

**TRABAJO DE INVESTIGACIÓN:
APLICACIONES OFIMÁTICAS PARA DISEÑO
UML**

Entornos de desarrollo

Sabrina Ojea Chapelet

05/03/2021

Índice de contenidos

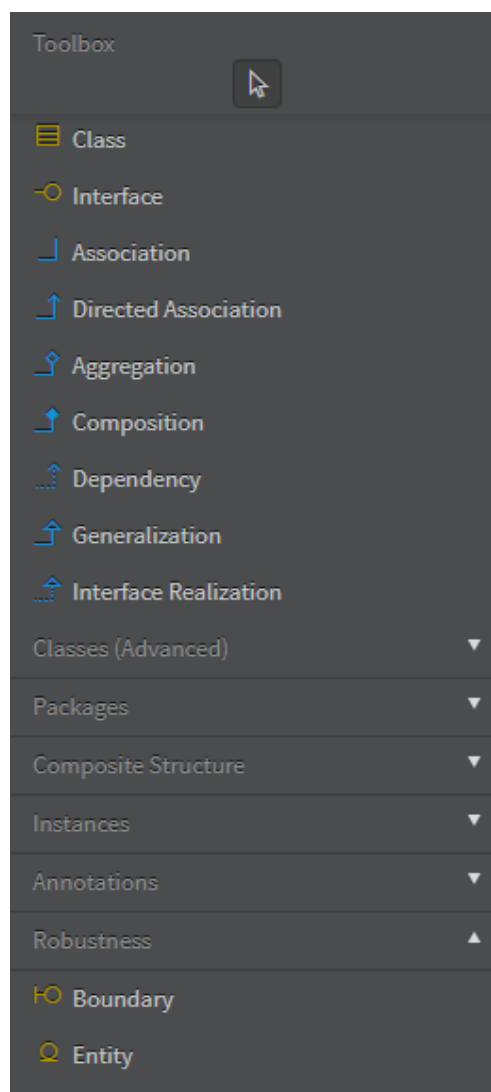
Introducción	3
StarUML	3
UMLet	5
Conclusión	7

1. Introducción

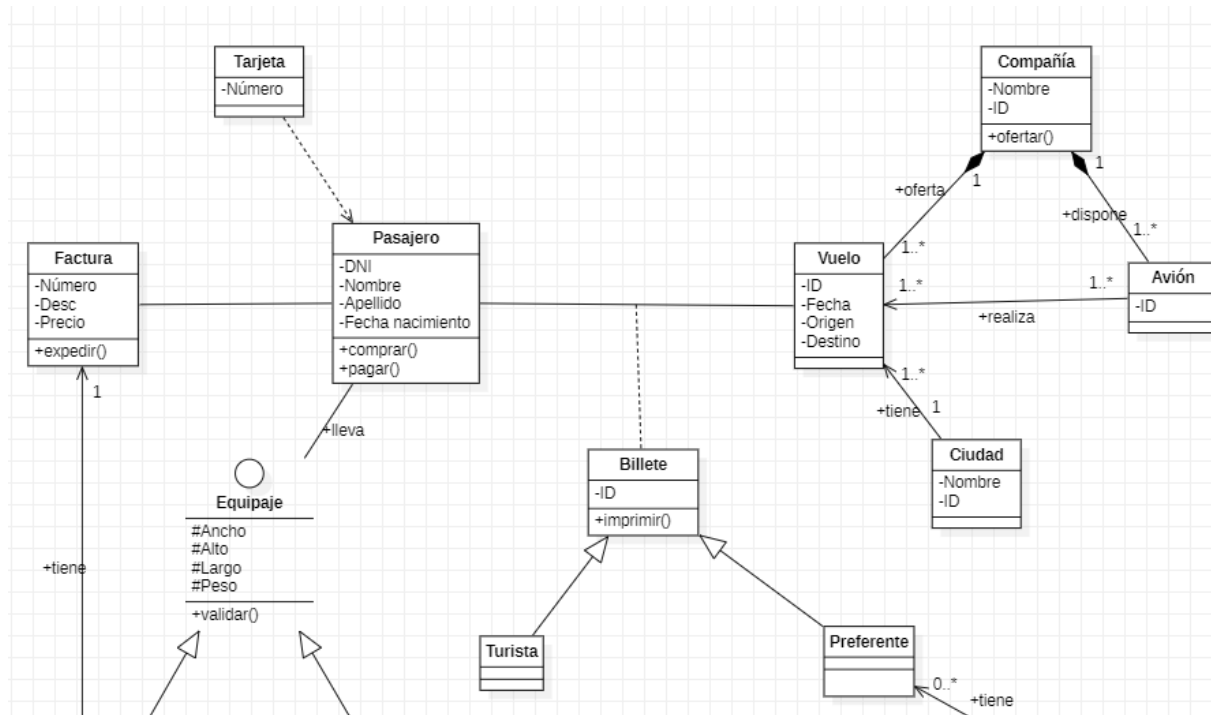
Para realizar el trabajo se analizaron y probaron diferentes opciones, las cuales se procederá a exponer en los siguientes apartados. A modo de introducción, cabe mencionar que es de gran ayuda para la realización de nuestro trabajo tener tan fácil acceso a diferentes herramientas y, aunque aquí se presentarán varias opciones, la variedad es tan amplia que no pueden recogerse todas en un trabajo de estas características.

