

Que es git?

Git es un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar en un proyecto de manera eficiente. Facilita el seguimiento de cambios en el código, la colaboración y la recuperación de versiones anteriores.

para qué sirve?

- **Control de versiones:** Mantiene un historial completo de cambios en los archivos.
- **Colaboración:** Permite trabajar en equipo en el mismo proyecto sin conflictos.
- **Seguridad:** Recupera versiones anteriores del proyecto si algo sale mal.
- **Flexibilidad:** Funciona en proyectos pequeños y grandes.

cómo funciona

1. **Inicialización:** Se crea un repositorio para rastrear los cambios.
2. **Snapshot de cambios:** Guarda el estado de los archivos en un punto específico.
3. **Áreas de trabajo:** Usa tres áreas clave:
 - **Working Directory:** Donde trabajas y editas los archivos.
 - **Staging Area:** Donde preparas los cambios antes de confirmarlos.
 - **Repository:** Donde se almacenan las versiones confirmadas.

explicar 5 comandos de git

¿Qué es Git?

Git es un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar en un proyecto de manera eficiente. Facilita el seguimiento de cambios en el código, la colaboración y la recuperación de versiones anteriores.

¿Para qué sirve Git?

- **Control de versiones:** Mantiene un historial completo de cambios en los archivos.
 - **Colaboración:** Permite trabajar en equipo en el mismo proyecto sin conflictos.
 - **Seguridad:** Recupera versiones anteriores del proyecto si algo sale mal.
 - **Flexibilidad:** Funciona en proyectos pequeños y grandes.
-

¿Cómo funciona Git?

1. **Inicialización:** Se crea un repositorio para rastrear los cambios.
2. **Snapshot de cambios:** Guarda el estado de los archivos en un punto específico.
3. **Áreas de trabajo:** Usa tres áreas clave:

- **Working Directory:** Donde trabajas y editas los archivos.
 - **Staging Area:** Donde preparas los cambios antes de confirmarlos.
 - **Repository:** Donde se almacenan las versiones confirmadas.
-

5 comandos de Git

1. **git init**

Inicializa un nuevo repositorio en el directorio actual.

bash:

```
git init
```

2. **git add**

Añade archivos al área de preparación para confirmarlos.

bash:

```
git add archivo.txt
```

```
git add . # Añade todos los cambios
```

3. **git commit**

Guarda los cambios en el historial del repositorio.

bash:

```
git commit -m "Mensaje descriptivo de los cambios"
```

4. **git log**

Muestra el historial de confirmaciones del proyecto.

bash:

```
git log
```

5. **git status**

Muestra el estado actual del repositorio (archivos modificados, sin seguimiento, etc.).

bash:

```
git status
```