Práctica 1 (25% nota final)

INTEGRANTES: Gerard Alcalde y Guillem Rochina

NOVIEMBRE 2022

1. **Contexto.** Explicar en qué contexto se ha recolectado la información. Explicar por qué el sitio web elegido proporciona dicha información. Indicar la dirección del sitio web.

   En el presente proyecto, nuestro equipo ha convenido llevar a cabo un proyecto de scraping del metabuscador de reservas de alojamiento conocido como **Booking.com.**

   Dicha plataforma proporciona información y permite hacer reservas en su plataforma en función de los criterios/preferencias que el usuario introduce en su buscador, siendo el número de reseñas positivas, la proximidad al destino indicado por el usuario y la disponibilidad en las fechas determinadas algunos de los criterios que más influencia tienen en la posición del ranking (explicado en el apartado de inspiración) entre los resultados de la búsqueda. Es decir, un alojamiento con más reseñas positivas, mejor nota y más próximo al lugar indicado por el usuario ocupará una posición en las primeras páginas del buscador.

   No obstante, no solo dichos factores influyen en la posición que ocupan cada oferta de alojamiento. ¿Qué elementos son relevantes? ¿Existen algunas características que pueden ser mejoradas para escalar posiciones en los rankings? Estas son preguntas que todo manager de hotel se ha realizado alguna vez en su carrera, y a su vez nos ofrece un contexto sobre el que trabajar en el presente proyecto.

   En concreto, la dirección del sitio web donde nuestro web scraper inicia su proceso es https://www.booking.com

2. **Título.** Definir un título que sea descriptivo para el dataset.

   Ofertas de alojamiento vacacional en las principales ciudades españolas: Principales indicadores y características.

3. **Descripción del dataset.** Desarrollar una descripción breve del conjunto de datos que se ha extraído. Es necesario que esta descripción tenga sentido con el título elegido.

Tal y como se observa en el título, el dataset presenta los datos e indicadores más relevantes de cada uno de los hoteles encontrados en función de determinados criterios de búsqueda (ciudad, fecha de check-in, fecha de check-out, número de adultos, niños y habitaciones). En este dataset en concreto presentamos datos para distintas fechas, ciudades, número de adultos, niños y habitaciones a fin de tener una muestra más amplia e informativa que la que nos daría una búsqueda con tan sólo unos parámetros fijos.

Dado que el enunciado de la práctica así lo indica, se ha optado por no realizar una limpieza del dataset resultante. Por tanto, encontramos una serie considerable de elementos que dan pie a un proyecto de limpieza bastante elaborado, por ejemplo, nos encontramos con algunas columnas conformadas por listas o diccionarios que a su vez presentan valores dentro de estos, así como datos faltantes en algunos registros del dataset que pueden ser objeto de “inputación” o eliminación en caso de ser oportuno.

Finalmente, cabe destacar que el formato escogido para el fichero resultantes es un csv, pues facilita su tratamiento y compartición.

4. **Representación gráfica.** Dibujar un esquema o diagrama que identifique el dataset visualmente y el proyecto elegido

