

Práctica #0: Uso de Repositorios Markdown

Montes Solis Kimberly

— .✦ Introducción

En este reporte se presentará la práctica realizada utilizando conceptos básicos de **Markdown** y **Git**, explorando su aplicación en la creación y gestión de contenido de manera eficiente.

⇒★ ¿Qué es Markdown?

Markdown es un **lenguaje de marcado ligero** que permite formatear texto de manera sencilla y rápida. Se usa principalmente para:

- ♥ Documentación (README en GitHub)
- ♥ Blogs y artículos
- ♥ Toma de notas
- ♥ Formateo de texto en plataformas web

Su simplicidad lo hace ideal para quienes desean escribir sin preocuparse por código HTML complicado.

☺ ¿Cómo se utiliza?

Markdown se escribe en archivos con **extensión .md** y puede visualizarse en diversas plataformas compatibles como **GitHub**, **GitLab**, **VS Code**, **Jupyter Notebook**, entre otros.

Textos:

```
# Título Principal
## Subtítulo
**Negrita** y *Cursiva*
```

Enlaces e imágenes:

```
[Texto del enlace](https://ejemplo.com)
![Imagen](ruta/imagen.jpg)
```

Tablas:

	Nombre		Edad	
	-----		-----	

Ana	23	
Pedro	30	

Listas:

- Elemento 1
- Elemento 2

Markdown es una herramienta poderosa y fácil de usar para formatear texto de manera rápida y organizada. Su combinación con **Git** permite gestionar contenido de manera eficiente y profesional. *'◇

⇒★ ¿Qué es Git?

Git es un **sistema de control de versiones distribuido** que permite gestionar cambios en proyectos de software de manera eficiente. Se usa para:

- ♥ **Controlar versiones de archivos**
- ♥ **Colaborar en equipo sin sobrescribir cambios**
- ♥ **Mantener un historial de modificaciones**

Git se utiliza en plataformas como **GitHub**, **GitLab** y **Bitbucket** para alojar y compartir código.

★. 🌀 Comandos Básicos de Git

A continuación, se presentan los comandos esenciales para trabajar con Git:

🌀 Configurar Git (solo la primera vez)

```
git config --global user.name "Tu Nombre"
git config --global user.email "tu@email.com"
```

🌀 Inicializar un repositorio

```
git init
```

🌀 Ver el estado de los archivos

```
git status
```

🌀 Agregar archivos al área de preparación

```
git add .
```

✿ Guardar cambios con un mensaje descriptivo

```
git commit -m "Mensaje del commit"
```

✿ Ver historial de commits

```
git log --oneline
```

✿ Conectar un repositorio local con GitHub

```
git remote add origin https://github.com/usuario/repositorio.git
```

✿ Subir cambios al repositorio remoto

```
git push origin main
```

✿ Descargar cambios del repositorio remoto

```
git pull origin main
```

Git es una herramienta esencial para el desarrollo de software, permitiendo trabajar de manera ordenada y colaborativa. Con estos comandos básicos, es posible gestionar versiones de código de forma eficiente.
