

# Clase N° 9: Git & Github



# Pull request

- La PR nos permite proponer cambios en una rama para que sean revisados e integrados en otra (**main**)
- Para crear una PR:
  - ◆ Hacer cambios en una feature branch.
  - ◆ Sube los cambios al repositorio remoto con `git push origin [branch-name]`.
  - ◆ En GitHub, ve a la pestaña Pull Requests y haz clic en New Pull Request.
  - ◆ Selecciona las ramas involucradas y proporciona un título/descripción clara de los cambios.

# Introducción a Git

## Beneficios:

- **Flujo de trabajo de rama de función**  
A diferencia de los sistemas de control de versiones centralizados, las ramas de Git son baratas y fáciles de fusionar. Esto facilita el flujo de trabajo de ramas de función tan popular entre muchos usuarios de Git.
- **Desarrollo distribuido**  
Cada desarrollador obtiene su propio repositorio local con un historial completo de confirmaciones.
- **Solicitudes de incorporación de cambios**  
permite a los desarrolladores iniciar debates sobre su trabajo antes de integrarlo con el resto del código base.
- **Comunidad**



# Documentaciones

Oficial: <https://git-scm.com/docs>

Atlassian: <https://www.atlassian.com/es/git/glossary#commands>

Gitlab: <https://about.gitlab.com/images/press/git-cheat-sheet.pdf>

# VS Code: Extensiones recomendadas



## GitLens — Git supercharged v16.0.3

GitKraken [gitkraken.com](https://gitkraken.com) | 36,934,778 | ★★★★★ (833)

Supercharge Git within VS Code — Visualize code authorship at a glance via Git blame annotations and CodeLens, seamlessly navigate and explore Git repositories, gain ...

[Restart Extensions](#)[Disable](#) ▼[Uninstall](#) ▼[Switch to Pre-Release Version](#)☒ Auto Update 

## Git Graph v1.30.0

mhutchie | 9,851,503 | ★★★★★ (619)

View a Git Graph of your repository, and perform Git actions from the graph.

[Install](#)☒ Auto Update

# Comandos

**git init** Inicializa un repositorio Git en el directorio actual

**git clone <url>** Clona un repositorio remoto al directorio local

**git status** Muestra el estado actual del repositorio, incluyendo archivos modificados y no rastreados

**git add <archivo>** Añade un archivo específico al área de preparación (staging area)

**git add .** Añade todos los archivos modificados al área de preparación

**git commit -m "<mensaje>"** Guarda los cambios en el historial con un mensaje descriptivo

**git log** Muestra el historial de commits del repositorio



# Comandos

`git branch` Lista las ramas locales en el repositorio

`git branch <nombre>` Crea una nueva rama

`git checkout <rama>` Cambia a la rama especificada

`git merge <rama>` Combina los cambios de una rama a la actual

`git pull` Actualiza el repositorio local con cambios remotos (fetch + merge)

`git push` Envía los commits locales al repositorio remoto



# Más utilizados diariamente

`git add <archivo>` Añade un archivo específico al área de preparación (staging area)

`git add .` Añade todos los archivos modificados al área de preparación

`git commit -m "<mensaje>"` Guarda los cambios en el historial con un mensaje descriptivo

`git log` Muestra el historial de commits del repositorio

`git checkout <rama>` Cambia a la rama especificada

`git pull` Actualiza el repositorio local con cambios remotos (fetch + merge)

`git push` Envía los commits locales al repositorio remoto





# Comandos avanzados

`git remote` Muestra o gestiona conexiones remotas

`git fetch` Descarga cambios del repositorio remoto sin fusionarlos

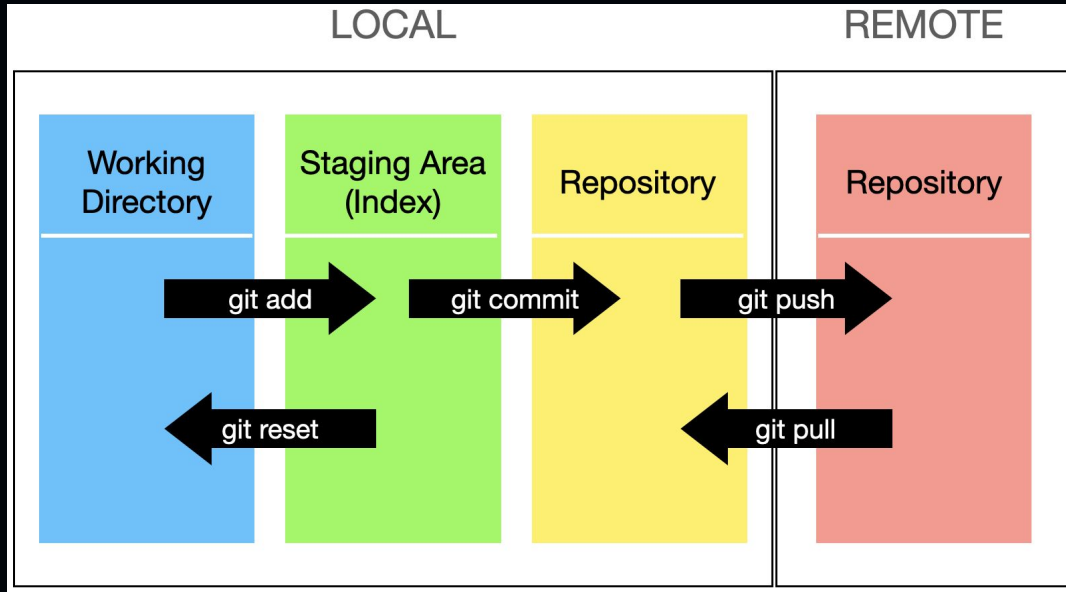
`git tag <nombre>` Crea un marcador en un commit específico (versión, checkpoint)

`git cherry-pick <hash>` Aplica un commit específico a la rama actual

`git revert <hash>` Crea un nuevo commit que revierte los cambios de un commit anterior



# Flujo de trabajo





IN CASE OF FIRE 🔥



1. git commit



2. git push



3. git out!

# Github

GitHub es una **plataforma de alojamiento basada en la nube** diseñada para gestionar repositorios Git. Permite a los desarrolladores almacenar, colaborar y controlar versiones de su código de forma eficiente.

## Funcionalidades principales:

**Control de versiones:** Guarda el historial de cambios del proyecto.

**Colaboración:** Equipos pueden trabajar en el mismo proyecto mediante ramas y solicitudes de cambios (Pull Requests).

**Alojamiento remoto:** Almacena repositorios en la nube, accesibles desde cualquier lugar.

**Integraciones:** Compatible con CI/CD, herramientas de automatización y despliegues.

**Comunidad:** Facilita el descubrimiento y contribución a proyectos de código abierto.



# Practiquemos!

Tiempo: 1 hora

- Crear una cuenta en Github
- Crearemos un repositorio público para nuestro ecommerce
- Descargar el repositorio
- Dividirnos en 4 equipos e implementar las siguientes features
  - Equipo 1: crear una página de About Us con la información de tu empresa
  - Equipo 2: crear la página para ver un producto, con imagen, título y precio
  - Equipo 3: agregar un dropdown en la barra de navegación para ir al perfil y cerrar la sesión (solo el botón)
  - Equipo 4: implementar un formulario para darme de alta en un Newsletter, con email y nombre



# Preguntas