



Introducción a Git y GitHub

¿Qué es Git?

¿Qué es Git?

Git es un sistema de control de versiones de código abierto que nos permite rastrear cambios en archivos y coordinar el trabajo en equipo.

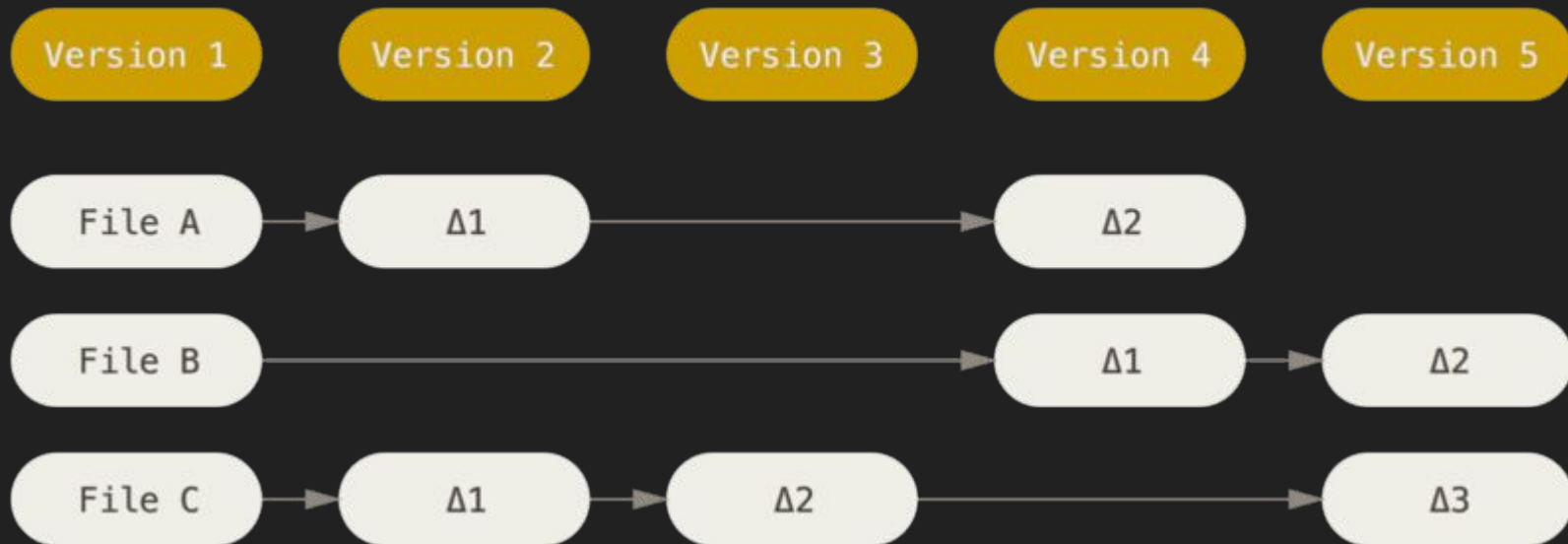
¿Por qué es importante Git?

- Permite tener un historial completo de los cambios realizados en el código, lo que facilita la colaboración y ayuda a evitar errores.
- Permite trabajar en ramas (branches) para probar nuevas funcionalidades o corregir errores sin afectar la versión principal (master).
- Facilita el trabajo en equipo al permitir la colaboración y el intercambio de cambios entre desarrolladores.

