ITESO
Proyecto PAP
Primavera 2026
Prof. Luis Alvarado

ITESO, Universidad Jesuita de Guadalajara

Manual de Git para trabajo colaborativo (GitHub)

0) Reglas del equipo (léelas antes de empezar)

1. Nadie hace push directo a main.
2. Cada alumno trabaja en una rama con su nombre (o usuario de GitHub).
    - Ejemplo: fajardo, anapau, arantza, ursula
3. Cada cambio se sube a su rama y se integra a main con Pull Request en GitHub.
4. Antes de empezar a trabajar cada día: actualiza tu rama con los cambios de main
5. Mensajes de commit claros: “Agrego función X”, “Corrijo bug Y”, “Actualizo README”, etc.

---

1) Clonar el repositorio (primera vez)

En una carpeta donde vas a trabajar:

git clone https://github.com/LuisCarlosAlvaradoG/PAP_P2026.git
cd PAP_P2026

Verifica el remoto:

git remote -v

Deberías ver:

origin https://github.com/LuisCarlosAlvaradoG/PAP_P2026.git (fetch)
origin https://github.com/LuisCarlosAlvaradoG/PAP_P2026.git (push)

---

2) Crear tu rama personal

Opción recomendada: rama con tu nombre.

1. Asegúrate de estar en main y actualizado:

git switch main
git pull origin main

2. Crea tu rama y cámbiate a ella:

git switch -c tu-nombre

3. Súbela a GitHub por primera vez:

git push -u origin tu-nombre

Desde ahora, trabajarás SIEMPRE en tu-nombre. (**“tu_nombre” debe de ser sustituido literalmente por tu nombre, no pongas espacios, mejor usa guiones bajos)**

---

3) Flujo diario de trabajo

A) Antes de empezar a programar (actualiza tu rama)

1. Trae lo nuevo del repo:

git fetch origin

2. Actualiza tu main local:

git switch main
git pull origin main

3. Regresa a tu rama y mézclale lo nuevo de main:

git switch tu-nombre
git merge main

Si todo sale bien, listo.

B) Hacer cambios y subirlos

1. Revisa qué cambió:

git status

2. Agrega archivos al “stage”:

- Para agregar todo:

git add .

3. Haz commit:

git commit -m "Mensaje claro del cambio"

4. Sube tu rama a GitHub:

git push

---

4) Integrar tu trabajo a main (Pull Request)

Cuando ya tengas una parte lista:

1. Asegúrate de haber actualizado tu rama con main (sección 3A).
2. Sube tus commits (git push).
3. En GitHub:

- Ve al repo → te aparecerá tu rama → "Compare & pull request"
- Describe qué hiciste
- Asigna a revisión
- Crea el PR

Cuando se aprueba el PR, se hace merge a main.

---

5) Después de que tu PR se integró

Para traer el main actualizado:

git switch main
git pull origin main

Y si quieres seguir trabajando en tu rama con la base nueva:

git switch tu-nombre
git merge main

Conflictos al hacer merge main

Un conflicto pasa cuando tú y otra persona editaron la misma parte del mismo archivo.

1. Git te avisará y marcará conflicto.
2. Abre el/los archivos con conflicto y busca marcas tipo:

<<<<<<< HEAD
=======
>>>>>>> main

3. Decide qué versión queda (o mezcla ambas) y borra las marcas.
4. Luego:

git add .
git commit -m "Resuelvo conflictos con main"
git push

Mi push fue rechazado
Significa: tu rama remota tiene cambios que tu rama local no tiene (o no actualizaste antes).
Haz:

git fetch origin
git merge origin/tu-nombre
git push

Recomendaciones para proyectos con notebooks

- No trabajen todos sobre el mismo .ipynb a la vez (poco eficiente para entregas donde todos deben editar).
- Recomendación personal:
  - Notebook solo para correr y documentar resultados. Mantengan fórmulas, funciones y desarrollo en archivos.py, y en el notebook únicamente muestre los resultados o plots.
  - Ejemplo: funciones.py, gráficos.py mantienen el código, en el notebook solo muestran los resultados.

Checklist
Inicio del día
git fetch origin
git switch main
git pull origin main
git switch tu-nombre
git merge main
Subir trabajo
git add .
git commit -m "mensaje"
git push
Entregar

- Crear Pull Request desde tu rama → main en GitHub.