Figura 1: Representación gráfica del proyecto escogido

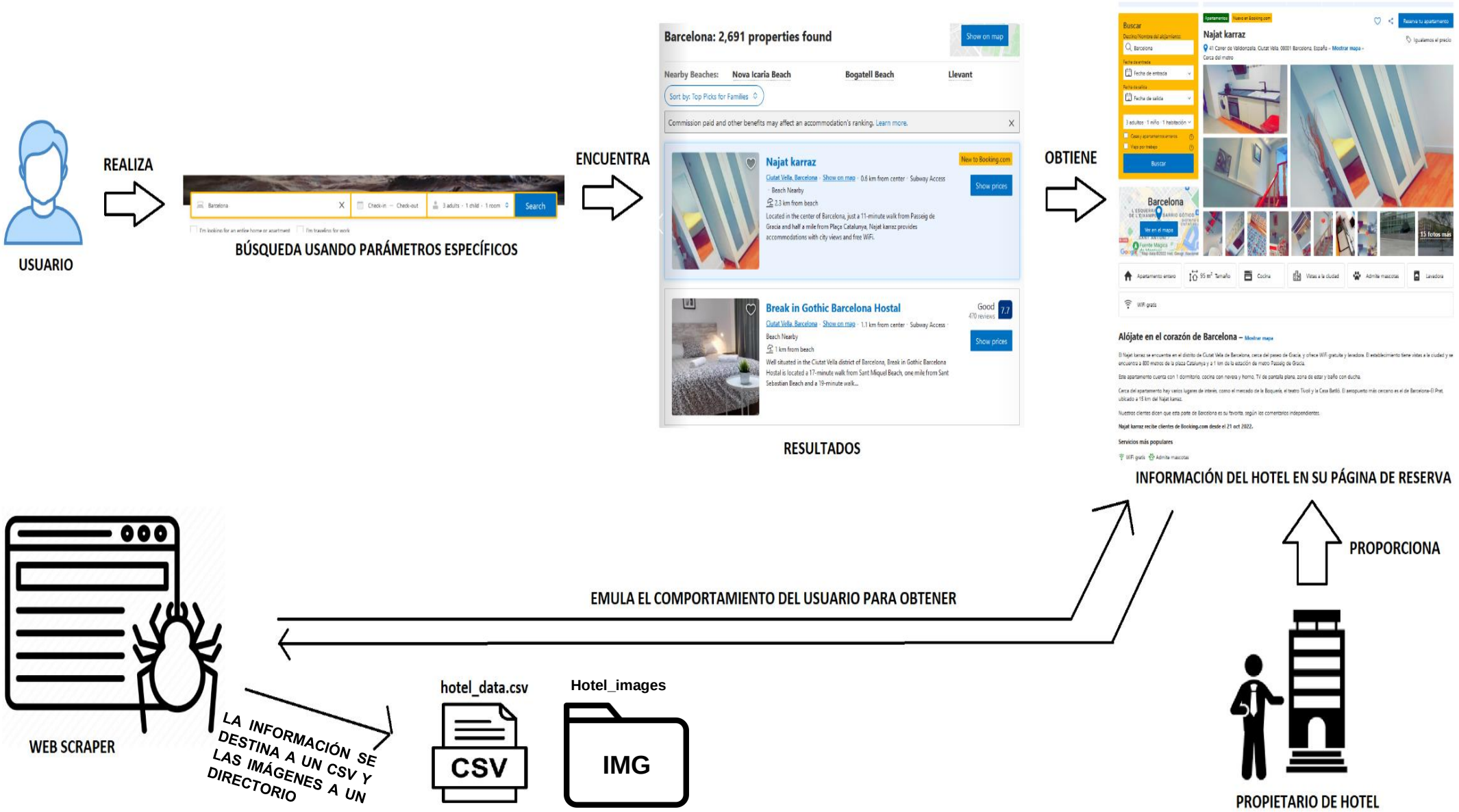
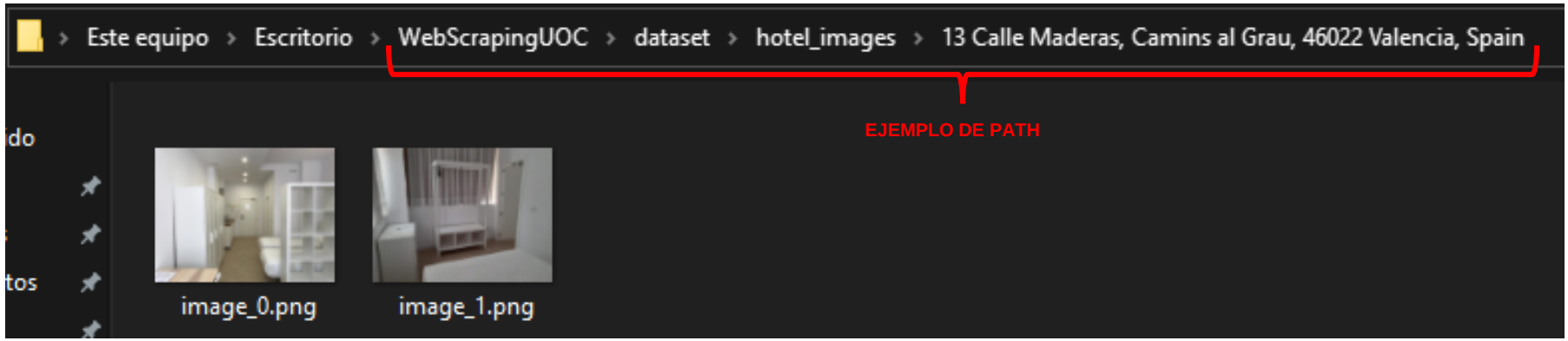


