

# Ejercicio Dirigido



## Scraping web booking.com



[Booking.com](https://www.booking.com) es el sitio de reserva de hotel más grande del mundo, con más **27 millones** listados reportados en **130,000 destinos** en 227 países de todo el mundo. Su vasta mina de datos disponibles públicamente sobre hoteles y resorts hace de Booking.com un recurso valioso para los mineros de datos y las respectivas OTA para observar las estrategias de fijación de precios de sus competidores.

En este tutorial, aprenderemos a raspar **Resultados de búsqueda de hoteles de Booking.com** usando Python y BeautifulSoup.

Primero, nos familiarizaremos con la estructura HTML de la página web que queremos eliminar. A partir de ahí, extraeremos información importante como el nombre del hotel, los precios, los enlaces y otra información relevante. Para concluir este tutorial, exploraremos una solución eficiente para eliminar los datos del hotel de Booking.com y discutiremos los beneficios de eliminar datos de sitios web de reservas de hoteles u OTA.

Al final de este tutorial, podrá obtener precios y otra información de Booking.com. Y también puede usar este conocimiento para crear una API de Hotel Scraper en el futuro para comparar el precio de diferentes proveedores en múltiples plataformas.

### ¿Por qué Python para raspar Booking.com?

Python es un lenguaje multipropósito de alto rendimiento utilizado en gran medida en tareas de raspado web, generalmente respaldado por bibliotecas diseñadas específicamente para raspado.

Python también ofrece varias características como excelente adaptabilidad y escalabilidad, lo que le permite manejar grandes cantidades de datos. En general, es el lenguaje más preferido para el raspado web con una gran comunidad de soporte activo, que puede utilizar para obtener soluciones para cualquier problema.

## Comencemos a desechar Booking.com

Antes de comenzar nuestro proyecto, analicemos algunos requisitos, incluida la instalación de bibliotecas para ayudarnos a extraer datos del hotel de Booking.com.

### Requisitos

Supongo que ya ha instalado Python en su computadora. A continuación, necesitamos instalar dos bibliotecas que usaremos para eliminar los datos más adelante.

1. [Solicitudes](#) — Usando esta biblioteca, estableceremos una conexión HTTP con Booking.com.
2. [BeautifulSoup](#) — Usando esta biblioteca, analizaremos los datos HTML extraídos para recopilar la información requerida.

### Configuración

A continuación, crearemos un nuevo directorio dentro del cual crearemos nuestro archivo Python e instalaremos las bibliotecas mencionadas anteriormente.

```
mkdir booking_scraper
pip install requests
pip install beautifulsoup4
```

Es importante decidir de antemano qué datos necesita extraer de la página web. Este tutorial nos enseñará a extraer los siguientes datos del objetivo [sitio web](#):

1. Nombre
2. Enlace
3. Ubicación
4. Calificación
5. Recuento de revisión
6. Precio
7. Miniatura

The screenshot shows the Booking.com search results page. On the left, there are filters for 'Travel Sustainable properties' (842), 'Property rating' (1 star to 5 stars, Unrated), 'Fun Things To Do' (Beach, Bicycle rental, Bike tours, Massage, Fishing), and 'Cancellation Policy'. The main content area displays two hotel listings. The first listing is 'GoanFiesta Ultra Luxury Suites CALANGUTE' with a 'Limited-time Deal' badge. The second listing is 'Ronnie's Studio Apartment' with a 'Free cancellation' badge. Red boxes and lines highlight specific data points for extraction: the hotel name, location, review count, rating, price, and thumbnail image for both listings. The first listing also highlights the 'Suite with Balcony' details and the 'Breakfast included' badge.

Thumbnail	Location	Title	Review Count	Rating	Price
	Calangute	GoanFiesta Ultra Luxury Suites CALANGUTE	10 reviews	Exceptional 10	US\$92
	Saligao	Ronnie's Studio Apartment	92 reviews	Very Good 8.3	US\$73

Utilizaremos los métodos BeautifulSoup `find ( )` y `find_all ( )` dependiendo de la estructura DOM para orientar los elementos DOM y extraer sus datos. Además, tomaremos la ayuda de las herramientas para desarrolladores para encontrar la ruta CSS para localizar los elementos DOM.

