



UNIVERSIDADE DA CORUÑA

MEMORIA PRÁCTICA BUSCADOR

BUSCA TU HOTEL

RECUPERACIÓN DE INFORMACIÓN Y WEB SEMÁNTICA

17 de noviembre de 2023

Samuel Deus García - samuel.deus@udc.es
Anxo Sánchez Rouco - anxo.sanchezr@udc.es
Adrián López Gude - adrian.lopez.gude@udc.es

Enlace al repositorio en GitHub: <https://github.com/adrian-gude/RI>

Índice

1. Introducción	3
2. Dominio	3
3. Tecnologías	5
3.1. Herramientas de Desarrollo	5
3.2. Lenguajes de programación	6
3.3. Librerías	6
4. Desarrollo	7
4.1. Crawler	7
4.2. ElasticSearch	9
4.3. Frontend	11

1. Introducción

Este proyecto consiste en la implementación de una aplicación web, que permitirá consultar información acerca de los principales hoteles españoles. Esta información se obtiene utilizando herramientas de web scraping de una de las plataformas turísticas líderes en el mundo. El resultado de la aplicación permite al usuario realizar búsquedas y consultas sobre hoteles españoles, como precios, valoraciones, etc.

2. Dominio

Como fuente de datos se ha utilizado la página web *Tripadvisor*, una página en línea en la que se recogen millones de opiniones y datos en torno a determinados negocios del sector turístico (hoteles, restaurantes, bares, discotecas, cruceros, etc). En concreto, se hizo uso de los datos relacionados con el sector turístico hotelero de España. Como se puede ver en la Figura 1, la página ofrece información en formato paginado horizontal sobre los distintos hoteles presentes para una comunidad introducida, e información detallada de cada uno en paginado vertical al seleccionarlo (Figura 2).

The figure displays three cards from the Tripadvisor website, each representing a hotel in Galicia:

- 1. Augusta Eco Wellness Resort**:
 - Rating: 4.5 stars (2247 reviews)
 - Rank: #1 of 3623 accommodations in Galicia
 - Review by De Annika: "La cena en el restaurante Potus fue igualmente buena, la comida era simple pero deliciosa, fresca y, a menudo, con un toque gallego (local). Recibí un servicio fantástico de Gustavo, Vanessa, Alex, Fran, Ernesto y Artem."
 - Link: [Visitar sitio web del hotel](#)
- 2. Parador de Santiago de Compostela**:
 - Rating: 4.5 stars (3013 reviews)
 - Rank: #2 of 3623 accommodations in Galicia
 - Review by De Paulina A: "Vale la pena quedarse al menos unas dos noches para disfrutar al máximo el hotel, que es un museo también, todos son muy amables, y el ambiente en este mágico lugar es de las mejores experiencias que hemos tenid..."
 - Link: [Visitar sitio web del hotel](#)
- 3. Occidental Vigo**:
 - Rating: 4.5 stars (490 reviews)
 - Rank: #3 of 3623 accommodations in Galicia

Figura 1: Listado de hoteles para la comunidad de Galicia

Gran Hotel Los Abetos Oferta especial Ofertas Información Ubicación Opiniones 72 € Booking.com Ver oferta

4,5 Excelente

15 de 72 hoteles en Santiago de Compostela

Ubicación
Limpieza
Servicio
Relación calidad-precio

Travellers' Choice

Gran Hotel Los Abetos es un confortable y acogedor hotel 4 estrellas superior ubicado en Santiago de Compostela, a 5 minutos del aeropuerto de Lavacolla, de la Catedral de Santiago y del estadio de San Lázaro. Entre sus servicios, dispone de 149 habitaciones, más de 10.000 m² de jardín botánico, villas exclusivas, diversos salones, capilla, auditorio, piscina, gimnasio, sauna o pista de tenis. El hotel ofrece, de manera gratuita, WiFi en todo el recinto, parking privado y servicio de transfer al centro de la ciudad. Por si esto fuera poco, el Gran Hotel Los Abetos cuenta con uno de los mejores restaurantes de Santiago de Compostela: El Leer más ▾

Sugiere modificaciones para mejorar lo que mostramos.