Figura 2: Representación gráfica del dataset resultante

	name	city	check-in	adults	children	check-out	num_rooms	address	hotel_coordinates	hotel_score	hotel_scores	hotel_description	features	room_data	page_count	current_page	in_page_count	search_date
632	Islas Canarias - Best Apartments Valencia	Valencia	12-December-2022	2	3	16-December-2022	2	Calle Islas Canarias, 189, Camins al Grau, 460...	39.46448780,-0.34885200	7.7	{'': '', 'Staff': '8.6', 'Facilities': '7.8', ...	["You're eligible for a Genius discount at Isl...	['Whole apartment', 'Size', 'Kitchen', 'View',...	{'541223401_285190958_5_0_0': {'room_name': 'T...	25	2	0	2022-11-18
130	Acta CITY47	Barcelona	12-December-2022	2	3	16-December-2022	2	Nicaragua, 47, Sants-Montjuïc, 08029 Barcelona...	41.38308780,2.14120746	8.2	{'': '', 'Staff': '9.0', 'Facilities': '8.3', ...	['Acta CITY47 is 5 minutes walk from Barcelon...	['Free WiFi', 'Bathtub', 'Air conditioning', '...	{'9004506_138954531_0_2_0': {'room_name': 'Tri...	130	5	25	2022-11-18
730	CASA ESPECTACULAR DE DISEÑO CON JARDIN INTERIOR	Valencia	12-December-2022	2	3	16-December-2022	2	Carrer Isabel la Catolica, 12 (Mislata) VALENC...	39.47250590,-0.41545640	10.0	{'': '', 'Staff': '10', 'Facilities': '10', 'C...	['Located in Valencia, one kilometer from Biop...	['Whole house', 'Size', 'Kitchen', 'Garden', '...	{'888411801_358874169_5_0_0': {'room_name': 'T...	123	5	18	2022-11-18
533	Carretas Apartments	Madrid	12-December-2022	5	3	16-December-2022	3	Carretas 8, Madrid City Center, 28012 Madrid, ...	40.41564180,-3.70329750	8.8	{'': '', 'Staff': '8.8', 'Facilities': '8.9', ...	["You're eligible for a Genius discount at Car...	['Apartments', 'Kitchen', 'City view', 'Washin...	{'53681707_329634573_5_0_0': {'room_name': 'Tw...	3	1	3	2022-11-18
726	Venecia Plaza Centro	Valencia	12-December-2022	2	3	16-December-2022	2	Plaza del Ayuntamiento, 3, Ciutat Vella, 46002...	39.47083946,-0.37695631	8.9	{'': '', 'Staff': '9.2', 'Facilities': '8.8', ...	["You're eligible for a Genius discount at Ven...	[]	{'9288314_269099915_0_2_0': {'room_name': 'Eco...	119	5	14	2022-11-18
351	Hotel Concordia Barcelona	Barcelona	12-December-2022	5	3	16-December-2022	3	Paral·lel, 115, Sants-Montjuïc, 08004 Barcelon...	41.37461874,2.16093779	8.1	{'': '', 'Staff': '8.9', 'Facilities': '8.1', ...	['Hotel Concordia Barcelona is located 801 m f...	[]	{'9118504_158812557_0_2_6047313952768': {'room...	85	4	7	2022-11-18



5. **Contenido.** Explicar los campos que incluye el dataset y el periodo de tiempo de los datos.

A continuación, se realiza una breve descripción de cada una de las columnas:

Columna	Tipo	Descripción
Name	**Str**	Nombre del hotel
City	**Str**	Nombre de la ciudad donde se realiza la búsqueda (parámetro de búsqueda inputado)
Check-in	**Str**	Fecha de entrada al hotel (parámetro de búsqueda inputado)
Check-out	**Str**	Fecha de salida del hotel (parámetro de búsqueda inputado)
Adults	**Int**	Número de adultos para los que se realiza la reserva (parámetro de búsqueda inputado)
Children	**Int**	Número de niños para los que se realiza la reserva (parámetro de búsqueda inputado)
Num_rooms	**Int**	Número de habitaciones reservadas (parámetro de búsqueda inputado)
Address	**Str**	Dirección postal del hotel, incluyendo calle, número, zona de la ciudad, código postal, ciudad y país
Hotel_coordinates	**Str**	Coordenadas del hotel formadas por longitud y latitud separadas por una coma
Hotel_score	**Int**	Nota general que recibe el hotel por los usuarios
Hotel_scores	**Dict**	Diccionario que recoge la puntuación del hotel para distintas categorías. Las categorías incluidas son: Staff, Facilities, Cleanliness, Comfort, Value for money, Location, Free WiFi.
Hotel_description	**List**	Descripción aportada por el propietario del hotel. Cada elemento de la lista es una línea de la descripción en formato string
Features	**List**	Lista de las distintas características del hotel
Room_data	**Dict**	Diccionario para cada una de las distintas habitaciones disponibles en el hotel los siguientes atributos: ○ **room_name** (string): Nombre de la habitación. ○ **room_price** (string): Precio de la habitación. ○ **room_capacity** (string): Texto con la capacidad de la habitación indicando número de adultos y de niños. ○ **room_options** (list of strings): Lista de opciones del hotel asociadas a la habitación (cancelación gratuita, tipo de pensión, etc)

		○ **room_facilities** (list of strings): Lista de características de la habitación. ○ **room_bed_type** (string): Tipo de camas en la habitación.
Page_count	**Int**	Posición en la que ha aparecido el hotel al buscar en booking
Current_page	**Int**	Página en la que ha aparecido el hotel al buscar en booking.
In_Page_count	**Int**	Número de la posición en la que se encuentra el hotel dentro de la página de resultados
Search_date	**Str**	Fecha en la que se realizó la búsqueda

El objetivo de este dataset es obtener las características que hacen que un hotel esté mejor posicionado que otro. El posicionamiento de los hoteles suele ser algo estático y por ello con una sola lectura de la página web puede ser suficiente. No obstante, se podría realizar un scrapping con diferentes fechas que nos permitiese obtener hoteles que puedan no aparecer en la búsqueda debido a que no tengan disponibilidad en las fechas seleccionadas.

