

MANUAL COMPLETO DE MARKDOWN

Guía definitiva para dominar el lenguaje de marcado ligero

AUTOR: R de Carrasco Escritor

FECHA: 2025

Créditos

Escritor y redactado por R de Carrasco Escritor

© 2025 R de Carrasco Escritor

Todos los derechos reservados

Tabla de Contenidos

1. [Introducción a Markdown](#)
2. [¿Por qué aprender Markdown?](#)
3. [Sintaxis básica](#)
4. [Encabezados](#)
5. [Énfasis y formato de texto](#)
6. [Listas](#)
7. [Enlaces](#)
8. [Imágenes](#)
9. [Citas](#)
10. [Código](#)
11. [Tablas](#)
12. [Líneas horizontales](#)
13. [Saltos de línea](#)
14. [Caracteres de escape](#)
15. [Markdown extendido](#)
16. [Notas al pie](#)
17. [Listas de tareas](#)
18. [Emojis](#)
19. [Resaltado de sintaxis](#)

20. [HTML en Markdown](#)
21. [Mejores prácticas](#)
22. [Herramientas y editores](#)
23. [Conversión de Markdown](#)
24. [Casos de uso prácticos](#)
25. [Recursos adicionales](#)

Introducción a Markdown

Markdown es un lenguaje de marcado ligero creado por John Gruber en 2004 con la colaboración de Aaron Swartz. Su objetivo principal es permitir a las personas escribir usando un formato de texto plano fácil de leer y escribir, que pueda convertirse en HTML y otros formatos.

El nombre "Markdown" es un juego de palabras con "markup" (marcado), sugiriendo que es una versión simplificada de los lenguajes de marcado tradicionales como HTML o XML.

Filosofía de diseño

Markdown se diseñó con estos principios fundamentales:

- **Legibilidad:** El texto en Markdown debe ser legible como está, sin parecer que ha sido marcado con etiquetas o instrucciones de formato.
- **Simplicidad:** La sintaxis es intuitiva y fácil de aprender.
- **Versatilidad:** Puede convertirse a múltiples formatos de salida.
- **Portabilidad:** Es texto plano, funciona en cualquier plataforma.

¿Por qué aprender Markdown?

Markdown se ha convertido en el estándar de facto para:

Documentación técnica

- README de repositorios GitHub
- Wikis y documentación de proyectos
- Comentarios en plataformas de desarrollo

Escritura y publicación

- Blogs y sitios web estáticos
- Libros electrónicos
- Artículos y publicaciones

Comunicación profesional

- Notas y documentación personal
- Correos electrónicos (en algunas plataformas)
- Presentaciones y slides

Ventajas principales

1. **Independencia de plataforma:** Funciona en cualquier sistema operativo
2. **Control de versiones:** Ideal para usar con Git
3. **Rapidez:** Escribir es más rápido que formatear con un procesador de textos
4. **Enfoque en el contenido:** Te concentras en escribir, no en el formato
5. **Futuro-proof:** Los archivos de texto plano nunca quedan obsoletos

Sintaxis básica

Extensión de archivo

Los archivos Markdown utilizan la extensión `.md` o `.markdown`:

documento.`.md`

README.`.markdown`

Estructura básica

Un documento Markdown es simplemente un archivo de texto plano. No requiere encabezados especiales ni declaraciones, simplemente comienza a escribir.

Encabezados

Los encabezados se crean usando el símbolo `#`. El número de símbolos determina el nivel del encabezado (de 1 a 6).

Sintaxis

```
# Encabezado nivel 1
## Encabezado nivel 2
### Encabezado nivel 3
#### Encabezado nivel 4
##### Encabezado nivel 5
##### Encabezado nivel 6
```

Resultado

Encabezado nivel 1

Encabezado nivel 2

Encabezado nivel 3

Encabezado nivel 4

Encabezado nivel 5

Encabezado nivel 6

Sintaxis alternativa (solo niveles 1 y 2)

Encabezado nivel 1

=====

Encabezado nivel 2

Mejores prácticas

- Siempre deja una línea en blanco antes y después de un encabezado
- Usa solo un encabezado de nivel 1 por documento
- Mantén una jerarquía lógica (no saltes de nivel 1 a nivel 3)
- Los encabezados deben ser descriptivos y concisos