[Mejora este perfil](#)

Servicios de la propiedad

- Aparcamiento gratuito
- Internet de alta velocidad gratuito (wifi)
- Piscina exterior
- Gimnasio / Sala de entrenamiento
- Bar/salón
- Pista de tenis
- Servicios gratuitos de traslado en autobús o taxi
- Centro de negocios con acceso a Internet

[Mostrar más](#)

Servicios de habitación

- Habitaciones insonorizadas
- Aire acondicionado
- Balcón privado
- Servicio de habitaciones
- Caja fuerte
- Minibar
- Televisor de pantalla plana
- Secador de pelo

[Mostrar más](#)

Tipos de habitación

- Habitaciones de no fumadores
- Suites
- Habitaciones para familias

Información relevante

CATEGORÍA DEL HOTEL ★★★★☆

IDIOMAS QUE SE HABLAN
Español, Inglés, Francés, Italiano y 1 más

Figura 2: Información completa sobre un hotel.

Para cada hotel se han extraído los siguientes datos:

- **Nombre:** Campo que contiene el nombre del hotel.
- **Comunidad:** Este campo representa la comunidad autónoma donde se ubica el hotel.
- **Precio:** Representa el precio de lo que cuesta una habitación por noche.
- **Localización:** Dirección completa del lugar donde se localiza el hotel.
- **Número de opiniones:** Representa el número de usuarios que han añadido una opinión al hotel.
- **Puntuación:** Es la media de las puntuaciones de los usuarios en un escala de [1-5].
- **Categoría:** Representa la clasificación por estrellas de un hotel en una escala de [1-5] en función al nivel de lujo del hotel.
- **Idiomas:** Representa que idiomas se hablan en la recepción del hotel. Se han seleccionado Español, Inglés, Francés, Italiano y Portugués, como idiomas de interés.
- **Servicios:** Representa los servicios que oferta el hotel. Se han seleccionado Aparcamiento público de pago cerca, Wifi, Gimnasio, Restaurante, Sauna y Hoteles de no fumadores como servicios de interés.
- **Dirección web de la página de detalles:** Representa el enlace a la página web de tripadvisor de información detallada del hotel.
- **Dirección web de la imagen :** Representa la URL de la imagen de cada hotel.

3. Tecnologías

3.1. Herramientas de Desarrollo

Elastic Search

Elasticsearch es un servidor de búsqueda basado en *Apache Lucene*, que provee un motor de búsqueda.

Se ha optado por su empleo debido a su alto rendimiento para procesar tanto las consultas como para la carga de datos, sumado a la simpleza a la hora de manejar las peticiones. En concreto, se hace uso de las versiones 8.10 para los ordenadores en *Windows* y 7.17 para el sistema *macOS*.

Postman

Postman es una herramienta *software* utilizada por los desarrolladores principalmente para testear colecciones y catálogos APIs (tanto a nivel front-end como back-end) y para gestionar el ciclo de vida de las APIs.

En este proyecto se ha utilizado con el fin de facilitar la comunicación entre el API de Elasticsearch y los datos extraídos con Scrapy, además de para realizar diferentes consultas contra el índice.

Multi Elasticsearch head

Multi Elasticsearch head es una extensión de navegador que integra el API de Elasticsearch. Contiene un ClusterOverview, que muestra la topología del clúster y le permite realizar operaciones a nivel de índice y nodo. Contiene un par de interfaces de búsqueda que permite consultar el clúster y recuperar resultados en formato json sin formato o tabular.

Se utiliza debido a la facilidad que ofrece para visualizar las peticiones al API RESTful de Elastic, además de ayudar en la creación de consultas.

NPM

Npm es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.

Se obtó por su uso para la creación y gestión de dependencias en la parte del *front-end* de la aplicación web.

GitHub

GitHub es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git, y que se utiliza principalmente para la creación de código fuente de programas de ordenador.

En este caso, se usó un repositorio de Git para almacenar los documentos del proyecto. Así, se puede localizar en el repositorio el código utilizado en la extracción de datos, Json de los datos extraídos o las llamadas a la Api de Elastic con Postman, entre otros.

Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado que incluye un sin fin de funcionalidades.

En este proyecto, se ha optado por su uso como principal medio de edición de código.

3.2. Lenguajes de programación

Python

Python es un lenguaje de programación interpretado, de alto nivel y de propósito general. Además, cuenta con una gran cantidad de bibliotecas y módulos que permiten ampliar su funcionalidad y simplificar el desarrollo de aplicaciones.

En este proyecto, se hizo uso de este lenguaje junto a la librería *Scrapy*, entre otras, para extraer los datos que posteriormente serán almacenados como índices invertidos haciendo uso del motor de búsqueda *Elastic Search*.

JavaScript

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas, por lo que en este proyecto se ha utilizado en la implementación de la aplicación web.

