

<Talento  
Tech />

Formaciones de verano

# Git y GitHub

PLAN DE ESTUDIOS

Buenos Aires  
aprende:  
Agencia de Habilidades para el Futuro



MODALIDAD VIRTUAL

DURACIÓN 2 sesiones

*2 hs cada una*

#### FECHAS DE COMIENZO DE CURSADA

**Mañana  
(9–11 h)**

- ▶ 19/01 y 21/01
- ▶ 03/02 y 05/02

**Noche  
(19–21h)**

- ▶ 19/01 y 21/01
- ▶ 20/01 y 22/01
- ▶ 02/02 y 04/02

## Fundamentación

En los entornos de desarrollo actuales, resulta fundamental contar con herramientas que faciliten la organización, la trazabilidad y el trabajo colaborativo. Git y GitHub se han consolidado como los estándares del control de versiones tanto en proyectos individuales como en equipos profesionales. Su adopción no solo permite gestionar cambios de manera eficiente, sino también construir buenas prácticas de documentación, experimentación y colaboración. Este curso está dirigido a personas con **alfabetización digital** y una primera aproximación a áreas como **desarrollo web, análisis de datos, diseño digital o producción de contenidos**, que necesitan incorporar el uso profesional de Git y GitHub dentro de su flujo de trabajo.

El propósito del curso es acompañar a quienes desean **formalizar sus competencias en control de versiones**, aprendiendo a organizar, documentar y gestionar proyectos de manera ordenada y colaborativa. A lo largo del taller, las y los participantes aplicarán un enfoque **práctico, progresivo y contextualizado en escenarios reales**, comprendiendo cómo Git y GitHub se integran en procesos de desarrollo, creación y mantenimiento de proyectos digitales.

## Objetivo general

Adquirir los conocimientos y habilidades necesarias para utilizar Git y GitHub como herramientas de control de versiones y colaboración, aplicándolos en proyectos web individuales y grupales mediante flujos de trabajo claros, ordenados y profesionales.

# Objetivos específicos

*de cada sesión*

## ► **Sesión 1:**

- Comprender los fundamentos del control de versiones.
- Configurar Git en el entorno local de trabajo.
- Ejecutar el flujo completo de trabajo local: creación de repositorios, seguimiento de cambios y registro mediante commits.
- Interpretar el historial de versiones y comprender la relación entre el directorio de trabajo, el área de preparación y el repositorio.

## ► **Sesión 2:**

- Comprender el rol del repositorio remoto y su articulación con el trabajo local.
- Utilizar GitHub para subir proyectos, gestionar ramas y trabajar de manera colaborativa.
- Resolver situaciones comunes tales como fusiones y conflictos.
- Aplicar los conocimientos adquiridos en un proyecto integrador utilizando ramificación y pull requests.

# Materiales necesarios

1. Computadora con Internet.
2. Git instalado en el sistema operativo.
3. Cuenta activa en GitHub.
4. VS Code u otro editor de código.
5. Terminal o consola.
6. Proyecto HTML/CSS de base (provisto por el docente).

# Temario

## SESIÓN 1 | Teoría + Práctica

1. Qué es el control de versiones y cuál es su utilidad en proyectos individuales y colaborativos.
2. Concepto de repositorio local y repositorio remoto.
3. Flujo fundamental de Git: working directory, staging area y commit.
4. Primeros principios de trazabilidad y registro de cambios.
5. Instalación y configuración inicial de Git (nombre y correo).
6. Comandos esenciales: **git init, git status, git add, git commit -m**.
7. Uso básico de la terminal para ejecutar operaciones con Git.
8. Visualización de historial con **git log** y exploración de versiones.
9. Integración de Git con un editor de código (VS Code).
10. Actividad de creación de un repositorio local y realización de los primeros commits.
11. Actividad de análisis del historial, revisión de cambios y modificación del proyecto.

## SESIÓN 2 | Profundización + Proyecto

1. Rol del repositorio remoto dentro del flujo de trabajo.
2. Concepto de rama y su utilidad para el desarrollo paralelo.
3. Sincronización entre trabajo local y remoto: push, pull y fetch.
4. Qué son los conflictos y por qué ocurren.
5. Creación y configuración de repositorios en GitHub.
6. Subida de un proyecto local a un repositorio remoto mediante **git remote add** y **git push**.
7. Creación, cambio y fusión de ramas (**git branch, git checkout, git merge**).
8. Presentación y uso inicial de GitHub Desktop como alternativa visual.
9. Introducción a los pull requests y revisión de cambios.
10. Actividad de creación de una rama, realización de cambios y merge controlado.
11. Actividad de simulación de colaboración mediante la creación de un pull request.

## Cierre del curso

- Reconstrucción del flujo completo: desde la creación del repositorio hasta la colaboración.
- Síntesis de los comandos esenciales en ambos encuentros.
- Buenas prácticas para continuar trabajando con control de versiones.
- Resolución de dudas y recomendaciones finales.

## Requisitos para la Micro credencial

Para completar el curso, el estudiante deberá cumplir con los siguientes requisitos:

1. Asistencia obligatoria a los dos encuentros del curso.
2. Realización y aprobación de los cuestionarios disponibles en el campus virtual, correspondientes a los contenidos trabajados en cada encuentro.

### RECURSOS PARA CONTINUAR APRENDIENDO

1. Guía con comandos básicos.
2. Repositorio de ejemplo.
3. Material complementario sugerido.

# **Resultados del Aprendizaje**

- ▶ Comprender los fundamentos del control de versiones y su importancia en proyectos individuales y colaborativos.
- ▶ Configurar Git correctamente en el entorno local, incluyendo identidad y editor.
- ▶ Utilizar el flujo esencial de Git:
  - ▶ working directory → staging area → commit, interpretando el historial y los cambios.
- ▶ Crear repositorios locales, realizar commits descriptivos y gestionar versiones de manera profesional.
- ▶ Conectar proyectos con repositorios remotos en GitHub mediante git remote, push y pull.
- ▶ Crear y gestionar ramas para organizar el desarrollo paralelo, fusionarlas y resolver conflictos.
- ▶ Subir proyectos, abrir pull requests y participar en un flujo de colaboración simulado.
- ▶ Construir un repositorio ordenado, con mensajes claros y un proceso de control de versiones completo aplicado a un proyecto simple de desarrollo web.

# Actividad Final

## **No condiciona la aprobación del curso**

Esta actividad propone aplicar de manera completa el flujo de trabajo con Git y GitHub desarrollado durante el el curso. Su propósito es que practiques la construcción, gestión y documentación de un repositorio en un entorno simulado de colaboración. Es opcional, no condiciona la aprobación del curso, pero recomendada para consolidar la práctica adquirida.

**Objetivo:** Implementar un proyecto práctico que incluya creación de repositorio, manejo de ramas, commits claros, subida al repositorio remoto y apertura de un pull request, siguiendo buenas prácticas de control de versiones.

## REQUERIMIENTOS DEL PROYECTO

### **1. Construcción inicial del repositorio**

- Crear un repositorio local con un proyecto web simple (HTML y CSS).
- Realizar los primeros commits describiendo los cambios de manera clara y específica.

### **2. Conexión con GitHub**

- Configurar el remoto y subir el proyecto utilizando git remote y git push.

### **3. Trabajo con ramas**

- Crear una rama secundaria para implementar una mejora puntual del proyecto.
- Mantener la rama separada hasta finalizar la funcionalidad.

### **4. Integración mediante pull request**

- Unificar los cambios a través de un pull request en GitHub.
- Redactar mensajes de commit adecuados y comentarios descriptivos en el pull request que expliquen la modificación realizada.