Énfasis y formato de texto

Negrita

Para texto en **negrita**, usa dos asteriscos o dos guiones bajos:

****Este texto está en negrita****

__Este también está en negrita__

Resultado: Este texto está en negrita

Cursiva

Para texto en *cursiva*, usa un asterisco o un guion bajo:

Este texto está en cursiva

Este también está en cursiva

Resultado: *Este texto está en cursiva*

Negrita y cursiva combinadas

```
***Este texto está en negrita y cursiva***  
__Este también__  
**_0 puedes combinarlos así_**  
*_0 de esta manera_*
```

Resultado: ***Este texto está en negrita y cursiva***

Tachado

Usa dos virgulillas para texto tachado:

```
~~Este texto está tachado~~
```

Resultado: ~~Este texto está tachado~~

Mejores prácticas

- Para negrita, prefiere ****** sobre **__** (más compatible)
- Para cursiva, prefiere *** sobre *_*
- No coloques espacios entre los símbolos y el texto
- Evita usar énfasis excesivo, úsalo solo cuando sea necesario

Listas

Listas desordenadas

Usa **-**, ***** o **+** para crear listas desordenadas:

```
- Elemento 1  
- Elemento 2  
- Elemento 3  
  
* También funciona con asterisco  
* Segundo elemento  
  
+ 0 con el signo más  
+ Otro elemento
```

Resultado:

- Elemento 1
- Elemento 2
- Elemento 3

Listas ordenadas

Usa números seguidos de un punto:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Resultado:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Listas anidadas

Indentar con 2 o 4 espacios para crear sublistas:

1. Elemento principal
 - Sub-elemento 1
 - Sub-elemento 2
2. Otro elemento principal
 1. Sub-elemento numerado
 2. Otro sub-elemento numerado

Resultado:

1. Elemento principal
 - Sub-elemento 1
 - Sub-elemento 2
2. Otro elemento principal
 1. Sub-elemento numerado
 2. Otro sub-elemento numerado

Listas con párrafos

Puedes incluir párrafos dentro de elementos de lista:

1. Primer elemento

Este es un párrafo dentro del primer elemento.
2. Segundo elemento

Otro párrafo aquí.

Mejores prácticas

- Mantén consistencia en el símbolo usado (`-` es el más común)
- Usa sangría de 4 espacios para sublistas
- Deja líneas en blanco entre elementos si contienen párrafos
- Los números en listas ordenadas pueden ser todos "1." (se numerarán automáticamente)

Enlaces

Enlaces básicos

[Texto del enlace] (<https://www.ejemplo.com>)

Resultado: [Texto del enlace](https://www.ejemplo.com)

Enlaces con título

[Texto del enlace] (<https://www.ejemplo.com> "Título que aparece al pasar el ratón")

Enlaces de referencia

Útiles cuando usas el mismo enlace varias veces:

[Texto del enlace] [etiqueta]

[etiqueta]: <https://www.ejemplo.com> "Título opcional"

Enlaces automáticos

<<https://www.ejemplo.com>>

<correo@ejemplo.com>

Resultado: <https://www.ejemplo.com>

Enlaces a secciones del documento

[Ir a la sección de tablas] (<#tablas>)

URLs y emails sin formato

Simplemente escribe la URL y muchos procesadores la convertirán automáticamente:

<https://www.ejemplo.com>

Mejores prácticas

- Usa texto descriptivo para los enlaces, evita "haz clic aquí"
- Verifica que los enlaces funcionen antes de publicar
- Usa enlaces de referencia para mantener el texto limpio
- Incluye títulos informativos cuando sea útil

Imágenes

Sintaxis básica

```
![Texto alternativo](ruta/a/imagen.jpg)
```

Imagen con título

```
![Texto alternativo](ruta/a/imagen.jpg "Título de la imagen")
```

Imagen de referencia

```
![Texto alternativo][etiqueta-imagen]
```

```
[etiqueta-imagen]: ruta/a/imagen.jpg "Título opcional"
```

Imágenes con enlaces

Para hacer una imagen clicable:

```
[![Texto alternativo](ruta/a/imagen.jpg)](https://www.ejemplo.com)
```