## Proceso

A medida que hayamos completado la configuración, es hora de hacer una solicitud HTTP GET a la URL de destino, que será la primera y básica parte de nuestro código.

```
import requests
from bs4 import BeautifulSoup

url = "https://www.booking.com/searchresults.html?ss=Goa&lang=en-us&dest_id=4127&dest_type=region&checkin=2024-05-30&checkout=2024-06-03&group_adults=2&no_rooms=1&group_children=0&selected_currency=USD"

headers={"User-Agent":"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5042.108 Safari/537.36"}

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.content, 'html.parser')

print(response.status_code)

hotel_results = []
```

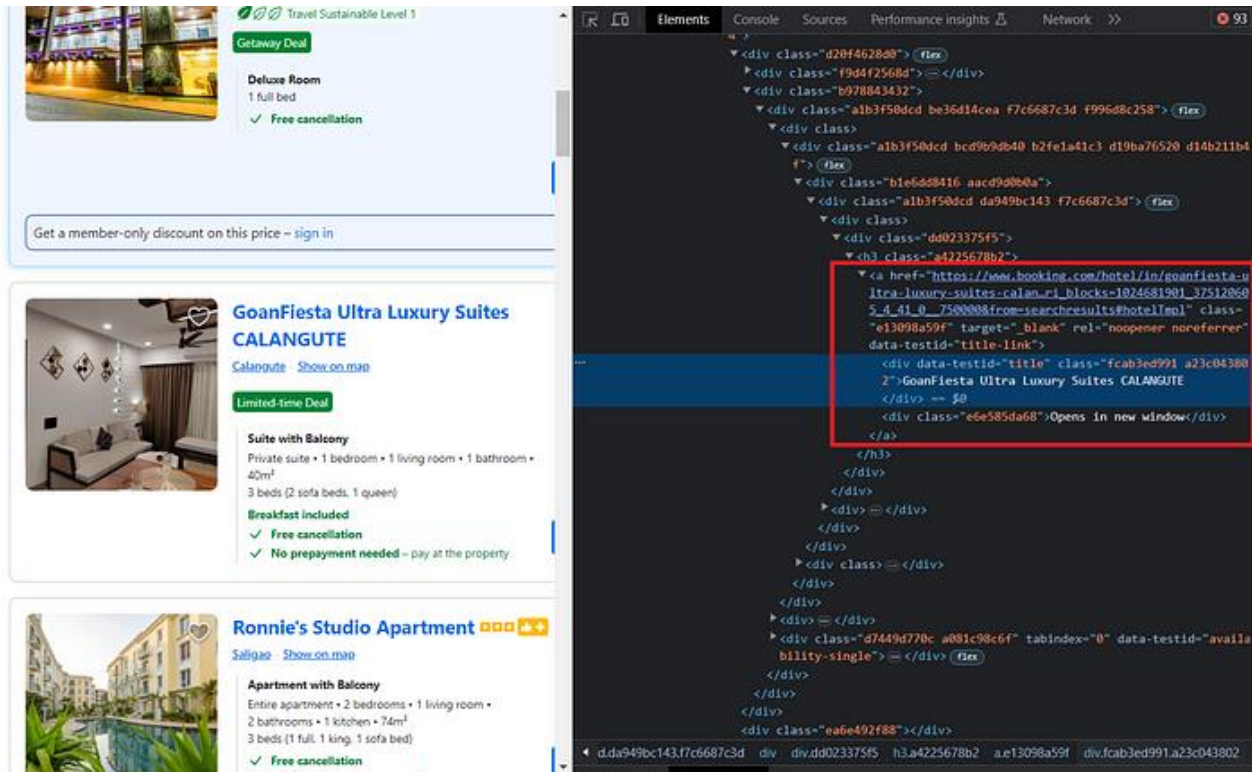
Primero, importamos las dos bibliotecas que instalamos. Luego, inicializamos la URL a la página de destino y el encabezado al Agente de usuario, lo que ayudará a nuestro bot a imitar a un usuario orgánico.

