

Convertir VSCode en un editor de Markdown con el poder de las extensiones.

- [Convertir VSCode en un editor de Markdown con el poder de las extensiones.](#)
 - [1. ¿Qué es Markdown y cómo se utiliza?](#)
 - [Sintaxis básica de Markdown](#)
 - [2. Utilizar Markdown en Visual Studio Code](#)
 - [Instalación de Visual Studio Code](#)
 - [Extensiones importantes](#)
 - [Comandos útiles dentro de VSCode usando Markdown All In One](#)
 - [Memoria de entrega](#)

1. ¿Qué es Markdown y cómo se utiliza?

Markdown es un lenguaje de marcado ligero que permite formatear texto de manera sencilla e intuitiva. Es muy utilizado para escribir documentación, blogs o cualquier otro tipo de texto que luego se convierte a HTML, ya que facilita mucho su legibilidad. Un documento Markdown es esencialmente texto plano que se puede leer fácilmente y que luego se puede exportar a otros formatos como HTML o PDF.

Estos apuntes están escritos combinando Markdown con plantuml y mermaidjs. Además, Markdown es el lenguaje de marcado más empleado a la hora de elaborar **documentación técnica**, y también es el lenguaje de marcado que emplean los chatbots como ChatGPT o Gemini para imprimir sus resultados.

Además, la sencillez de Markdown se puede combinar con el poder de HTML5, permitiendo intercalar etiquetas HTML en documentos Markdown para así conseguir cosas complejas, como aplicar un estilo CSS o usar `<details>` para ocultar elementos (como se hace en estos apuntes con los códigos en plantuml).

Sintaxis básica de Markdown

1. Títulos y subtítulos

Los títulos en Markdown se crean utilizando el símbolo de almohadilla `#`. Cuantas más

almohadillas se colocan, menor es el nivel del título:

- # Título 1
- ## Título 2
- ### Título 3

2. Texto en negrita

El texto se puede poner en negrita rodeándolo con dos asteriscos ****** o dos guiones bajos **__** :

- ****Este texto está en negrita****
- **__Este texto también está en negrita__**

3. Texto en cursiva

Para poner el texto en cursiva, se utiliza un solo asterisco *** o un solo guion bajo *_* :

- **Texto en cursiva**
- *_Texto en cursiva_*

4. Listas

Para crear una **lista no ordenada** (puntos), se utiliza un asterisco *** , un signo más *+* , o un guion *-* :

- ** Elemento 1*
- *- Elemento 2*

Para **listas numeradas**:

- 1. Primer elemento
- 2. Segundo elemento

5. Enlaces

Los enlaces se crean utilizando la sintaxis `[texto](url)` :

- `[Google](https://www.google.com)`

6. Imágenes

Las imágenes se pueden insertar de manera similar a los enlaces, añadiendo un signo de exclamación al principio `![texto alternativo](url)` :

- `![Logo](https://url-de-la-imagen.com/logo.png)`

7. Código

Si quieres mostrar un fragmento de código dentro de una línea de texto, puedes utilizar las comillas invertidas ``` . Para **bloques de código**, se pueden usar tres comillas invertidas `````:

- inline code: `\ echo "Hola"`
- Bloques de código:
- Puedes especificar incluso el lenguaje de programación para que resalte la sintaxis si quieres (no funcionará en todos los entornos).

```
echo "Esto es un bloque de código"
```

8. Citas

Se pueden añadir citas utilizando `>` al principio de la línea:

- `>` Esto es una cita

2. Utilizar Markdown en Visual Studio Code

Instalación de Visual Studio Code

Primero que todo y si no lo has hecho aún, es necesario instalar **VSCode**, que se puede descargar desde su [página oficial](#). Una vez instalado, ya estaremos listos para personalizarlo y utilizarlo para editar Markdown.

Extensiones importantes

Para mejorar la experiencia trabajando con Markdown en VSCode, es necesario instalar algunas **extensiones muy útiles**:

1. Markdown All in One

Esta extensión proporciona herramientas útiles para editar Markdown, como vista previa, cierre automático de listas y otros atajos de teclado.

- Para instalarla, ve a la sección de extensiones dentro de VSCode y busca "Markdown All in One".

2. Markdown PDF

Esta extensión permite convertir ficheros `.md` a PDF, HTML u otros formatos.

- Busca "Markdown PDF" e instálala para poder exportar tus documentos.

3. Markdown Enhanced Preview (RECOMENDADA)

Esta extensión es ideal para trabajar con Markdown, Mermaid y Plantuml. Incorpora funciones de todo lo anterior, por lo que vas a poder exportar el resultado a pdf o incluso HTML. Puedes instalar solamente esta extensión y funcionará, es una especie de sustituta de las otras dos.

4. Markdownlint

Añade un **linter** que verifica si tu Markdown sigue buenas prácticas, ayudándote a corregir errores comunes y a seguir una sintaxis pura de Markdown (cosa que no siempre te será útil)

- Busca "Markdownlint" en la sección de extensiones.

Comandos útiles dentro de VSCode usando Markdown All In One

1. Vista previa de Markdown

Para ver una vista previa del fichero Markdown que estás editando, puedes abrir el fichero `.md`

y hacer clic en `Ctrl+Shift+V` (Windows/Linux) o `Cmd+Shift+V` (Mac). También puedes hacer clic con el botón derecho sobre el fichero y seleccionar "Open Preview".

2. Exportar a HTML o PDF

Una vez tienes el documento editado, puedes exportarlo a HTML o PDF utilizando la extensión Markdown PDF:

- Para exportarlo, abre la paleta de comandos (`Ctrl+Shift+P` o `Cmd+Shift+P`) y escribe "Markdown PDF: Export". Selecciona el formato deseado.
- También puedes escribir `>` en la barra de búsqueda. Tiene el mismo efecto y puede ser más fácil de recordar.

Otra herramienta de diagramas que está ganando popularidad recientemente, en parte gracias a su compatibilidad con diferentes frameworks de aplicaciones web como `astro.build`, es `d2`. Puedes encontrar más información [aquí](#).

Memoria de entrega

Para entregar la actividad, crea una memoria en la que repases paso a paso el proceso de instalación y preparación de las extensiones necesarias para poder editar `plantuml` y `mermaid.js` en VS Code. Crea un documento con capturas de pantalla de todo el proceso.