Rutas de imágenes

Puedes usar rutas relativas o absolutas:

```
![Logo](./imagenes/logo.png)
![Banner](/assets/banner.jpg)
![Remota](https://ejemplo.com/imagen.jpg)
```

Mejores prácticas

- Siempre incluye texto alternativo descriptivo
- Usa formatos web optimizados (JPEG, PNG, WebP)
- Considera el tamaño de archivo de las imágenes
- Usa rutas relativas para portabilidad

- Organiza las imágenes en una carpeta dedicada

Citas

Citas simples

Usa el símbolo `>` para crear citas:

```
> Esta es una cita.  
> Puede ocupar varias líneas.
```

Resultado:

```
    Esta es una cita.  
    Puede ocupar varias líneas.
```

Citas anidadas

```
> Cita de primer nivel.  
>  
> > Cita anidada.  
> >  
> > > Cita doblemente anidada.
```

Resultado:

```
    Cita de primer nivel.  
      
      
    Cita anidada.
```

Citas con otros elementos

Puedes incluir otros elementos Markdown dentro de citas:

```
> ### Encabezado en cita  
>  
> - Lista en cita  
> - Otro elemento  
>  
> Texto en negrita y cursiva.  
>  
> ```  
> Código en cita  
> ```
```

Mejores prácticas

- Usa citas para destacar texto importante o citas literales
- Mantén las citas cortas y relevantes
- Deja líneas en blanco entre párrafos dentro de citas
- Atribuye la fuente cuando cites a otros

Código

Código en línea

Usa acentos graves para código en línea:

Usa la función ``print()`` para mostrar texto.

Resultado: Usa la función `print()` para mostrar texto.

Bloques de código

Usa tres acentos graves para bloques de código:

```
def hola_mundo():  
    print("¡Hola, mundo!")
```

Resultado:

```
def hola_mundo():  
    print("¡Hola, mundo!")
```

Bloques de código con lenguaje

Especifica el lenguaje para resaltado de sintaxis:

```
```python  
def hola_mundo():
 print("¡Hola, mundo!")
```

**\*\*Resultado\*\*:**

```
```python  
def hola_mundo():  
    print("¡Hola, mundo!")
```

Otros lenguajes soportados

```
```javascript
function holaMundo() {
 console.log("¡Hola, mundo!");
}
```

```
<div class="contenedor">
 <h1>Título</h1>
</div>
```

```
.contenedor {
 background-color: #f0f0f0;
 padding: 20px;
}
```

```
#!/bin/bash
echo "¡Hola, mundo!"
```

```
{
 "nombre": "Juan",
 "edad": 30
}
```

## *Bloques de código indentados*

Alternativa: indentar con 4 espacios:

```
```markdown
def hola_mundo():
    print("¡Hola, mundo!")
```

Mejores prácticas

- Siempre especifica el lenguaje para mejor resaltado
- Usa código en línea para nombres de variables y funciones
- Mantén los bloques de código concisos
- Asegúrate de que el código sea correcto y funcional

Tablas

Las tablas se crean usando pipes `|` y guiones `-`:

Sintaxis básica

```
| Encabezado 1 | Encabezado 2 | Encabezado 3 |
|-----|-----|-----|
| Celda 1      | Celda 2      | Celda 3      |
| Celda 4      | Celda 5      | Celda 6      |
```

Resultado:

Encabezado 1	Encabezado 2	Encabezado 3
Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6

Alineación de columnas

```
| Izquierda | Centrado | Derecha |
|:-----|:-----:|-----:|
| Texto    | Texto    | Texto    |
| Más texto| Más      | Más      |
```

Resultado:

Izquierda	Centrado	Derecha
Texto	Texto	Texto
Más texto	Más	Más

- `:---` alinea a la izquierda
- `:---:` centra
- `---:` alinea a la derecha

Formato en tablas

Puedes usar formato Markdown dentro de celdas:

```
| Nombre      | Descripción                | Estado |
|-----|-----|-----|
| **Tarea** | *Descripción importante* | ✓      |
| Otra       | `codigo()`                 | ✗      |
```

Mejores prácticas

- Mantén las tablas simples y legibles
- Alinea visualmente las columnas en el código para facilitar edición