Por último, hicimos una solicitud GET a la URL de destino utilizando la biblioteca de Solicitudes y creamos una instancia de BeautifulSoup para atravesar el HTML y extraer información de él.

Esto completa la primera parte del código. Ahora, encontraremos los selectores CSS del HTML para obtener acceso a los datos.

## Extracción del nombre y enlace del hotel

Comencemos ahora extrayendo el título y el enlace de los hoteles del HTML. Apunte el mouse sobre el título y haga clic con el botón derecho, que abrirá un menú. Seleccionar **Inspeccionar** desde el menú, que abrirá las Herramientas para desarrolladores.



The image shows a side-by-side comparison of a hotel booking website and its browser's developer tools. On the left, the website displays three hotel listings: 'Deluxe Room', 'GoanFiesta Ultra Luxury Suites CALANGUTE', and 'Ronnie's Studio Apartment'. The 'GoanFiesta Ultra Luxury Suites CALANGUTE' listing is highlighted. On the right, the browser's developer tools are open, showing the 'Elements' panel. The DOM tree is expanded to the 'GoanFiesta Ultra Luxury Suites CALANGUTE' listing. A red box highlights the following HTML structure:

```
<a href="https://www.booking.com/hotel/in/goanfiesta-ultra-luxury-suites-calangute.html" data-testid="title-link">
  <div data-testid="title" class="fcab3ed991 a23c043802">GoanFiesta Ultra Luxury Suites CALANGUTE</div>
</a>
<div class="e6e585da68">Opens in new window</div>
```

En la imagen de arriba, puede ver que el nombre se encuentra debajo de la etiqueta de anclaje. La etiqueta de anclaje consiste en el enlace del hotel y se puede identificar en la estructura DOM utilizando su atributo `data-testid=title-link`. La etiqueta `div` debajo del enlace de anclaje también tiene un atributo `data-testid=title` que se puede utilizar para extraer los nombres de los hoteles. Pero, no solo los rasparemos directamente. Por simplicidad, recorreremos cada tarjeta de propiedad en la lista y extraeremos cada entidad declarada paso a paso.

Esto es lo que quise decir con la tarjeta de propiedad.

The screenshot displays a travel website interface on the left and its DOM structure in a browser's developer tools on the right. The website shows three property cards: 'GoanFiesta Ultra Luxury Suites CALANGUTE', 'Audio Apartment', and 'Lime Tree Hotel & Resort Goa'. The developer tools on the right show the DOM tree with the 'Elements' panel expanded. A red box highlights a specific DOM element: `<div data-testid="property-card" class="a826ba81c4 fa2f36ad22 afd256fc79 d08f526e0d ed11e24d01 ef9845d4b3 da89aeb942" style="--bui_box_padding--s: 4">`. This element is a child of a `<div class="bea018f16c">` container. The DOM structure shows a repeating pattern of these property cards, indicating a list of properties.

Utilizaremos `find_all()` de BS4 para orientar todas las tarjetas de propiedad.

```
for el in soup.find_all("div", {"data-testid": "property-card"}):
```

