

# Guía completa de formato Markdown

## 1. Negrita

Sintaxis:

```
**texto** o __texto__
```

Resultado: **texto**

## 2. Cursiva

```
Sintaxis: *texto* o _texto_
```

Resultado: *texto*

## 3. Tachado

```
Sintaxis: ~~texto~~
```

Resultado: ~~texto~~

## 4. Monoespaciado (inline)

```
Sintaxis: `texto`
```

Resultado:

```
texto
```

## 5. Enlace

```
Sintaxis: [texto] (https://url.com)
```

Resultado: [texto \(https://url.com\)](https://url.com)

## 6. Imagen

```
Sintaxis: ![alt] (imagen.png)
```

Resultado:



## 7. Emoji

```
Sintaxis: :rocket:
```

Resultado: 🚀 (emojins)

## CÓDIGO

Tipo	Sintaxis Markdown	Resultado
Bloque de código	<code>```\ncódigo aquí\n```</code>	bloque de código con formato
Código en Python	<code>```python\nprint("Hola")\n```</code>	resaltado en Python
Código en HTML	<code>```html\n...\n```</code>	resaltado en HTML
Código en JS	<code>```js\nconsole.log()\n```</code>	resaltado en JavaScript
En línea	<code>`variable`</code>	variable

## ➡ LISTAS

- Elemento
    - Sub-elemento
  - \* También sirve
1. Numeradas
  2. Paso 2

## SEPARADORES

texto ---

## ↩ SALTOS DE LÍNEA

Qué lograr	Cómo hacerlo	
-----	-----	
Salto suave	Dos espacios al final + Enter	
Párrafo nuevo	Línea en blanco entre bloques	
Espacio visible	Línea con `&nbsp;` entre bloques	

## TABLAS / CUADROS

Tabla básica: (sin el punto)

.  Campo	Valor	
.  -----	-----	
.  Nombre		
.  Edad		

### Campo Valor

Nombre

Edad

Caja vacía tipo ficha:

## ESTILO VISUAL (usando CSS)

Personalización	CSS sugerido	
-----	-----	
Tamaño de letra	font-size: 12px;	
Color de fondo	background-color: #f9f9f9;	
Color de texto	color: #111;	
Código resaltado	estilo para pre y code	
Columnas	usar tablas	

## EXPORTAR A PDF (recomendaciones)

- Usa fuente monoespaciada (Courier, Fira Code) para código.
- Usa ``&nbsp;`` para controlar espacios.
- Tablas vacías mejor que dibujar cuadros con `|`, `┌`, etc.
- Usa extensión Markdown PDF para mejor control visual.

## EJEMPLO FINAL

### ¿Qué hace esta función?

**remove()** elimina elementos de una lista.

**len()** cuenta cuántos elementos hay.

Sintaxis: `&nbsp;` SALTO DE LINEA

### ⇒ Ejemplo:

```
frutas = ["manzana", "pera", "banana"]
frutas.remove("pera")
print(len(frutas)) # 2
```