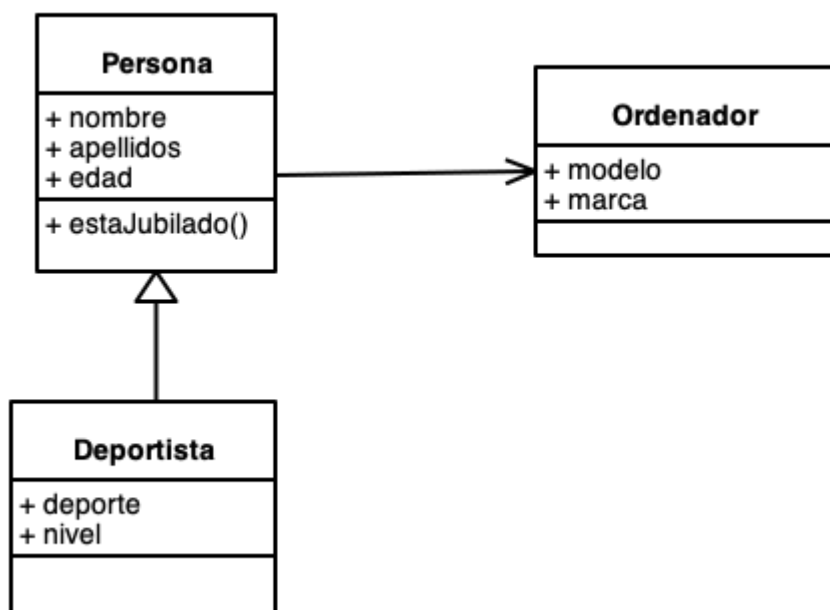


Gaphor UML es una de las herramientas Open Source para desarrollar diagramas UML y es una de las que a mí me parece que hace un trabajo razonable. Muchas veces nos encontramos con la necesidad de elegir una herramienta y hay muchas quizá demasiadas. El gran problema es que la mayoría son de pago y un pago a veces con un importe bastante alto . Gaphor es capaz de realizar la mayor parte de los diagramas UML de una manera bastante sencilla y amigable siendo una herramienta opensource y clara a la hora de utilizarse. Vamos a ver algunos diagramas con esta herramienta.

Gaphor UML y Clases

El diagrama más habitual que podemos necesitar construir es un diagrama de clases. En este caso he construido 3 clases con propiedades y funciones y Gaphor se ha comportado muy bien a la hora de presentar menus y opciones para construir el diagrama.



El sistema de Menus me ha parecido francamente intuitivo a la hora de construir las clases con sus propiedades:

Name

Persona

Abstract ☐

▼ **Attributes**

Show Attributes ☒

Attributes	S
+ nombre	☐
+ apellidos	☐
+ edad	☐
	☐

OBJECT Help

▼ **Operations**

Show Operations ☒

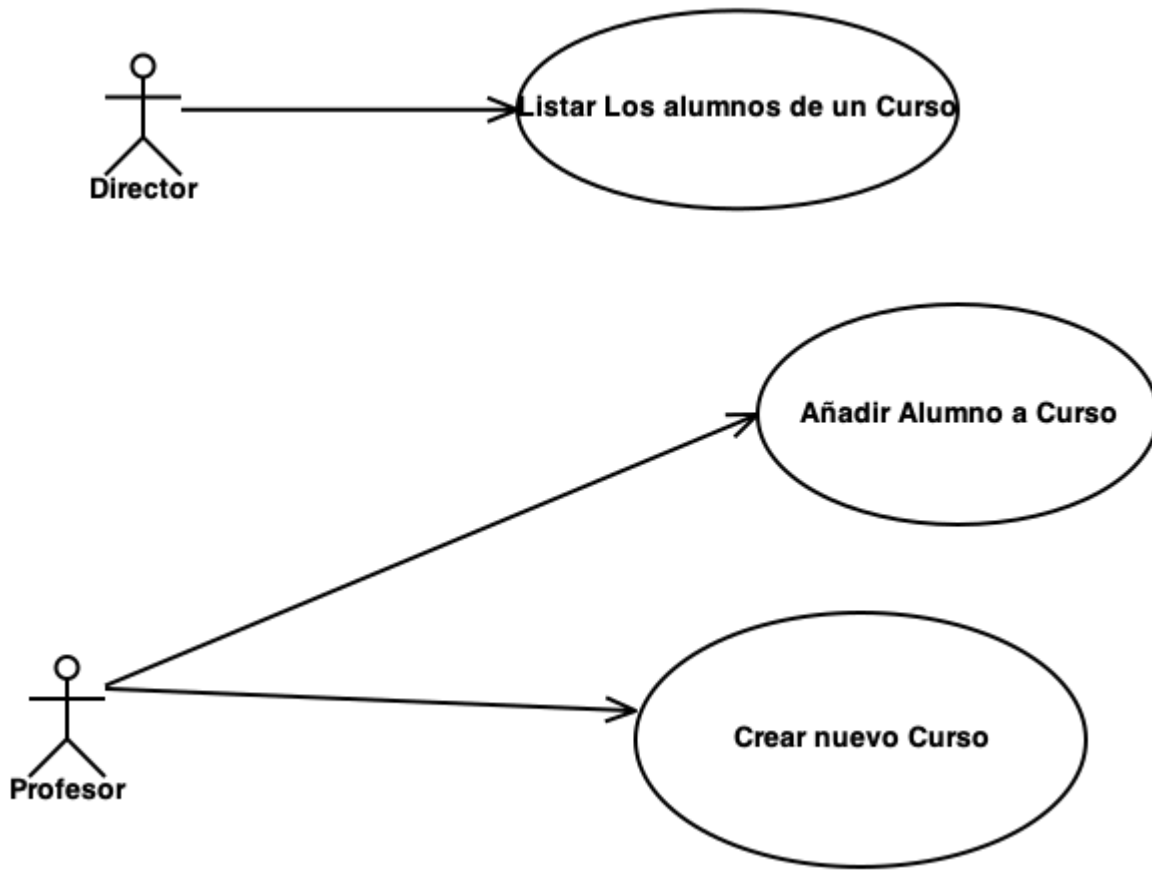
Operations	A	S
+ estaJubilado()	☐	☐
	☐	☐