CSS

CSS es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Así, en este proyecto se empleó junto con *JavaScript* para desarrollar la aplicación web, definiendo con *CSS* los diferentes estilos de la aplicación.

3.3. Librerías

Scrapy

Scrapy es un *framework* de rastreo web de alto nivel, que se utiliza para rastrear sitios web y extraer datos estructurados, además de poder usarse para monitorear o pruebas automatizadas. Por tanto, en este proyecto se optó por su uso para la tarea de extracción de información.

Re

Es un módulo escrito en lenguaje Python, que permiten comprobar si una determinada cadena coincide con una expresión regular. Se ha hecho uso en este proyecto para variar una cadena correspondiente a una clase *HTML* y así poder seleccionar un elemento por dicha clase, para luego extraer alguna información de este.

Random

Random es un módulo nativo de Python que implementa generadores de números pseudo-aleatorios para varias distribuciones. En concreto, se hizo uso para asignar una imagen a un hotel que no se ha conseguido extraer, debido a que disponemos de una colección de imágenes a usar en estos casos y se selecciona una aleatoriamente para evitar en lo mayor posible dar una sensación de duplicidad en la aplicación web.

BeautifulSoup

Beautiful Soup es una biblioteca de Python para analizar documentos HTML. Esta biblioteca crea un árbol con todos los elementos del documento y puede ser utilizado para extraer información.

En el proyecto, es empleada para extraer la información de la web de *Tripadvisor*, debido a su facilidad de uso y a su versatilidad.

React

React es una librería Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.

En el proyecto, se ha optado por su uso debido a que, junto a la librería *ReactiveSearch*, tiene

una gran variedad de componentes que facilitan la tarea de realizar consultas sobre un índice de *ElasticSearch*.

ReactiveSearch

Reactivesearch es una librería para React de código abierto diseñada para facilitar el trabajo contra un índice de Elasticsearch, motivo por el cual se decidió su uso frente a otras alternativas. En concreto se utilizó la versión 3.45, ya que con versiones posteriores encontramos problemas de conexión con *ElasticSearch*.

Material Icons

Los Material Icons es parte de *Material Design*, el cual es un lenguaje de programación creado por Google. Así, *Material Icons* es un amplio catálogo de iconos importables, y de la cual se han usado algunos en la aplicación web.

4. Desarrollo

Para llevar a cabo este proyecto se han seguido las siguientes fases:

1. Seleccionar una página de dónde obtener la información, analizar qué datos sería interesante capturar/extraer de esta, cómo se podría hacer y, por último desarrollar un *crawler* para extraer estos datos.
2. Componer un índice invertido con los datos extraídos en el anterior paso.
3. Crear una página web en dónde se pueda visualizar los datos del índice y hacer diferentes consultas contra este.

4.1. Crawler

Para la extracción de los datos, se ha empleado un *Spyder* creado con la herramienta de *Scrapy*. Este *script*, analiza todas las *URLs* de hoteles que pertenezcan a las comunidades de España a excepción de Ceuta y Melilla que se desestimaron en el proceso de selección al no tener una cantidad suficiente de hoteles. Por otra parte, las comunidades de Andalucía y Cataluña presentaron problemas durante el proceso de extracción de los datos, por lo que decidió omitirlos. De manera análoga, se descartan los hoteles de los que no se consiga extraer el valor de alguno de sus atributos, a excepción de la categoría, los idiomas y los servicios. Estos atributos se guardarán con valor *0* en el caso de la categoría, y como *arrays* vacíos si hablamos de idiomas o servicios.

De manera general, por cada comunidad existe un *URL* única que las identifica en las que se muestran los 30 hoteles más relevantes. Esta *URL* se componen de un código con un prefijo -g , y el número de paginación compuesto por el prefijo -oa. Se ha limitado el número de páginas a extraer por cada enlace hasta un máximo de 15. Así, por cada comunidad se analizarán 15 direcciones web, en donde cada una muestra una lista de 30 hoteles.

```
https://www.tripadvisor.es/Hotels-g187506-Galicia-Hotels.html
https://www.tripadvisor.es/Hotels-g187506-oa30-Galicia-Hotels.html
https://www.tripadvisor.es/Hotels-g187506-oa60-Galicia-Hotels.html
https://www.tripadvisor.es/Hotels-g187506-oa90-Galicia-Hotels.html
https://www.tripadvisor.es/Hotels-g187506-oa120-Galicia-Hotels.html
https://www.tripadvisor.es/Hotels-g187506-oa150-Galicia-Hotels.html
```

