

Bienvenidos a la practica 9

En esta práctica vamos a realizar diferentes cosas:

- Estamos utilizando una extensión de VS
- Esta extensión está basada en Markdown
- Vamos a realizar tres actividades:
 - Diapositiva con MARP
 - Mapa Mental con MARKMAP
 - Diagramas UML

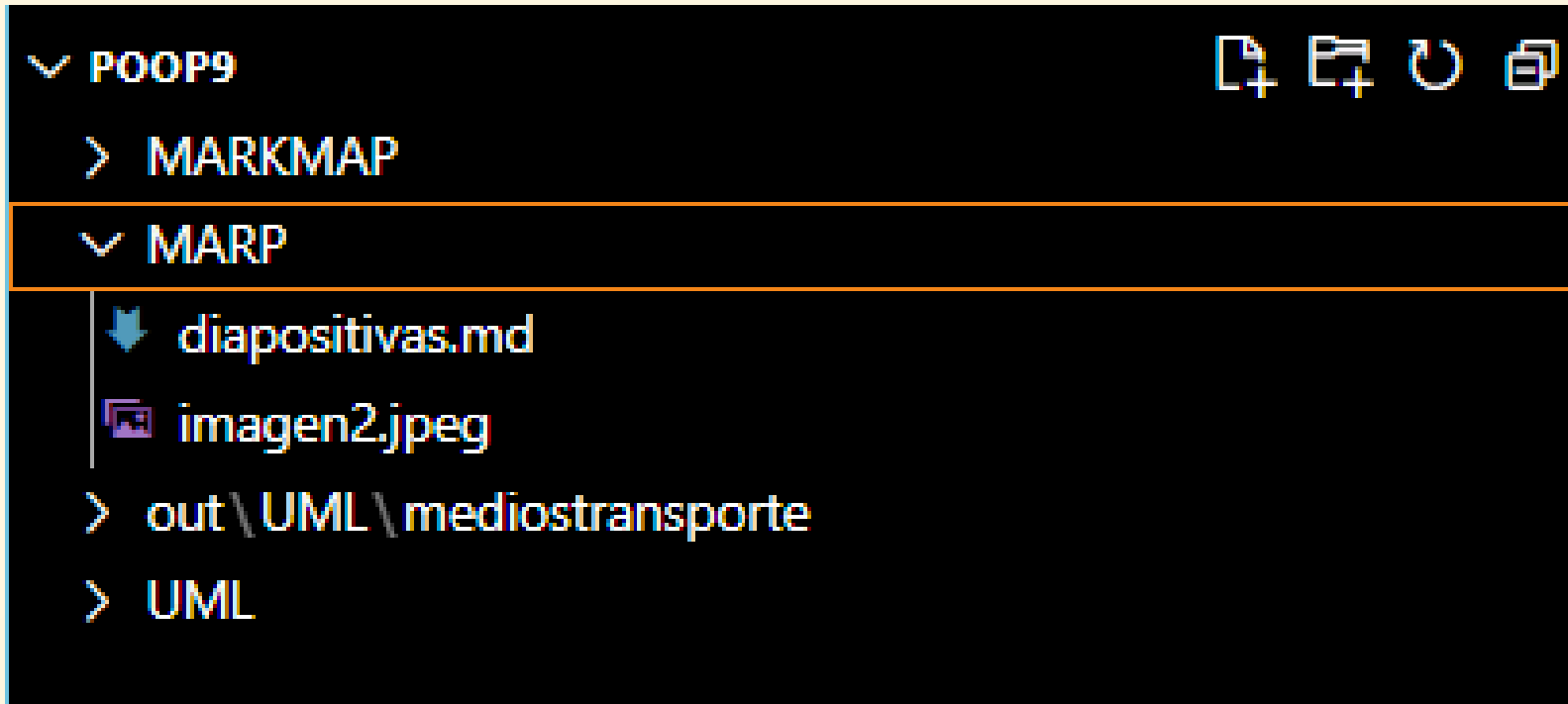
Iniciando en VS

Para poder realizar estas actividades primero vamos a tener que instalar tres extensiones en VS, las cuales son:

- Marp
- Markmap
- PlantUML Previewer

Después de instalarlas vamos a crear una carpeta nueva, en donde vamos a crear mas sub carpetas donde vamos a realizar cada una de las actividades, primero vamos a iniciar con la diapositiva en MARP.

