Introducción a Git

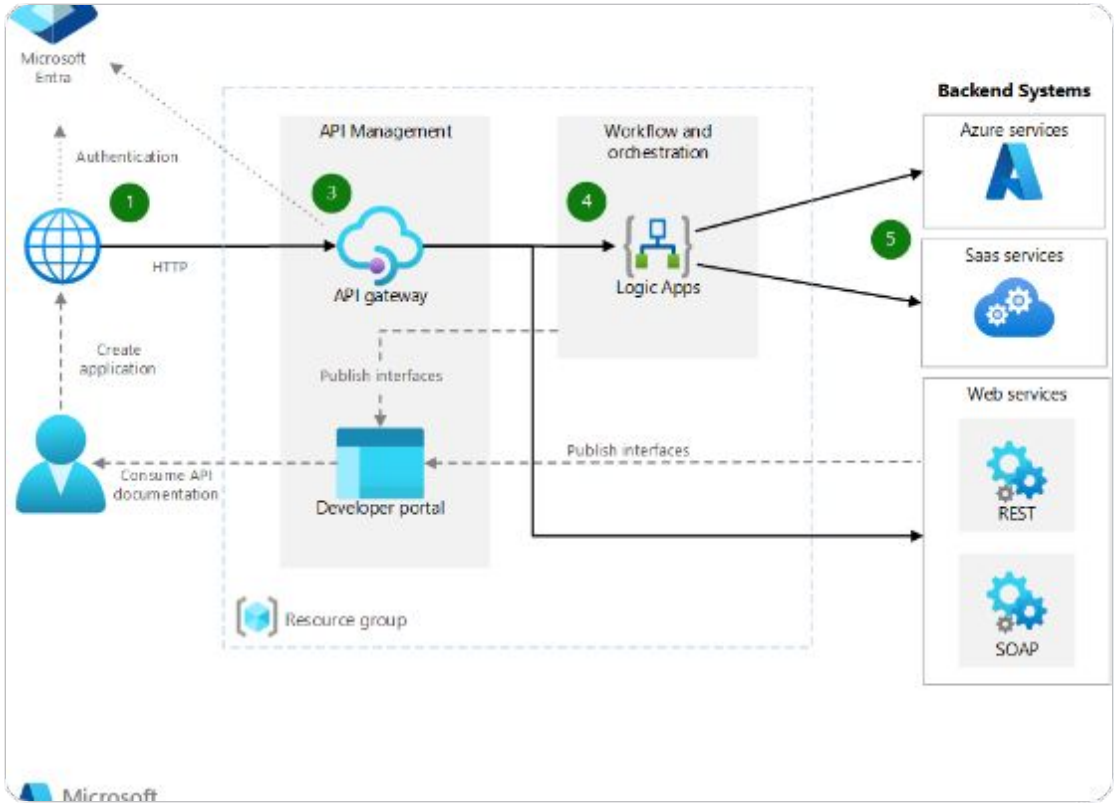
Guía para Principiantes

¿Qué es Git?

¿Qué es Git?

Git es un **sistema de control de versiones distribuido** (DVCS). Piensa en él como un botón de 'Guardar' súper avanzado para tu código.

- Registra "instantáneas" (snapshots) de tu proyecto.
- Te permite trabajar en equipo sin sobrescribir el trabajo de otros.
- Puedes volver a cualquier versión anterior de tu proyecto.



Instalación

Descarga e instala Git desde el sitio web oficial o usa un gestor de paquetes.

Windows: Descarga el instalador desde git-scm.com o usa winget install git.

macOS: Usa Homebrew: brew install git.

Linux (Ubuntu/Debian): sudo apt-get install git

Verifica la instalación con: git --version

Configuración Inicial (¡Solo una vez!)

Preséntate a Git

Antes de usar Git, necesitas presentarte. Esto es importante para que Git sepa quién está haciendo cada cambio.

Esta información se guarda en tus "commits" (guardados).