Listing 1: Ejemplo de urls para la comunidad de Galicia

Entonces, se crea una clase *Item* que tiene todos los campos citados en la Sección 2 (Figura 3).

```
class HotelItem(scrapy.Item):
    nombre = scrapy.Field(output_processor=TakeFirst())
    comunidad = scrapy.Field(output_processor=TakeFirst())
    precio = scrapy.Field(output_processor=TakeFirst())
    comunidad = scrapy.Field(output_processor=TakeFirst())
    localizacion = scrapy.Field(output_processor=TakeFirst())
    n_opiniones = scrapy.Field(output_processor=TakeFirst())
    puntuacion = scrapy.Field(output_processor=TakeFirst())
    categoria = scrapy.Field(output_processor=TakeFirst())
    idiomas = scrapy.Field()
    servicios = scrapy.Field()
    url = scrapy.Field(output_processor=TakeFirst())
    imageUrl = scrapy.Field(output_processor=TakeFirst())
```

Figura 3: Clase *HotelItem*

Una vez se ejecuta el código del crawler, se consigue como salida un archivo en formato *JSON* con la información de aproximadamente 3000 hoteles. Es importante destacar, que aproximadamente dos tercios de los hoteles carecen de imágenes asociadas. Para mejorar la experiencia del usuario, se ha implementado la estrategia de añadir imágenes de manera aleatoria como se puede ver en la figura 4. Esta decisión busca optimizar la presentación visual de los hoteles en la plataforma, contribuyendo así a una experiencia más atractiva y completa para los usuarios.

```
for div in hotel_divs:
    try:
        hotel_link = div.select('div[data-automation="hotel-card-title"] a')
        if len(hotel_link)>0:
            hotel_link = hotel_link[0]
            hotel_url = hotel_link.get('href')
            if not hotel_url.startswith('http'):
                hotel_url = response.urljoin(hotel_url)

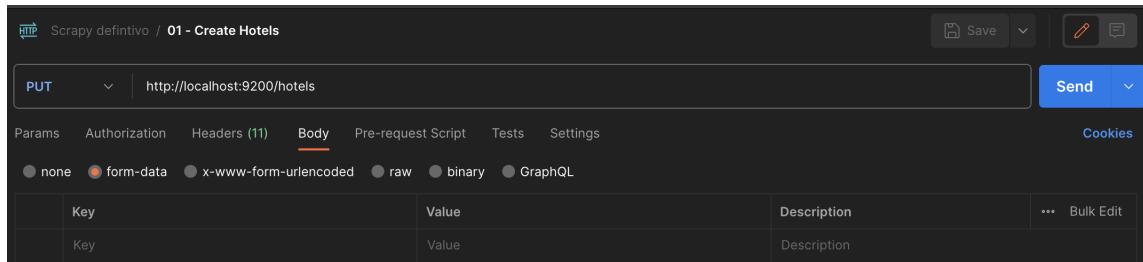
            image_link = div.find_all(class_='_C')
            if image_link:
                image_url = image_link[0].get('src')
            else:
                image_url = random.choice(self.IMAGENES)

        yield scrapy.Request(url=hotel_url, callback=self.parse_hotel_details, cb_kwargs={'comunidad': nombreComunidad, 'imageUrl': image_url})
    except:
        # Si no se ha podido obtener un campo, se muestra un warning si la variable MOSTRAR_WARNINGS es True
        if MOSTRAR_WARNINGS:
            self.logger.warning(f'No se pudo encontrar información de un hotel en {response.url}')
```