2. StarUML

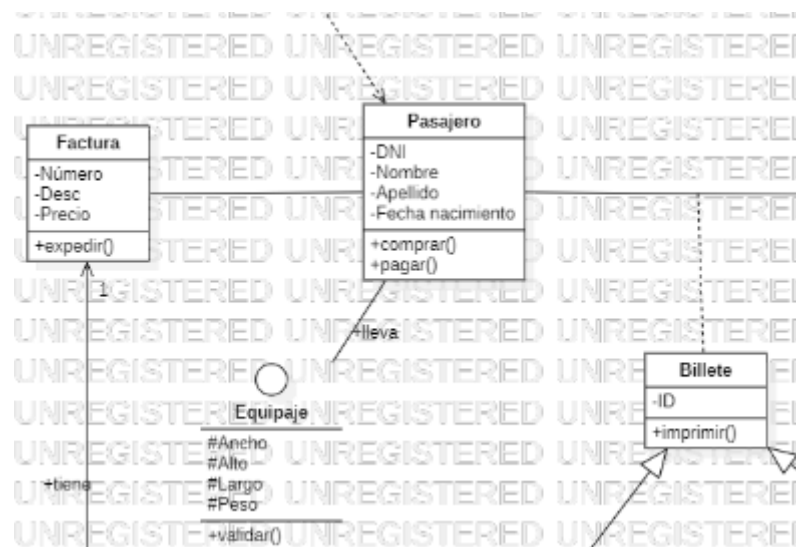
Una aplicación muy correcta y completa, tanto para la realización de diagramas de clases como de diagramas de uso. Presenta una interfaz muy intuitiva y amigable, lo que ayuda en gran manera, como se muestra en la siguiente imagen:



Todo esto nos permite realizar un trabajo muy adecuado y con una presentación bastante profesional sin excesivas complicaciones.



Esta herramienta puede ser descargada muy fácilmente y en un primer momento no necesita de pagos ni licencias. No obstante si se quiere exportar el documento creado para compartirlo con personas que quizás no tengan StarUML instalado y, por tanto, no puedan abrir un archivo con extensión .mdj nos encontraremos con el problema de documentos con marca de agua, ofreciendo, por tanto, una imagen muy poco profesional.