El dataset abarca las búsquedas para los meses de diciembre, marzo y junio de la temporada 2022-2023 para añadir un componente temporal al análisis posterior de los datos.

6. **Propietario.** Presentar al propietario del conjunto de datos. Es necesario incluir citas de análisis anteriores o, en caso de no haberlas, justificar esta búsqueda con análisis similares. Justificar qué pasos se han seguido para actuar de acuerdo a los principios éticos y legales en el contexto del proyecto.

Como se ha presentado en el apartado de contexto, los datos han sido recopilados de la web de Booking.com estableciendo en su motor de búsqueda distintos parámetros, por lo que el propietario del conjunto de datos es **Booking Holdings,** una compañía norteamericana propietaria de gran número de metabuscadores como Priceline, Agoda, Kaya, Rentalcars, y por supuesto, el que nos interesa particularmente, Booking.com

Como metabuscador, esta plataforma se encarga desde hace más de 20 años de agregar tarifas de habitaciones/apartamentos en un mismo sitio web a fin de ofrecer facilidades a los individuos en su búsqueda por un sitio donde alojarse durante sus viajes. Actualmente, miles de viajeros ocasionales o aquellos que viajan por negocios

hacen uso de esta plataforma para reservar hoteles, hostales o incluso barcos, por lo que esta web supone una herramienta muy potente tanto para huéspedes como para las cadenas hoteles o propietarios que deciden hacer uso de sus propiedades como alojamientos vacacionales.

Scrapear la web de Booking.com no es una tarea que no se haya hecho nunca. De hecho, existen muchas páginas que ofrecen sus servicios de scraping para esta pagina como por ejemplo ***Bright Data*** *(*https://cutt.ly/IMDMFbx*).* Por otro lado, también encontramos repositorios de Github con la misma finalidad, como por ejemplo el repositorio de ***Zoran Padovski***: https://github.com/ZoranPandovski/BookingScraper o el de ***Ennio Campagna***: https://github.com/HexNio/booking_scraper. Cabe destacar el artículo publicado por **Mariia Potapova** en el portal ***Scraping* Ant** llamado *"Better real estate decicions with Bookijng.com data scraping"* (disponible en el siguiente link: https://scrapingant.com/blog/booking-data-scraping). En este, Marriia destaca como los datos scrapeados desde Booking.com pueden servir para responder a cuestiones como qué precio determinar para tus servicios de alojamiento, mejorar el *customer service* o simplemente hacer análisis del mercado y tus competidores. Asimismo, existen muchos posts donde se ofrecen recomendaciones sobre cómo mejorar, como en la web de ***Hostaway*** (https://www.hostaway.com/how-to-rank-1-on-booking-com/). Sin embargo, no hemos sido capaces de encontrar ningún proyecto que junte las dos vertientes, scrapear los datos y analizarlos a fin de obtener recomendaciones sobre como posicionarse mejor en Booking.com. Por tanto, esta búsqueda se justifica como la herramienta que poner fin al "gap" entre estas dos actividades (el scrapeo web de Booking y las recomendaciones de páginas web especializadas en posicionamiento web de hoteles).

Finalmente, con el fin de seguir con los principios éticos y legales previamente se investigó el archivo: https://www.booking.com/robots.txt y la página de términos y condiciones de Booking.com: https://www.booking.com/content/terms.es.html, cuyo contenido de interés en nuestro caso se encuentra en el apartado A14.2 (véase Figura 3). En este se indica que no se puede scrapear ningún dato con fines comerciales sin permiso escrito. No obstante, dado que nuestro proyecto tan solo persigue fines académicos y no comerciales, consideramos que legalmente estamos cubiertos en este aspecto y, por lo tanto, no existe necesidad de solicitar un permiso implícito para realizar el scrapeo objeto del presente proyecto.

Como medida adicional, se añade un apartado de Disclaimer en el documento del README del repositorio, en el que se especifica la finalidad académica del proyecto y se advierte que no se debe hacer uso comercial de los datos obtenidos, eximiendo de responsabilidad alguna a los integrantes del equipo en el caso de que un usuario externo decida hacer algún uso no permitido del dataset o el código incluido en el repositorio.