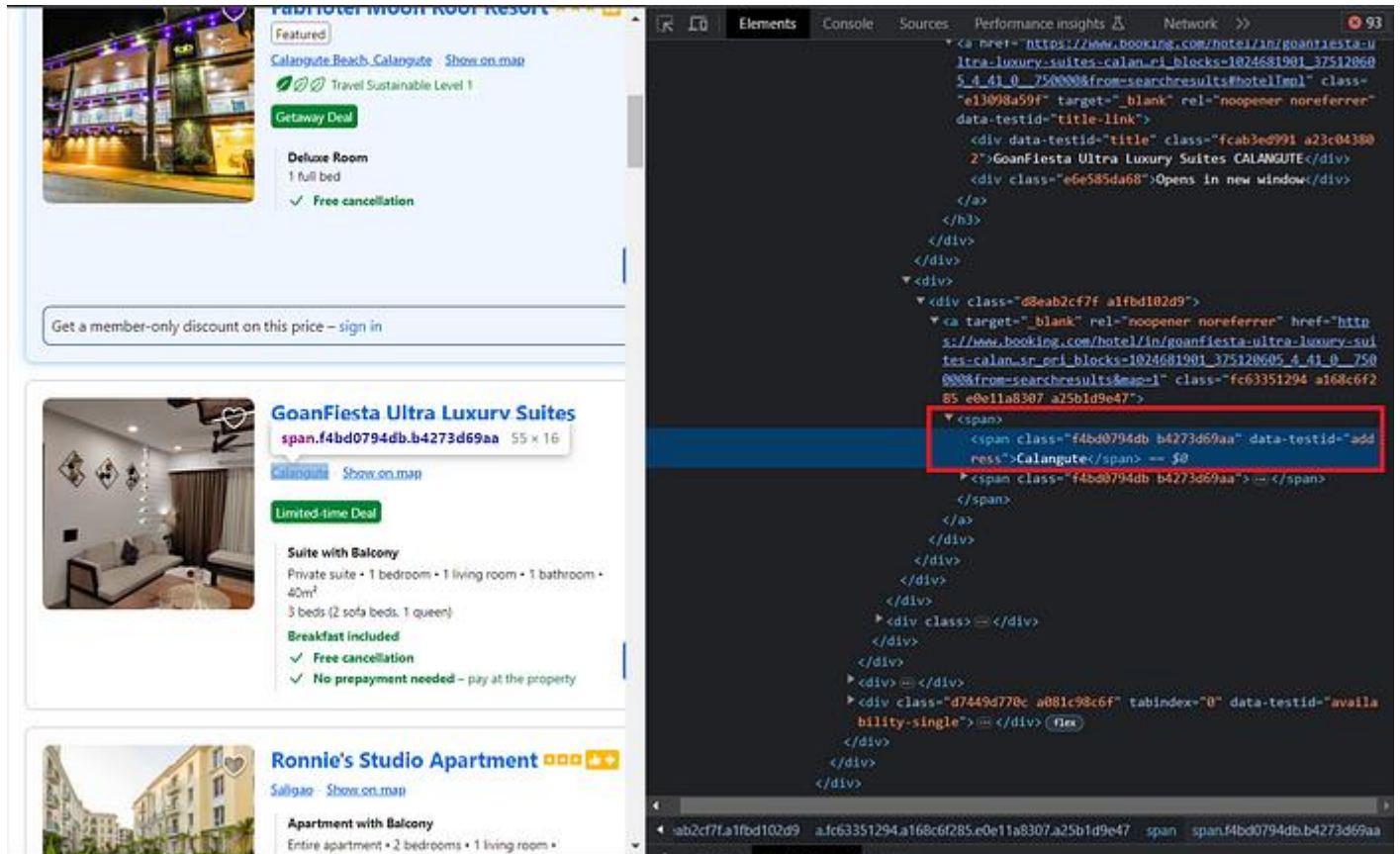
A continuación, extraeremos el nombre y el enlace de las propiedades respectivas.

```
name = el.find("div", {"data-testid": "title"}).text.strip()
link = el.find("a", {"data-testid": "title-link"})["href"]
```



## Extracción de la ubicación y precios del hotel.

Del mismo modo, podemos extraer la ubicación del hotel y los precios. Después de inspeccionar la ubicación, encontrará que también tiene un data-testid atributo igual a address.



The image shows a screenshot of a Booking.com website on the left and its corresponding HTML structure in the browser's developer tools on the right. The website displays three hotel listings: 'FabHotel Moon Roof Resort', 'GoanFiesta Ultra Luxury Suites', and 'Ronnie's Studio Apartment'. The developer tools show the HTML elements for the 'GoanFiesta Ultra Luxury Suites' listing, with a red box highlighting the data-testid="address" data-bbox="675 350 965 395">address attribute.

