

1. Necesidad de un repositorio de código fuente:

- Un repositorio de código fuente es crucial para gestionar versiones, historial y colaboración en el desarrollo de software. Proporciona un lugar centralizado para almacenar y organizar el código, facilitando la colaboración entre desarrolladores al permitirles trabajar en paralelo, fusionar cambios y mantener un historial de revisiones.

2. Instalación, configuración y comandos básicos:

- Instalación de Git: Descarga e instala Git desde [\[https://git-scm.com/\]\(https://git-scm.com/\)](https://git-scm.com/).
- Configuración global:
git config --global user.name "Tu Nombre"
git config --global user.email "tu@email.com"
- Iniciar un repositorio local: git init
- Añadir archivos al área de preparación:
git add nombre_archivo
git add . (añade todo lo modificado y guardado)
- Realizar un commit: git commit -m "Mensaje del commit".

3. Commits y restauración de archivos:

- Crear archivo: touch archivo.txt
- Realizar commits: git add archivo.txt y git commit -m "Mensaje"
- Restaurar versión anterior: "git log" para obtener el hash del commit y luego "git checkout hash_commit archivo.txt"

4. Cambios de nombres:

- Renombrar archivo: git mv viejo_nombre nuevo_nombre
- Git maneja cambios de nombres automáticamente al usar "git mv"

5. Ignorando archivos:

- Crear ".gitignore" y agregar nombres de archivos o patrones.
- Ejemplo .gitignore:
...
*.log
carpeta_a_ignorar/
...

6. Ramas, uniones, conflictos y tags:

- Crear rama: "git branch nombre_rama"
- Cambiar a rama: "git checkout nombre_rama"
- Fusionar ramas: "git merge nombre_rama_destino".
- Provocar conflictos: Modificar el mismo archivo en dos ramas diferentes.
- Resolver conflictos: Editar manualmente el archivo con conflictos y hacer commit.
- Etiquetar versión: "git tag -a v1.0 -m "Versión 1.0""

7. Stash y Rebase:

- Stash: "git stash" para almacenar temporalmente cambios.
- Rebase interactivo: "git rebase -i HEAD~n", donde 'n' es el número de commits a rebase.

8. Repositorios remotos, Push y Pull:

- Configurar repositorio remoto: "git remote add origin url_del_repositorio"
- Push: "git push -u origin nombre_rama"
- Pull: "git pull origin nombre_rama"

9. Fetch v/s Pull:

- "git fetch" descarga cambios del repositorio remoto pero no los fusiona automáticamente.
- "git pull" descarga y fusiona automáticamente los cambios del repositorio remoto.

10. Clonando un repositorio:

- Clonar: "git clone url_del_repositorio"

11. Documentando un proyecto con Markdown:

- Crear archivo README.md con formato Markdown.

12. Administrando Pull Request:

- Crear Pull Request en GitHub.
- Realizar comentarios y revisiones.
- Fusionar cambios en la rama principal.

13. Flujos de trabajo con GitHub:

- Modelo de bifurcación: Cada colaborador trabaja en su propia bifurcación y envía Pull Requests.
- Modelo de rama por característica: Cada nueva característica se desarrolla en una rama separada antes de fusionarse en la rama principal.