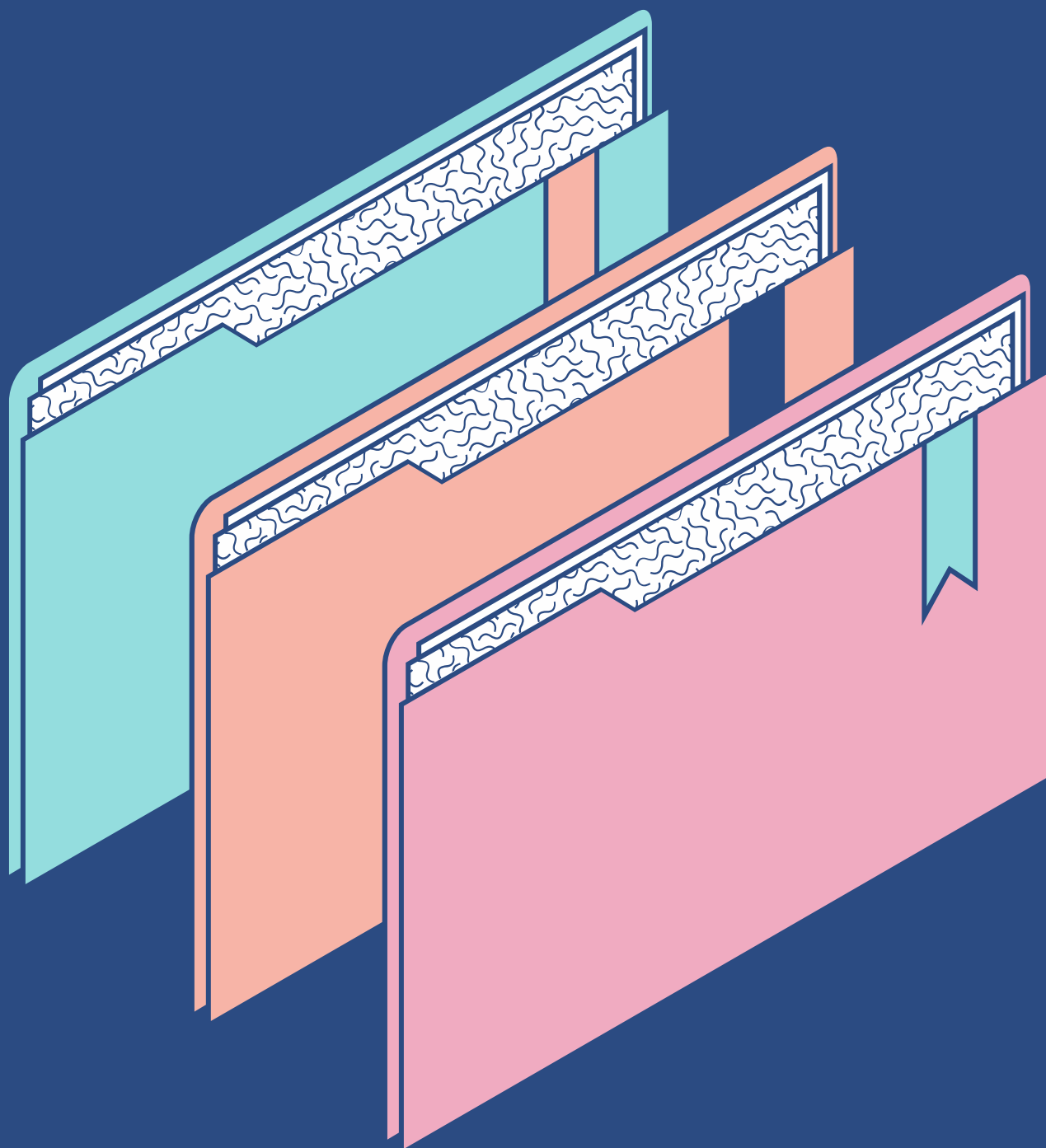


# GIT Y SUS CONCEPTOS BASICOS

ALBERTO ESCAMILLA JASSO



# Introduccion

GIT ES UN SISTEMA DE CONTROL DE VERSIONES DISTRIBUIDO QUE GESTIONA CAMBIOS EN EL CÓDIGO FUENTE Y FACILITA LA COLABORACIÓN ENTRE DESARROLLADORES. PERMITE RASTREAR EL HISTORIAL DE MODIFICACIONES, TRABAJAR EN PARALELO MEDIANTE RAMAS Y FUSIONAR CAMBIOS DE MANERA EFICIENTE. GIT ASEGURA QUE CADA DESARROLLADOR TENGA UNA COPIA COMPLETA DEL PROYECTO, LO QUE FACILITA EL TRABAJO COLABORATIVO Y LA GESTIÓN DE VERSIONES.

# Repositorios

Un repositorio en Git es un espacio donde se almacena el código fuente de un proyecto junto con su historial de cambios. Puede ser local, en la máquina del desarrollador, o remoto, en un servidor en línea. El repositorio local permite trabajar de manera independiente y gestionar el historial de versiones del proyecto, mientras que el repositorio remoto facilita la colaboración al permitir que varios desarrolladores sincronicen y compartan cambios.





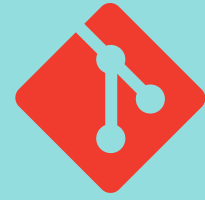
# Ventajas

- **Control Completo del Historial:** Git registra todos los cambios realizados en el código, permitiendo a los desarrolladores revisar y revertir a versiones anteriores fácilmente.
- **Colaboración Eficiente:** Permite a varios desarrolladores trabajar simultáneamente en diferentes ramas y fusionar sus cambios sin sobrescribir el trabajo de otros.
- **Desconexión y Autonomía:** Cada desarrollador tiene una copia completa del repositorio, lo que permite trabajar sin conexión a internet y proporciona una copia de seguridad completa del proyecto.



# Desventajas

- **Curva de Aprendizaje:** Git puede ser complejo para principiantes, con una variedad de comandos y conceptos que pueden resultar difíciles de dominar.
- **Conflictos de Fusión:** En proyectos con muchos colaboradores, pueden surgir conflictos al fusionar cambios, lo que requiere resolución manual y puede ser propenso a errores.
- **Consumo de Recursos:** Git almacena todo el historial del proyecto, lo que puede hacer que los repositorios sean grandes y consuman más espacio en disco y ancho de banda.



# Comandos basicos



1

2

3

4

5

## GIT INIT

Inicializa un nuevo repositorio Git en el directorio actual, creando la estructura necesaria para el control de versiones.

## GIT CLONE <URL>

Crea una copia local de un repositorio remoto usando la URL proporcionada.

## GIT ADD

Agrega archivos al área de preparación (staging area), preparándolos para el próximo commit.

## GIT COMMIT -M

Guarda los cambios en el repositorio con un mensaje descriptivo, creando un nuevo "commit" en el historial.

## GIT PUSH

Envía los commits locales al repositorio remoto, actualizando la versión en el servidor.