



***Universidad Tecnológica de Tijuana***

**Tema:**

Strategy Versioning

**Materia:**

Desarrollo móvil Integral

**Grupo:**

10-B

**Nombre del Alumno:**

Sanchez Zamudio Guadalupe

**Docente:**

Ray Brunette Parra Galaviz

**Fecha de Elaboración:**

10/01/2024

## ¿Qué es Git?

Git es un sistema de control de versiones distribuido diseñado para realizar un seguimiento de los cambios en los archivos, especialmente en proyectos de desarrollo de software. Fue creado por Linus Torvalds en 2005 y se ha convertido en una herramienta fundamental para programadores y equipos de desarrollo.

Git permite a los usuarios trabajar de manera organizada, registrar cada modificación en los archivos y colaborar con otros desarrolladores de forma eficiente, incluso si están en ubicaciones diferentes.

### Características principales de Git

#### 1. Control de versiones distribuido:

Cada usuario tiene una copia completa del repositorio, lo que asegura que los datos no se pierdan si el servidor remoto falla.

#### 2. Seguimiento de cambios:

Registra un historial detallado de todas las modificaciones realizadas en los archivos, facilitando el análisis de cambios y la resolución de problemas.

#### 3. Velocidad:

Las operaciones locales, como los commits o las comparaciones de cambios, son muy rápidas gracias al diseño de Git.

#### 4. Soporte para ramas y fusiones:

Permite trabajar en varias ramas de un proyecto de manera simultánea, facilitando la experimentación y el desarrollo de nuevas funcionalidades sin afectar la versión principal.

#### 5. Colaboración eficiente:

Git se integra con plataformas como GitHub, GitLab o Bitbucket, lo que permite a los equipos colaborar en tiempo real y revisar cambios.

## Ventajas de usar Git

**Organización:** Facilita el manejo de proyectos, manteniendo un historial claro de todos los cambios realizados.

**Seguridad:** Cada copia del repositorio incluye todo el historial, lo que protege contra pérdida de datos.

**Trabajo en equipo:** Permite que varios desarrolladores trabajen simultáneamente en el mismo proyecto sin conflictos.

**Flexibilidad:** Admite múltiples flujos de trabajo, como Git Flow, GitHub Flow o Trunk-Based Development.