Asimismo, el apartado A14.3 indica la posibilidad de bloqueo a cualquier usuario que muestre indicios de hacer web scraping (usar software para recopilar precios o realizar una cantidad irrazonable de búsquedas). A fin de cubrirnos en este aspecto, se han incluido varios retardos a lo largo del código para no sobrecargar la página y se ha diseñado el proyecto de forma que las búsquedas que realizar el programa sean lo más fidedignas posibles a como actuaria un usuario real, además de cambiar el header del user agent.

Aunque no podemos afirmarlo con un 100% de confianza, sospechamos que las medidas adoptadas han sido suficientes, pues a pesar de dejar correr el script durante horas y para distintas búsquedas, en ningún momento nos hemos con bloqueo alguno por parte de Booking.com

Figura 3: Términos y condiciones de Booking.com

A14. Derechos de propiedad intelectual

1. A menos que se indique lo contrario, todos los derechos de nuestra Plataforma (tecnología, contenido, marcas comerciales, apariencia, etc.) son propiedad de Booking.com (o sus licenciantes) y, al utilizar nuestra Plataforma, aceptas hacerlo solo para el propósito previsto y respetando las condiciones establecidas a continuación en los párrafos A14.2 y A14.3.

2. No puedes monitorear, copiar, raspar/rastrear, descargar, reproducir o usar cualquier cosa en nuestra Plataforma para ningún propósito comercial sin el permiso por escrito de Booking.com o sus licenciantes.

3. Vigilamos de cerca cada visita a nuestra Plataforma y bloquearemos a cualquier persona (y a cualquier sistema automatizado) que sospechemos que:

- realiza una cantidad irrazonable de búsquedas;
- usa cualquier dispositivo o software para recopilar precios u otra información;
- hace cualquier cosa que suponga una tensión indebida en nuestra Plataforma.

4. Al cargar cualquier imagen en nuestra Plataforma (con un comentario, por ejemplo), estás confirmando que cumples con nuestros criterios y que:

- es veraz (no has alterado la imagen, por ejemplo, ni has subido una imagen de un alojamiento diferente);
- no contiene virus;
- puedes compartirla con nosotros;

7. **Inspiración.** Explicar por qué es interesante este conjunto de datos y qué preguntas se pretenden responder. Es necesario comparar con los análisis anteriores presentados en el apartado 6.

Aparecer en los primeros resultados de una búsqueda de un usuario por internet se ha convertido en los últimos años en una de las preocupaciones principales de cualquier negocio que busca hacerse un hueco en el mundo digital. Este fenómeno explica la proliferación de herramientas como el SEO y el SEM, mediante las cuales se intenta innovar (más allá del pago de tarifas para aparecer en las primeras páginas de búsqueda) a fin de lograr un mejor "posicionamiento" en buscadores como Google o Bing. El negocio hotelero no es una excepción en este caso.

No obstante, el uso de dichas herramientas no resulta útil si nos enfrentamos a un buscador dentro de una página web, como el caso que nos ocupa con Booking.com. ¿Qué hacer entonces? ¿Cómo mejorar el posicionamiento dentro de la web?

En la web del metabuscador de alojamientos encontramos que define a dichas posiciones entre los resultados como "ranking", descrito por ellos mismos como el orden en el que se muestran los alojamientos disponibles en los resultados de búsqueda. En su blog "Booking.com Partner Hub" indican que los resultados son ordenados según la relevancia en base a las preferencias particulares de cada "cliente", incluyendo en este caso también las "dinámicas del mercado", el "rendimiento del alojamiento", etc. Variable que poca información revela al interesado en que sus hoteles/alojamientos aparezcan en posiciones más "atractivas" para este.

En concreto, en el mismo post analizado se destaca la relevancia de revisar regularmente las "condiciones flexibles", los comentarios, los precios externos, las promociones, la puntuación de la página del alojamiento… a fin de lograr escalar puestos en la clasificación.

Dado el impacto que puede tener en el negocio de un hotel el aparecer en posiciones más inmediatas del buscador, el presente proyecto pretende ser un paso inicial (el de recopilación de datos) que finalice realizando un análisis inferencial que destaque que factores tienen un mayor peso en el posicionamiento de un alojamiento en **Booking.com.**

Una vez más, cabe destacar que no hemos encontrado un análisis que combine el scrapeo web con el análisis que se propone en el presente documento y, por tanto, no podemos ofrecer una comparación con algún análisis anterior.

Sin embargo, la herramienta desarrollada puede tener múltiples funcionalidades más allá de la presentada. A saber, puede ser utilizada por un usuario que busque el mejor momento en términos económicos para realizar un viaje a una ciudad determinada (si lo cruza con datos relacionados con el precio de vuelos, viajes en barco o el coste de cualquier otro modo de transporte) o, por otro lado, un hotel puede utilizarla para hacer llevar a cabo una comparación de sus servicios con respecto a la competencia (en este caso, dejamos a consideración del propio usuario determinar si los datos están siendo o no utilizados con fines comerciales, pues, como se ha visto en el apartado 6 está prohibido por las políticas de Booking.com.)