Figura 4: Extracción de las URLs de las imágenes

4.2. ElasticSearch

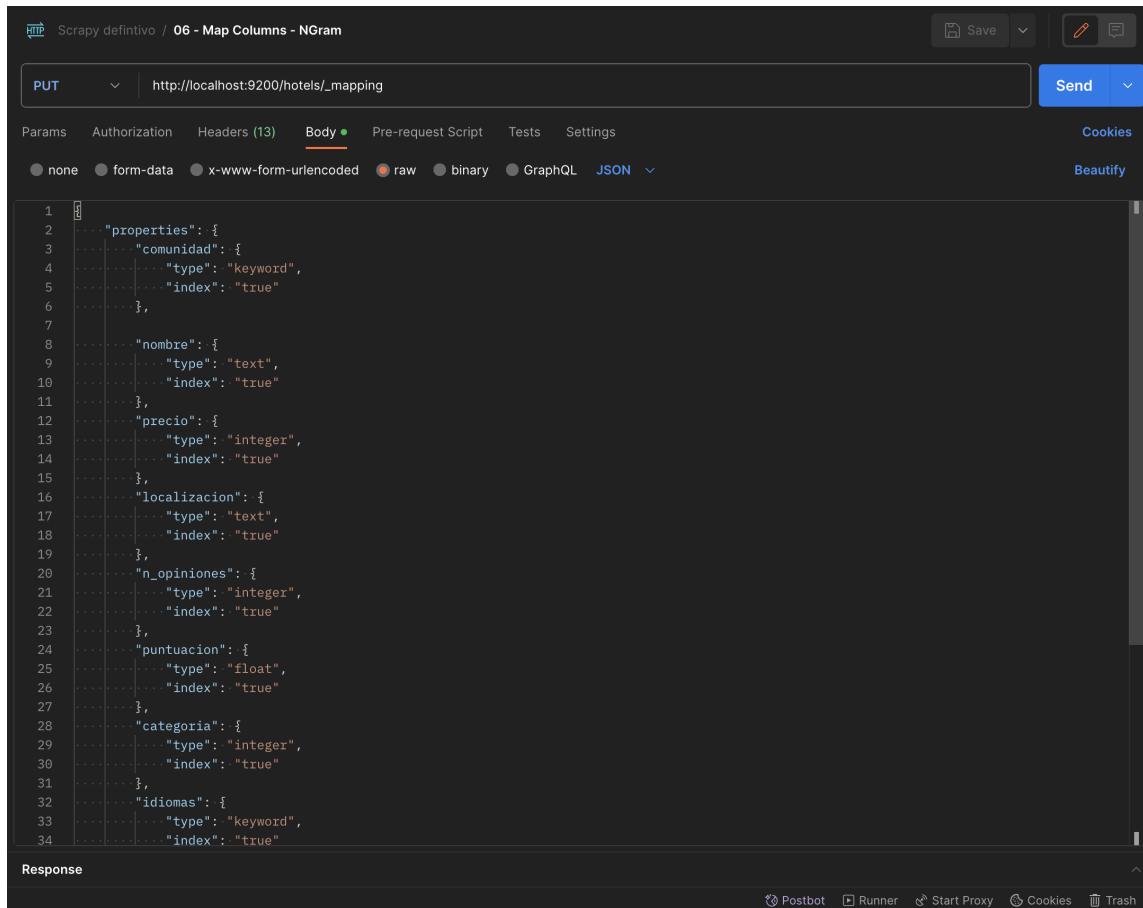
Ya con los datos extraídos, el siguiente paso fue construir un índice invertido en local con los datos extraídos anteriormente haciendo uso de la herramienta *Elasticsearch*. Para conseguirlo, se hace uso de la herramienta *Postman* para interactuar con el *API* de *Elastic* de modo que queda accesible en la dirección web *localhost:9200*. Para ello, primero es necesario, con postman, crear un índice con el nombre *hotels* con una operación *PUT* como se aprecia en la Figura 5.



The screenshot shows a Postman interface with the following details:

- Method: PUT
- URL: http://localhost:9200/hotels
- Body tab is selected.
- Form-data key: Key (Value: Value)
- Other tabs: Params, Authorization, Headers (11), Pre-request Script, Tests, Settings, Cookies.

Figura 5: Petición al API de elastic para crear un índice invertido



The screenshot shows a Postman interface with the following details:

- Method: PUT
- URL: http://localhost:9200/hotels/_mapping
- Body tab is selected, showing a JSON mapping definition:

```
1 {
2     "properties": {
3         "comunidad": {
4             "type": "keyword",
5             "index": "true"
6         },
7         "nombre": {
8             "type": "text",
9             "index": "true"
10        },
11        "precio": {
12            "type": "integer",
13            "index": "true"
14        },
15        "localizacion": {
16            "type": "text",
17            "index": "true"
18        },
19        "n_opiniones": {
20            "type": "integer",
21            "index": "true"
22        },
23        "puntuacion": {
24            "type": "float",
25            "index": "true"
26        },
27        "categoria": {
28            "type": "integer",
29            "index": "true"
30        },
31        "idiomas": {
32            "type": "keyword",
33            "index": "true"
34        }
35    }
36}
```
- Other tabs: Params, Authorization, Headers (13), Pre-request Script, Tests, Settings, Cookies, Beautify.

Figura 6: Campos que conforman el índice invertido.

