Informe del Grupo

1 Pasos Realizados:

- se configuro la firma en Git
- se Sincronizo el git con el cvisual
- Creamos el Repositorio < ToDoApp-Grupo-JavKevMat >
- Creamos los archivos main y task-model
- Realizamos un git add y un git commit
- Subimos los archivos al repositorio (push)
- Cada miembro creo una rama (estudiante1, estudiante2, estudiante3)
- Cada miembro Realizo sus respectivo cambios en sus ramas
- Uno del grupo creo otra Rama llamada < group-2 >
- Se realizaron 3 merge en la rama group-2 merge estudiante1, estudiante2 y estudiante3
- Implementacion funcionalidades y resolvimos conflictos

- Realizamos un PR en el Repositorio
- Realizamos un merge del main con la rama grupo-2, para fusionar los cambios

2. Comandos Git:

- git config --global core.editor "code --wait"
- git init
- git status
- git commit -m "text"
- git pull
- git commit --amend
- git branch y git branch -r
- git log --oneline
- git switch "nombre de la Rama"
- git add .
- git push
- git merge
- git merge --abort
- git checkout -b

- git push origin -- delete estudiante1
- git commit -- reset hard
- git push -f
- git branch -D

4. Contribuciones:

- Estudiante1 (Kevin):
  - Creo el Repositorio
  - Añadio mark_as_completo y actualizo main.py
- Estudiante2 (Javier):
  - Añadio delete_task y actualizo README.md, y añadio el Report.txt
- Estudiante3 (Mateo):
  - Simulo conflicto con set_done/remove_task y lo resolvio

Conflictos:

- Hubo conflictos en el "git merge estudiante1" cuando fusionamos "estudiante1" en la rama "grupo2", habían funciones en la clase task_model que no debían estar, y lo solucionamos fusionando las funciones

- Hubo conflictos en el "git merge estudiante2" clase main.py cuando fusionamos "estudiante2" en la rama "grupo2", habían funciones que faltaban y añadimos esos cambios para solucionar el conflicto

- Hubo conflictos en el "git merge estudiante3" clase task_model cuando fusionamos "estudiante3" en la rama "grupo2" habían 4 funciones "def" que faltaban, solucionamos este error añadiendo las clases "def" que faltaban y modificando algunos que ya estaban.

... que yo estaban.

Reflexión:

Durante la creación y gestión de las ramas surgieron pequeños problemas como commites sin mensajes errores al hacer push por no configurar el upstream y confusiones con la sintaxis. Estos inconvenientes muestran la importancia de seguir un flujo ordenado: Preparar bien los cambios, escribir mensajes claros y configurar la rama remota desde el inicio. Son detalles simples, pero que marcan la diferencia para evitar retrasos y mantener un manejo de git mas fluido y eficiente.