OBJECT Help

▼ **Note or Remark**

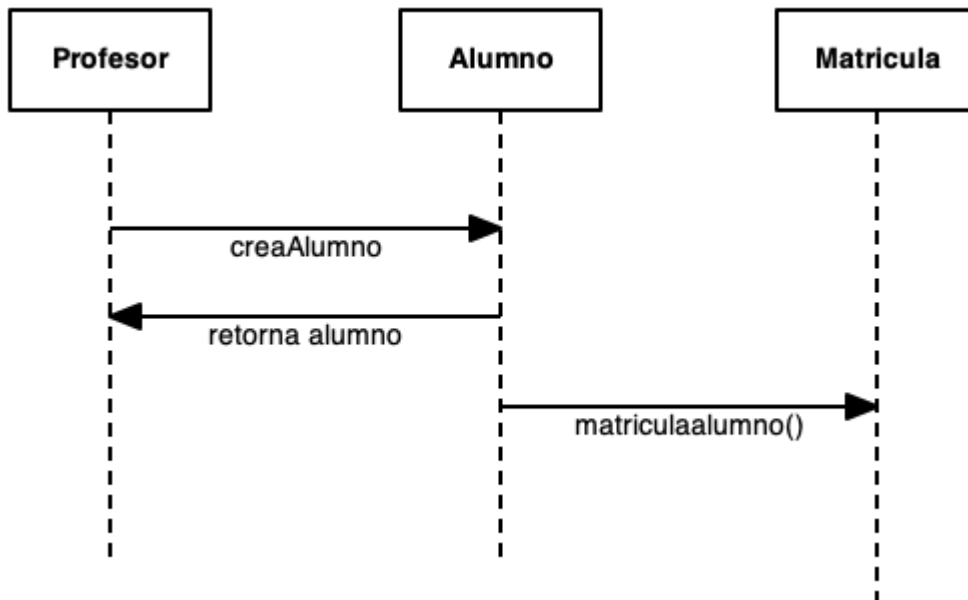
Gaphor UML y diagramas de casos de uso

Vamos a ver otro de los diagramas más sencillos de UML que es el diagrama de casos de Uso que nos permite ver a vista de pájaro las tareas que tenemos que implementar a nivel de desarrollo.



Gaphor UML y diagramas de secuencia

Todos los diagramas están soportados y los de secuencia no son una excepción nos permite construir de una forma rápida un diagrama de Secuencia que nos sirva para comunicar cual es el flujo de trabajo entre varios elementos de forma rápida:



La herramienta me ha gustado y me experiencia es que siempre cuesta encontrar una herramienta UML que de un soporte razonable y que sea gratuita . Echala un vistazo y espero que te aporte en tu día a día para solventar estos temas de diagramas y documentación. La web es : [Gaphor](#)

Otros artículos relacionados

1. [El patron Repository y la explosión de métodos](#)
2. [Spring @Service , usando el patrón Servicio](#)
3. [El patrón MVC , arquitectura cliente vs servidor](#)