

# Artículo de Práctica: Colaboración con Git y LaTeX

Mauricio<sup>1</sup>

José Miguel<sup>1</sup>

Rodrigo<sup>1</sup>

6 de febrero de 2026

## Resumen

Este documento es un ejercicio práctico para aprender a colaborar en artículos académicos usando Git, GitHub, Overleaf y VS Code. Cada autor contribuirá una sección diferente.

## 1. Introducción

La colaboración efectiva es fundamental en la investigación académica moderna. Los proyectos de investigación involucran cada vez más a equipos distribuidos geográficamente, lo que hace necesario contar con herramientas que faciliten el trabajo conjunto.

En particular, la escritura de artículos académicos presenta desafíos únicos: múltiples autores necesitan editar el mismo documento, mantener un historial de cambios, y asegurar que todos trabajen sobre la versión más reciente.

### 1.1. Motivación

El flujo de trabajo tradicional basado en enviar archivos por correo electrónico presenta numerosos problemas: versiones duplicadas, pérdida de cambios, y dificultad para rastrear quién modificó qué.

### 1.2. Objetivos

Este artículo presenta un flujo de trabajo colaborativo que combina:

- Git para control de versiones
- GitHub para almacenamiento y revisión
- Overleaf para compilación en la nube
- VS Code para edición local eficiente

## 2. Trabajos Relacionados

Existen diversas aproximaciones a la colaboración en documentos académicos. A continuación revisamos las más relevantes.

## 2.1. Herramientas de edición colaborativa

Google Docs y Microsoft Word Online permiten edición simultánea, pero carecen de soporte nativo para LaTeX. Overleaf (Overleaf, 2024) resuelve esto parcialmente al ofrecer un editor LaTeX en línea con colaboración en tiempo real.

## 2.2. Control de versiones en academia

El uso de Git en investigación ha crecido significativamente. Pérez et al. (2024) demuestran que GitHub facilita la reproducibilidad y colaboración en laboratorios de investigación.

## 2.3. Flujos de trabajo híbridos

Algunos equipos combinan múltiples herramientas. El enfoque que presentamos en este artículo sigue esta línea, integrando Overleaf, GitHub y editores locales.

# 3. Metodología

En esta sección describimos las herramientas utilizadas y el flujo de trabajo propuesto para la colaboración en artículos académicos con LaTeX.

## 3.1. Herramientas utilizadas

El flujo de trabajo propuesto se basa en cuatro herramientas principales:

- **Git**: Sistema de control de versiones distribuido que permite rastrear cambios en archivos de texto, crear ramas para desarrollo paralelo, y fusionar contribuciones de múltiples autores.
- **GitHub**: Plataforma en la nube que aloja repositorios Git, proporcionando respaldo automático, Pull Requests para revisión de cambios, e Issues para gestión de tareas.
- **VS Code**: Editor de código con extensión LaTeX Workshop que permite compilación local automática y vista previa del PDF en tiempo real.
- **Overleaf**: Editor LaTeX en línea con compilador en la nube, utilizado como capa de verificación final mediante sincronización con GitHub.

## 3.2. Flujo de trabajo

El ciclo diario de trabajo sigue el orden: **Local** → **GitHub** → **Overleaf**.

1. **Obtener cambios**: `git pull origin main`
2. **Editar localmente**: Escribir y compilar en VS Code
3. **Preparar cambios**: `git add archivo.tex`
4. **Guardar snapshot**: `git commit -m "descripción"`
5. **Subir a GitHub**: `git push origin main`
6. **Verificación final**: Sincronizar Overleaf con GitHub

## 4. Resultados

Implementamos el flujo de trabajo propuesto durante el desarrollo de este artículo. A continuación presentamos los resultados observados.

### 4.1. Beneficios de la colaboración con Git

Durante las dos horas del taller, el equipo logró:

- Configurar un entorno de trabajo compartido
- Realizar múltiples contribuciones en paralelo
- Resolver conflictos de manera sistemática
- Mantener un historial completo de cambios

### 4.2. Comparación con métodos tradicionales

La Tabla 1 muestra las diferencias entre el flujo tradicional (envío de archivos por correo) y el flujo propuesto.

Cuadro 1: Comparación de flujos de trabajo colaborativo

Característica	Tradicional	Git + Overleaf
Control de versiones	Manual	Automático
Historial de cambios	Limitado	Completo
Trabajo simultáneo	Difícil	Fácil
Resolución de conflictos	Ad-hoc	Sistemática
Backup	Manual	Automático
Revisión de cambios	Por correo	Pull Requests

### 4.3. Observaciones

El principal desafío fue la curva de aprendizaje inicial de Git. Sin embargo, una vez dominados los comandos básicos, el flujo de trabajo resultó más eficiente que los métodos tradicionales.

## 5. Conclusión

En este artículo presentamos un flujo de trabajo colaborativo para la escritura de artículos académicos en LaTeX, combinando Overleaf, GitHub y VS Code.

### 5.1. Contribuciones principales

Las principales contribuciones de este trabajo son:

1. Un flujo de trabajo que aprovecha las fortalezas de cada herramienta
2. Guías paso a paso para la configuración del entorno

3. Estrategias para resolver conflictos de manera sistemática
4. Uso de ramas para mantener versiones alternativas del documento

## 5.2. Limitaciones

El flujo propuesto requiere que al menos un miembro del equipo tenga cuenta Premium de Overleaf para la sincronización con GitHub. Además, existe una curva de aprendizaje inicial para usuarios no familiarizados con Git.

## 5.3. Trabajo futuro

Como trabajo futuro, se podría explorar:

- Integración con sistemas de gestión de referencias como Zotero
- Automatización de compilación con GitHub Actions
- Plantillas pre-configuradas para diferentes journals

## Agradecimientos

Agradecemos a todos los participantes del taller por su entusiasmo y colaboración durante el desarrollo de este ejercicio práctico.

## Referencias

Overleaf (2024). Overleaf documentation. <https://www.overleaf.com/learn>.

Pérez, F. et al. (2024). Github is an effective platform for collaborative and reproducible laboratory research. *arXiv preprint arXiv:2408.09344*.