

GIT

- INSTRUCTOR
- I.S.C. E.T.W. M.T.W. CARLOS URIEL DE JESÚS SÁNCHEZ GONZÁLEZ
- [HTTPS://WWW.LINKEDIN.COM/IN/CARLOS-URIEL-DE-JESUS-SANCHEZ-GONZALEZ-8920A1372/](https://www.linkedin.com/in/carlos-Uriel-de-Jesus-Sanchez-Gonzalez-8920A1372/)

GIT

- Git es un sistema de control de versiones distribuido.
- Fue creado por Linus Torvalds (el creador de Linux) en 2005.
- Su objetivo es gestionar el historial de cambios en el código fuente de un proyecto.
- En otras palabras:
- Imagina que escribes un documento y quieres guardar versiones anteriores por si te equivocas.
- Git hace eso mismo, pero de manera muy eficiente y colaborativa para proyectos de software.

¿CÓMO FUNCIONA GIT?

- Git organiza la información en tres “áreas principales”:
- Directorio de trabajo (Working Directory)
 - Es tu carpeta local donde editas archivos. Aquí haces cambios normales (crear, editar, borrar).
- Área de preparación (Staging Area / Index)
 - Es como una “lista de pendientes” de los archivos que quieres incluir en la próxima versión (commit).
 - Se llena con git add.
- Repositorio (Repository / .git)
- Es la base de datos interna de Git donde se guardan los commits (historial de versiones).
 - Los envías aquí con git commit.

FLUJO TÍPICO DE GIT

- Modificas archivos en tu working directory.
- Los agregas al staging con:
 - `git add archivo.txt`
- Creas un commit (guardar versión en el repo local):
 - `git commit -m "Agregué nueva función"`
- Si trabajas con un servidor remoto (GitHub, GitLab, Bitbucket), subes tus commits:
 - `git push origin main`
- Otros colaboradores pueden traer los cambios:
 - `git pull origin main`

CONCEPTOS CLAVES

- Commit → una versión del proyecto en un momento dado.
- Branch → una “rama” de desarrollo independiente.
- Merge → unir ramas distintas.
- Remote → repositorio en la nube (GitHub, GitLab, etc).
- Clone → copiar un repositorio remoto a tu máquina.