Markdown: documentación técnica de proyectos

1. ¿Qué es Markdown?

Markdown es un lenguaje de marcado ligero que permite dar formato al texto utilizando una sintaxis sencilla. Se escribe como texto plano y, posteriormente, puede convertirse en un documento con títulos, listas, tablas, enlaces, imágenes y otros elementos.

Su principal ventaja es que permite crear documentos estructurados sin necesidad de utilizar un editor complejo ni conocer HTML.

Por ejemplo, para escribir un título en Markdown se utiliza el símbolo `#`:

```
# Mi primer documento
## Apartado
### Subapartado
```

El documento se puede leer directamente como texto o visualizar con formato mediante un editor compatible.

Este documento que estás leyendo, está hecho en Markdown y posteriormente exportado a PDF.

1.1. Características principales

- **Sencillo:** utiliza símbolos y convenciones fáciles de aprender.
- **Legible:** el contenido sigue siendo comprensible aunque se abra como texto plano.
- **Portable:** los archivos se pueden abrir con cualquier editor de texto.
- **Compatible:** muchas plataformas y herramientas permiten visualizar Markdown.
- **Versátil:** sirve para elaborar documentación técnica, apuntes, manuales, informes y archivos de presentación de proyectos.
- **Fácil de mantener:** permite modificar la documentación sin depender de un formato propietario.

Los archivos Markdown suelen guardarse con las extensiones `.md` o `.markdown`. Se pueden exportar a HTML o PDF.

2. ¿Para qué se utiliza Markdown?

Markdown se utiliza especialmente cuando es necesario escribir documentación clara, organizada y fácil de compartir.

Algunos usos habituales son:

- **Documentación de proyectos:** descripción del proyecto, objetivos, alcance y requisitos.
- **Documentación técnica:** instalación, configuración y administración de sistemas y servicios.
- **Archivos README:** presentación de repositorios de código, instrucciones de instalación y requisitos de uso.
- **Manuales de usuario:** instrucciones para utilizar una aplicación o un servicio.
- **Manuales de instalación:** pasos necesarios para desplegar una infraestructura o configurar un servidor.
- **Informes de pruebas:** descripción de las pruebas realizadas y de sus resultados.
- **Notas y apuntes:** organización de contenidos técnicos y académicos.

3. ¿Cómo se utiliza Markdown?

Markdown se escribe con un editor de texto o con una aplicación que permita editar archivos `.md`. Posteriormente, se puede exportar a otro formato como HTML o PDF si la herramienta utilizada ofrece esa posibilidad.

3.1. Editores y herramientas

Algunas herramientas que permiten trabajar con Markdown son:

- **Visual Studio Code:** editor de código que permite escribir Markdown y previsualizar el resultado.
- **Obsidian:** aplicación orientada a notas y documentación enlazada.
- **Typora:** editor que ofrece una visualización integrada del documento.
- **GitHub:** permite mostrar archivos Markdown dentro de los repositorios.
- **Onedrive:** en una reciente actualización OneDrive, desde la cuenta educativa, permite crear y editar ficheros Markdown aunque con ciertas limitaciones.

También se puede escribir Markdown en cualquier editor de texto plano, aunque la previsualización requiere una herramienta compatible.

3.2. Sintaxis y visualización

Markdown utiliza símbolos para indicar cómo debe interpretarse cada elemento.

Por ejemplo, el siguiente texto:

```
# Servidor web

El servidor utiliza **Apache** para publicar la aplicación.

- Instalar el servicio.
- Configurar el sitio.
- Comprobar el acceso.
```

Se visualizará con un título, una palabra en negrita y una lista de tres elementos.

4. Sintaxis básica de Markdown

4.1. Títulos y encabezados

Los encabezados permiten organizar el documento en diferentes niveles.

Se utiliza el símbolo `#` delante del texto. Cuantos más símbolos se escriben, menor es el nivel del encabezado.

```
# Título principal
## Apartado
### Subapartado
#### Nivel 4
##### Nivel 5
###### Nivel 6
```

Por ejemplo:

Esto es un título de nivel 4

Esto un título de nivel 5

Y esto uno de nivel 6

Se recomienda mantener una jerarquía coherente. Por ejemplo, después de un título de nivel 2, los apartados dependientes deberían utilizar el nivel 3. No es recomendable utilizar los

encabezados únicamente para cambiar el tamaño del texto: deben representar la estructura real del documento.