Agregue el siguiente código para extraer la ubicación.

```
location = el.find("span", {"data-testid": "address"}).text.strip()
```

A continuación, con el mismo proceso, extraeremos los precios.

The screenshot displays a travel website interface with three hotel listings. The first listing is a 'Deluxe Room' for \$121. The second is 'GoanFiesta Ultra Luxury Suites' for \$92. The third is 'Ronnie's Studio Apartment' for \$73. To the right, a browser's developer console is open, showing the DOM tree. A red box highlights the following HTML snippet:

```

<span data-testid="price-and-discounted-price" aria-hidden="true" class="fcab3ed991 fbdd3038c e729ed5ab6">US$92</span>

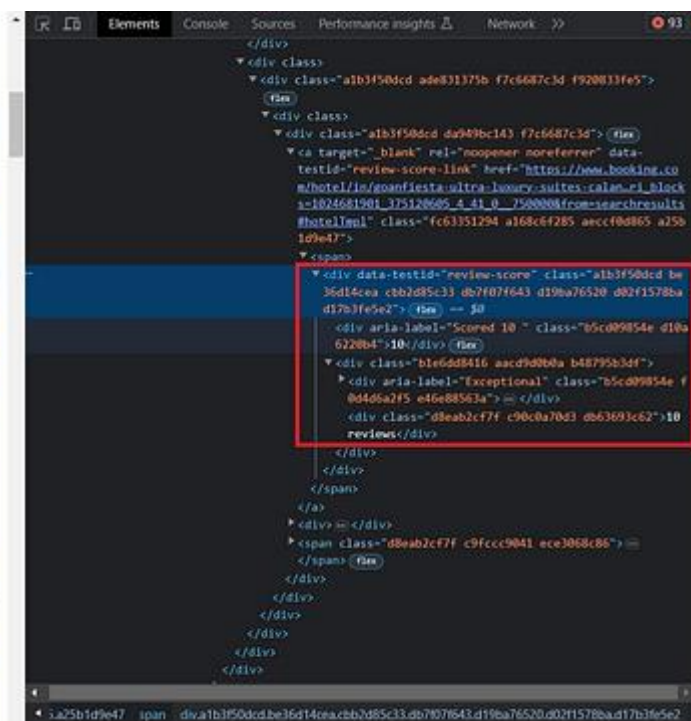
```

Seleccionaremos el precio después del descuento con el atributo `data-testid=price-and-discounted-price`. Si desea obtener otra información como precios antes del descuento y después de impuestos, puede agregarla a su código siguiendo el mismo proceso.

A continuación, agregue el siguiente código para extraer el precio.

```
pricing = el.find("span", {"data-testid": "price-and-discounted-price"}).text.strip()
```

La información de la revisión del hotel está encapsulada en la etiqueta div con el atributo data-testid=review-score.