Ejemplo de visualización de la carpeta:



Aquí se pueden ver las carpetas creadas, así como también dentro de marp se crea un archivo con extensión .md

Diapositiva con MARP

Para poder realizar una diapositiva con Marp se necesita iniciar con la siguiente sintaxis:

```
1  ---  
2  marp: true  
3  author: Santiago Ruiz  
4  size: 4:3  
5  theme: gaia  
6  ---
```

En esta se define el autor, el tamaño de la hoja y el tema predeterminado que se usara.

Como también se puede apreciar cada tres guiones significan que empieza una nueva hoja.

Personalización de la Letra

Después de tener las especificaciones anteriormente mostradas completas puedes crear tu diapositiva libremente, sin embargo estas se pueden personalizar aún más con los siguientes aditamentos:

- Se puede cambiar la letra del texto de estas 4 maneras:
 - **Negrita**----- ****Negrita****
 - *Italica*----- ***Italica***
 - ~~Tachado~~----- **~~Tachado~~**
 - `codigo`----- **'codigo'** (Este ultimo va con acentos invertidos)

Personalización del Título

Aquí se muestra como los diferentes titulos pueden ser utilizados, así como su escritura en el código:

Título 1-----# Título 1

Título 2-----## Título 2

Título 3-----### Título 3

Título 4-----#### Título 4

Agregando imagenes

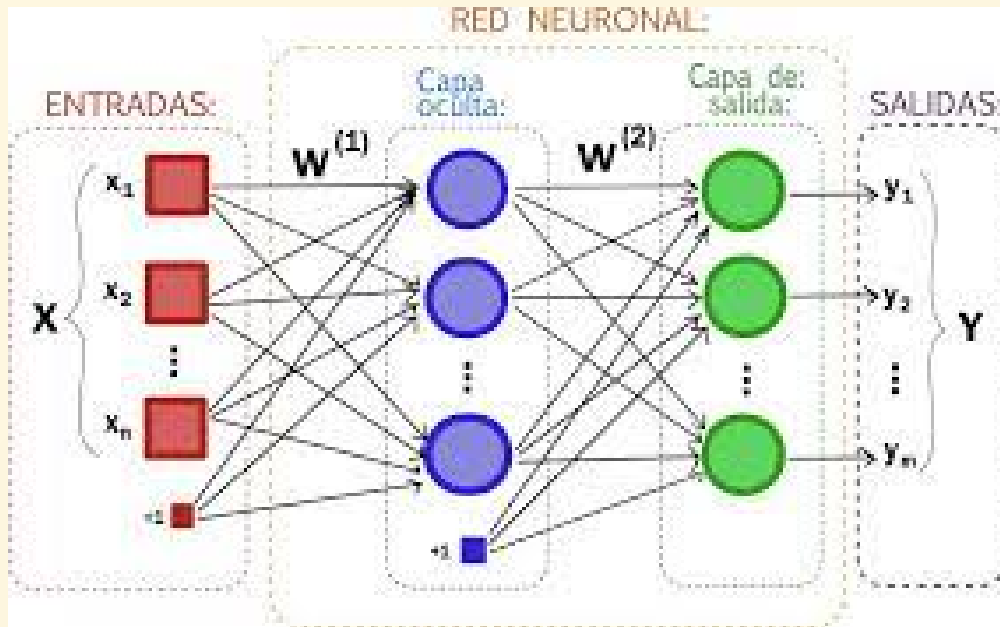
Para agregar imagenes solo de debe descargar una imagen en la carpeta de tu dispositivo, después esta la mueves dentro de la carpeta del programa, en nuestro caso la carpeta MARP, por ultimo se agrega la siguiente línea la código:

```
![width: 000px](nombreImagen.png)
```

Donde se elige su anchura en pixeles, así como se le agrega el nombre de la imagen descargada junto con su extensión (.png).