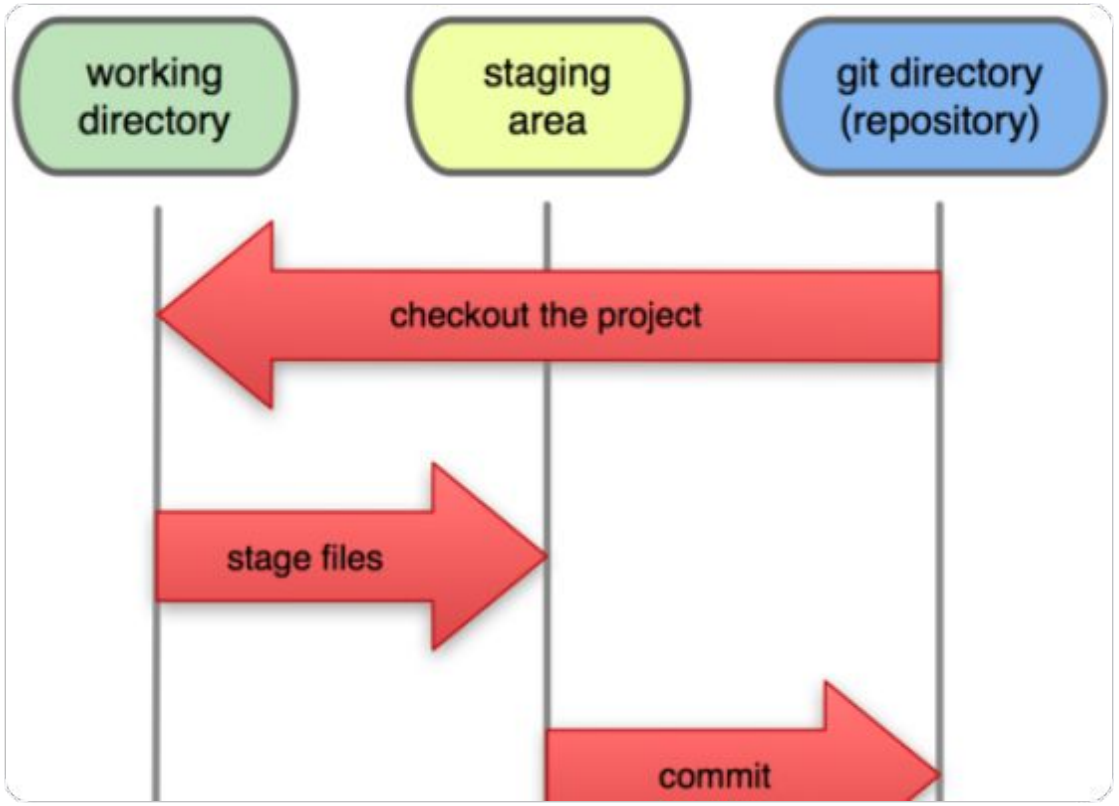
Comandos de Terminal

```
git config --global user.name "Tu Nombre"


git config --global user.email
"tu@email.com"
```

¿Cómo Funciona Git? Las 3 Zonas Clave

- **1. Directorio de Trabajo (Working Directory):** Los archivos que ves y editas en tu carpeta.
- **2. Área de Preparación (Staging Area):** Un área intermedia donde agrupas los cambios que quieres guardar.
- **3. Repositorio (.git):** La base de datos donde Git guarda permanentemente tu historial de cambios (commits).



Flujo de Trabajo Básico (Local)

git init

Inicializa un nuevo repositorio de Git en tu carpeta.

git add .

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

git commit -m "..."

Guarda la "instantánea" del área de preparación en tu repositorio con un mensaje.

Comandos para Verificar tu Trabajo

git status

El comando más importante. Te dice qué archivos están modificados, en staging, o sin seguimiento.

