

Git y GitHub son herramientas fundamentales para el *desarrollo de software colaborativo* y el *control de versiones*.

Git

Git es un *sistema de control de versiones distribuido* que permite gestionar los cambios en el código fuente de un proyecto. Con Git se pueden guardar distintas versiones de trabajo y revertir a versiones anteriores si es necesario. Es especialmente útil cuando se trabaja en equipo, ya que permite que varios desarrolladores trabajen simultáneamente en el mismo proyecto sin conflictos.

Conceptos Clave en Git:

- **Repositorio**: es el espacio donde Git almacena el proyecto y su historial de cambios. Se puede almacenar localmente (en una computadora) o de manera remota.
- **Commit**: es un registro de cambios en el repositorio. Cada commit representa una versión del proyecto y contiene un mensaje descriptivo sobre los cambios realizados.
- **Branch**: es una línea de desarrollo independiente dentro de un proyecto. Permite trabajar en nuevas características o arreglos sin afectar la rama principal del código (normalmente llamada `main` o `master`).
- **Merge**: combina los cambios de una rama con otra, es muy común fusionar ramas secundarias en la rama principal después de revisar y aprobar los cambios.
- **Pull**: trae los cambios desde un repositorio remoto al repositorio local, sincronizando ambas versiones.
- **Push**: envía los cambios del repositorio local al remoto.

Comandos Básicos de Git:

1. ``git init``: inicializa un nuevo repositorio en una carpeta local.
2. ``git add <archivo>``: agrega cambios en archivos específicos al "stage" para ser confirmados en el próximo commit.
3. ``git commit -m "mensaje"``: crea un commit con un mensaje descriptivo.
4. ``git status``: muestra el estado actual del repositorio y los archivos modificados.
5. ``git log``: muestra el historial de commits.
6. ``git branch <nombre_rama>``: crea una nueva rama.
7. ``git checkout <nombre_rama>``: cambia de rama.
8. ``git merge <nombre_rama>``: fusiona una rama en la rama actual.
9. ``git pull``: actualiza el repositorio local con cambios del repositorio remoto.
10. ``git push``: envía los commits del repositorio local al repositorio remoto.

GitHub

GitHub es una plataforma en línea que almacena repositorios Git de manera remota. Su propósito principal es permitir la colaboración y compartir el código de forma segura. Aunque GitHub es una de las plataformas más populares, existen otras como GitLab y Bitbucket que ofrecen funcionalidades similares.

Funcionalidades de GitHub:

- Repositorios remotos: almacena el código en la nube, permitiendo acceder a él desde cualquier lugar.
- Control de acceso: define permisos para los colaboradores, cosa que es esencial en proyectos a gran escala.
- Pull Requests: permite a los colaboradores proponer cambios en el código. Una pull request incluye los commits hechos en una rama y facilita su revisión antes de fusionarse con la rama principal.
- Issues: herramientas de seguimiento de problemas o tareas. Ayudan a organizar y priorizar el trabajo en equipo.
- GitHub Actions: herramientas de automatización para ejecutar pruebas, desplegar aplicaciones, entre otros, cuando se cumplen ciertas condiciones en el repositorio.
- GitHub Pages: funcionalidad que permite a los usuarios desplegar sitios web estáticos directamente desde un repositorio.
-

Flujo de Trabajo Básico en Git y GitHub

1. Clonar el Repositorio: se ``git clone`` para copiar un repositorio de GitHub en la computadora.
2. Crear una rama: trabajar en una rama separada del código principal para hacer cambios.
3. Hacer Cambios y Commits: realizar cambios y confirmarlos con ``git add`` y ``git commit``.
4. Push a GitHub: subir la rama a GitHub usando ``git push``.
5. Abrir una Pull Request: en GitHub, abrir una pull request para que otros revisen tu código.
6. Revisión y Fusión: después de la revisión, se fusionan la rama con los cambios en la rama principal.