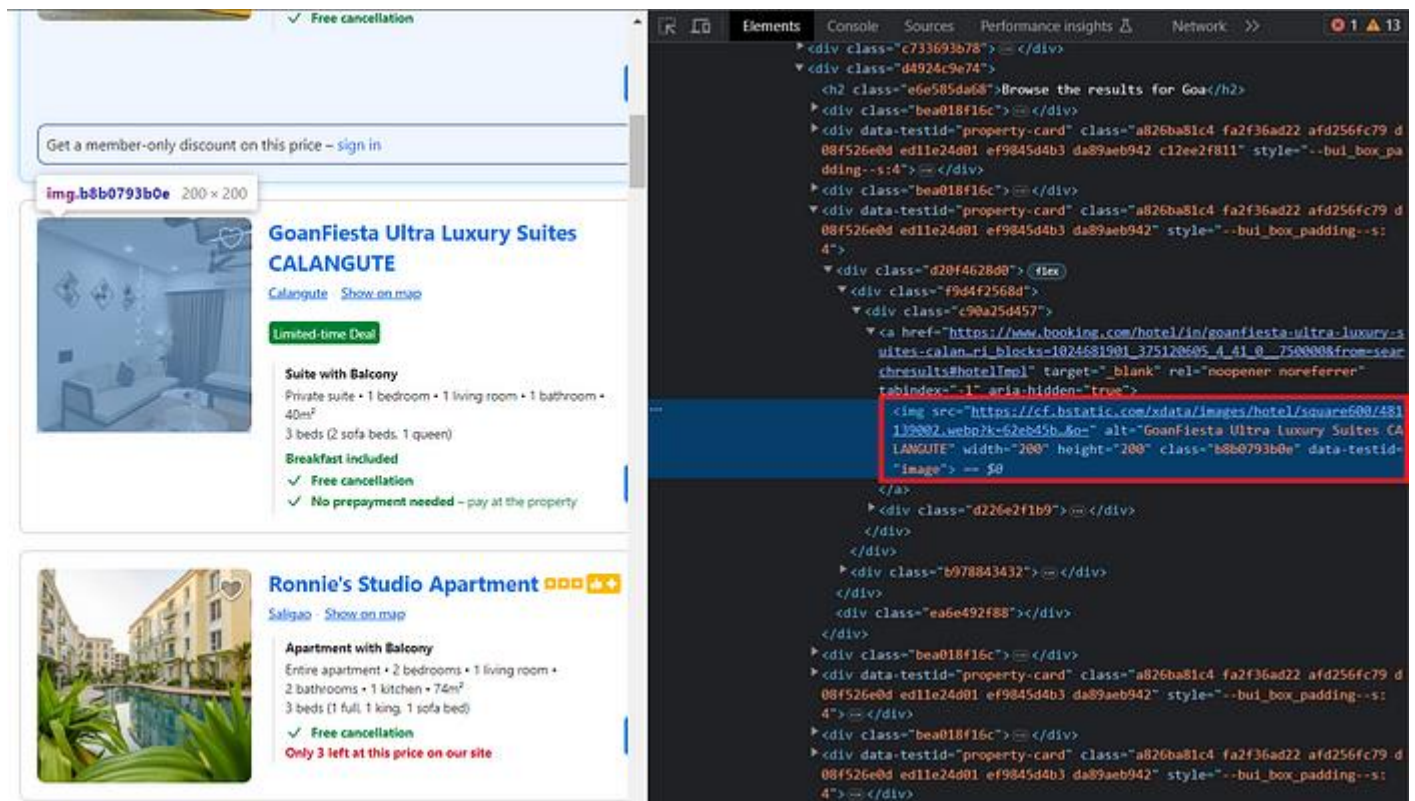
```
rating = el.find("div", {"data-testid": "review-score"}).find("div",
recursive=False).text.strip().split(" ")[0]
review_count = el.find("div", {"data-testid": "review-
score"}).text.strip().split(" ")[1]
```

Estamos extrayendo la calificación y el recuento de revisiones utilizando la función `split()`. Esto nos ayuda a obtener los resultados por separado en el formato deseado. También puede tirar de ellos individualmente apuntando específicamente al div en el que se encuentran.



## Extrayendo la miniatura del hotel

Finalmente, extraeremos la miniatura del hotel. La miniatura es fácil de encontrar y se puede ubicar dentro del img etiqueta con atributo data-testid=image.



El siguiente código le devolverá la fuente de la imagen.

```
thumbnail = el.find("img", {"data-testid": "image"})['src']
```

Hemos extraído con éxito toda la información deseada de la página de resultados de búsqueda en Booking.com.

## Código completo

Ahora, también puede obtener un conjunto adicional de información, como unidades recomendadas que consisten en servicios prestados por el Hotel, disponibilidad, y la información de la cinta que son servicios adicionales que se brindan en la miniatura del hotel. También puede cambiar la URL de acuerdo con los datos respectivos que desee.

