

Informe del proyecto - Ivan Riaño, Jhoser Pabuenza y Cristian Villarreal

1. Pasos realizados:

- Creamos el repositorio público TodoAPP-Group99 en Github.
- Ivan Riaño (estudiante 1) inicializó el proyecto con los archivos main.py, task-model.py y .gitignore.
- Cada miembro del grupo creó su propia rama: ~~main~~
Jhoser Pabuenza y Cristian Villarreal. Ivan Riaño
- Cada estudiante trabajó una funcionalidad distinta en su rama individual.
- Se creó una rama de grupo (group-99) donde se funcionaron las ramas individuales.
- Durante el proceso, se presentaron conflictos en el archivo task-model.py, que fueron resueltos manualmente.
- Finalmente, se creó un Pull Request para funcionar group-99 con main, completando así el trabajo colaborativo.

2. Comandos Git Utilizados:

- `git clone <https://github.com/IvanRiaño170904/ToDoApp-Group99>`: Clonar el repositorio del Grupo.
- `git checkout -b <IvanRiaño, JhasePabvena, CristianVillurreal>`
Crear nueva rama.
- `git add`: Añadir cambios al Área de Preparación (staging).
- `git commit -m "mensaje"`: Guardar cambios con descripción.
- `git push origin <rama>`: Subir una rama al repositorio remoto.
- `git fetch` y `git pull`: Traer los últimos cambios del repositorio remoto.
- `git merge <rama>`: Unir ramas y cambios.
- `git push origin --delete <rama>`: Eliminar ramas remotas después de fusionar.
- `git log --oneline --graph --all`: Ver la historia visual del Proyecto.

3. Conflictos:

- Se produjo un conflicto en el archivo `task_model.py` al fusionar métodos similares: `mark-as-completed()` y `set-done()`, así como los atributos `is-completed` y `is-done`.
- También hubo conflicto con métodos ~~para~~ ~~eliminar~~ ~~tareas~~ para eliminar tareas: `delete-task()` y `remove-task()`.
- Solución: Se unificaron los métodos y se usó `is-completed` como el atributo principal para indicar el estado de la tarea.

4. Contribuciones:

- Iván Riaño (Estudiante 1): Inicializó el repositorio, implementó `mark-as-completed()`, resolvió los conflictos de fusión, y realizó el trabajo de estudiante 3.
- Jhayer Pubuera (Estudiante 2): Implementó `delete-task()` y redactó/actualizó el `README.md`.
- Cristian Villarruel (Estudiante 3): No pudo participar directamente en GitHub por no tener acceso a un computador, pero estuvo presente en todas las reuniones del grupo, colaboró activamente en la planeación, y fue el responsable de redactar el informe a mano.

5. Reflexiones:

- Aprendimos a trabajar en equipo usando Github como herramienta de control de versiones.
- Las ramas individuales permitieron dividir el trabajo sin conflictos al inicio.
- Resolver conflictos nos enseñó a combinar código de forma ordenada y cuidadosa.
- La comunicación fue clave para coordinarnos, especialmente cuando un compañero no tenía acceso técnico.
- Git y Github resultan herramientas poderosas para el trabajo colaborativo y el seguimiento de cambios.