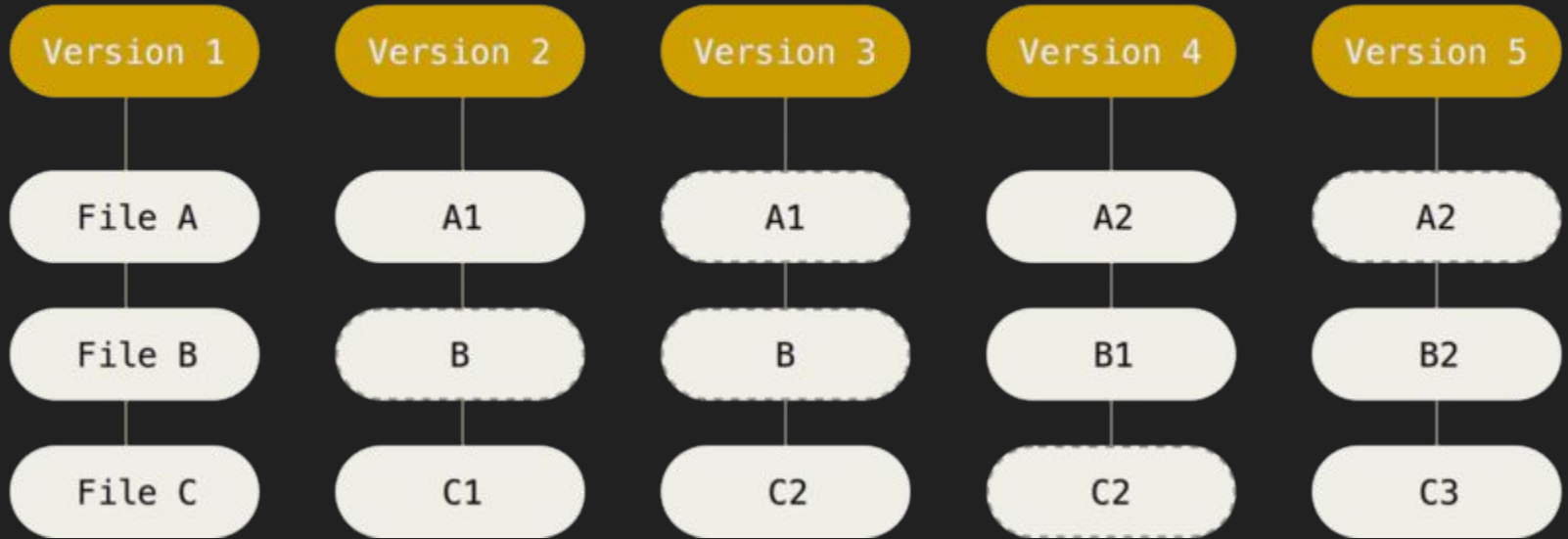
Fundamentos de GIT

Copias instantáneas, no diferencias

Checkins Over Time



Checkins Over Time



Casi todas las operaciones son locales

Los Tres Estados

Working
Directory

Staging
Area

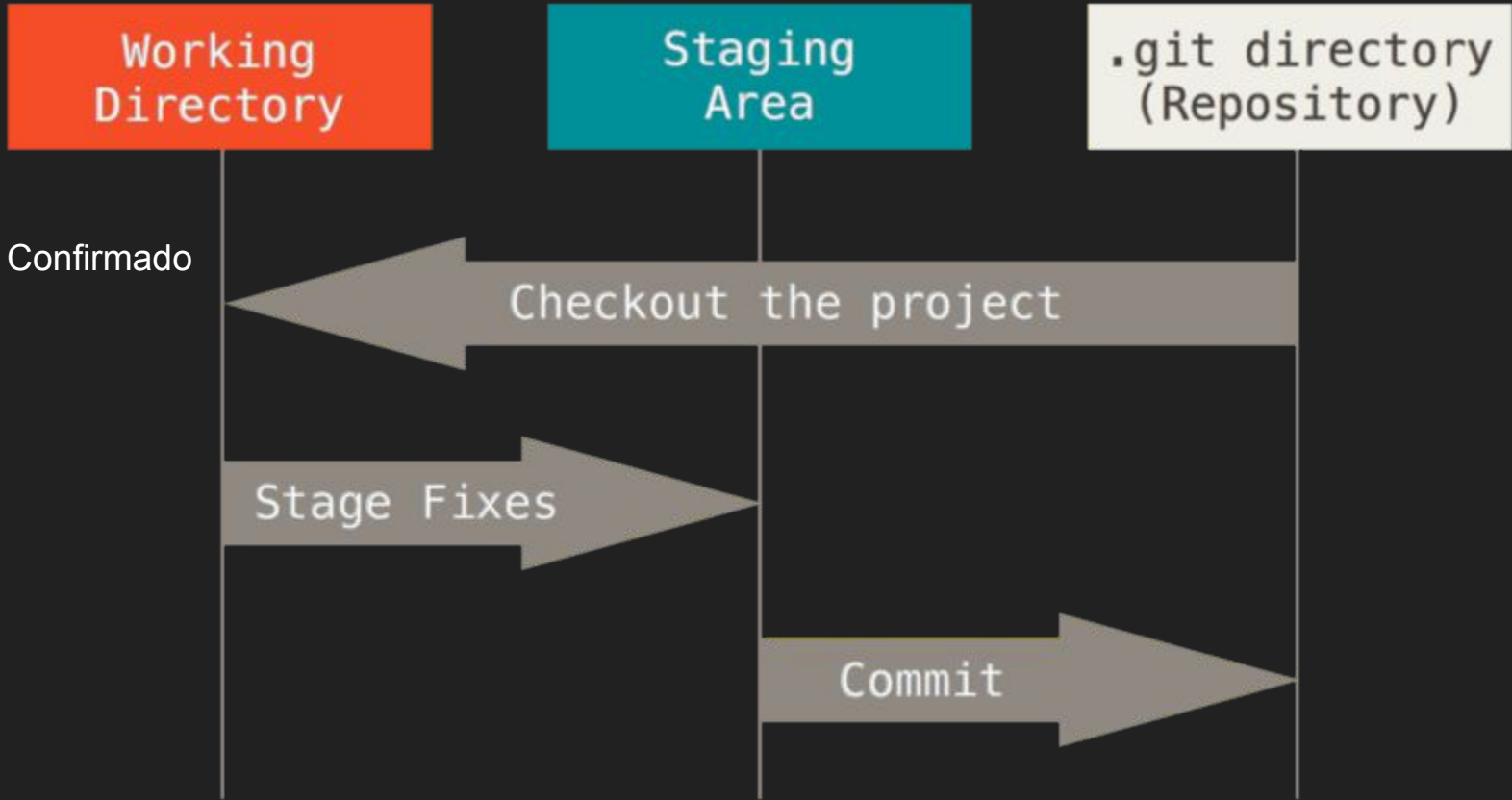
.git directory
(Repository)

1. Confirmado

Checkout the project

Stage Fixes

Commit



Working
Directory

Staging
Area

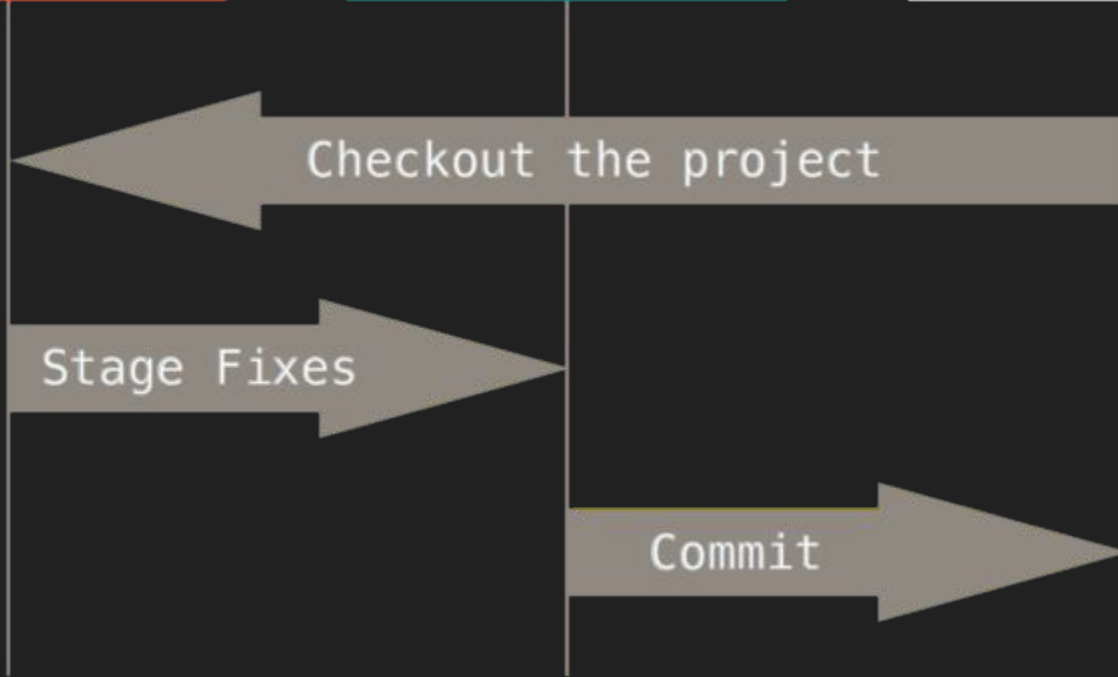
.git directory
(Repository)

