

## Documento - "Octavio Posada - Jandy Tete - Randy Fornari"

### 1- Actividades Realizadas

- Se creó el reposito TODOAPP-GrupoX en la plataforma.
- Se invitó a los miembros del equipo y se descargo en cada maquina.
- Cada estudiante del grupo creo su propia rama.
- Las funcionalidades fueron realizadas individualmente dentro de cada una de las ramas.
- Se creo una rama colectiva llamada grupo con el fin de integrar todo el trabajo.
- Se resolvieron los conflictos generados por cambios simultaneos en los archivos
- Finalmente, se eliminaron tanto las ramas individuales como la rama del grupo

### 2- Comandos GIT

- git clone <URL> → clonar el reposito desde Github
- git checkout -b nombre-rama → Crear y cambiar a una nueva rama
- git commit -m "mensaje" → Confirmar los cambios con un mensaje descriptivo
- git Push origin nombre-rama → subir la rama al repositorio remoto
- git branch se utiliza para gestionar las ramas
- git branch -m se utiliza para renombrar una rama local

### 3- Conflicto

- Surgieron conflictos en el archivo task\_model.py al intentar hacer merge, debido a diferencias en el contenido de dicho archivo
- El conflicto fue detectado en el mensaje  
"CONFLICT (content): Merge conflict in Task\_model.py"
- Tambien se identificaron conflictos en otras ramas como main.py, ocasionados por edición



- Para solucionar, se combinaron las contribuciones de los desarrolladores, dejando `is-completed` como atributo personal, y se conservaron todas las funcionalidades

- Finalmente, los archivos fueron marcados como resuelto

#### 4- Aportes individuales

- Integrante 1: se encargo de desarrollar la opción para marcar las tareas como finalizadas, usando la función `mark-as-complete`, y realizó las modificaciones necesarias en el archivo `main.py`

- Integrante 2 implemento la funcionalidad para eliminar tareas con la función `delete-Task`, además de actualizar el contenido del archivo `Readme.md`.

- Integrante 3 Provoca manualmente conflicto obligando los métodos `set-done` y `remote-task` y posteriormente gestiona y resuelve dicho conflicto durante el proceso de integración

#### 5- Pregunta de reflexión

1- ¿Cómo organizaron el trabajo en equipo para evitar conflicto?

R1= Nos organizamos dividiendo la tarea en diferentes ramas y manteniendo buena comunicación. Ayuda entre todos

2- ¿Qué lección obtuvieron sobre como resolver conflicto en git?

R1= entendimos que los conflictos son comunes cuando se trabaja en equipo, pero git ofrece herramientas efectivas para solucionarlos.

3- ¿Por qué consideras útil emplear Pull Requests en Proyecto Grupales?



4- ¿De que manera herramientas como CI/CD pueden optimizar el flujo de trabajo?

R/- utilizando CI/CD es posible automatizar tanto las pruebas como el proceso de despliegue, lo cual garantiza que el código funcione correctamente antes de integrarlo al proyecto final.