

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

Proyecto nomnoml



Introducción

El presente proyecto, se llevó a cabo a través del proyecto nomnoml, que nos proporciona contribuciones de diferentes personas, que hace que este proyecto sea más completo.

La contribución a este desarrollo de proyecto nos permitió aplicar conocimientos adquiridos, al igual que nos permite el desarrollo de diagramas UML a través de código, así como su exportación en diferentes lenguajes de programación.

Este proyecto busca dar una buena implementación del desarrollo de modelado UML, para que a través de este se obtenga un buen código, en diferente lenguaje de programación.



nomnoml

Es una herramienta para realizar diagramas UML basados en una sintaxis simple.

Está basado en código, a través del código se crean los diagramas .

Exportar diagrama/código.

Importar diagrama/código.

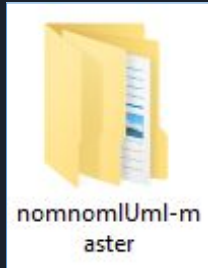
Instalación

Entrar a la página y descargar el proyecto :

<https://github.com/Euwe-QlqI71/nomnomlUml>

Descomprimir carpeta

Abrir carpeta



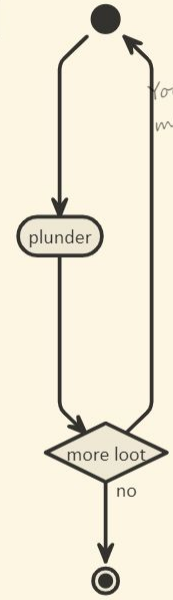
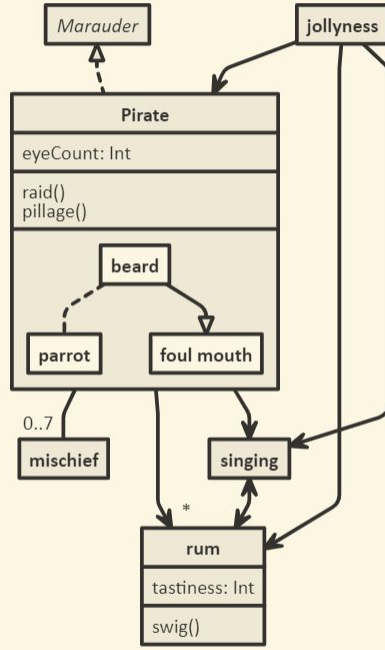
Abrir el siguiente archivo

Nombre	Fecha de modificación	Tipo	Tamaño
.cache	21/10/2019 07:47 p. m.	Carpeta de archivos	
addon	21/10/2019 07:47 p. m.	Carpeta de archivos	
build	21/10/2019 07:47 p. m.	Carpeta de archivos	
codemirror	21/10/2019 07:47 p. m.	Carpeta de archivos	
dist	21/10/2019 07:47 p. m.	Carpeta de archivos	
img	21/10/2019 07:47 p. m.	Carpeta de archivos	
lib	21/10/2019 07:47 p. m.	Carpeta de archivos	
node_modules	21/10/2019 07:47 p. m.	Carpeta de archivos	
src	21/10/2019 07:47 p. m.	Carpeta de archivos	
test	21/10/2019 07:47 p. m.	Carpeta de archivos	
webapp	21/10/2019 07:47 p. m.	Carpeta de archivos	
.gitattributes	21/10/2019 07:47 p. m.	Documento de te...	1 KB
changelog.md	21/10/2019 07:47 p. m.	Archivo MD	1 KB
CONTRIBUTING.md	21/10/2019 07:47 p. m.	Archivo MD	1 KB
deploy	21/10/2019 07:47 p. m.	Archivo JavaScript	2 KB
favicon	21/10/2019 07:47 p. m.	Archivo PNG	1 KB
index	26/11/2019 07:46 p. m.	Archivo HTML	17 KB
LICENSE	21/10/2019 07:47 p. m.	Archivo	2 KB
nomnoml	21/10/2019 07:47 p. m.	Archivo CSS	6 KB
package	21/10/2019 07:47 p. m.	JavaScript Object ...	2 KB
package-lock	21/10/2019 07:47 p. m.	JavaScript Object ...	7 KB
README.md	21/10/2019 07:47 p. m.	Archivo MD	5 KB

```

1 [Pirate|eyeCount: Int|raid();pillage()]
2   [beard]--[parrot]
3   [beard]->[foul mouth]
4 ]
5
6 [<abstract>Marauder]<:--[Pirate]
7 [Pirate]- 0..7[mischief]
8 [jollyness]->[Pirate]
9 [jollyness]->[rum]
10 [jollyness]->[singing]
11 [Pirate]-> *[rum|tastiness: Int|swig()]
12 [Pirate]->[singing]
13 [singing]<->[rum]
14
15 [<start>st]->[<state>plunder]
16 [plunder]->[<choice>more loot]
17 [more loot]->[st]
18 [more loot] no ->[<end>e]
19
20 [<actor>Sailor] - [<usecase>shiver me;timbers]
21 [ <Actores> mas] - [<usecase> tiembla;maderas]

```



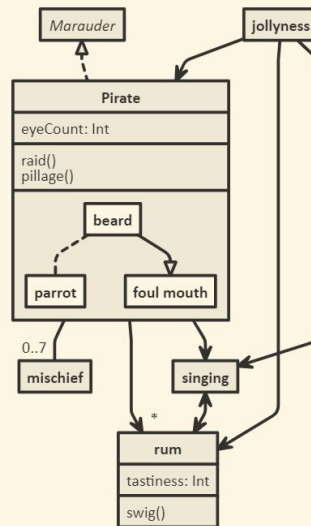
You can now have multiple diagrams!



