Elaborado por: Alexander Moises Bautista Mamani

Git y GitHub

- Configuración de nombre de usuario en **Git**.

```
$ git config user.name "nombreUsuario"
```

- Verificar el nombre de usuario configurado en **Git**.

```
$ git config user.name
```

- Configuración del correo en **Git**.

```
$ git config user.email "correo@gmail"
```

- Verificar el correo configurado en **Git**.

```
$ git config user.email
```

- Crear un repositorio en **Git**.

```
$ git init
```

- Cambiar el nombre por defecto de la rama principal (de “master” a “main” u otro nombre) en **Git**.

```
$ git config --global init.defaultBranch
    main
```

- Asociar el editor de texto Visual Studio Code para realizar los mensajes de commits en **Git**.

```
$ git config --global core.editor "code
  --wait"
```

- Asociar el editor de texto Visual Studio Code para realizar los mensajes de commits en **Git**.

```
$ git config --global core.editor "code
  --wait"
```

- ÁREAS DE **Git**.
  - ✓ Directorio de trabajo ➔ Working Directory Modificada ➔ Modified
  - ✓ Área de preparación ➔ Staging Area Preparada ➔ Staged
  - ✓ Repositorio (directorio .git) ➔ Repository Confirmada ➔ Committed

- Verificar el estado de un repositorio en **Git**.

```
$ git status
```

- LLevar al área de preparacion un cambio (el archivo modificado estará señalado con una “U”, para después para a ser señalado con una “A”).

```
$ git add nombreArchivo
```

Elaborado por: Alexander Moises Bautista Mamani

- LLevar al área de preparacion todos los archivos modificados.

```
$ git add .
```

- Eliminar del área de preparación del Staged.

```
$ git rm --cached nombreArchivo
```

- Crear COMMIT para pasar del área de preparación al repositorio.

```
$ git commit -m "escribir mensaje"
```

- Crear COMMIT para pasar del área de preparación al repositorio, pero el mensaje se realizará mediante el editor VSCode.

```
$ git commit
```

- Listar los COMMITS de un repositorio en **Git** y para salir presionar **:** y después **Q**.

```
$ git log
```

- Listar los COMMITS de un repositorio en **Git** de manera concisa.

```
$ git log --oneline
```

- Modificar el último COMMIT (recomendado para repositorios locales “**Git**” y no para repositorios remotos “**GitHub**”).

```
$ git log --amend
```

- Revertir el último COMMIT.

```
$ git reset --soft HEAD~1
$ git add .
```

- Crear una rama (branch) en **Git**.

```
$ git branch nombreRamaNueva
```

- Listar todas las ramas de un repositorio en **Git**.

```
$ git branch
```

- Cambiar de rama (ir de una rama a otra) en **Git**.

```
$ git checkout nombreRamaDestino
```

- Crear una rama (branch) y cambiar a esa rama directamente en **Git**.

```
$ git checkout -b branch nombreRamaNueva
```

- Cambiar el nombre de una rama (estando dentro de esa rama) en **Git**.

```
$ git branch -m nuevoNombreRama
```

Elaborado por: Alexander Moises Bautista Mamani

- Cambiar el nombre de una rama (estando fuera de esa rama) en **Git**.

```
$ git branch -m antigNomRama nueNomRama
```

- Eliminar una rama solo para repositorios locales en **Git**.

```
$ git branch -d nombreRama
```

- Fusionar una rama con otra en **Git** (se debe estar en la rama que recibirá la fusión).

```
$ git merge nombreRamaFusionar
```

- En caso de presentarse un conflicto en el momento de fusionar dos ramas escribir la siguinte linea esto una vez haber solucionado dicho conflicto en VS-Code.

```
$ git merge --continue
```

- Clonar un repositorio remoto de **GitHub** a un repositorio local en **Git** (el nombre estandar es “origin”).

```
$ git clone rutaGitHubHTTPS
```

- Ver el repositorio clonado de **GitHub** en **Git**.

```
$ git remote
```

- Enivar cambios de repositorio local de **Git** al repositorio remoto de **GitHub**.

```
$ git push origin nombreRamaEnviar
```

- Incorporar cambios del repositorio remoto de **GitHub** al repositorio local de **Git** (“origin” es la rama local y “main” es la rama del remoto).

```
$ git pull origin main
```

- Verificar si existe cambios en el repositorio remoto de **GitHub**.

```
$ git fetch
```

- Observar los cambios después ejecutar **fetch**.

```
$ git checkout origin/main
```

- Enviar un repositorio local a un repositorio remoto (para ello primeramente crear un repositorio vacio en **GitHub**).

```
$ git remote add origin https://nomUs/
    nomRepVacio
$ git fetch
$ git push -u origin main
```

- **Bifurcar ("Fork")** Copiar un repositorio remoto de otra cuenta **GitHub** a un nuetra cuenta de **GitHub**. Este procedimiento se realiza desde la página de GitHub (Click en Fork).
- Crear una copia local en **Git** un repositorio bifurcado en **GitHub**.

```
$ git clone rutaGitHubHTTPS
```

- Solicitar cambiar un repositorio que no es propio en **Git**.

```
$ git pull request
```

- Enviar una rama del repositorio local de **Git** al repositorio remoto de **GitHUb**.

```
$ git push origin nombreRama
```

- Ver ramas remotas desde el repositorio local de **Git**.

```
$ git branch -a
```

- Eliminar una rama del repositorio remoto de **GitHub**, desde el repositorio local de **Git**.

```
$ git push origin -d nombreRama
```

- Para mostrar una lista de los archivo Modificados (M), agregados (A) o los archivos que aun no se agregaron a la zona de preparación (staged) el cual tendrá un simbolo ?? en **Git**.

```
$ git status -s
```

- Para ver los cambios que existen en un archivo que fue modificado y que no esta no esta en zona de preparación.

```
$ git diff
```

- Para realizar un tag o etiqueta a un commit (mayormente se utiliza para señalar las versines ej. v1.0.0).

```
$ git tag nombreEtiqueta
```

- Para ir a esa etiqueta (tag) en **Git**.

```
$ git checkout nombreEtiqueta
```

- Otra forma de ir a otra rama en **Git**.

```
$ git switch nombreRamaDestino
```

- Eliminan lineas de codigo que no esten con un commit dejando todo como estaba antes de modificar algun archivo.

```
$ git stash
```

- Restaura lo que el comando stash elimino.

```
$ git stash pop
```

Elaborado por: Alexander Moises Bautista Mamani

- Lista los stash relizados.

```
$ git stash list
```

- Para eliminar una rama del repositorio remoto de **GitHub** desde el repositorio local de **Git** .

```
$ git push origin --delete nombreRama
```

- Para ver todos los commits del repositorio remoto de **GitHub** y del repositorio local de **Git**.

```
$ git log --oneline --all
```

- Para volver a la version anterior (al anterior commit) del archivo.

```
$ git checkout nombreArchivo
```

- Elimina de la zona de preparación o staged los archivos modificados, pero aun estan listos para ser adicinados nuevamente a esa zona.

```
$ git reset
```

- Vuelve al ultimo commit eliminando así todos los cambios realizados.

```
$ git reset --hard
```

Elaborado por: Alexander Moises Bautista Mamani