

# Git

## Nivel – Profesional

(Duración: 18 hrs)

### DESCRIPCIÓN:

Este curso ofrece una formación completa en Git, la herramienta de control de versiones más utilizada en la industria del software. A lo largo del programa, los participantes aprenderán desde los fundamentos básicos para gestionar repositorios locales y sincronizarlos con repositorios remotos, hasta técnicas avanzadas de trabajo colaborativo, resolución de conflictos, versionado y flujos de desarrollo profesional. También se abordarán temas como la reescritura de historial, el uso de ramas, etiquetas, submódulos, automatización con hooks e integración con pipelines de CI/CD. Con un enfoque práctico, el curso prepara a los participantes para trabajar de manera individual y en equipo, optimizando la gestión de proyectos y asegurando buenas prácticas en el desarrollo de software.

### OBJETIVO:

Brindar a los participantes los conocimientos y habilidades necesarios para utilizar Git como herramienta de control de versiones en proyectos de software, desde la gestión individual de repositorios hasta la colaboración en equipo y la implementación de flujos avanzados de desarrollo, fomentando buenas prácticas que optimicen la organización, trazabilidad y calidad del código.

### A QUIÉN VA DIRIGIDO:

- Estudiantes de carreras relacionadas con tecnología que desean iniciar en el uso de control de versiones.
- Desarrolladores principiantes que quieren organizar y dar seguimiento a sus proyectos.
- Profesionales de TI que requieren integrar Git en su flujo de trabajo.
- Personas interesadas en dar sus primeros pasos en el trabajo colaborativo de proyectos de software.

## REQUISITOS:

Para aprovechar al máximo este curso, es recomendable que los participantes cuenten con:

- Conocimientos básicos de programación o desarrollo de software, para contextualizar el uso de Git en proyectos reales.
- Familiaridad con la línea de comandos, ya que gran parte de las operaciones se realizan desde la terminal.
- Comprensión de conceptos básicos de proyectos de software, como archivos, directorios, dependencias y ciclos de desarrollo.
- Nociones de trabajo en equipo en entornos digitales, lo cual facilitará la comprensión de los flujos colaborativos y la integración con plataformas como GitHub o GitLab.

## TEMARIO:

### 1. Introducción y fundamentos

- ¿Qué es el control de versiones y por qué usarlo?
- Conceptos clave: repositorio local vs remoto, área de staging, commit

### 2. Instalación y configuración

- Instalación en diferentes sistemas operativos
- Configuración inicial (git config)
- Llaves SSH y autenticación con GitHub/GitLab

### 3. Creación y conexión de repositorios

- Crear un repositorio local (git init)
- Clonar un repositorio remoto (git clone)
- Conectar repositorio local con GitHub/GitLab (remote add)

### 4. Flujo básico de trabajo

- Ver estado del repositorio (status)
- Añadir cambios (add)
- Guardar cambios (commit)
- Enviar y recibir cambios (push, pull)

### 5. Exploración y recuperación básica

- Revisar historial (log)
- Ver diferencias (diff)
- Deshacer cambios simples (restore, checkout)

## 6. Práctica integradora

- Crear un proyecto, conectarlo a GitHub
- Hacer modificaciones, subir y bajar cambios

## 7. Trabajo con ramas

- Creación y cambio de ramas (branch, switch)
- Combinar ramas con merge
- Estrategias básicas de ramificación (main/dev/feature)

## 8. Colaboración en equipo

- Forks y clones
- Pull requests en GitHub / Merge requests en GitLab
- Buenas prácticas en mensajes de commit

## 9. Resolviendo conflictos

- Qué son los conflictos
- Cómo detectarlos
- Resolución manual y con herramientas gráficas (VS Code, GitKraken)

## 10. Versionado y etiquetas

- Crear y administrar tags (tag)
- Uso de tags para releases

## 11. Stash y reflog

- Guardar cambios temporales (stash)
- Recuperar commits perdidos (reflog)

## 12. Práctica integradora

- Simulación de trabajo colaborativo
- Un integrante abre un PR y otro lo revisa y fusiona

## 13. Reescritura del historial

- Modificar commits (commit --amend)
- Reorganización con rebase
- Reset y revert

## 14. Técnicas avanzadas

- Cherry-pick
- Bisect para localizar errores

### 15. Submódulos y monorepos

- Uso de submódulos
- Alternativa con subárboles

### 16. Automatización y hooks

- Qué son los hooks
- Ejemplos de pre-commit y pre-push
- Integración con linters o pruebas automáticas

### 17. Git en entornos profesionales

- Flujos avanzados: Git Flow vs GitHub Flow
- Integración con CI/CD (GitHub Actions, GitLab CI, Jenkins)

### 18. Mantenimiento y seguridad

- Optimización de repos grandes (gc, filter-repo)
- Uso de .gitignore avanzado
- Seguridad: gestión de credenciales, tokens, llaves SSH

## ¿QUÉ INCLUYE?

- 9 sesiones (2 horas por sesión)
- Clases *live-streaming* impartidas por un instructor certificado
- Acceso a plataforma MTI 24x7 durante la duración del curso:
  - Material electrónico en español
  - Sesiones grabadas
  - Evaluaciones
  - Reconocimiento electrónico MTI

## REQUISITOS DE HARDWARE PARA EL CURSO VIRTUAL

- Conexión a Internet
- Computadora o Laptop
- Tener la aplicación de Zoom instalada
- Aurífonos y micrófono para una comunicación clara (el micrófono integrado de la laptop es suficiente).
- Cámara web opcional pero recomendada para una experiencia más interactiva.

## BENEFICIOS

Adquiriendo nuestros cursos obtendrás las herramientas y formación profesional, gracias a nuestra metodología de aprendizaje que está orientada a que adquieras los conocimientos de una manera práctica y sencilla aumentando tu desempeño laboral, mejorando sus habilidades profesionales y potenciando el crecimiento laboral.



### Descubrimiento

Guiar al participantes para que descubra los conocimientos del tema a tratar.



### Comprensión

Desarrollo y comprensión del tema a tratar.



### Desafíos

Elementos prácticos mediante actividades complementarias a los ejercicios realizados en clase.



### Evaluación

¿Qué he aprendido?



### Prácticas

Actividades para realizar en casa que fortalezcan el conocimiento adquirido.



### Encuesta de satisfacción

Medir el nivel de satisfacción del cliente.