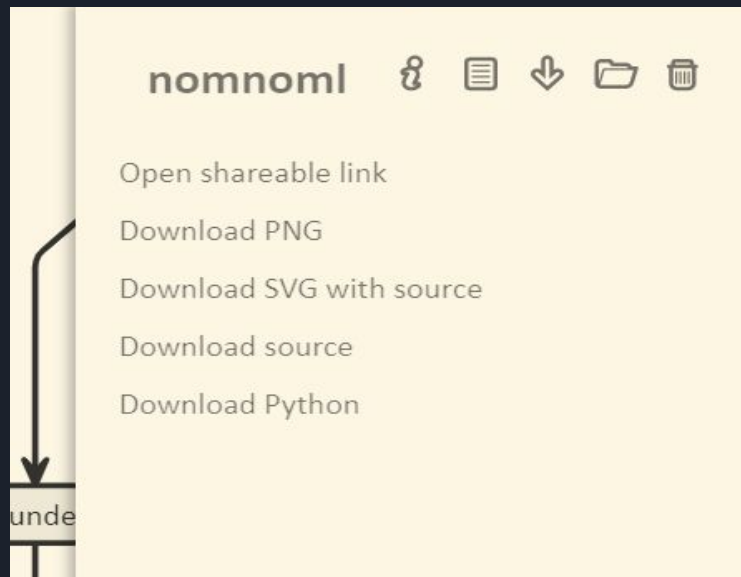
Funciones

Escribir código, creación de diagrama

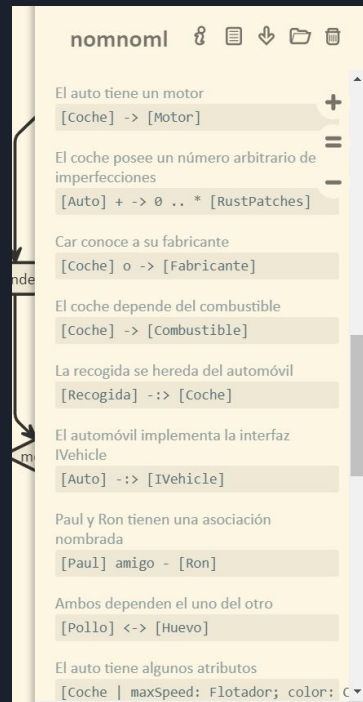
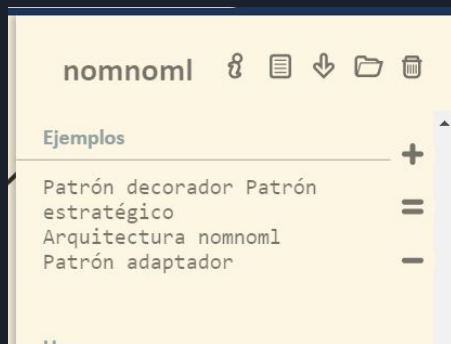
```
1 [Pirate|eyeCount: Int|raid();pillage()|
2   [beard]--[parrot]
3   [beard]->[foul mouth]
4 ]
5
6 [<abstract>Marauder]<--[Pirate]
7 [Pirate]- 0..7[mischief]
8 [jollyness]->[Pirate]
9 [jollyness]->[rum]
10 [jollyness]->[singing]
11 [Pirate]->*[rum|tastiness: Int|swig()]
12 [Pirate]->[singing]
13 [singing]<->[rum]
14
15 [<start>st]->[<state>plunder]
16 [plunder]->[<choice>more loot]
17 [more loot]->[st]
18 [more loot] no ->[<end>e]
19
20 [<actor>Sailor] - [<usecase>shiver me;timbers]
21 [ <Actores> mas] - [<usecase> tiembla;maderas]
```



Exportación



Ejemplos y notación





Ventajas hacia el proyecto

Con un buen desarrollo de modelado UML, puedes generar un buen código.

Con un conocimiento de programación puedes contribuir a proyectos , para una mejor aplicación hacia ellos.



Conclusión

El diagrama de clases recoge las clases de objetos y sus asociaciones, es por ellos que al hacer un buen modelado UML, se puede generar un buen código en cualquier lenguaje de su preferencia.

Es por ello que a lo que concluimos en este proyecto es que con la contribución de proyectos externos, se puede mejorar y hacer una herramienta más fácil de utilizar y a la vez facilitar el desarrollo de código a través de buenos diagramas UML.