4.2. Texto en negrita y cursiva

Para destacar palabras o expresiones se pueden utilizar la negrita y la cursiva.

```
**Texto en negrita**

*Texto en cursiva*

***Texto en negrita y cursiva***

~~Texto tachado~~
```

Usos habituales:

- **Negrita:** conceptos importantes, nombres de opciones o advertencias.
- *Cursiva:* términos técnicos, expresiones destacadas o títulos de obras.
- ~~Tachado~~: contenido que se considera eliminado o que ya no está vigente, cuando la plataforma lo admite.

4.3. Párrafos y saltos de línea

Los párrafos se separan dejando una línea en blanco entre ellos.

```
Este es el primer párrafo.

Este es el segundo párrafo.
```

Si se escribe todo el texto seguido, sin separar los párrafos, el resultado puede no ser el esperado.

4.4. Listas sin ordenar

Se utilizan para enumerar elementos cuando no importa su orden.

Se pueden emplear guiones, asteriscos o signos `+`.

```
- Servidor web
- Servidor de bases de datos
```

```
- Servidor DNS
```

Se muestran así:

- Elemento 1
- Elemento 2

También se pueden crear listas anidadas:

```
- Servidor web
  - Instalación
  - Configuración
  - Pruebas
- Servidor de bases de datos
  - Instalación
  - Copias de seguridad
```

Por ejemplo:

- Elemento 1
    - Elemento anidado 1
    - Elemento anidado 2
- Elemento 2

4.5. Listas ordenadas

Se utilizan cuando los elementos deben seguir una secuencia, por ejemplo, en un procedimiento de instalación.

```
1. Instalar el sistema operativo.
2. Configurar la red.
3. Instalar los servicios.
4. Comprobar el funcionamiento.
```

Por ejemplo:

1. Primer elemento.

2. Elemento.

Son especialmente útiles para explicar procedimientos técnicos paso a paso.

4.6. Listas de tareas

Markdown permite representar tareas pendientes y tareas completadas mediante casillas.

```
- [x] Instalar el servidor.
- [x] Configurar la dirección IP.
- [ ] Configurar el cortafuegos.
- [ ] Realizar las pruebas.
```

Por ejemplo:

- [x] Configurar la dirección IP.
- [ ] Configurar el cortafuegos.

La marca `x` indica una tarea completada, mientras que una casilla vacía representa una tarea pendiente.

Esta sintaxis es útil para listas de comprobación (*checklists*), planificación y seguimiento del proyecto. La posibilidad de marcar las casillas de forma interactiva depende de la herramienta utilizada.

4.7. Enlaces

Los enlaces permiten acceder a páginas web, documentación externa u otros recursos.

La sintaxis es:

```
[Texto que se muestra](https://www.ejemplo.com)
```

Por ejemplo, esto es un enlace.

También se pueden enlazar archivos o apartados del propio documento, siempre que la ruta o el destino sean válidos y la herramienta los admita.

4.8. Imágenes

Para insertar una imagen se utiliza una sintaxis similar a la de los enlaces, añadiendo un signo de exclamación al principio.

```
![Descripción de la imagen](imagenes/arquitectura.png)
```

La ruta puede hacer referencia a una imagen local del proyecto o a una imagen accesible mediante una URL.

La descripción alternativa permite identificar el contenido de la imagen y mejora la accesibilidad.

En un proyecto de ASIR, las imágenes pueden ser capturas de pantalla, diagramas de red, esquemas de arquitectura o evidencias de configuración.

Es importante comprobar que los archivos de imagen se encuentran en la ubicación indicada y que se incluyen al entregar el proyecto.

4.9. Código en línea

Cuando se menciona un comando, una variable, una ruta o un fragmento de código dentro de un párrafo, se utilizan comillas invertidas.

```
Para comprobar la conectividad se utiliza el comando `ping`.

El archivo de configuración se encuentra en `/etc/hosts`.

La variable `PATH` contiene las rutas de búsqueda de ejecutables.
```

Por ejemplo, `esto es código en línea`.

