

Nombre: \_\_\_\_\_ Fecha: \_\_\_\_\_

Jornada: \_\_\_\_\_

Realizar la siguiente actividad, según los pasos indicados a continuación:

1. Ingresar a la carpeta donde desea realizar el proyecto. (Desktop, documents, etc.).
2. Crear la carpeta a través de comandos.
3. Inicializar Git.
4. Abrir el editor de texto usando el comando Code .
5. Estando en el editor de código crear los siguientes archivos\_
  - Informe.txt
  - Clasehoy.txt
  - Conclusión.txt
6. Escribir en informe.txt: lo visto en la clase anterior fue:
7. Escribir en clasehoy.txt: en el día de hoy los temas tratados son:
8. Escribir en conclusión.txt: usted va a escribir su opinión sobre la clase y temas vistos.
9. Ver el estado del repositorio en git bash.
10. Agregar los archivos al repositorio (área de ensayo)
11. Agregar en clasehoy.txt: que es un sistema de control de versiones
12. Ver nuevamente el estado del repositorio y debe mostrar que un archivo fue modificado.
13. Adicionar la modificación del archivo a el área de ensayo, usando el comando correspondiente.
14. Realizar un commit con el título “modificado el archivo clase hoy”.
15. Listar o mostrar los archivos que se encuentren en la carpeta.
16. Eliminar el archivo informe.txt
17. Ver el estado del repositorio en git bash
18. Adicionar la eliminación al área de ensayo, porque este archivo no se va a restaurar.
19. Realizar otro commit con el título “archivo informe fue eliminado”.
20. Cambiar el nombre del archivo clasehoy.txt a informehoy.txt
21. Ver el estado del repositorio y agregar el cambio realizado al área de ensayo de ser necesario. (debe agregar el archivo borrado y el archivo con el nuevo nombre)
22. Realizar el commit asociado a este cambio de nombre.
23. Crear el archivo Datosusuario.txt
24. Escribir en Datos de usuario.txt: su nombre como dato de usuario y una contraseña inventada
25. Crear un archivo llamado .gitignore en este se escribe el nombre del archivo a ignorar, es decir, Datosusuario.txt
26. Guardar los cambios, adicionar el archivo gitignore al área de ensayo (repositorio) y realizar commit con el título correspondiente”
27. Ver el historial de commit
28. Verificar en que rama se encuentran con el comando **git branch**.
29. Crear una nueva rama con **git checkout -b** y el nombre que usted le va asignar a la rama.

30. Modificamos el archivo informehoy.txt escribiendo “ramas  
Como ignorar archivos para no ser comprometidos en el  
área de trabajo.  
Cambiar de ramas y volver a la principal”
31. Guardar los cambios
32. Realizar un commit con un título correspondiente a lo  
realizado.
33. Ver el contenido de este archivo.
34. Cambiar a la rama principal y ver el contenido del mismo  
archivo. (debe mostrar diferente información.
35. Mostrar contenido del informehoy.txt
36. Cambiarse a la rama principal con el siguiente comando:

Git checkout nombre de la rama principal

37. Ver el contenido nuevamente del archivo informe.txt
38. Traer los cambios de la rama creada, a la rama principal con  
el comando correspondiente. (verificar que este ubicado en  
la rama principal)
39. Subir el repositorio a github