- No uses tablas complejas, considera alternativas
- Las tablas no son parte de Markdown básico, pero están ampliamente soportadas

Líneas horizontales

Las líneas horizontales se usan para separar secciones:

```
---  
  
***  
  
---
```

Todos producen el mismo resultado:

Requisitos

- Usa al menos 3 guiones, asteriscos o guiones bajos
- Pueden tener espacios entre ellos: `- - -`
- Deja líneas en blanco antes y después

Mejores prácticas

- Usa `---` por claridad (menos confusión con énfasis)
- No abuses de las líneas horizontales
- Úsalas solo para separaciones importantes de contenido

Salto de línea

Salto de línea simple

Para crear un salto de línea sin comenzar un nuevo párrafo:

```
Primera línea  
Segunda línea (nota los dos espacios al final de la línea anterior)
```

O usa `
`:

```
Primera línea<br>  
Segunda línea
```

Nuevo párrafo

Deja una línea en blanco para crear un nuevo párrafo:

Primer párrafo.

Segundo párrafo.

Mejores prácticas

- Usa líneas en blanco para párrafos (más claro)
- Los dos espacios al final pueden ser confusos
- Considera usar `
` cuando necesites control preciso

Caracteres de escape

Usa la barra invertida `\` para escapar caracteres especiales:

```
\* No es una lista
\# No es un encabezado
\[No es un enlace\]
```

Resultado:

```
* No es una lista
# No es un encabezado
[No es un enlace]
```

Caracteres que necesitan escape

```
\  barra invertida
`  acento grave
*  asterisco
_  guion bajo
{}  llaves
[]  corchetes
()  paréntesis
#  numeral
+  signo más
-  guion
.  punto
!  exclamación
|  pipe
```

Ejemplo práctico

El precio es `\$100` (no `$100` sin escape)
Para usar `*asteriscos*` sin cursiva

Markdown extendido

Estas características no están en Markdown básico, pero son ampliamente soportadas:

Markdown sabores

- **CommonMark:** Especificación estándar
- **GitHub Flavored Markdown (GFM):** Usado en GitHub
- **Markdown Extra:** Extensiones de PHP
- **MultiMarkdown:** Características avanzadas
- **Pandoc Markdown:** Para conversiones

Notas al pie

Las notas al pie agregan referencias y explicaciones:

Este es un texto con una nota al pie^[1].

Aquí hay otra referencia^[^nota-explicativa].

[¹]: [Esta es la primera nota al pie.](#)

[^{^nota-explicativa}]: [Esta es una nota al pie con identificador descriptivo.](#)

Características:

- Las notas aparecen al final del documento
- Los números se generan automáticamente
- Puedes usar identificadores descriptivos
- Soportado en: Pandoc, MultiMarkdown, algunos editores

Listas de tareas

Útiles para checklists y tareas pendientes:

- [x] Tarea completada
- [] Tarea pendiente
- [] Otra tarea
 - [x] Sub-tarea completada
 - [] Sub-tarea pendiente

Resultado:

- ☒ Tarea completada
- ☐ Tarea pendiente
- ☐ Otra tarea
 - ☒ Sub-tarea completada
 - ☐ Sub-tarea pendiente

Soportado en: GitHub, GitLab, algunos editores

Emojis

Códigos de emoji

Algunos procesadores soportan emojis con código:

```
:smile: :heart: :thumbsup: :rocket: :fire:
```

Resultado (si está soportado):

```
:smile: :heart: :thumbsup: :rocket: :fire:
```

Emojis Unicode

También puedes copiar y pegar emojis directamente:

😊 ❤️ 👍 🚀 🔥

Soportado en: GitHub, Slack, Discord, algunos editores

Resaltado de sintaxis

Sintaxis avanzada

Algunos procesadores permiten resaltar líneas específicas:

```
```python {1,3-4}
def funcion():
 print("Línea 1 resaltada")
 print("No resaltada")
 print("Líneas 3-4 resaltadas")
```

```
Números de línea
```

```
```markdown
```python showLineNumbers
```



```
def funcion():
 return "Con números de línea"
```

**\*\*Soportado en\*\*:** Algunos procesadores web, editores especializados

# HTML en Markdown

Markdown permite HTML cuando necesites más control:

## Casos de uso

```markdown

```
<div style="text-align: center">  
    Texto centrado  
</div>
```

```
<span style="color: red">Texto rojo</span>
```

```
<details>  
    <summary>Haz clic para expandir</summary>  
    Contenido oculto que se revela al hacer clic.  
</details>
```

Elementos HTML comunes

```
<!-- Salto de página -->  
<div style="page-break-after: always;"></div>
```

```
<!-- Tabla con estilos -->  
<table>  
    <tr style="background-color: #f0f0f0">  
        <th>Encabezado</th>  
    </tr>  
</table>
```

```
<!-- Alineación de imágenes -->  

```

```
<!-- Anclajes personalizados -->  
<a name="seccion-especial"></a>
```

Limitaciones

- No todo el HTML funciona en todos los procesadores

- Algunos procesadores sanitizan HTML por seguridad
- Puede reducir la portabilidad
- Usa HTML solo cuando Markdown no sea suficiente

Mejores prácticas

Organización del documento

1. **Usa encabezados jerárquicos:** Mantén una estructura lógica
2. **Una idea por párrafo:** Facilita la lectura
3. **Listas para elementos relacionados:** Mejor que párrafos largos
4. **Espaciado consistente:** Líneas en blanco para separar secciones

Estilo de escritura

1. **Sé claro y conciso:** Evita la verbosidad innecesaria
2. **Voz activa:** Preferible a la voz pasiva
3. **Términos técnicos:** Define al usarlos por primera vez
4. **Ejemplos:** Ilustra conceptos con ejemplos prácticos

Formato

1. **Consistencia:** Mantén el mismo estilo en todo el documento
2. **Énfasis moderado:** No abuses de negritas y cursivas
3. **Código legible:** Usa bloques de código con lenguaje especificado
4. **Enlaces descriptivos:** Texto significativo, no "clic aquí"

Accesibilidad

1. **Texto alternativo:** Describe las imágenes para lectores de pantalla
2. **Encabezados descriptivos:** Ayudan a la navegación
3. **Contraste de colores:** Si usas HTML con estilos
4. **Estructura lógica:** Facilita la comprensión

Mantenimiento

1. **Control de versiones:** Usa Git para rastrear cambios
2. **Nombres de archivo:** Usa nombres descriptivos y consistentes
3. **Comentarios HTML:** Para notas que no deben aparecer
4. **Revisión:** Verifica enlaces y formato antes de publicar

Herramientas y editores

Editores dedicados

Multiplataforma

- **Typora**: Editor WYSIWYG elegante y minimalista
- **Mark Text**: Editor gratuito y de código abierto
- **Obsidian**: Para notas vinculadas y gestión del conocimiento
- **Zettlr**: Orientado a escritura académica
- **Notable**: Editor potente con soporte de tags

Online

- **Dillinger**: Editor en línea simple y efectivo
- **StackEdit**: Editor con sincronización en la nube
- **HackMD**: Colaboración en tiempo real
- **CodiMD**: Editor colaborativo de código abierto

Editores de código

Con soporte Markdown

- **Visual Studio Code**: Con extensiones de Markdown
 - Markdown All in One
 - Markdown Preview Enhanced
 - markdownlint
- **Sublime Text**: Con paquetes de Markdown
- **Atom**: Editor de GitHub con soporte nativo
- **Vim/Neovim**: Para usuarios avanzados

Conversores

Pandoc

El convertidor universal de documentos:

```
# Markdown a HTML
pandoc archivo.md -o archivo.html
```

```
# Markdown a PDF
pandoc archivo.md -o archivo.pdf
```

```
# Markdown a DOCX
```

```
pandoc archivo.md -o archivo.docx
```

```
# Markdown a EPUB
```

```
pandoc archivo.md -o libro.epub
```

Otros conversores

- **kramdown**: Convertidor rápido en Ruby
- **markdown-it**: Parser rápido en JavaScript
- **Marked**: Biblioteca JavaScript popular
- **Python-Markdown**: Implementación en Python

Conversión de Markdown

A HTML

Usando Pandoc

```
pandoc entrada.md -o salida.html --standalone
```

Con plantilla personalizada:

```
pandoc entrada.md -o salida.html --template=plantilla.html
```

