



# Scraper

DEVELOPING A SCRAPER THAT PUBLISHES DATA THROUGH AN  
API

| DESARROLLO DE APLICACIONES PARA CIENCIA DE DATOS |  
| 2º | GCID | ESCUELA DE INGENIERÍA INFORMÁTICA | ULP GC |

# Resumen

Este trabajo se centra en el uso de Java para realizar dos funciones, la primera sería realizar un “scraping” de una página de opiniones acerca de hoteles (en este caso se ha elegido a *booking.com*), mientras que la segunda función sería la creación de una API Rest que sea capaz de mostrar información variada sobre un hotel (ubicación, servicios, comentarios y valoraciones). Para la primera función se ha usado principalmente la librería de *Jsoup*, mientras que para la segunda se ha usado *Spark*.

En cuanto al funcionamiento del programa, este está dividido en tres paquetes:

- El paquete “api”, que contiene la clase “WebService” dedicada a la creación del *web service* encargado de solicitar la información del hotel deseado a la API haciendo uso de *Spark*. Además, contiene una interfaz que se encarga de empezar dicho *web service*.

- El paquete “model”, que contiene las clases “Comment”, “Hotel” y “Location”, cada una manejando objetos de su propia clase mediante constructores y *getters*.

- El paquete “scraper”, que contiene tanto el *main* del proyecto como su controlador. En el *main* solamente debes colocar la *url* a la página de bookin de la zona que deseas investigar (en mi caso, he elegido la zona de Puerto de Mogán), asegurandose que al final de la *url* se añade “&offset=”.

# Índice

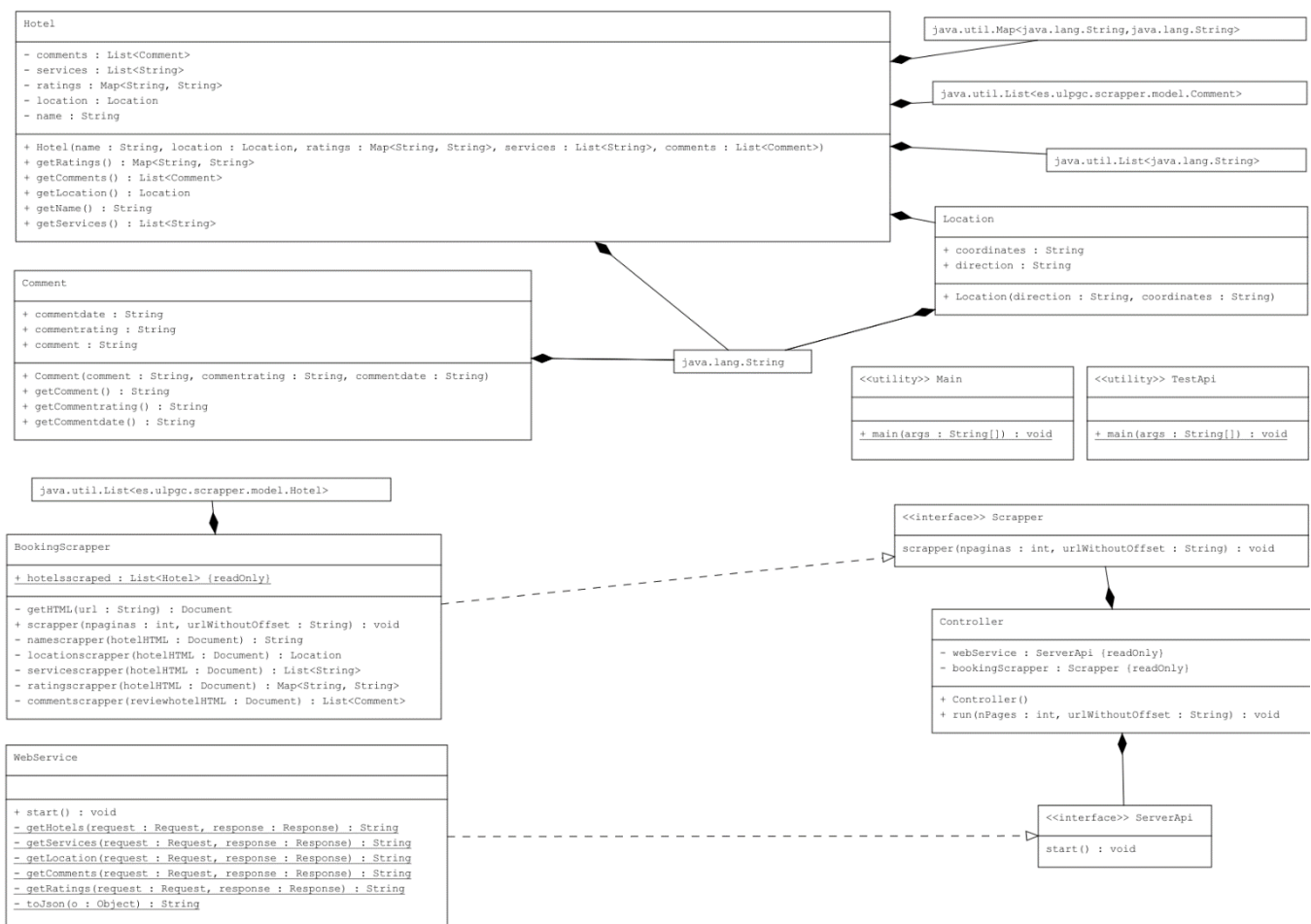
· Resumen.....	1
· Recursos Utilizados.....	3
· Diseño.....	3
· Conclusiones.....	4
· Líneas Futuras.....	4
· Bibliografía.....	5

# Recursos Utilizados

Para la realización de este trabajo se ha usado principalmente el entorno de desarrollo de *IntelliJ IDEA Community Edition 2022.2.1* junto a *Talend API Tester*, mientras que para su documentación se ha usado el software de *Microsoft Word 2019*. Finalmete para subir el programa se ha hecho uso de *GitHub*.

## Diseño

A lo largo de este trabajo se ha tenido en mente el principio de “Responsabilidad Única”, donde una clase solo debe tener una única responsabilidad y solo debería tener una razon para cambiar. Además, se ha hecho uso del principio de inversión de la dependencia al usar interfaces.



# Conclusiones

Este trabajo me ha ayudado a comprender mejor el funcionamiento de una API Rest, y como implementar un Webservice, al igual que, me ha permitido entender mejor como se ha de estructurar un proyecto para mejorar su comprensión tanto para mi como para otros. Tras los errores cometidos en el primer trabajo he podido mejorar mi forma de enfocar este tipo de trabajos, para que en un futuro no vuelva a ocurrir.

# Líneas Futuras

Como yo lo veo, el futuro de este producto va enfocado a los propios hoteles. Los cuales, usando este programa, podrán saber que es lo que verdaderamente opinan sus clientes. Incluso, en un futuro, se podrían predecir que cambios tienen que hacer los hoteles para aumentar tanto la satisfacción de sus clientes, como sus beneficios. Todo esto, podría verse implementado en una web dedicada a ello, en la que por una subscripción mensual, los hoteles podrán ver todo esto.

# Bibliografía

- <https://stackoverflow.com/>
- <https://www.booking.com/>

Autor: Gabriel Felipe Bernal Pinto

Fecha: 13/01/2023

Version: 1.0