Por ejemplo:

Diagrama de una red neuronal



En este caso la línea de código quedo de la sig manera:

```
![width: 300px](imagen2.jpeg)
```


Tablas

Para agregar tablas solo se debe agregar la siguiente sintaxis:

```
|Titulo 1|Titulo 2|
```

```
|---|---|
```

```
|celda 1|celda 2|
```

```
|celda 3 |celda 4|
```

Al final se mostrara de la sig manera en pantalla:

Titulo 1 Titulo 2	
celda 1	celda 2
celda 3	celda 4

Comentarios

Para finalizar el tutorial de las diapositivas se encuentran también los comentarios, estos solo se ven en el código y sirven para hacer alguna notación extra.

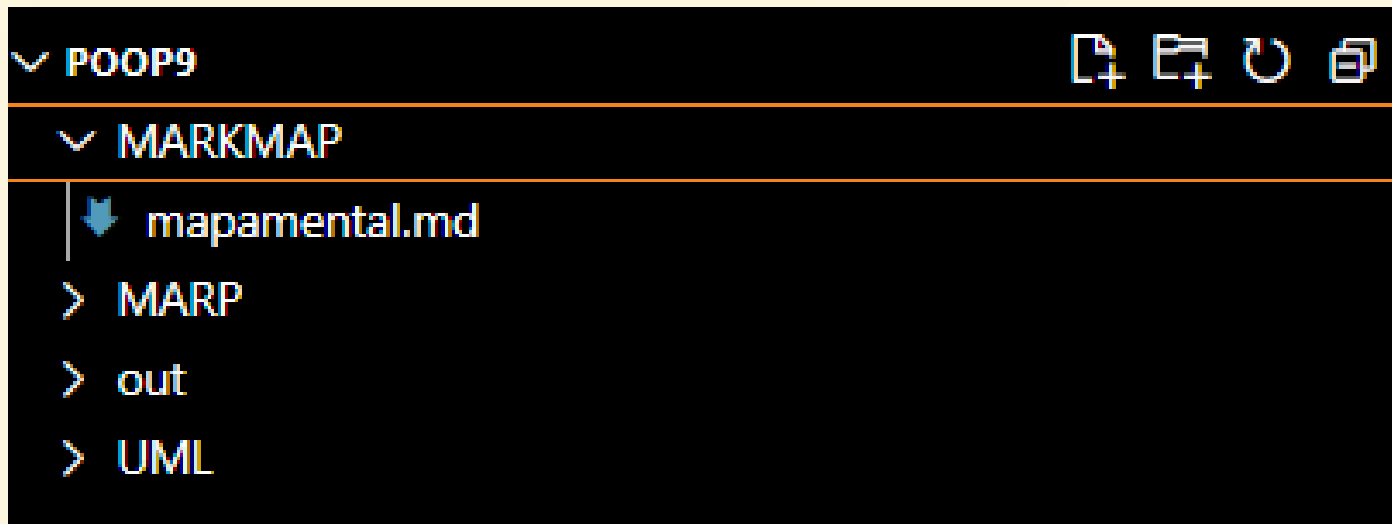
Hay dos maneras:

```
[comment]: <> (This is a comment, it will not be included)
```

```
<!--Este es un comentario-->
```

Mapa Mental con MARKMAP

Para continuar con las actividades, ahora realizaremos un mapa mental con la extensión MARKMAP, en la carpeta anteriormente creada de MARKMAP, vamos a crear un nuevo documento, con el nombre mapamental.md.



Posteriormente dentro del documento crearemos el mapa deseado, su estructura es muy sencilla, ya que solo utilizaremos el simbolo de # para definir de que manera lo vamos a organizar.

Por ejemplo:

```
# Mapa mental
```

```
## Rama 1
```

```
### Subrama 1.1
```

```
### Subrama 1.2
```

