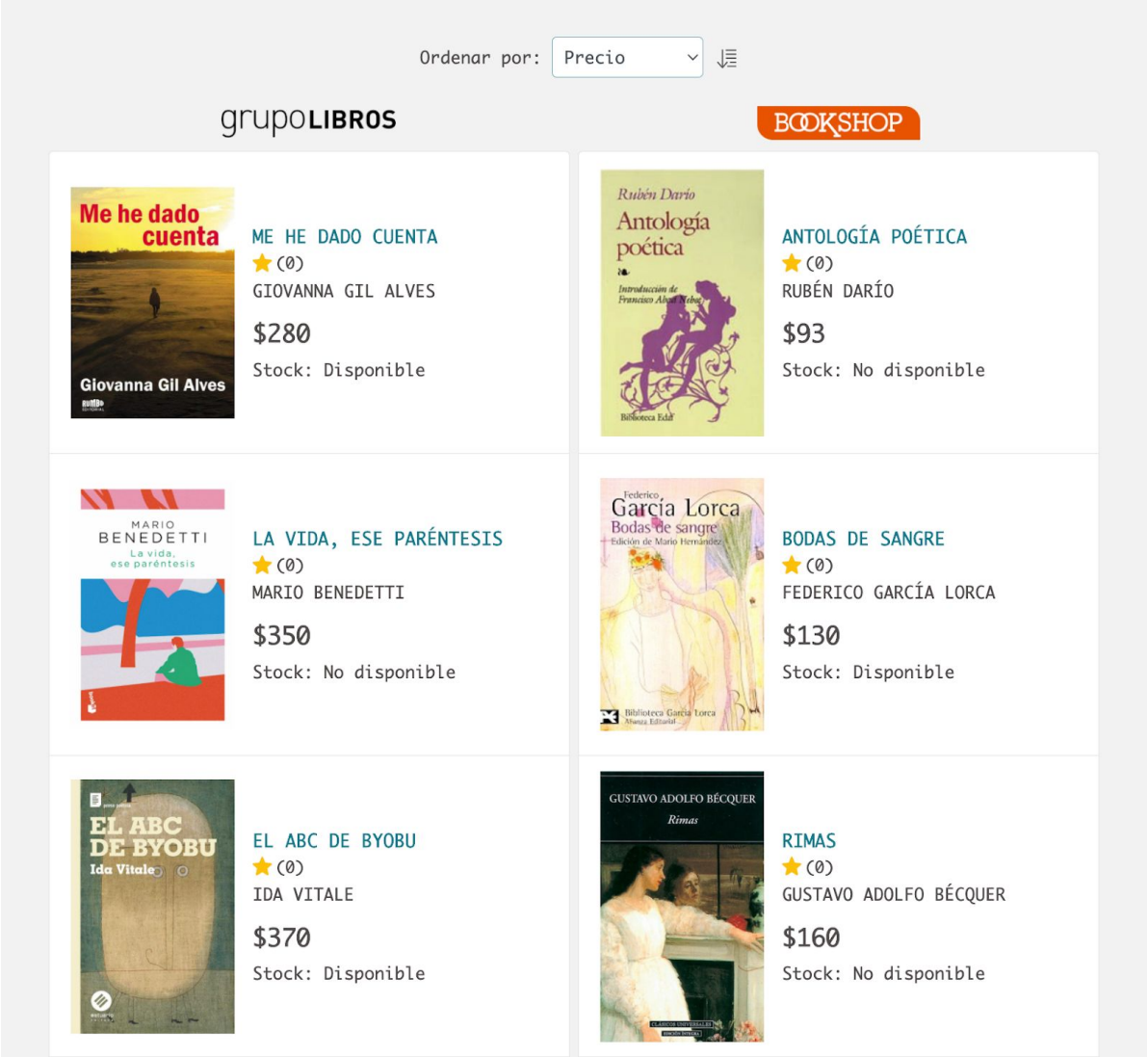
Luego de realizar una búsqueda es posible ordenar los resultados obtenidos por varios criterios, estos son:

- Nombre del libro
- Autor
- Precio
- Stock
    - Disponible ó No disponible
- Cantidad de reviews.

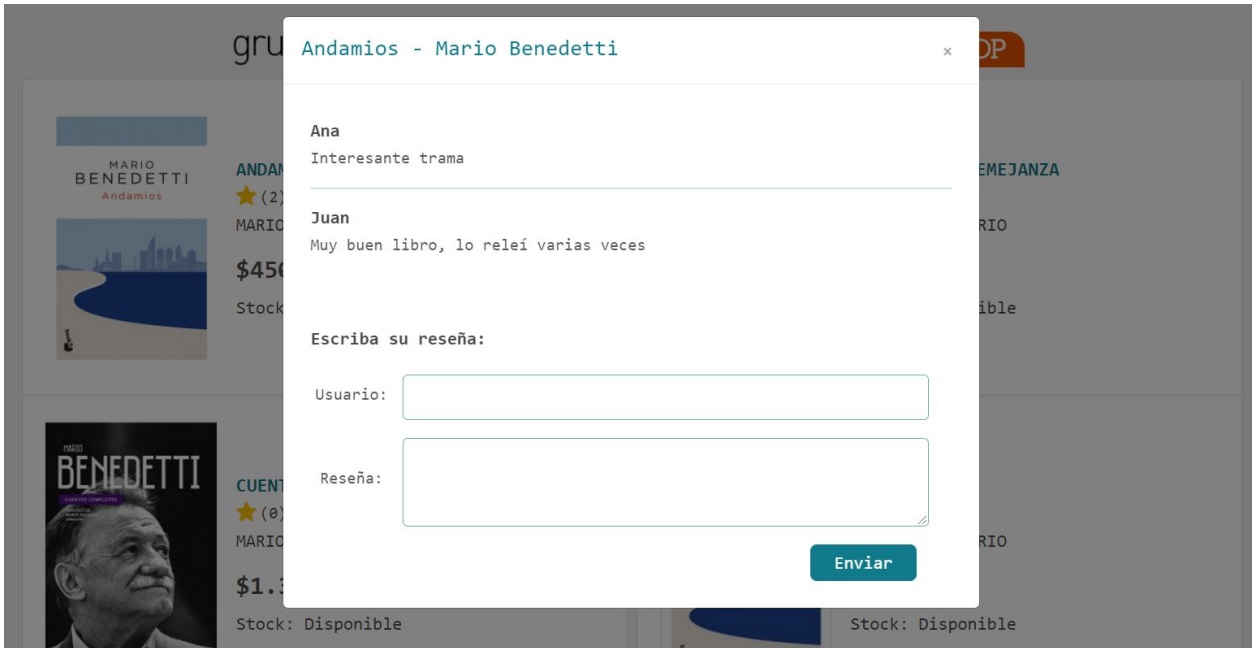
Además se cuenta con un botón que permite elegir si el orden es ascendente o descendente.



