Informe de Actividad: Git y Python

Autor: Santiago de Hoyos Hernández

Materia: Programación en Python y Control de Versiones con Git

# Introducción

En este informe se describe el procedimiento para crear un repositorio en GitHub, clonarlo en un equipo local, trabajar con ramas y unir los cambios realizados en la rama principal (master). Además, se documenta con imágenes cada uno de los pasos realizados en el proceso.

# Desarrollo

1. En primer lugar, el líder del equipo debe crear el repositorio en GitHub.  
2. Posteriormente, los integrantes deben aceptar la invitación enviada por el líder para unirse al repositorio.  
3. Una vez aceptada la invitación, ingresamos a GitHub.  
4. El líder habrá creado previamente la estructura inicial del repositorio, con el menú de opciones y algunos archivos base. En nuestro caso, trabajaremos con las operaciones de multiplicación y división.  
5. Copiamos la URL del repositorio proporcionada por GitHub.  
6. Nos dirigimos al gestor de archivos de nuestro computador y creamos una carpeta donde alojaremos el proyecto.  
7. Abrimos la carpeta y ejecutamos la consola Git desde allí.  
8. En la consola, escribimos el comando: git clone <URL del repositorio>.  
9. Al ejecutar el comando, se descargará el repositorio en la carpeta local.  
10. Ingresamos a la carpeta clonada, donde estarán los archivos creados por el líder.  
11. Desde la carpeta, abrimos Visual Studio Code (VS Code) y cargamos el proyecto.  
12. Creamos un archivo con extensión .py para desarrollar nuestra parte de la calculadora.  
13. Escribimos y verificamos que el código funcione correctamente.  
14. Una vez guardado, el archivo aparecerá en la carpeta clonada.  
  
Trabajo con ramas en Git:  
  
15. Para trabajar en nuestra propia rama, abrimos la consola Git y ejecutamos el comando: git branch santiago-operaciones.  
16. Cambiamos a nuestra rama con: git checkout santiago-operaciones.  
17. Verificamos el estado del repositorio con: git status.  
18. El archivo aparecerá en rojo; lo añadimos con: git add operaciones.py.  
19. Registramos los cambios con un comentario utilizando: git commit -m "Creación de operaciones".  
20. Cambiamos nuevamente a la rama principal con: git checkout master.  
21. Actualizamos la rama con: git pull.  
22. Unimos los cambios realizados con: git merge santiago-operaciones.  
23. Finalmente, subimos los cambios al repositorio remoto con: git push -u origin master.  
  
De esta manera, nuestro trabajo quedará integrado en la rama principal (master), y la contribución será visible en el repositorio compartido del equipo.

# Conclusión

El uso de Git y GitHub permite trabajar de manera colaborativa y organizada en proyectos de programación. Gracias a las ramas, cada integrante puede realizar sus aportes sin afectar el trabajo de los demás, y posteriormente integrar los cambios en la rama principal. Este informe documenta de forma clara y detallada cada paso realizado durante la práctica.





