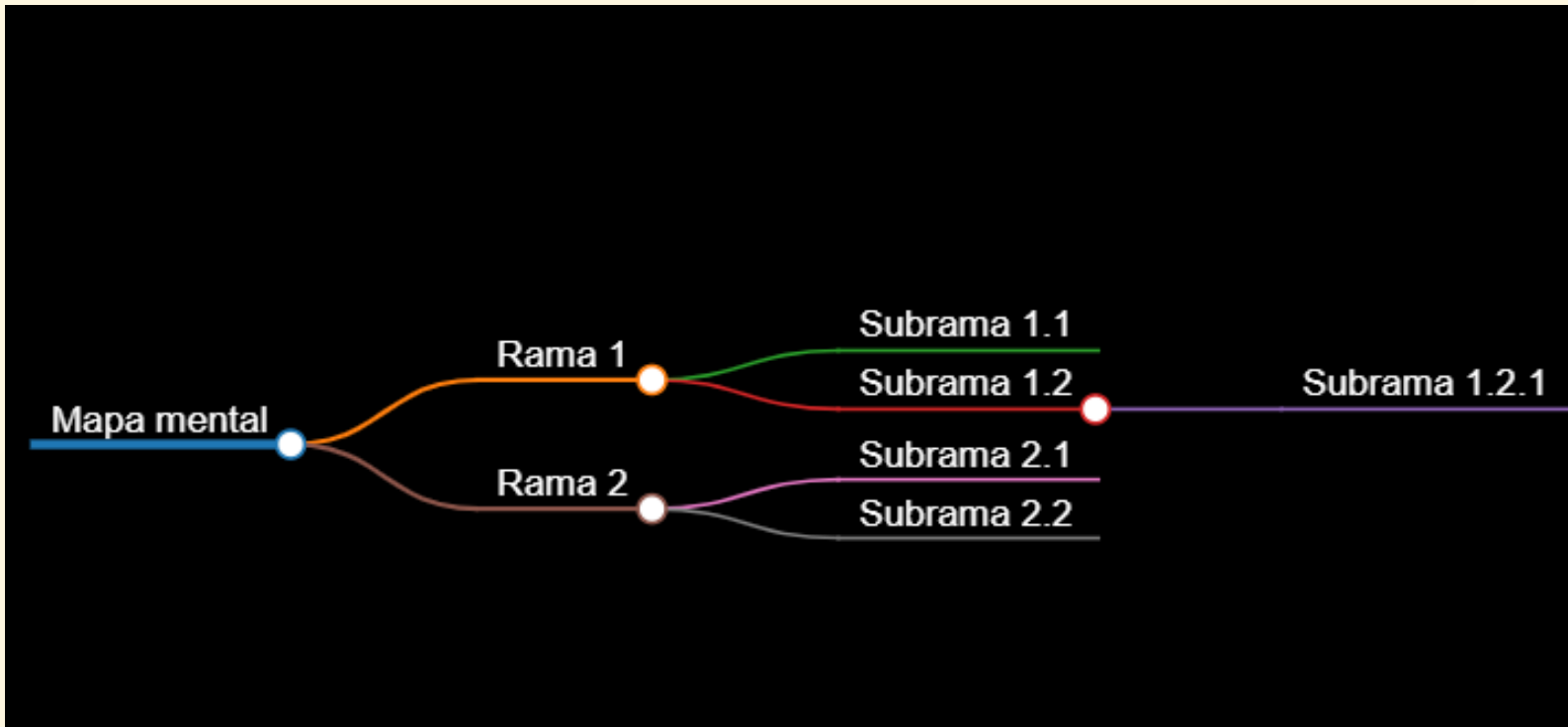
```
#### Subrama 1.2.1
```

```
## Rama 2
```

```
### Subrama 2.1
```

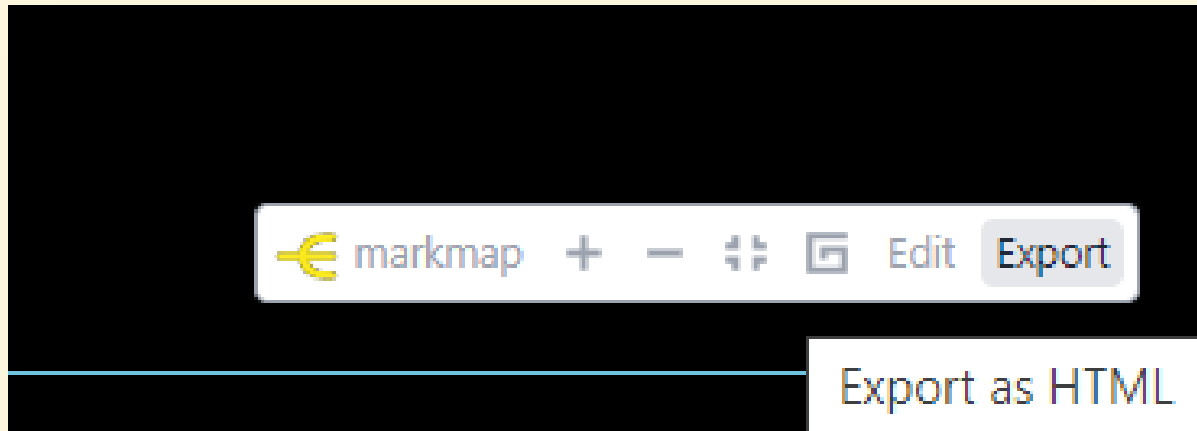
```
### Subrama 2.2
```

De esta manera, al presionar el boton con forma de tridente de la parte superior derecha de VS, generará el mapa:

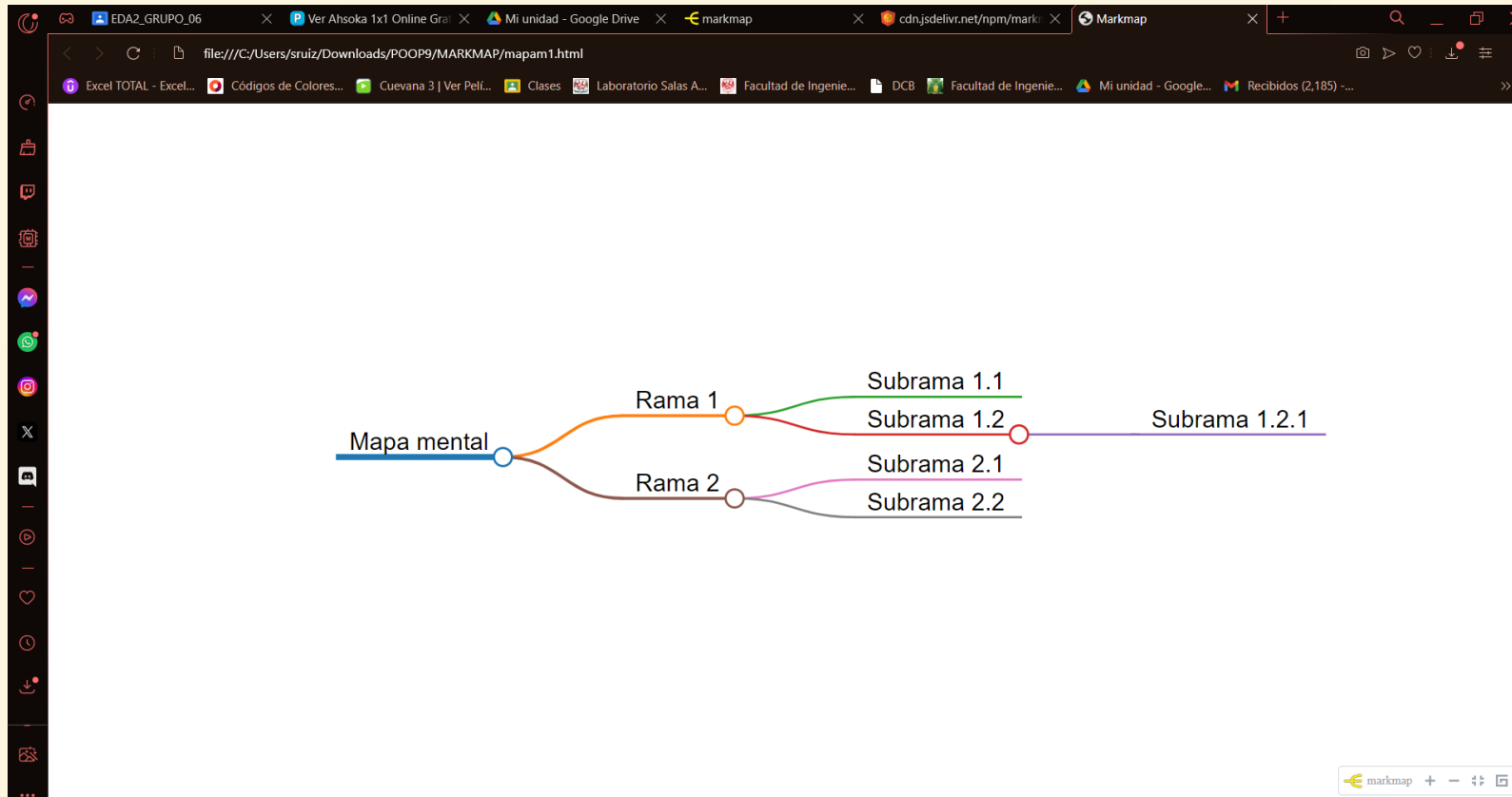


Exportación

Este mapa se puede exportar para que sea visible en una página *.html*, para hacerlo es de igual manera muy sencillo, en la página donde generamos el mapa hace un momento, en la parte inferior derecha se encuentra unos botones.