En la siguiente captura se muestran los resultados de una búsqueda ordenados por su precio de forma ascendente. De esta forma es posible encontrar fácilmente libros más económicos.



Se puede ver que cada libro tiene un ícono de estrella. Este se utiliza para visualizar las reseñas ingresadas por los usuarios, el cero representa que aún no hay reseñas para ese libro. Haciendo click sobre el número, se pueden ver los diferentes comentarios, desde ese mismo pop up el usuario puede dejar su reseña, agregando su nombre.



Evaluación y resultado

La aplicación es muy sencilla de utilizar y hace de forma efectiva su trabajo que es devolverle al usuario la información disponible. Adicionalmente, las reseñas resultan útiles a aquellos usuarios que no buscan un libro específico.

Sería bueno incluir más fuentes, pero la mayoría de las páginas de librerías no incluyen toda la información buscada, como por ejemplo la disponibilidad.

El principal problema es la demora en la búsqueda, dado que el scraping se realiza en tiempo real. Para la página de Bookshop, la demora es particularmente notoria; como se comentó anteriormente esto llevó a limitar la cantidad de libros mostrados.

Conclusiones

El scraping tiene dos desventajas importantes, no es escalable ni fácilmente mantenible. En el marco de este proyecto ninguna de las dos cosas afectan demasiado, pero son dos puntos decisivos al momento de considerarlo en proyectos de mediano y gran porte. Si las páginas de las que se está obteniendo información renovaran su apariencia, se debe cambiar completamente el código de los módulos de scraping. Además, dado su funcionamiento, tampoco se puede generalizar de un sitio a otro, si se quisieran agregar nuevas fuentes de información, se debe implementar desde cero el scraping correspondiente.

La falta de homogeneidad entre sitios que ofrecen un mismo producto también es un problema. En este caso particular, los datos de cada libro ofrecidos por distintas páginas de librerías eran muy distintos, y resultó difícil encontrar un punto de conexión. Cierto es que existen iniciativas, como el uso de metadatos (y existen estándares para libros), pero falta mucho camino en ese sentido.

Trabajo a futuro

El scraping demora un tiempo considerable en analizar todas las páginas, y la aplicación realiza este scraping en el momento en que el usuario hace una búsqueda, para obtener información lo más actualizada posible. Esto lleva a que el tiempo de respuesta sea un poco más alto de lo acostumbrado por los usuarios modernos. Sería bueno considerar algún otro mecanismo para tener información actualizada evitando esta demora. Una forma sería almacenar la información obtenida de las páginas en una base de datos (se podría usar, por ejemplo, Elasticsearch, que es rápida y permite buscar en archivos JSON[7]), y al momento de realizarse una búsqueda, hacerlo en dicha base, evitando de este modo tener que analizar varias páginas cada vez que se hace una búsqueda. Para asegurar la frescura de la información, se podría tener en el servidor de Django una tarea programada que se ejecute cada cierto tiempo. Esta tarea realizaría el scraping y compararía la información obtenida con la almacenada, actualizando la base de datos en los casos que sea necesario. Como las páginas de las librerías no van a ser actualizadas permanentemente, la tarea sólo se ejecutaría, por ejemplo, cada algunas horas, garantizando de este modo que no va a ser una carga para el servidor que ya está atendiendo las búsquedas de los usuarios.

Otra alternativa, si se contara con varios equipos, sería hacer el scraping en el momento, pero paralelamente en distintos equipos para cada sitio, permitiendo que la información se vaya cargando a medida que está disponible. Esto reduciría el tiempo que se demora en obtener un resultado, no tanto como la opción anterior, pero la información se obtendría en el momento.

Dado lo anterior se podría no limitar la visualización a 40 libros o en otro caso ofrecer la opción de “ver más” para que el usuario obtenga la mayor cantidad de información posible.

En cuanto a las recomendaciones y reseñas, la aplicación permite únicamente dejar una reseña con un nombre, en un futuro se podría agregar la valoración de otros usuarios como un criterio de búsqueda.
También, se podría tener un usuario registrado para poder realizar recomendaciones personalizadas en base a los libros que ya leyó y reseñó. Además, podría guardar una lista de libros en los que está interesado para comprar más adelante.

Referencias

[1] https://grupolibros.com.uy/
[2] https://www.bookshop.com.uy/
[3] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[4] https://www.djangoproject.com/
[5] https://docs.djangoproject.com/en/3.1/intro/tutorial02/
[6] https://es.wikipedia.org/wiki/Web_scraping
[7] https://www.elastic.co/what-is/elasticsearch