Práctica de Desarrollo Colaborativo en Git

En esta práctica vamos a ver cómo se desarrolla un proyecto de manera colaborativa usando GitHub.
El primer paso de la práctica es armar un equipo de 2 personas.

Consigna

El objetivo será desarrollar un programa por línea de comandos con las siguientes funciones:

- Se reciben como argumentos 2 números y el nombre de una operación que puede ser suma, resta, multiplicación o división luego el programa muestra por pantalla el resultado de llevar a cabo dicha operación utilizando los números provistos.
- Opcionalmente el programa puede recibir un nombre de archivo. De ser así, el resultado debe guardarse en dicho archivo.

Ejemplo de uso:

```
$ calc.py 1 2 --op suma
3
```

1. Setup

Un miembro cualquiera del equipo tiene que crear en GitHub (https://github.com/new) el repositorio donde se llevará a cabo la práctica.
Pueden seguir estas instrucciones de la documentación de GitHub.

Este repositorio debe ser compartido con el otro compañero de equipo.
Para eso, dentro de GitHub, ir a Settings → Colaborators -> Manage access -> Add people y agregar al compañero (por su usuario GitHub). Esto le dará permisos necesarios para llevar a cabo la práctica.

2. Cargar issues

En el repo creado chequear tener activado el uso de Issues en Settings -> General -> Features -> Issues.
Luego, ingresar a la solapa de Issues -> New Issue
Crear al menos 3 issues que describan las tareas necesarias para desarrollar las funcionalidades pedidas.
Tener en cuenta que todos los miembros del equipo van a estar trabajando en conjunto.
Pueden seguir estas instrucciones de la documentación de GitHub.

3. Desarrollar funcionalidades

En este paso los miembros del equipo tienen que implementar las funcionalidades descritas en los issues en un branch local.
Recuerden clonar localmente el repositorio y crear y ubicarse en el branch con git checkout -b nombre_de_su_branch
Luego ir haciendo los cambios con los commits y push **en el branch** paso a paso.
Recordar:
git add, git commit -m "comentario" y git push -u origin nombre_de_su_branch

En este punto **es clave la coordinación** para evitar duplicar trabajo y que el código de ambos opere correctamente.

Quienes tengan menos experiencia programando pueden realizar un proceso iterativo en el que la primera versión sea solo definir la función que reciba dos valores y una operación y se la invoque en el main. Luego, podrá repetirse el proceso de desarrollo incorporando el guardado y el pasaje de parámetros desde la ejecución por línea de comandos utilizando argparse para leerlos.

4 Enviar pull requests, hacerles review, validar funcionamiento y mergearlos

Una vez que las funcionalidades están implementadas en un branch local crear un pull request y pedir al compañero que lo revise, comente y acepte los cambios para que el branch de desarrollo sea emergido contra el main.
Para eso:
En la interfaz web del Repositorio, ingresar a la pestaña de pull request o a la de código, aparecerá un botón "Compare & pull request" para el branch pusheado. O la pestaña "Pull requests" → "New pull request". Asignar reviewer (el compañero)
Tip: En el cuerpo del PR se puede referenciar una issue: Closes #1 para que al mergear se cierre automáticamente.
Luego, el compañero que validar, podrá revisar, comentar y aceptar los cambios
Desde GitHub → botón *Merge pull request* → *Confirm merge*
Para ver los cambios reflejados en el repo local, no olviden hacer pull desde el main.

Consideraciones a tener en cuenta:

- Puede ser que sea necesario traer cambios del branch **main** al branch donde se está realizando el PR.
- Es importante **validar que los cambios no rompan el desarrollo existente**.