Ejercicios de comandos 2

Creación de un repositorio local y estados de los archivos

Objetivos:

- Creación de archivos en un repositorio local y cambios en su estado.

Pre-requisitos:

- Trabajar con un usuario de nombre *alumno*.
- Tener instalado el entorno Git.

Comandos Git utilizados:

- `git init`
- `git status`
- `git add`
- `git ls-files`
- `git show`
- `git config`
- `git gui`

Comandos Linux utilizados:

- `cd`
- `ls -la`
- `less`
- `touch`
- `vi`
- `nano`
- `rm`

1) Posicionarse en el directorio de repositorios locales.
   Mantener a la vez abierta una ventana con un explorador de archivos.
   ```
   cd /home/alumno/Documentos/gitRepos
   ```

2) Crear un repositorio llamado repo_uno.
   ```
   git init repo_uno
   ```
   Entrar dentro del directorio creado.
   ```
   cd repo_uno
   ```

3) Comprobar el estado del repositorio.
   ```
   git status
   ```
   Revisar el contenido del directorio `.git/objetcs`.

4) Crear un archivo arch_01 vacío.
   ```
   touch arch_01
   ```

5) Comprobar el estado del repositorio.
   ```
   git status
   ```
   Se informa que en el directorio de trabajo hay archivos sin seguimiento (*Untracked files*) y se sugiere ponerlos bajo seguimiento (`git add`).
   Revisar el contenido del directorio `.git/objects`.

6) Ejecutar la interfaz gráfica integrada (si existe) y comprobar qué ha pasado.
```
git gui
```

7) Añadir el archivo al área de preparado (*Staging área* o SA).
```
git add arch_01
```

8) Comprobar el estado del repositorio.
```
git status
```
Se informa que en el directorio de trabajo hay cambios para ser confirmados (*changes to be commited*) y también se ofrece la posibilidad de deshacerlos (`git rm --cached <archivo>`).

9) Revisar el contenido del directorio `.git/objects`.
Ha aparecido un directorio nombrado con dos caracteres alfanuméricos. Entrar dentro, y comprobar si contiene archivos y, en caso afirmativo, lo que éstos contienen.

10) Listar los archivos existentes en el SA, mostrando su hash.
```
git ls-files –s
```
Comprobar:
   - El código del archivo: 10.0.644 (revisar documentación).
   - Los dos primeros dígitos del hash del archivo.

11) Crear un nuevo archivo, arch_02, editándolo con `vi`, (o `nano`) escribiendo algo en su interior, guardando los cambios y salir de `vi`.
```
vi arch_02
```
Algunos comandos de vi.
   - A → entra en el modo de inserción `INSERT`
   - ESC → sale del modo `INSERT`
   - :q → sale del programa desde el modo inicial
   - :q! → sale forzadamente del programa desde el modo inicial
   - :w → guarda los cambios desde el modo inicial
   - :wq! → guarda los cambios y sale forzadamente desde el modo inicial

12) Añadir este nuevo archivo al SA.
```
git add arch_01
```

13) Revisar el contenido del directorio `.git/objetcs`.

14) Listar los archivos existentes en el área de montado, mostrando su hash.
```
git ls-files –s
```
Comprobar:
   - El código de los archivos
   - El hash de los archivos

15) Crear un nuevo archivo arch_03 vacío.
```
touch arch_01
```

16) Añadir el archivo al SA.
```
git add arch_03
```

17) Eliminar el archivo.

```
rm arch_03
```

Comprobar que el archivo se borró correctamente.

```
ls -la
```

18) Comprobar el estado del repositorio.

```
git status
```

Se informa que hay nuevos archivos con seguimiento, algunos con cambios listos para confirmar (*Changes to be committed*) y otros no montados (*Changes not staged for commit*) para confirmar. Con cada uno de ellos se ofrecen posibles acciones.

19) Listar los archivos existentes en el SA, mostrando su hash.

```
git ls-files –s
```

Comprobar:

- El archivo borrado sigue apareciendo

20) Elimina el archivo “eliminado” del índice (del árbol de Git).

```
git rm arch_03
```

21) Comprobar el estado del repositorio.

```
git status
```

22) Listar los archivos existentes en el SA, mostrando su hash.

```
git ls-files –s
```