8. **Licencia.** Seleccionar una licencia adecuada para el dataset resultante y justificar el motivo de su elección.

A la hora de escoger una licencia para el dataset resultante se ha optado por una licencia ampliamente conocida como es la Creative Commons, ya que esta permitirá conocer los requisitos establecidos de una forma sencilla.

Creative Commons recoge distintos tipos de licencia basada en distintos parámetros como son la atribución, el uso a realizar de los datos o como se deben tratar estos. En este sentido se ha optado por una licencia: **Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**, esta licencia estipula:

- Se pueden compartir los datos en cualquier medio o formato
- Se permite modificar los datos o trabajar sobre ellos, siempre y cuando se indique que esta modificación se ha llevado a cabo.
- Se debe indicar quien es el autor de los datos.
- Si se modifican o se trabaja sobre los datos, estos se deberán distribuir bajo la misma licencia.

Se ha optado por esta licencia ya que permite que cualquiera pueda acceder al dataset que hemos generado y utilizarlo con la finalidad deseada, pero con el compromiso de compartir sus modificaciones de forma pública y gratuita promoviendo así que aumente el contenido de calidad y gratuito en la red.

9. Código.

El código desarrollado en esta práctica se encuentra en el enlace del Github: https://github.com/guiruha/WebScrapingUOC

Concretamente, el código, desarrollado principalmente en Python, se encuentra en la carpeta ***src*** tal como indican las instrucciones de práctica, agrupados en dos ficheros:

- **Main.py:** script que se encarga de ejecutar el proceso diseñado en la clase ***BookingSpider.*** El código dentro de dicho script es bastante simple, pues se encarga de importar la clase, instanciarla y ejecutar las funciones principales.
- **BookingScraper.**py: script que concentra todo el desarrollo e implementación de la clase ***BookingSpider,*** el cual se encarga de realizar las búsquedas dentro del portal de Booking.com con los parámetros introducidos al ejecutar el script previamente descrito, recorrer toda la lista de resultados, obtener la información de la página de reserva de cada hotel encontrado y almacenarlo todo en un fichero csv y una carpeta donde guardar las imágenes.

Como añadido, se ha incluido un script en lenguaje ***bash, sampleDataCollection.sh****,* para automatizar el proceso de ejecución del ***Main.py*** con varios parámetros de búsqueda distintos.

A continuación, se realiza una descripción a grandes rasgos del código elaborado, así como de los principales problemas encontrados a lo largo del desarrollo del proyecto. No se adjuntan imágenes del código en el presente documento debido a la longitud del código desarrollado y el límite de páginas impuesto en el presente pdf.

El código se estructura en varios grandes bloques:

1. **Definición de la clase y setup del webdriver:** En las primeras 100 líneas del script ***BookingScraper.py*** se define el init de la clase ***BookingSpider*** con los atributos correspondientes: parámetros de búsqueda, la url de Booking.com, el driver (creado con la función descrita a continuación) y la lista donde se recopilará en un principio la información de los hoteles. La función **set_selenium_driver** configura el webdriver que se utiliza para navegar por las webs. En ella introducimos como opción el user agent del buscador de Firefox. Finalmente, el

webdriver se encarga de entrar en la primera página de Booking.com y hacer los preparativos necesarios para dar paso a la introducción de los parámetros de búsqueda.

2. **Introducción de los parámetros de búsqueda y recorrido de los resultados:** Tras el set-up y entrar en la página principal de Booking.com, el código ha sido diseñado para que extraiga los parámetros de búsqueda por defecto (con la función **get_selection_numbers**) de la página e introduzca los "inputados" en la ejecución del script (**introduce_selection_numbers**), siguiendo dos formas alternativas que se explicaran en la subsección de problemas encontrados. Una vez avanzando a la página de resultados, el script se encarga de seleccionar los días de reserva que se han determinado al correr el código, gestionando la manipulación del calendario como haría un usuario real (con las funciones **set_month_year** y **set_date**). Con los resultados ya filtrados por la fecha y demás parámetros, el script obtiene todos los "bloques" donde se sitúa la información de los hoteles en la página actual, el número de la página actual y de la última página de los resultados. Finalmente, una vez se han scrapeado los datos de la página actual, se clica en el botón de "Siguiente" para obtener nuevos resultados. Salvo la última parte descrita, todas las funcionalidades se incluyen en una función más general llamada **search_listings**.