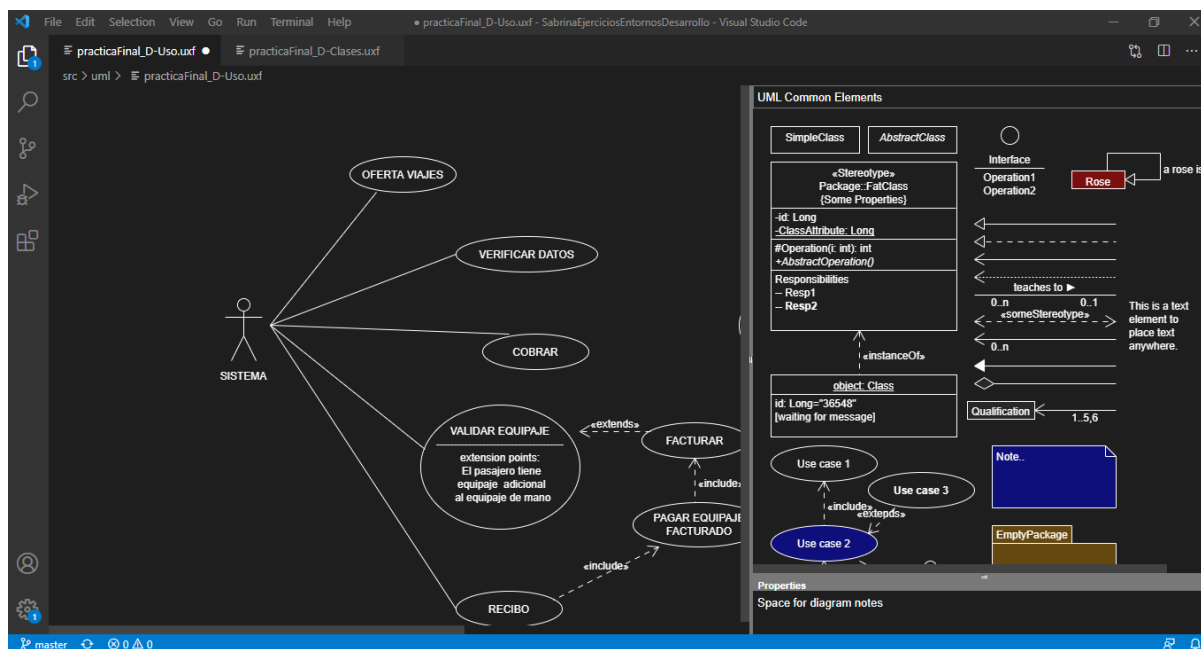


A pesar de sus destacables prestaciones, tras un periodo de prueba me decidí a seguir buscando otras herramientas para la realización del mismo trabajo, dado que no me convencía su utilización para diseñar diagramas de uso, quizás por encontrarlo menos intuitivo.