Al darle en el boton *Export* te dara la opcion de guardar en tu dispositivo la página *.html*, esta se genera automaticamente y queda de la sig manera:



Así como también el código de esta página `.html` se queda en la carpeta MARKMAP.

POOP9

MARKMAP

mapam1.html

mapamental.md

MARP

out

UML

```
MARKMAP > mapam1.html > html > head > meta
1 |<!DOCTYPE html>
2 |<html>
3 |<head>
4 |<meta charset="UTF-8">
5 |<meta name="viewport" content="width=device-width, initial-scale=1.0">
6 |<meta http-equiv="X-UA-Compatible" content="ie=edge">
7 |<title>Markmap</title>
8 |<style>
9 | * {
10 |   margin: 0;
11 |   padding: 0;
12 | }
13 | #mindmap {
14 |   display: block;
15 |   width: 100vw;
16 |   height: 100vh;
17 | }
18 |</style>
19 |<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/markmap-toolbar@0
20 |</head>
21 |<body>
22 |<svg id="mindmap"></svg>
23 |<script src="https://cdn.jsdelivr.net/npm/d3@7.8.5/dist/d3.min.js"></script>
24 | | | | | | | setTimeout(r);
25 | | | | | | | })(() => {
26 | const {
27 |   markmap,
28 |   mm
29 | } = window;
30 | const {
31 |   el
32 | } = markmap.Toolbar.create(mm);
33 | el.setAttribute('style', 'position:absolute;bottom:20px;right:20px');
34 | document.body.append(el);
35 |}</script><script>((getMarkmap, getOptions, root, jsonOptions) => {
36 | | | | | const markmap = getMarkmap();
37 | | | | |   window.mm = markmap.Markmap.create('svg#mindmap', (getOptions ||
38 | | | | |   ))( () => window.markmap,null,{ "type": "heading", "depth": 0, "payload":
39 |</body>
40 |</html>
```


Diagramas UML

La ultima actividad consiste en realizar diagramas en la extensión UML, para este caso en especifico vamos a utilizar como ejemplo clases que hemos creado anteriormente en practicas pasadas, para empezar vamos a ir a la ultima carpeta que creamos, UML, dentro de esta misma se crea un nuevo documento el cual llamaremos *mediostransporte.md*, donde empezaremos con el sig código:

```
---
```

```
marp: false
```

```
---
```

Para iniciar a poner los diagrmas se debe agregar al inicio:

```
@startuml
```

Posteriormente se agrega su escala y la información de cada clase, es decir sus atributos y sus metodos, para al final agregar la relacion de cada uno de ellos, quien es 'padre' y quien es 'hijo', así como si estas llegan a ser una interfaz.

Por ejemplo:

```
@startuml
```

```
scale 3
```

```
abstract class MediosTransporte{}
```

```
class TransporteAcuatico{  
    . -velocidad: int  
    . -capacidad: String  
    . +aumentarVelocidad():void  
}
```

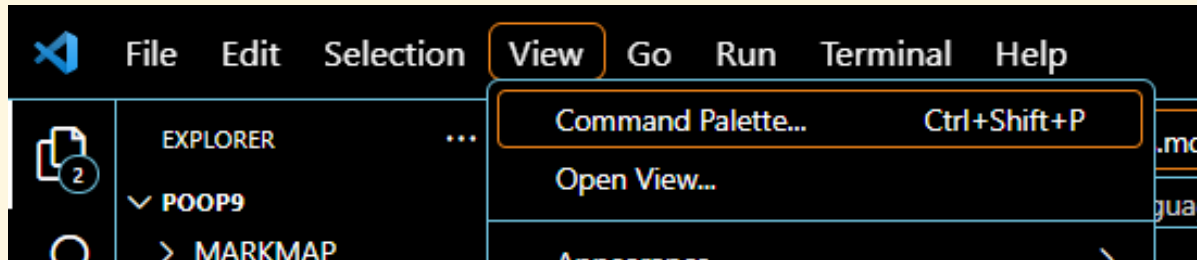
```
class Barco{  
    . -puertoOrigen: String  
    . -puertoDestino: String  
    . +abordarPasajeros():void  
}
```

```
MediosTransporte <|--TransporteAcuatico
```