3. **Extracción de la información y las imágenes de los hoteles encontrados:** Dentro de cada una de las páginas que recorre el script, este hace click en cada bloque obtenido con la función **open_hotel,** accediendo así a la información completa de cada hotel. Una vez se ha accedido al hotel, la función **get_hotel_data** y **save_photos_random** se encargan de extraer toda la información que consideramos relevante, así como las fotos de algunos de los hoteles obtenidos (de forma aleatoria, pues no queremos subir un archivo demasiado pesado a Github, aunque siempre cabe la posibilidad de modificarlo para que descargue todas las fotos de todos los hoteles).

4. **Guardar los datos en el directorio correspondiente:** Una vez obtenidos todos los datos de la página de resultados actual, la función **data_to_csv** se encarga de crear un directorio (en caso de que no exista ya) e introducir todos los datos dentro de un archivo csv llamado **hotels_data.csv**. En el caso de las imágenes estas se guardan en el mismo directorio cada vez que se visita una página de hotel, no la página de resultados.

5. **Función main:** En la parte final del código se define una función main que se encargue de ejecutar todos los procesos previamente definidos a fin de lograr un código más limpio y fácil de debugguear.

Problemas o retos encontrados a lo largo del proceso de creación de los scripts:

1. **Diferentes landing page en la página principal de Booking.com:** Durante las ejecuciones de prueba de los scripts hemos encontrado una serie de errores en los que no se detectaba el xpath definido. Tras investigar el código detrás de la página web (con “inspeccionar elemento”) encontramos que Booking.com utiliza varias alternativas para sus páginas web, codificadas de forma distinta (suponemos que son utilizadas para hacer A/B test relacionados con cambios de interfaz de usuario y otras pruebas de diseño frontend)Por tanto, para acceder a los distintos elementos se debe utilizar xpaths diferentes. Como solución se implementaron dentro de las distintas funciones definidas en el subapartado 2: “**Introducción de los parámetros de búsqueda y recorrido de los resultados”** dos vertientes distintas de código para gestionar correctamente las dos landing page alternativas.

2. **El parámetro de nº de niños no ha podido ser introducido en una de las landing page alternativas:** Una de las páginas alternativas de Booking.com presenta un comportamiento inusual a la hora de introducir el número de niños. Si hay niños entre los parámetros de búsqueda, Booking.com te solicita que introduzcas la edad de cada uno de ellos. En el caso de la primera página alternativa esto se gestiona sin mayor problema. No obstante, en el otro tipo de página una vez introducida la edad del primer niño, esta bloquea el desplegable y no puede volver a clicarse para abrirlo (o lo cierra automáticamente). En este caso no hemos encontrado solución, por lo que en ocasiones los datos scrapeados no llegan a estar completamente filtrados por todos los parámetros (aunque tras comprobar los resultados, el hecho de tener niños no influye demasiado en los hoteles encontrados y la información aportada).

3. **El warning de cookies oculta (“obscures”) los botones de las páginas:** ·En muchas ocasiones el mensaje para aceptar las cookies aparece de repente y oculta el resto de botones que se deben pulsar para avanzar a la página de resultados o activar los desplegables como el calendario. Como solución se buscó

el xpath del botón de aceptar cookies, el código espera 10 segundos (tiempo razonable para que aparezca el mensaje de política de cookies) y lo pulsa.

Figura 4: Código encargado de realizar el proceso de aceptar cookies

```
try:
    #Wait until warning appears and accept cookies
    wait = WebDriverWait(driver, 5).until(EC.presence_of_element_located((By.ID, 'onetrust-accept-btn-handler')))
    driver.find_element(By.ID, "onetrust-accept-btn-handler").click()
# IN case the cookies policy button is not found the script waits until it can introduce the search criteria in the web page searcher
# In particular id "ss" references the place where the name of the city is introduced.
except:
    try:
        wait = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, 'ss')))
    except:
        wait = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, '__bui-c3765876-1')))
```

4. **El pop-up de inicio de sesión con la cuenta de Google oculta y dificulta la introducción de los parámetros de búsqueda:** Al igual que el warning de cookies, en ciertas ocasiones el pop up de iniciarse sesión de Google impide que el scraper encuentre los xpaths correspondientes. Se proporciona una solución similar a la anterior, el scraper espera hasta encontrar el banner de Google y lo cierra.

Figura 5: Código encargado de realizar el proceso de cerrar el Google banner

```
#Close google banner
try:
    #Change to the pop-up window
    driver.switch_to.frame(driver.find_element(by="xpath", value="//div[@id='google-one-tap-wrapper']//iframe"))
    driver.find_element(by="xpath", value='//div[@id="close"]').click()

    #Return to the original window
    driver.switch_to.default_content()

    time.sleep(2)

except Exception as e:
    print(e)

print("\n [{}] | INFO | Driver prepared\n\n".format(str(datetime.datetime.now())[:-7]))

return driver
```

5. **El script en ocasiones no encuentra los xpath a pesar de estar presentes:** De forma aleatoria el script lanza errores porque no encuentra un xpath a pesar de que este se encuentra en la página que está scrapeando. Como solución se han introducido varios try and except a fin de que busque varios elementos distintos del xpath de forma independiente. Gracias a este método de “fuerza bruta” se ha

minimizado el número de errores (aunque todavía aparece alguno de vez en cuando, sobre todo si se deja correr el script durante horas y con varias búsquedas distintas).

