









Glosario de Git para Estudiantes 📖

Términos Básicos (Para Empezar)





1. Git

-  No tiene traducción
-  Sistema que guarda todas las versiones de tu proyecto
-  Como guardar diferentes versiones de tu tarea
-  `git --version` (ver si está instalado)

2. GitHub





-  No tiene traducción
-  Sitio web donde guardas tus proyectos
-  Como una "nube" para tus proyectos
-  Crear cuenta en github.com

3. Repository (repositorio)





-  Repositorio
-  Carpeta de tu proyecto con todo su historial
-  Como una carpeta mágica que recuerda todos los cambios
-  `git init mi-primer-proyecto`

Comandos de Uso Diario





4. **init** (inicializar)

-  Inicializar
-  Crea un nuevo repositorio
-  Como crear un nuevo cuaderno para empezar a escribir
-  git init (dentro de tu carpeta del proyecto)





5. **add** (añadir)

-  Añadir
-  Prepara archivos para guardarlos
-  Como seleccionar qué tareas quieres entregar
-  git add index.html (añade solo ese archivo)





6. **add .** (añadir todo)

-  Añadir todo
-  Prepara todos los archivos modificados
-  Como seleccionar todas las tareas de golpe
-  git add . (añade todos los archivos)





7. **commit** (confirmar)

-  Confirmar cambios
-  Guarda los cambios en el historial
-  Como tomar una foto de tu tarea terminada
-  git commit -m "Agregué el menú de navegación"

8. **push** (empujar)





-  Empujar/subir
-  Sube tus cambios a GitHub
-  Como entregar tu tarea al profesor
-  git push origin main

9. **pull** (tirar)





-  Traer/descargar
-  Descarga cambios de GitHub
-  Como recibir las correcciones del profesor
-  `git pull origin main`

Trabajo con Ramas





10. **branch** (rama)

-  Rama
-  Versión paralela de tu proyecto
-  Como hacer un borrador antes de la versión final
-  `git branch mi-nueva-funcion`

11. **main/master** (principal)





-  Principal
-  Rama principal de tu proyecto
-  Como la versión oficial de tu tarea
-  `git checkout main`

12. **merge** (fusionar)





-  Fusionar
-  Une los cambios de diferentes ramas
-  Como combinar el borrador con la versión final
-  `git merge mi-nueva-funcion`

Comandos de Revisión

13. **status** (estado)





-  Estado
-  Muestra qué archivos has modificado
-  Como ver qué tareas has cambiado
-  `git status`

14. **log** (registro)





-  Registro
-  Muestra el historial de cambios
-  Como ver el historial de versiones de tu tarea
-  git log

Comandos Avanzados





15. **clone** (clonar)

-  Clonar
-  Copia un proyecto de GitHub a tu computadora
-  Como descargar la tarea del profesor
-  git clone https://github.com/usuario/proyecto.git

16. **fork** (bifurcar)





-  Bifurcar
-  Copia un proyecto a tu GitHub
-  Como hacer tu propia copia de un proyecto
-  Se hace desde el botón "Fork" en GitHub

17. **pull request** (solicitud de cambios)





-  Solicitud de extracción
-  Propone cambios a un proyecto
-  Como entregar tu tarea para revisión
-  Se crea desde GitHub después de hacer push

Estados de los Archivos





18. **staged** (preparado)

-  Preparado
-  Archivo listo para commit
-  Como tener la tarea lista para entregar
-  Archivos después de git add

19. **modified** (modificado)

-  Modificado
-  Archivo con cambios sin preparar
-  Como una tarea con correcciones pendientes
-  Archivos que aparecen en rojo en git status

20. **committed** (confirmado)

-  Confirmado
-  Cambios guardados en el historial
-  Como una tarea ya entregada
-  Archivos después de git commit

Consejos

1. **Antes de cada commit:**

- Usa git status para ver qué has cambiado
- Revisa que los archivos sean los correctos

2. **Buenos mensajes de commit:**

- Escribe mensajes claros que expliquen qué cambiaste
- Ejemplo: git commit -m "Agregué el formulario de contacto"

3. **Frecuencia de commits:**

- Haz commit cuando completes una tarea específica. No esperes a tener muchos cambios

4. **Antes de entregar:**

- Usa git push para subir todos tus cambios
- Verifica en GitHub que todo esté actualizado