Usando herramientas online

- Markdown to HTML (markdowntohtml.com)
- Dingus (daringfireball.net)

A PDF

Pandoc con LaTeX

```
# Requiere instalación de LaTeX
```

```
pandoc entrada.md -o salida.pdf --pdf-engine=xelatex
```

Con opciones:

```
pandoc entrada.md -o salida.pdf \  
  --pdf-engine=xelatex \  
  --variable mainfont="Arial" \  
  --variable fontsize=12pt \  
  --variable geometry:margin=2.5cm
```

Pandoc con wkhtmltopdf

```
pandoc entrada.md -o salida.pdf --pdf-engine=wkhtmltopdf
```

A EPUB

```
pandoc entrada.md -o libro.epub \  
  --metadata title="Mi Libro" \  
  --metadata author="R de Carrasco" \  
  --metadata lang=es \  
  --toc \  
  --epub-cover-image=portada.jpg
```

A DOCX

```
pandoc entrada.md -o documento.docx
```

Con plantilla personalizada:

```
pandoc entrada.md -o documento.docx \  
  --reference-doc=plantilla.docx
```

A presentaciones

Reveal.js

```
pandoc presentacion.md -o presentacion.html \  
  -t revealjs \  
  --standalone \  
  --slide-level=2
```

Beamer (LaTeX)

```
pandoc presentacion.md -o presentacion.pdf \  
  -t beamer
```

Casos de uso prácticos

Documentación de software

README.md típico

```
# Nombre del Proyecto
```

Breve descripción del proyecto.

```
## Características
```

- Característica 1
- Característica 2

– Característica 3

Instalación

```
```bash
npm install mi-proyecto
```

## Uso

```
import { funcion } from 'mi-proyecto';
funcion();
```

## Contribuir

Las contribuciones son bienvenidas. Por favor:

1. Fork el proyecto
2. Crea una rama (`git checkout -b feature/AmazingFeature`)
3. Commit tus cambios (`git commit -m 'Add some AmazingFeature'`)
4. Push a la rama (`git push origin feature/AmazingFeature`)
5. Abre un Pull Request

## Licencia

MIT License - ver [LICENSE.md](#) para detalles

```
Blog personal
```

```
```markdown
```

```
---
title: "Mi Primer Post"
```

```
date: 2025-01-15
```

```
tags: [markdown, escritura, blog]
---
```

```
# Mi Primer Post
```

Hoy comienzo mi blog usando Markdown. Las ventajas son:

1. **Simplicidad**: Solo necesito un editor de texto
2. **Portabilidad**: Puedo mover mis posts fácilmente
3. **Control**: Tengo control total sobre el formato

```
## Por qué Markdown
```

Markdown me permite concentrarme en escribir sin distraerme con el formato. Es perfecto para bloguear.

![[Imagen del post]](imagenes/featured.jpg)

[Continuar leyendo...](#)

Notas de clase

Historia Universal – Clase 5

****Fecha****: 2025-01-15

****Tema****: La Revolución Industrial

Puntos clave

- [] Revisar causas de la Revolución Industrial
- [] Estudiar impacto social
- [] Preparar ensayo sobre innovaciones tecnológicas

Notas

Causas principales

1. ****Revolución agrícola****

- Aumento de productividad
- Liberación de mano de obra

2. ****Acumulación de capital****

- Comercio colonial
- Sistema bancario

Cita importante

> "La Revolución Industrial fue el cambio más fundamental
> en la forma de vida humana desde la invención de la
> agricultura." – Profesor García

Tarea

Leer capítulos 7–9 del libro de texto.

Recetas de cocina

Tarta de Manzana Clásica

 ****Tiempo****: 1 hora 30 minutos

 ****Porciones****: 8

 ****Dificultad****: Media

Ingredientes

Para la masa

- 250g de harina
- 125g de mantequilla fría
- 50g de azúcar
- 1 huevo
- Pizca de sal

Para el relleno

- 6 manzanas
- 100g de azúcar
- 1 cucharadita de canela
- Jugo de medio limón

Preparación

1. ****Preparar la masa****

- Mezclar harina, azúcar y sal
- Incorporar mantequilla hasta obtener textura arenosa
- Agregar huevo y formar bola
- Refrigerar 30 minutos



2. ****Preparar el relleno****

- Pelar y cortar manzanas en rodajas
- Mezclar con azúcar, canela y limón

3. ****Hornear****




- Extender masa en molde
- Colocar manzanas
- Hornear 180°C por 45 minutos

Consejos

-  Usa manzanas Granny Smith para mejor textura
-  Sirve tibia con helado de vainilla

Currículum vitae

Juan Pérez García

 juan.perez@email.com |  +34 600 000 000 |  [LinkedIn]
(<https://linkedin.com/in/juanperez>)

Perfil profesional

Desarrollador Full Stack con 5 años de experiencia en aplicaciones web modernas.