Figura 6: Ejemplo de try and except en serie

```
try:
    hotel_name = self.driver.find_element(by="xpath", value="//h2[contains(@class, ' pp-header__title']").text
except:
    try:
        hotel_name = self.driver.find_element(by = "xpath", value = "//h2[contains(@class, 'd2fee87262 pp-header__title')]").text
    except:
        try:
            hotel_name = self.driver.find_element(by="xpath", value='//h2[@class="d2fee87262 pp-header__title"]').text
        except:
            try:
                hotel_name = self.driver.find_element(by="xpath", value="//div[@data-capla-component='b-property-web-property-page/PropertyHeaderName']").text
            except:
                hotel_name = f"Unkown{count}"

# Hotel address
try:
    hotel_address = self.driver.find_element(by="xpath", value='//*[contains(@class, "hp_address_subtitle")]').text
except:
    try:
        hotel_address = self.driver.find_element(by = "xpath", value = "//*[contains(@class, 'hp_address_subtitle js-hp_address_subtitle jq_tooltip')]").text
    except:
        try:
            hotel_address = self.driver.find_element(by = "xpath", value = "//*[contains(@data-node_tt_id, 'location_score_tooltip')]").text
        except:
            hotel_address = f"Unknown{count}"

# Hotel score
try:
    hotel_score = self.driver.find_element(by="xpath", value="//div[@class='page-section js-k2-hp--block k2-hp--featured_reviews']//div[@class='b5cd09854e d10a6220b4']").text
except:
    try:
        hotel_score = self.driver.find_element(by="xpath", value='//*[@class="b5cd09854e d10a6220b4"]').text
    except:
        try:
            hotel_score = self.driver.find_element(by="xpath", value="//div[@class='b5cd09854e d10a6220b4']").text
        except:
            hotel_score = -1

# Hotel coordinates
try:
    hotel_coordinates = self.driver.find_element(by="xpath", value='//a[@id="hotel_address"]').get_attribute(
        "data-atlas-latlng")
except:
    hotel_coordinates = "NA"
```

Cabe destacar que se han añadido algunos logs a lo largo del código para comprobar en que parte de la “run” se encuentra el código en cada momento, así como retardos en algunos puntos clave a fin de no sobrecargar la página de Booking.com. Las secciones presentadas líneas arriba son fácilmente identificables en el script ***BookingScraper.py,*** ya que han sido destacadas con comentarios para que se detecten más rápidamente.

10. **Dataset.** Publicar el dataset obtenido en formato CSV en Zenodo, incluyendo una breve descripción. Obtener y adjuntar el enlace del DOI del dataset (https://doi.org/…). El dataset también deberá incluirse en la carpeta **/dataset** del repositorio.

El ejemplo de dataset resultante del proceso previamente descrito se encuentra de nuevo el repositorio de Github nombrado. Se incluye tanto el csv obtenido como varias carpetas de ejemplo con imágenes de algunos hoteles scrapeados.

El dataset está disponible en Zenodo a través del siguiente enlace del DOI: https://doi.org/10.5281/zenodo.7337369

11. **Vídeo**. Realizar un breve vídeo explicativo de la práctica **(máximo 10 minutos)**, que deberá contar con la participación de los dos integrantes del grupo. En el vídeo se deberá realizar una presentación del proyecto, destacando los puntos más relevantes, tanto de las respuestas a los apartados como del código utilizado para extraer los datos. Indicar el enlace del vídeo (https://drive.google.com/…), que deberá ubicarse en el Google Drive de la UOC.

El video se encuentra en el siguiente link:

https://drive.google.com/drive/folders/1erPPaez9yOk8HkLQLRkV_2L1SLRTWHi2?usp=share_link

12. **Recursos.**

Lawson, R. (2015). *Web Scraping with Python. Packt Publishing Ltd.*

Mitchel, R. (2015). *Web Scraping with Python: Collecting Data from the Modern Web.* O'Reilly Media, Inc.

Calvo, M. & Subirats, L. (2019). *Web Scraping.* Editorial UOC.

Contribuciones	Firma
Investigación previa	Gerard Alcalde, Guillem Rochina
Redacción de las respuestas	Gerard Alcalde, Guillem Rochina
Desarrollo del código	Gerard Alcalde, Guillem Rochina
Participación en el vídeo	Gerard Alcalde, Guillem Rochina