

Tutorial MARP para Visual Studio

- **Paso 1: Instalación:**

En Visual Studio Code, vamos a la pestaña "Extensions" en la barra lateral izquierda (icono de cuatro cuadrados apilados) y buscamos *Marp for VS Code*. Instalamos esta extensión. Ahora, haremos esa misma búsqueda e instalación para los visualizadores *Markmap*, *Markdown Preview Markmap Support* y *PlantUML*.

Markdown: Para crear diagramas:

Para crear una presentación:

Paso 2: Crea tu presentación

Crear un archivo con la extensión .md y agregar elementos.

- Estilos de texto:

Estilo	Sintaxis	Resultado
Negrita	** texto **	Negrita
Cursiva	<i>* texto *</i>	<i>Cursiva</i>

- Crear una tabla

- Se utilizan |, para separar las columnas y crean los bordes de la tabla.
- Las líneas inferiores -, se utilizan para separar las filas de los encabezados.
- El contenido de cada celda se coloca entre los separadores |.

- Ejemplo:

```
|tituloa1|titulo2|
```

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

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

```
|celda 3|celda4|
```

Resultado de la tabla:

tituloa1	titulo2
celda 1	celda 2
celda 3	celda 4

- Jerarquizamos títulos con #:
 - Ejemplo:

Titulo 1 = # Titulo 1

Titulo 2 = # Titulo 2

- Sintaxis para agregar una imagen:
"! [Descripción de la imagen] (imagen.png)"

Paso 3: Previsualización

- Para previsualizar la presentación, hacemos clic con el botón derecho en el archivo .md en la barra lateral izquierda y selecciona "Marp: Open Preview".

Markmap: Para visualizar.

Paso 1: Creamos un archivo nuevo con la extensión .md.

Paso 2: Uso de extensión Markmap para crear mapa.

- Para crear un mapa, hacemos clic con el botón derecho en el archivo .md en la barra lateral izquierda y selecciona "Reopen Editor With...", buscamos y seleccionamos la extensión Markmap.

- Una vez abierta este editor es posible visualizar la contrucción de nuestro mapa, a la vez que lo modificamos.

Paso 3: Creamos nuestro mapa. Para esto, usamos # para general los niveles de las ramas:

- Ejemplo de sintaxis (omitiendo comillas):
"# Mapa mental"
"## Rama"
"### subrama 1.1"
"### subrama 1.2"
"## Rama 2"
"### subrama 2.1"

PlantUML

PlantUML es una herramienta que nos permite crear diagramas UML y otros tipos de diagramas utilizando una sintaxis basada en texto.

Paso 1: Creamos un archivo nuevo con la extensión PlantUML

- Formatos admitidos:
.wsd, .pu, .puml, .plantuml, .iuml

Paso 2: Creamos nuestro diagrama de clases:

- Comenzamos con un *@startuml* y finalizamos con un *@enduml*.
- Para la vista previa del diagrama, presionamos Alt + D.
- Este complemento integra componentes de diagramas. Se dividen en 9 secciones:
 - Diagram: snippets de elementos de diagramas generales.
 - activity: snippets de diagramas de actividades.

- ○ class: snippets de diagramas de clases.
- component: snippets de diagramas de componentes.
- state: snippets de diagramas de estado.
- usecase: snippets de diagramas de casos de uso.
- sequence: snippets de diagramas de secuencia.
- ui: snippets de diagramas de sal.
- egg: snippets de algunos diagramas divertidos, como sudoku.

- Uso de flechas paar generar el diagrama: <|--
- Ejemplo de uso:

```
@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
@enduml