A continuación, se configuran los tipos de campos que necesitamos almacenar en nuestro índice (Figura 6). Hemos configurado 4 tipos distintos de campos para almacenar en el índice invertido:

- **keyword:** Se ha empleado para los campos comunidad, servicios e idioma debido a que en el *frontend* se muestran ya las opciones y el usuario solo tiene que hacer *click* en la que necesite. No se contempla en el buscador.
- **text:** Se ha empleado para poder hacer un búsqueda por ese campo de manera que podamos obtener un ranking en base al *score* generado por Elastic de los documentos. Serían los campos correspondientes al nombre y localización. Las URLs también se guardarán como texto.
- **integer:** Se ha empleado para poder realizar consultas de rango para los campos precio, categoría y número de opiniones.
- **float:** Se ha empleado para poder realizar consultas de rango en la puntuación.

Una vez configurados los campos del índice, es necesario volcar la información recopilada al índice. Para ello, hemos conformado un archivo con extensión *JSON*, que se adjunta en el repositorio de *Git* (Figura 7).

```
{
  "create": {},
  {"comunidad": "La Rioja", "nombre": "Hotel Castillo El Collado", "precio": 176, "localizacion": "Paseo el Collado, 1, 01300 Laguardia-España", "n_opiniones": 364, "puntuacion": 4.5, "categoria": 2, "idiomas": ["Español", "Inglés", "Francés"], "servicios": ["Wifi", "Restaurante", "Hotel de no fumadores", "Habitaciones de no fumadores"], "url": "https://www.tripadvisor.es/Hotel_Review-g1069097-d267212-Reviews-Hotel_Castillo_El_Collado-Laguardia_Province_of_Alava_Basque_Country.html", "imageUrl": "https://a.loveholidays.com/horizon/public/default-hotel-img.jpg"},
  {"create": {},
  {"comunidad": "País Vasco", "nombre": "Hotel Berazadi Berri", "precio": 117, "localizacion": "Talaimendi Auzoa 723, 20800 Zarautz-España", "n_opiniones": 62, "puntuacion": 5.0, "categoria": 0, "idiomas": ["Español", "Inglés"], "servicios": ["Restaurante", "Hotel de no fumadores", "Habitaciones de no fumadores"], "url": "https://www.tripadvisor.es/Hotel_Review-g897148-d6483312-Reviews-Hotel_Berazadi_Berri-Zarautz_Province_of_Gipuzkoa_Basque_Country.html", "imageUrl": "https://mediacdn.tripadvisor.com/media/photo-s/24/dc/23/3e/berazadi-berri.jpg"},
  {"create": {},
  {"comunidad": "Madrid", "nombre": "Hotel Anaco", "precio": 70, "localizacion": "Calle Tres Cruces 3, 28013 Madrid España", "n_opiniones": 380, "puntuacion": 3.0, "categoria": 3, "idiomas": ["Español", "Inglés"], "servicios": ["Wifi", "Hotel de no fumadores", "Habitaciones de no fumadores"], "url": "https://www.tripadvisor.es/Hotel_Review-g187514-d232827-Reviews-Hotel_Anaco-Madrid.html", "imageUrl": "https://a.loveholidays.com/horizon/public/default-hotel-img.jpg"}}
}
```

Figura 7: Archivo Json con los datos extraídos.

Una vez creado el índice con los datos extraídos se prueban diferentes consultas con *Postman* y la extensión de Google Chrome *Multi Elastic Search* para comprobar que los documentos devueltos coincidan con la necesidad de información.

The screenshot shows the Elasticsearch interface with the following details:

- Header:** Elasticsearch, New, elasticsearch_adrianlopezgude (http://localhost:9200/) (7.17.4), cluster health: yellow (7 of 13)
- Search Bar:** Search: hotels (2743 docs) for documents where: must _doc.localizacion query_string Sanxenxo
- Output Results:** Number of Results: 10, Show query source
- Table:** Shows search results with columns: _score, comunidad, nombre, precio, localizacion, n_opiniones, puntuacion, categoria.