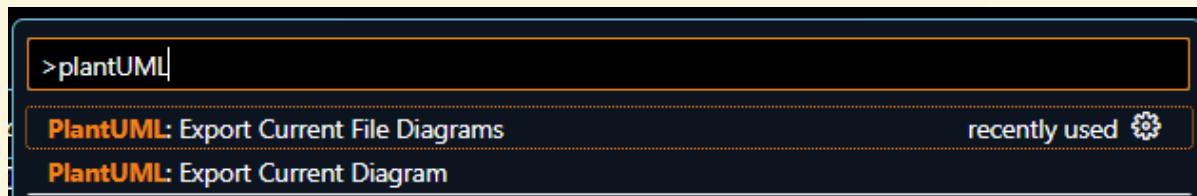
```
TransporteAcuatico<|--Barco : interfaz
```

```
@endum1
```

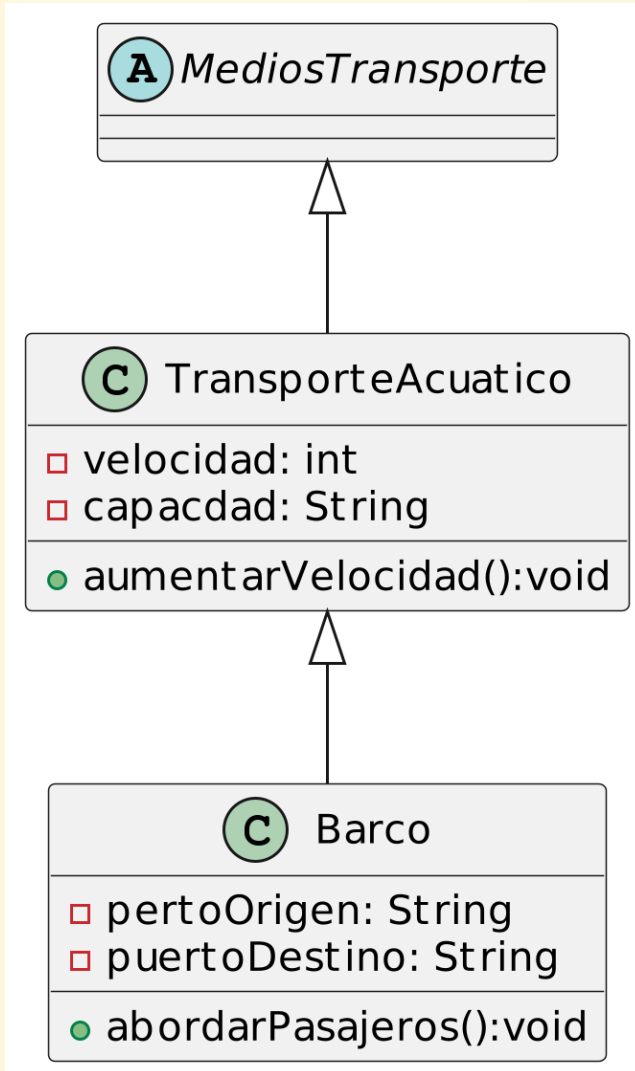
Con esto ya hecho se puede realizar el diagrama llendo a la pestaña *View*, en esta se selecciona *Command Palette*.



De ahí vas a buscar plantUML y seleccionas la primera opción.



Lo guardas en tu dispositivo y se generará la imagen correspondiente a tu código:



En la imagen se puede apreciar como es que cada clase tiene sus respectivos atributos y metodos, así como que clase es 'padre' y cual es 'hijo'.

Esta imagen se puede guardar facilmente en su dispositivo, para poder usarla en algun documento.

Con esto finaliza el tutorial de como utilizar MARKDOWN, MARP y UML, en base a estos conocimientos basicos se puede crear algo mucho más elaborado para un trabajo o tarea asignado.