

Git y GitHub: El primer commit a GitHub Actions

Taller • 6 horas • Zoom

Aprende a controlar la historia de tu código y a colaborar como lo hacen los equipos de software profesionales: a guardar cada versión de tu trabajo, moverte entre ideas sin romper lo que ya funciona y recuperar lo que creías perdido. Partirás desde cero —instalación, primer commit y repositorios remotos— y avanzarás hasta los flujos de trabajo que usan las empresas todos los días: ramas, Pull Requests, resolución de conflictos y automatización con GitHub Actions. Todo se hace a mano, en la terminal y en VS Code, sobre repositorios reales. Al terminar no memorizas comandos sueltos, sino que entiendes qué hace cada uno, cuándo usarlo y cómo salir del apuro cuando algo se rompe.

Lo que harás:

- Configurar desde cero tu entorno completo: Git, GitHub, SSH y VS Code.
- Crear tu primer repositorio y llevar un historial de cambios limpio y legible.
- Clonar proyectos y sincronizarlos con push, pull y fetch.
- Trabajar con ramas y resolver conflictos de merge cuando dos cambios chocan.
- Deshacer, corregir y recuperar trabajo con reset, revert, amend, reflog y stash.
- Hacer un fork, abrir un Pull Request y revisar el de alguien más.
- Documentar un proyecto con README y CONTRIBUTING y aplicar Conventional Commits.
- Proteger ramas y montar un flujo de trabajo de equipo (Feature Branches, Git Flow, Trunk-Based).
- Crear tu primer workflow de CI/CD con GitHub Actions.

Herramientas a aprender a usar: Git, GitHub, VS Code, la terminal, SSH y GitHub Actions.

Contenido

Módulo 0: Preparación del entorno

Antes de versionar una sola línea, dejas todo listo para trabajar cómodo: Git instalado, tu cuenta de GitHub creada y VS Code configurado. Estableces el acceso por SSH para no volver a teclear tu contraseña en cada operación y añades las extensiones que te ahorran tiempo desde el primer día.

- a. Instalación de Git
- b. Creación de cuenta en GitHub
- c. Configuración básica con VS Code
- d. Extensiones recomendadas
- e. Configuración SSH

Módulo 1: Fundamentos de Git

El corazón del taller: qué es Git y qué guarda realmente cada vez que haces un commit. Aprendes el ciclo que repetirás miles de veces —init, add, commit— y a leer el estado de tu proyecto: qué cambió, qué falta por guardar y cómo comparar versiones. Con .gitignore mantienes fuera lo que no debe versionarse.

- a. ¿Qué es Git y cómo funciona internamente?
- b. Ciclo básico: init → add → commit
- c. .gitignore
- d. Estado y comparación de cambios: status, diff
- e. Historial: log, log --oneline

Módulo 2: Repositorios remotos y GitHub

Tu código deja de vivir solo en tu máquina. Aprendes a clonar proyectos, a subir y bajar cambios con push, pull y fetch, y a entender la diferencia entre origin y upstream. Decides cuándo conviene SSH y cuándo HTTPS, y empiezas a moverte por GitHub como plataforma: repositorios, Issues y Releases.

- a. Clonar repositorios
- b. push, pull y fetch
- c. Manejo de remotos: origin y upstream
- d. SSH vs HTTPS
- e. Uso práctico de GitHub
- f. Repositorios, Issues y Releases

Módulo 3: Ramas y flujos de trabajo

Las ramas son lo que te permite trabajar en varias cosas a la vez sin romper lo que ya funciona. Aprendes a crearlas, cambiarlas y fusionarlas, a resolver los conflictos que aparecen cuando dos cambios chocan, y conoces los flujos que usan los equipos reales para organizarse: Feature Branches, Git Flow y Trunk-Based Development.

- a. Crear, cambiar y eliminar ramas
- b. Merge
- c. Resolución de conflictos
- d. Visualización de ramas
- e. Feature Branches
- f. Git Flow
- g. Trunk-Based Development

Módulo 4: Reescritura de historial

Git no solo guarda tu trabajo: te deja rehacerlo. Aprendes a deshacer cambios de forma segura (reset vs revert), a corregir el último commit, a rescatar trabajo que creías perdido con reflog y a guardar cambios a medias con stash. Con cherry-pick tomas commits sueltos de una rama y los llevas a otra.

