

Guía de Trabajo Colaborativo: Git & GitHub

Cátedra de Tecnología Financiera - UMA

Versión 1.1 / Febrero 2026

1. Introducción: El Control de Versiones

En el desarrollo de software financiero, la trazabilidad es crítica. No podemos depender de archivos llamados `Modelo_Final_v2.xls`.

Utilizaremos Git (el sistema) y GitHub (la plataforma) para crear un **Historial Inmutable** del proyecto. Esto nos permite:

- Colaborar: Trabajar simultáneamente sin sobrescribir el código de otro.
- Auditar: Saber quién hizo qué cambio y cuándo.
- Revertir: Volver a una versión anterior si un cambio rompe el sistema.

PASO 1: Configuración de Cuenta (GitHub.com)

1. Regístrate en github.com usando tu correo institucional (@uma.es).
2. **Importante:** Activa la Autenticación en Dos Pasos (2FA) en `Settings > Password and authentication`. GitHub suele bloquear cuentas nuevas que no lo tengan activado.
3. Si tienes carnet de estudiante, solicita el **GitHub Student Pack** para obtener acceso gratuito a herramientas profesionales.

2. Entorno de Trabajo (GitHub Desktop)

Para facilitar la gestión sin usar comandos de terminal, usaremos la interfaz gráfica oficial.

1. Descarga e instala **GitHub Desktop**.
2. Inicia sesión (**Sign in**) autorizando la conexión desde el navegador.
3. **Configuración de Git:** Asegúrate de poner tu Nombre y Apellidos reales. Esto es lo que aparecerá en el registro de cambios del proyecto.

3. Primeros Pasos: Clonar el Repositorio

«Clonar» significa descargar una copia vinculada del proyecto a tu ordenador local.

1. En GitHub Desktop: `File > Clone repository`.
2. Pestaña **URL**: Pega el enlace del repositorio facilitado por los coordinadores.
3. **Local Path**: Selecciona una carpeta segura (ej: `Documentos/Proyectos_UMA`). Evita carpetas temporales o de descargas.

4. Flujo de Trabajo Diario (Workflow)

El ciclo de vida de un cambio consta de tres estados. Es vital entender este orden:

A. El Commit (Guardado Local)

Cuando modificas un archivo en tu ordenador, GitHub Desktop lo detecta automáticamente.

1. Revisa los cambios en la barra lateral (Verde = Añadido, Rojo = Eliminado).
2. Escribe un Resumen descriptivo (ej: «Ajuste de parámetros de volatilidad»).
3. Pulsa el botón azul **Commit to main**.

Nota: En este punto, el cambio solo existe en TU ordenador.

B. El Push (Sincronización con la Nube)

Para que el equipo vea tu trabajo, debes «empujar» los cambios.

1. Pulsa **Push origin** en la barra superior.
2. Espera a que la barra de progreso termine.

C. El Pull (Actualización)

Antes de empezar a trabajar, siempre debes traer los cambios de tus compañeros.

1. Pulsa **Fetch origin**. Si hay cambios nuevos, el botón cambiará a **Pull origin**.
2. Púlsalo para actualizar tu carpeta local.

5. Diccionario de Conceptos (Equivalencias Financieras)

Para facilitar la comprensión, hemos mapeado los términos técnicos a conceptos de gestión:

Concepto Git	Equivalencia	Definición Operativa
Repository	Libro Mayor	La base de datos centralizada de todo el proyecto.
Commit	Asiento	Registro permanente de un cambio en tu copia local.
Push	Publicación	Envío de tus asientos locales al libro mayor central.
Pull	Conciliación	Actualización de tu copia con los últimos datos del servidor.
Conflict	Discrepancia	Cuando dos personas modifican la misma línea a la vez.

❓ Solución de Problemas Frecuentes

- **Error «Authentication Failed»:** Ve a Options > Accounts, cierra sesión y vuelve a entrar. A veces el token de seguridad caduca.
- **Error al hacer Push:** GitHub te dice que no puedes subir cambios. **Solución:** Casi siempre es porque alguien subió algo antes que tú. Dale a **Fetch/Pull** primero, y luego intenta el Push de nuevo.
- **Archivos extraños (.DS_Store, pycache):** Si ves archivos basura en la lista de cambios, haz clic derecho sobre ellos y selecciona **Ignore file**.

REGLA DE ORO DEL EQUIPO:

Siempre haz **Pull** antes de empezar a trabajar y **Push** al terminar la sesión.