Este formato permite distinguir los elementos técnicos del texto normal.

4.10. Bloques de código

Cuando se necesita mostrar varias líneas de código o un comando completo, se utilizan tres comillas invertidas al principio y al final del bloque.

Se puede indicar el lenguaje para facilitar el resaltado de sintaxis.

Por ejemplo, un bloque de comandos de Bash:

````
```bash
ip addr
ip route
ping 192.168.1.1
```
````

Un bloque de configuración:

````
```ini
[server]
port=8080
```
````

Un bloque SQL:

````
```sql
SELECT usuario, host
FROM mysql.user;
```
````

Los bloques de código son muy útiles para documentar comandos, configuraciones y consultas SQL. El contenido se muestra como código y no debe confundirse con las instrucciones que el propio documento explica.

4.11. Líneas horizontales

Se pueden utilizar tres guiones o más para introducir una separación visual entre secciones.

```
---
```

Se visualizan así:

Es recomendable utilizar este recurso con moderación, ya que los encabezados suelen ser suficientes para organizar el documento.

5. Tablas

Las tablas permiten presentar información estructurada en filas y columnas.

La primera fila contiene los encabezados. La segunda utiliza guiones para separar los encabezados de los datos.

```
| Servicio | Puerto | Protocolo |
|---|---:|---|
| HTTP | 80 | TCP |
```

```
| HTTPS | 443 | TCP |
| DNS | 53 | TCP/UDP |
| SSH | 22 | TCP |
```

El símbolo `|` separa las columnas y los guiones definen la fila de encabezados. La sintaxis de las tablas puede parecer compleja, pero hay múltiples herramientas que convierten una tabla de formato Excel en un

La tabla de arriba se visualiza como:

Servicio	Puerto	Protocolo
HTTP	80	TCP
HTTPS	443	TCP
DNS	53	TCP/UDP
SSH	22	TCP

Conviene no utilizar tablas excesivamente grandes, especialmente cuando el contenido incluye párrafos extensos.

6. Otros recursos útiles

6.1. Caracteres especiales

Algunos símbolos de Markdown tienen un significado especial. Si se necesita mostrarlos literalmente, se puede utilizar una barra invertida delante del símbolo.

```
\*Esto no aparecerá en cursiva\*
\# Esto no será un encabezado
```

La barra invertida permite evitar que determinados caracteres se interpreten como parte de la sintaxis.

7. Ventajas y limitaciones de Markdown

Ventajas

- Es fácil de aprender y utilizar.

- Los archivos ocupan poco espacio.
- Se puede editar con herramientas sencillas.
- Facilita el trabajo colaborativo.
- Se integra con plataformas de control de versiones como Git.
- Permite mantener la documentación cerca del código y de los archivos de configuración.
- Facilita la conversión a otros formatos mediante herramientas compatibles.

Limitaciones

- No todas las herramientas admiten exactamente las mismas funciones.
- El diseño visual depende de la aplicación que muestra el documento.
- Los diseños complejos pueden requerir HTML u otras herramientas.
- La generación de PDF, documentos Word u otros formatos puede necesitar herramientas adicionales.
- Las imágenes y los archivos enlazados deben mantenerse en las rutas adecuadas.
- No sustituye a herramientas especializadas cuando se necesitan diagramas complejos, hojas de cálculo o documentos con maquetación avanzada.

8. Resumen de la sintaxis

Elemento	Sintaxis
Encabezado de nivel 1	`# Título`
Encabezado de nivel 2	`## Título`
Negrita	`**texto**`
Cursiva	`*texto*`
Tachado	`~~texto~~`
Lista sin ordenar	`- Elemento`
Lista ordenada	`1. Elemento`
Tarea pendiente	`- [ ] Tarea`
Tarea completada	`- [x] Tarea`
Enlace	`[texto](URL)`
Imagen	`![descripción](ruta)`

Elemento	Sintaxis
Código en línea	`` `código` ``
Bloque de código	Tres comillas invertidas antes y después
Cita	`> Texto`
Línea horizontal	`---`
Tabla	Columnas separadas por `