_score	comunidad	nombre	precio	localizacion	n_opiniones	puntuacion	categoria
6.3966637	Galicia	Hotel O Son Do Mar	892	Rua Madrid 32, 36969, Sanxenxo España	119	4	1
6.3966637	Galicia	Pazo Revel Hotel & Resort	113	O Rebel 15, 36990, Sanxenxo España	44	4	2
6.3966637	Galicia	Nuevo Astur	96	Lg. Gondar, 45, 36990, Sanxenxo España	388	4	3
6.0604353	Galicia	Hotel Rotillo	90	Avenida del Puerto 7, 36960, Sanxenxo España	66	4	3
6.0604353	Galicia	Hotel Vimar	45	Escalinata Gondar 40 Villalonga, 36990, Sanxenxo España	34	4.5	1
5.7577877	Galicia	Hotel Portonovo	55	Rua Alegre, 3 Benitino, 36970 Portonovo, Sanxenxo España	77	4.5	2
5.7577877	Galicia	Hotel Florida Mar	42	Avenida Emilia Pardo Bazán 3, 36969, Sanxenxo España	130	4.5	2
5.7577877	Galicia	Hotel Pífeiro	894	Lugar A Lanzada, 104 Bajo, 36990, Sanxenxo España	207	4.5	2
5.7577877	Galicia	Hotel VIDA Playa Paxariñas	2727	Playa de Paxariñas, s/n, 36970, Sanxenxo España	230	4.5	2

Figura 8: Resultados de Elastic para una query por el campo localización

4.3. Frontend

Para crear el *front-end* se hace uso de *React* con la librería *ReactiveSearch*. Así, la aplicación web permite visualizar los elementos con paginación, realizar consultas mediante un buscador, filtrar los resultados en base a distintos criterios, y ordenar el listado de hotels por precio o categoría. Para obtener este resultado se utilizaron los siguientes componentes:

- **DataSearch** : Se ha usado en este contexto para implementar una búsqueda de texto respecto al índice invertido para los campos *nombre* y *localidad* (Figura 9).

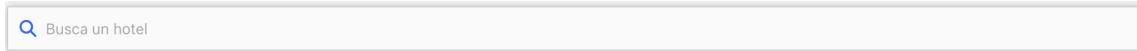


Figura 9: Componente *DataSearch*

- **DataList** : Se utiliza para filtrar resultados según las selecciones actuales de una lista de elementos de datos. En nuestro caso, se emplea para seleccionar filtros de comunidad, servicios e idioma (Figura 10).

Figura 10: Componente *MultiList*

- **RangeInput** : Crea un componente de interfaz de usuario de control deslizante de rango numérico con campos de entrada. Para este proyecto, se utiliza dicho componente para filtrar por precio (Figura 11) y número de opiniones.



Figura 11: Demostración de uso del componente *RangeInput*

De esta manera, la página web permitirá realizar búsquedas con cualquier tipo de filtro anteriormente mencionado. Así por ejemplo, se permite hacer una búsqueda compleja de la siguiente manera: Buscar los hoteles de la comunidad autónoma de Galicia, con localización en Santiago, sin un rango específico de precio, de 3 estrellas, con un 5 de valoración, que cuenten con servicio de *Wifi* y hablen portugués. Así, los resultados de aplicar todos estos filtros conjuntamente es el que se aprecia en la Figura 12.