1. Confirmado
2. Preparado

Checkout the project

Stage Fixes

Commit



Working
Directory

Staging
Area

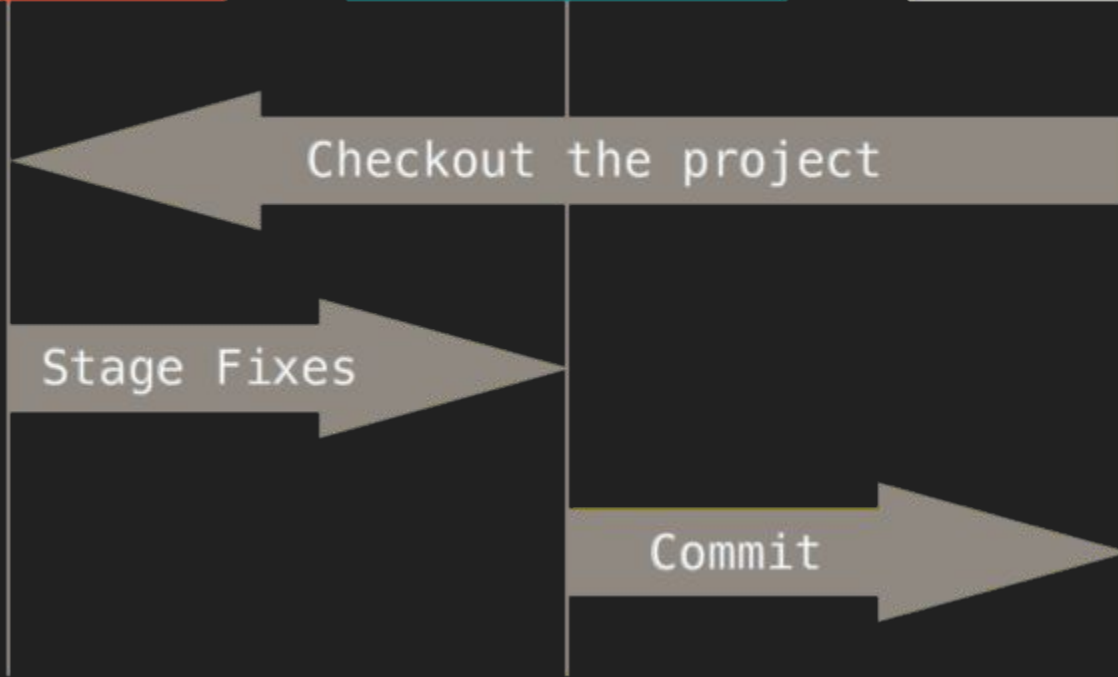
.git directory
(Repository)

1. Confirmado
2. Preparado
3. Modificado

Checkout the project

Stage Fixes

Commit



Instalación y configuración

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

Comandos básicos de Git

git init: inicializa un nuevo repositorio Git

git add: agrega un archivo o directorio al área de preparación (staging area)

git commit: guarda los cambios realizados en el área de preparación en el historial de versiones

¿Qué es GitHub?

¿Qué es GitHub?

GitHub es una plataforma web que utiliza Git para alojar y compartir proyectos de software. Permite a los desarrolladores trabajar en equipo, compartir código y realizar un seguimiento de los cambios.

git push: envía los cambios locales al
repositorio remoto

git pull: descarga los cambios del repositorio
remoto al repositorio local

git checkout: cambia de rama o de commit en el historial

git branch: lista las ramas del repositorio

Trabajo en equipo con Git y GitHub

Branching y merging: cómo trabajar en ramas y fusionar cambios de forma segura

Pull requests: cómo solicitar cambios y
revisar el trabajo de otros desarrolladores

Recomendaciones:

Escribir mensajes de commit descriptivos y significativos

Trabajar en ramas separadas para cada nueva funcionalidad o corrección