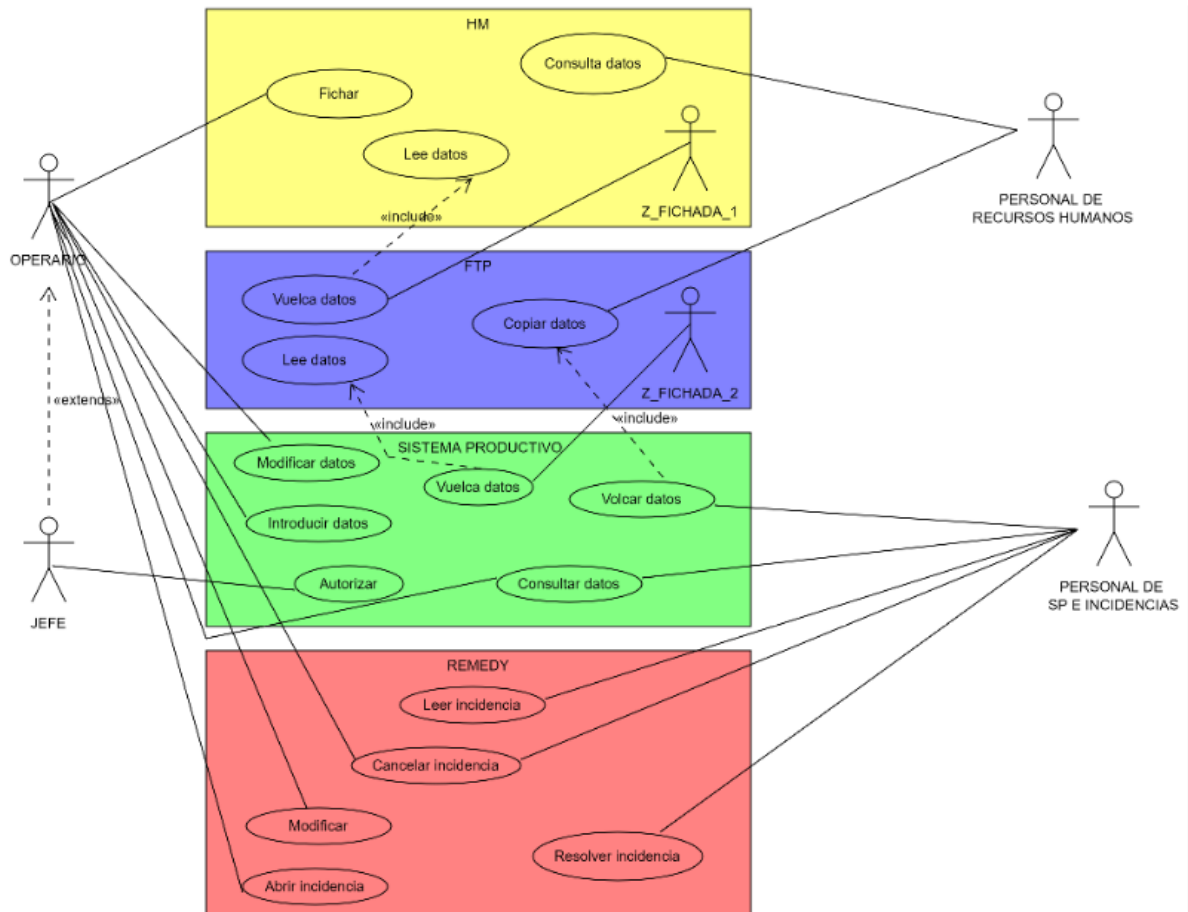
3. UMLet

Se trata de una herramienta para diseño y creación de diagramas de clases y de uso de código abierto. Presenta una interfaz muy simple, visualmente atractiva y de fácil utilización.

Por otra parte, entre sus principales ventajas destaca la posibilidad de ser instalada como extensión en Visual Studio Code (una de las herramientas que, personalmente, más utilizo), lo cual supone una enorme ventaja, pues desde el mismo programa tenemos acceso a todo lo que necesitamos, incluso podemos subir los diagramas a nuestros repositorios en Git con tan solo un par de clics.



Además de lo anteriormente mencionado y a diferencia de StarUML, esta herramienta sí que permite exportar los diagramas creados a formatos tan conocidos como .png para poder compartirlos fácilmente (sin marcas de agua y permitiendo detalles como la inclusión de colores), como se muestra a continuación:



4. Conclusión

Una vez expuestos brevemente las diferentes herramientas probadas para mejorar nuestro rendimiento a la hora de diseñar y crear diagramas tanto de clases como de uso, hemos de decir que no se tiene una clara preferencia por ninguna de ellas. Ambas opciones ofrecen sin duda buenas prestaciones pero ninguna gana en todos los campos y, por ello, se ha optado por la utilización de las dos, pero con diferentes objetivos. Como se mencionó en puntos anteriores, StarUML presenta una interfaz muy sencilla e intuitiva, sobre todo para la creación de diagramas de clases (fácil incorporación de cuadros de clases, tipos de relaciones, cardinalidades, etc.) no tanto así para diagramas de uso. Es por ello, que, en este apartado, se ha optado por la utilización del plugin de UMLet en Visual Studio Code, el cual ofrece un entorno muy amigable para la creación de diagramas de uso (tan simple como arrastrar elementos a la plantilla de trabajo) y por el contrario, parece menos preparado para el diseño de diagramas de clase (bastante tedioso a la hora de colocar flechas, relaciones y cardinalidades).