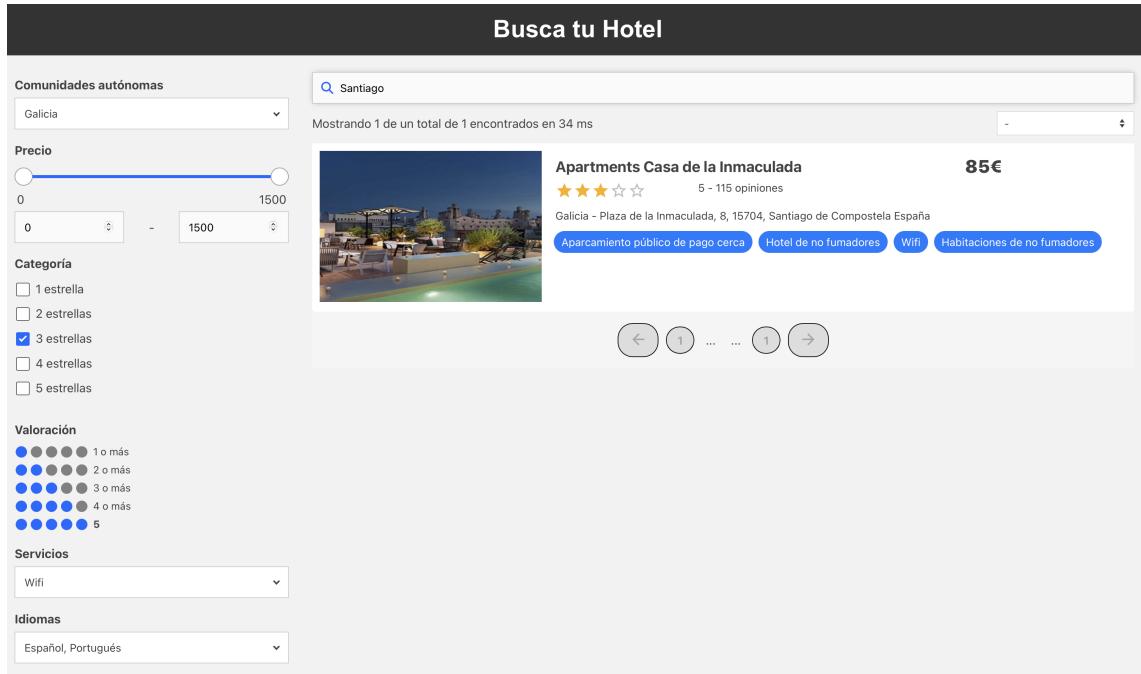


Figura 12: Resultado de todos los hoteles que cumplen los filtros descritos.

Otro ejemplo, sería la búsqueda de hoteles en la comunidad de Madrid que cueste la noche entre 500 y 1500 euros (Figura 13). Este resultado, se podrá ordenar en función de distintos parámetros, como puede ser el *score* de la búsqueda por defecto, precios ascendentes y descendentes o valoraciones ascendentes o descendentes.(Figura 14).

Busca tu Hotel

Comunidades autónomas

Precio

0

1500

500 ⬇ ⬆ 1500

Categoría

- 1 estrella
- 2 estrellas
- 3 estrellas
- 4 estrellas
- 5 estrellas

Valoración

●
●
●
●
●
●
1 o más

●
●
●
●
●
●
2 o más

●
●
●
●
●
●
3 o más

●
●
●
●
●
●
4 o más

●
●
●
●
●
●
5

Servicios

Idiomas

Mostrando 4 de un total de 4 encontrados en 35 ms

AC Palacio del Retiro, Autograph Collection 520€

★★★★★ 4.5 - 691 opiniones

Madrid - Calle Alfonso XII 14, 28014 Madrid España

Gimnasio / Sala de entrenamiento, Wifi, Sauna, Restaurante, Sauna, Hotel de no fumadores, Habitaciones de no fumadores

Hotel Único Madrid 738€

★★★★★ 5 - 1461 opiniones

Madrid - Calle Claudio Coello 67, 28001 Madrid España

Hotel de no fumadores, Habitaciones de no fumadores

Four Seasons Hotel Madrid 990€

★★★★★ 4.5 - 313 opiniones

Madrid - Calle de Sevilla 3, 28014 Madrid España

Gimnasio / Sala de entrenamiento, Wifi, Sauna, Restaurante, Sauna, Habitaciones de no fumadores

Figura 13: Listado de hoteles de Madrid con un precio de entre 500€ y 1500€.

Busca tu Hotel

Comunidades autónomas

Precio

0

1500

500 ⬇ ⬆ 1500

Categoría

- 1 estrella
- 2 estrellas
- 3 estrellas
- 4 estrellas
- 5 estrellas

Valoración

●
●
●
●
●
●
1 o más

●
●
●
●
●
●
2 o más

●
●
●
●
●
●
3 o más

●
●
●
●
●
●
4 o más

●
●
●
●
●
●
5

Servicios

Idiomas

Mostrando 4 de un total de 4 encontrados en 24 ms

Four Seasons Hotel Madrid 990€

★★★★★ 4.5 - 313 opiniones

Madrid - Calle de Sevilla 3, 28014 Madrid España

Gimnasio / Sala de entrenamiento, Wifi, Sauna, Restaurante, Sauna, Habitaciones de no fumadores

Rosewood Villa Magna 990€

★★★★★ 4.5 - 863 opiniones

Madrid - Paseo de la Castellana 22, 28046 Madrid España

Gimnasio / Sala de entrenamiento, Wifi, Sauna, Restaurante, Sauna, Habitaciones de no fumadores

Hotel Único Madrid 738€

★★★★★ 5 - 1461 opiniones

Madrid - Calle Claudio Coello 67, 28001 Madrid España

Hotel de no fumadores, Habitaciones de no fumadores

Valoración descendente
 Valoración ascendente
 Precio descendente
 Precio ascendente

Figura 14: Formas de ordenación del listado de resultados.

Finalmente, si se necesita hacer una reserva, se ha implementado un hipervínculo en el nombre de los hoteles listados, que traslada al usuario a la página original de *Tripadvisor*.