Ahora tiene el código para raspar nombres, enlaces, precios y revisiones de las propiedades respectivas. También estoy agregando un bloque de prueba y captura para cada propiedad para que el programa no devuelva ningún error por valores vacíos.

Nuestro raspador debería verse así:

```
import requests
from bs4 import BeautifulSoup

url = "https://www.booking.com/searchresults.html?ss=Goa&lang=en-us&dest_id=4127&dest_type=region&checkin=2024-05-30&checkout=2024-06-03&group_adults=2&no_rooms=1&group_children=0&selected_currency=USD"

headers = {"User-Agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5042.108 Safari/537.36"}

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.content, 'html.parser')

print(response.status_code)

hotel_results = []

for el in soup.find_all("div", {"data-testid": "property-card"}):
    try:
        name = el.find("div", {"data-testid": "title"}).text.strip()
    except:
        name = ""

    try:
        link = el.find("a", {"data-testid": "title-link"})["href"]
    except:
        link = ""

    try:
        location = el.find("span", {"data-testid": "address"}).text.strip()
    except:
        location = ""

    try:
        pricing = el.find("span", {"data-testid": "price-and-discounted-price"}).text.strip()
    except:
        pricing = ""

    try:
        rating = el.find("div", {"data-testid": "review-score"}).find("div", recursive=False).text.strip().split(" ")[0]
    except:
        rating = ""
```

```

        try:
            review_count = el.find("div", {"data-testid": "review-
score"}).text.strip().split(" ")[1]
        except:
            review_count = ""

        try:
            thumbnail = el.find("img", {"data-testid": "image"})['src']
        except:
            thumbnail = ""

        hotel_results.append({
            "name": name,
            "link": link,
            "location": location,
            "pricing": pricing,
            "rating": rating,
            "review_count": review_count,
            "thumbnail": thumbnail,
        })

print(hotel_results)

```

## Beneficios de Scraping Booking.com

Booking.com ha crecido a una capitalización de mercado de 111 mil millones \$ desde su lanzamiento en 1997. Su tamaño gigantesco ofrece una variedad de beneficios para los mineros de datos:

### Beneficios de Scraping Booking.com

1. **Acceso a una amplia gama de datos.** — Scraping Booking.com le permite acceder a una amplia gama de datos sobre hoteles, comentarios de clientes, ubicación, disponibilidad, y mucho más que se puede utilizar para recopilar información sobre el mercado y otra información relevante.
2. **Monitoreo de precios** — Puede eliminar Booking.com para comparar el precio de los hoteles en diferentes plataformas y seleccionar la opción más económica.
3. **Reseñas de los clientes** — Raspar las reseñas de hoteles de Booking.com permite a los usuarios identificar el mejor restaurante con las opciones disponibles. Las empresas pueden hacer un análisis sentimental basado en las revisiones de los clientes y determinar las áreas de mejora.

## Preguntas frecuentes

### Q1. ¿Puedo extraer datos de Booking.com?

Los datos del hotel en Booking.com están disponibles públicamente, y el desecho de los datos disponibles públicamente no es ilegal. Pero también es importante raspar el sitio web a un ritmo más lento y evitar escalar, ya que puede resultar en una sobrecarga del servidor del sitio web.

### Q2. ¿Cómo puedo raspar Booking.com sin ser bloqueado?

Puede raspar Booking.com sin bloquearse mediante el uso **Serpdog's [API de desguace web](#)**, que gira millones de proxies en su backend permitiendo a sus usuarios obtener datos de manera fluida y eficiente.

## Conclusión

El desguace de datos del hotel crecerá a medida que aumente el tamaño y la capitalización de mercado de las OTA y otros competidores con el aumento de la industria hotelera. Esta puede ser una gran oportunidad para los desarrolladores que desean ganar dinero creando un proyecto que obtenga datos de hoteles en tiempo real de diferentes plataformas como Expedia, MakeMyTrip, y más para la comparación de precios y otros fines relevantes.