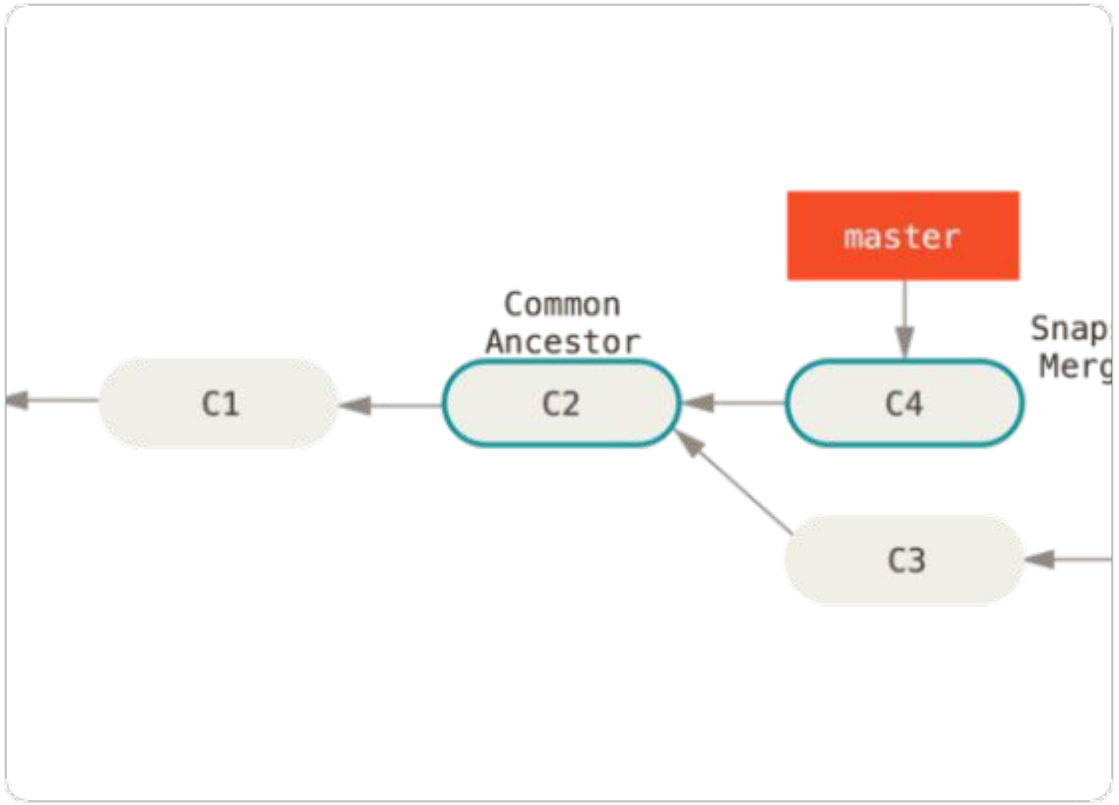
git log

Muestra el historial de todos los commits (guardados) que has hecho en el repositorio.

Ramas (Branches): Trabajar en Paralelo

Las ramas te permiten crear una línea de desarrollo separada para probar nuevas ideas o funciones sin afectar la rama principal (main).

- git branch : Crea una nueva rama.
- git checkout : Se mueve a esa rama para empezar a trabajar.
- git checkout -b : Crea y se mueve a la nueva rama en un solo paso.