- a. reset vs revert
- b. commit --amend
- c. reflog y recuperación de commits
- d. stash

e. cherry-pick

Módulo 5: Colaboración y Pull Requests

Aquí trabajas como en un equipo de software real. Haces un fork, propones cambios con un Pull Request y aprendes a revisar los de otros. Documentas el proyecto con README y CONTRIBUTING, aplicas Conventional Commits para que el historial se lea solo y proteges las ramas importantes para que nada llegue a producción sin revisión.

- a. Forks
- b. Crear y revisar Pull Requests
- c. Issues
- d. Documentación: README y CONTRIBUTING
- e. Conventional Commits
- f. Protección de ramas

Módulo 6: Temas avanzados y herramientas

Cierras con las herramientas que separan a quien usa Git de quien lo domina: submódulos para manejar proyectos dentro de proyectos, hooks para automatizar tareas, alias para escribir menos y una integración más profunda con VS Code. Terminas con una introducción a CI/CD y creas tu primer workflow con GitHub Actions.

- a. Submódulos
- b. Hooks
- c. Alias útiles
- d. Integración avanzada con VS Code
- e. Introducción a CI/CD
- f. GitHub Actions y creación de un workflow básico

Ya viste lo que vas a dominar.

Asegura tu lugar

Inscríbete ahora · \$600 MXN →

Constancia con registro STPS · Grabaciones por 1 año · Acceso a Zoom al instante

[¿Dudas antes de inscribirte? Escríbenos por WhatsApp](#)

Al terminar tendrás:

- Un flujo de trabajo completo con Git y GitHub, del primer commit hasta un Pull Request revisado y fusionado.
- El criterio para elegir el flujo correcto según el proyecto: cuándo usar Feature Branches, Git Flow o Trunk-Based Development.
- La habilidad de recuperar tu trabajo cuando algo sale mal —reset, revert, reflog, stash— en lugar de entrar en pánico.
- Tu primer workflow de CI/CD funcionando con GitHub Actions.

[Inscríbete aquí](#)

- Un GitHub presentable: proyectos documentados, historial limpio y contribuciones visibles.
- Constancia de participación verificable en línea.

¿Para quién es este taller?

- Desarrolladores y estudiantes que ya programan pero usan Git "a medias" —solo add, commit y push— y quieren entender de verdad lo que hacen.
- Personas que trabajan (o van a trabajar) en equipo y necesitan colaborar sin pisarse el código: ramas, Pull Requests y revisiones.
- Quienes quieren un GitHub presentable —proyectos documentados, historial limpio y contribuciones visibles— para su portafolio o su trabajo.

Este taller NO es para ti si:

- Ya usas Git con soltura a diario: ramas, resolución de conflictos y Pull Requests.
- Buscas un curso de programación. Aquí no aprendes a programar en ningún lenguaje; aprendes a versionar y colaborar sobre el código que ya escribes.
- Esperas administración de servidores Git, GitLab autoalojado o pipelines de CI/CD de escala industrial. Aquí llegas hasta un primer workflow con GitHub Actions, no a la infraestructura de despliegue completa.

Clases

Clases 3 sesiones de 2 horas (6 hrs en total), **martes 8, jueves 10 y viernes 11 de septiembre de 2026, de 20:00 a 22:00** (hora de la Ciudad de México). Cada sesión es práctica: ejecutas los comandos y trabajas sobre repositorios reales en vivo, con la teoría justa para entender qué está pasando. Todo se imparte por Zoom y queda grabado; tendrás acceso a las grabaciones durante 1 año para repasar a tu ritmo.

Instructor

Dr. Uriel Escalona — [LinkedIn](#)

Testimonios

Hemos capacitado a más de 2,000 personas en IA — visita [sitio anterior](#)

Medios de pago

PayPal, transferencia SPEI, Tarjetas de crédito/débito, Link, Apple Pay, Google Pay.

Ver [checkout](#)

Constancia de participación con folio único, verificable en línea

Al terminar recibirás una constancia de participación firmada por el instructor y el director de Actumlogos — agente capacitador registrado ante la Secretaría del Trabajo y Previsión Social. Cada constancia incluye un folio único con el que cualquier persona puede verificar su autenticidad en línea, y podrás compartirla directamente en LinkedIn.

Registro STPS: ZAGE-810930-FW2-0005

Este es un ejemplo para ilustrar que esperar:

