

# Informe del grupo

Cristian A. Jorge H. Jhon R.

## 1. Pasos realizados:

- Creamos el repositorio ToDoApp-Group2 en Github
- Cada miembro creó una rama (Cristian, Jhon, Jorge)
- Implementamos funcionalidades y resolvimos conflictos

## 2. Comandos Git

- `git clone <URL>`: Clonar el repositorio.
- `git checkout -b "nombre"`: crear rama
- `git add <archivos>`: Añadir cambios
- `git commit -m "mensaje"`: Hacer commit
- `git push origin "nombre"`: Empujar rama
- `git merge Jhon`: Fusionar rama
- `git push origin -delete Cristian`: Eliminar rama

## 3.

- Conflicto en `task_model.py` entre `is-complete/is-done` y `mark-as-complete/set-done`.
- Solución: combinamos métodos, manteniendo `is-completed` como atributo principal.

#### 4. Contribuciones

- Cristian: Añadió mark-as-complete y actualizó main.py
- Jorge: Añadió delete-task y actualizó README.md
- Jhon: Simuló conflicto con set-done/remove-task y lo resolvió

#### 5. Reflexiones

- Aprendimos a coordinar ramas y resolver conflictos en equipo
- Git facilita el trabajo colaborativo, pero requiere comunicación clara
- Los Pull Requests ayudan a revisar cambios antes de fusionar

#### 6. Preguntas Finales

a. ¿Cómo coordinaron el trabajo en equipo para evitar conflictos innecesarios?

R/ Asignamos tareas por rama y comunicamos los cambios antes de hacer merge

b. ¿Qué aprendieron sobre la resolución de conflictos en Git?

R/ Aprendimos a identificar, revisar y fusionar manualmente cambios en conflicto

c. ¿Por qué es importante usar Pull Requests en proyectos colaborativos?

R/ Permiten revisar, discutir y validar cambios antes de integrarlos al proyecto

d. ¿Cómo podría mejorarse el flujo de trabajo con herramientas como CI/CD? R/ Automatizando pruebas y despliegues para detectar errores