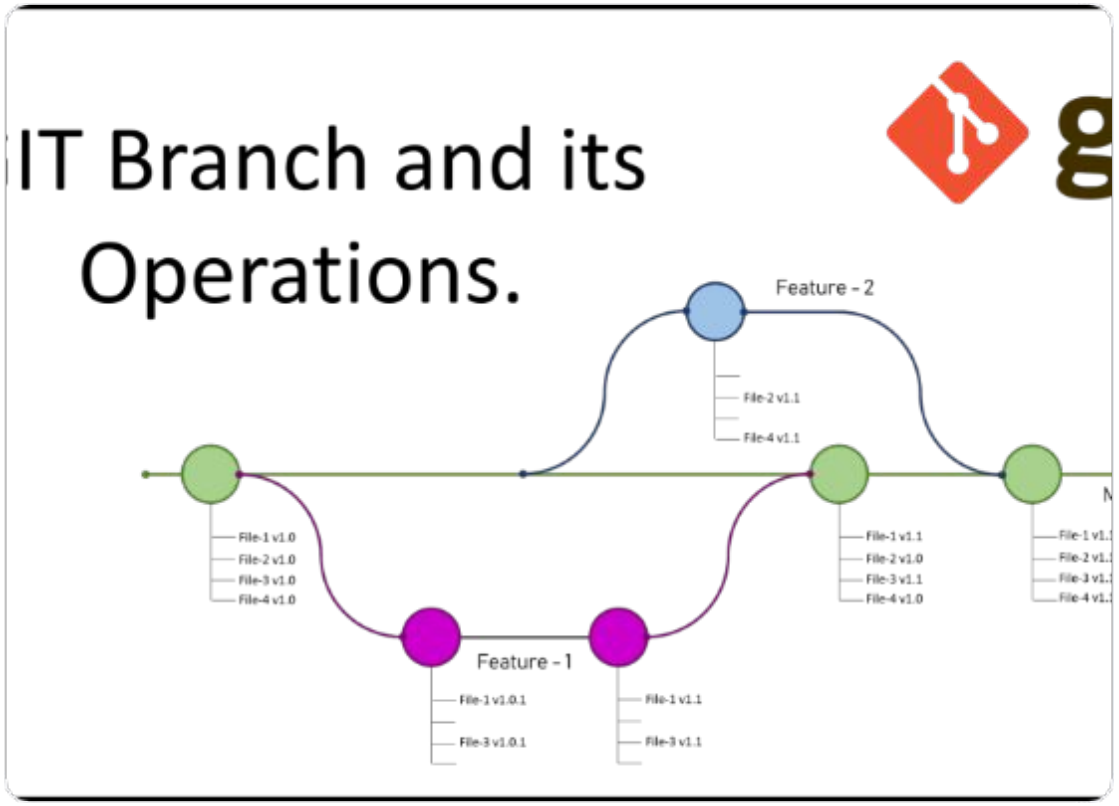


Fusionar (Merge): Unir el Trabajo

Una vez que terminas tu trabajo en una rama, la fusionas de nuevo con la rama principal.

1. Primero, vuelve a la rama principal:
   git checkout main
2. Luego, fusiona tu rama de feature:
   git merge mi-nueva-feature

¡Y listo! Tus cambios ahora están integrados en main.



Trabajando con Remotos (Ej: GitHub)

Un repositorio remoto es una copia de tu proyecto en un servidor (como GitHub, GitLab, etc.).

git clone

Descarga (clona) un repositorio remoto existente a tu máquina local. Esta es la forma más común de empezar a trabajar en un proyecto.

git remote add origin

Conecta tu repositorio local (creado con git init) a un repositorio remoto vacío en GitHub. (Solo se hace una vez).

Sincronización: Push & Pull

git push origin main

Sube tus commits (guardados) locales al repositorio remoto (en la rama 'main'). Comparte tus cambios con el equipo.

git pull origin main

Descarga y fusiona los cambios más recientes del remoto a tu repositorio local. Actualiza tu proyecto con el trabajo de otros.

¿Preguntas?

¡Gracias por tu atención!

Image Sources

https://learn.microsoft.com/es-es/azure/architecture/reference-architectures/enterprise-integration/_images/simple-enterprise-integration.png

Source: learn.microsoft.com

https://aulasoftwarelibre.github.io/taller-de-git/images/git-estados.png

Source: aulasoftwarelibre.github.io

https://git-scm.com/book/es/v2/images/basic-merging-1.png

Source: git-scm.com

https://www.chucksacademy.com/api/proxy/uploads/git_branches_cb50bf87af.webp

Source: www.chucksacademy.com