

# GIT Y GIT FLOW

# GIT

- Historial de Versiones
  - Seguimiento detallado de cambios en el código
  - Acceso a versiones anteriores en cualquier momento

# GIT

- Colaboración Eficiente
  - Facilita el trabajo en equipo en proyectos de desarrollo
  - Integración sencilla con plataformas de colaboración como GitHub y GitLab

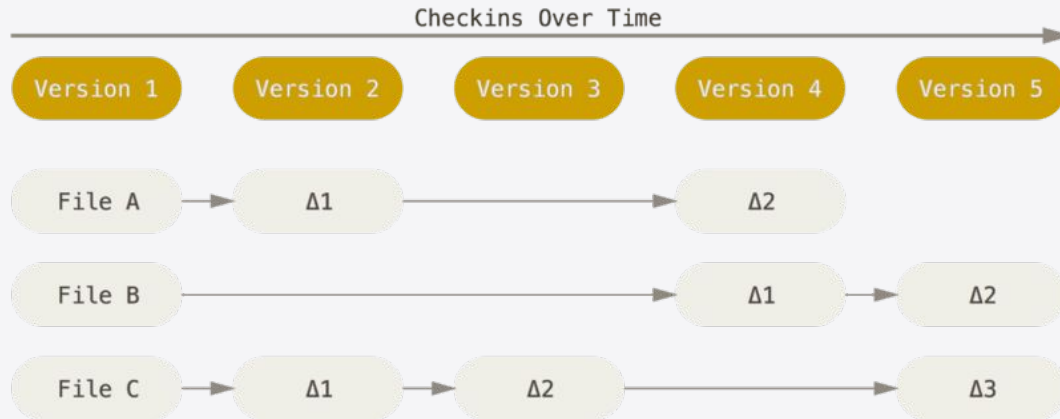
# GIT

- Ramificación (Branching)
  - Permite trabajar en nuevas características o arreglos sin afectar la rama principal
  - Facilita la implementación de metodologías como Git Flow

# GIT

- Distribuido y Descentralizado
  - Cada desarrollador tiene una copia completa del repositorio
  - Permite trabajar sin conexión a internet y fusionar cambios después

# HISTÓRICO DE GIT



# ESTADOS EN GIT

- **Modificado (Modified)**
  - El archivo ha sido modificado desde la última confirmación

# ESTADOS EN GIT

- **Preparado (Staged)**
  - Los cambios han sido marcados para ser incluidos en la próxima confirmación (`git add`)



# ESTADOS EN GIT

- **Confirmado (Committed)**
  - Los cambios han sido confirmados y guardados en el repositorio (git commit)

# GIT FLOW

- Organización de Ramas
  - Define una estructura clara de ramas: master, develop, feature, release y hotfix
  - Cada tipo de rama tiene un propósito específico en el ciclo de desarrollo de software

# GIT FLOW

- Organización de Ramas
  - Facilita un flujo de trabajo coherente y controlado
  - Mejora la calidad del software al separar el desarrollo de nuevas características, las pruebas y la entrega