Especializado en React, Node.js y bases de datos SQL/NoSQL.

Experiencia laboral

Senior Developer | **TechCorp S.A.** | 2022 – Presente

- Lideré equipo de 5 desarrolladores en proyecto de e-commerce
- Implementé arquitectura de microservicios
- Reduje tiempos de carga en 40%

Developer | **StartupXYZ** | 2020 – 2022

- Desarrollé aplicación móvil con React Native
- Integré APIs de terceros (Stripe, Firebase)
- Participé en todas las fases del ciclo de desarrollo

Educación

Ingeniería en Informática | Universidad Politécnica | 2016 – 2020

Habilidades técnicas

- **Frontend**: React, Vue.js, HTML5, CSS3, TypeScript
- **Backend**: Node.js, Express, Python, Django
- **Bases de datos**: PostgreSQL, MongoDB, Redis
- **DevOps**: Docker, Kubernetes, CI/CD, AWS

Idiomas

- Español: Nativo
- Inglés: Avanzado (C1)
- Francés: Intermedio (B1)

Recursos adicionales

Especificaciones oficiales

- **CommonMark:** commonmark.org
- **GitHub Flavored Markdown:** github.github.com/gfm
- **Markdown Guide:** markdownguide.org

Tutoriales interactivos

- **Markdown Tutorial:** markdowntutorial.com
- **Learn Markdown:** learnxinyminutes.com/docs/markdown

Cheat sheets

- **GitHub Markdown Cheat Sheet**
- **Markdown Here Cheatsheet**
- **Adam Pritchard's Markdown Cheatsheet**

Comunidades

- **Reddit:** [r/markdown](https://www.reddit.com/r/markdown)
- **Stack Overflow:** Etiqueta [markdown]
- **GitHub Discussions:** En proyectos de Markdown

Libros recomendados

- "The Markdown Guide" por Matt Cone
- "Learning Markdown" por Dean Kaplan
- Documentación de Pandoc para conversiones avanzadas

Herramientas de validación

- **markdownlint:** Linter para mantener calidad
- **markdown-link-check:** Verificar enlaces rotos
- **doctoc:** Generar tabla de contenidos automáticamente

Extensiones del navegador

- **Markdown Here:** Escribe emails en Markdown
- **Markdown Viewer:** Visualiza archivos .md en el navegador

Conclusión

Markdown es una herramienta poderosa y versátil que, una vez dominada, mejorará significativamente tu flujo de trabajo de escritura y documentación. Su simplicidad no limita su potencia: desde documentación técnica hasta libros completos, Markdown puede adaptarse a prácticamente cualquier necesidad.

Próximos pasos

1. **Practica regularmente:** La mejor forma de aprender es escribir
2. **Experimenta:** Prueba diferentes editores y herramientas
3. **Personaliza:** Adapta Markdown a tu flujo de trabajo
4. **Comparte:** Enseña a otros lo que has aprendido

Recuerda

- **Simplicidad es poder:** No necesitas herramientas complejas
- **Contenido primero:** El formato debe servir al contenido, no al revés
- **Portabilidad:** Tus documentos funcionarán en cualquier plataforma
- **Comunidad:** Miles de personas usan y mejoran Markdown cada día

¡Feliz escritura con Markdown!

Créditos finales

Escritor y redactado por R de Carrasco Escritor

Este manual está diseñado para ser una guía completa y práctica de Markdown, desde los conceptos básicos hasta técnicas avanzadas. Puede ser distribuido, modificado y utilizado libremente para fines educativos.

© 2025 R de Carrasco Escritor
Todos los derechos reservados

Versión 1.0 - Enero 2025