

git hub

iniciar un repositorio
validar el usuario
saber el usuario
validar el correo
saber el correo
Configurar del todo el usuario en el cpu
Configurar del todo el email en el cpu

Configurar repositorio en git hub
Indica los repositorios vinculados a git hub
Añadir un archivo al repositorio
Mirar el status del archivo
Añadir todos los archivos
Comitear= etiquetar el estado de un archivo en un momento dado
historial de cambios de los archivos
Subir los archivos a git hub
Llevar los archivos del repositorio local al remoto
crear una copia de todos los archivos
actualizar los archivos exietentes ya clonados
Descomitiar
Descomitiar forzado

Enumera los brazos
Crea un brazo llamado branch
Elimina la rama llamada branch

Elimina la rama llamada branch de manera forzada

para moverse de una rama a la otra

Enviar los cambios a git hub
traer los cambios de git hub

```
git init
git config user.name "usuario"
git config user.name
git config user.email "email"
git config user.email
git config global --user.name "name"
git config global --user.email "email"
```

```
git remote add origin https://github.com/miguelpinedag/intro.git
git remote -v
git add "nombre del archivo"
git status "nombre del archivo"
git add .
```

```
git commit -m "mensaje"
git log
git push
git push origin master
git clone url
git pull origin main
git reset --soft head~1
git reset --hard head~1 forzada
git branch -M main
git branch
git branch <branch>
git branch -d <branch>
```

```
git branch -D <branch>
```

```
git checkout nombre_rama
```

```
git push origin <branch>
git pull origin <branch>
```

```
rm -rf .git
git config --list
```

no sale error todo bb
sale el usuario registrado
no sale error todo bb
sale el email registrado

Nos podemos vover si no hay cambios, silos hay debemos eliminarlos o hacer commit
Una vez que terminamos de realizar los cambios que queremos en nuestra branch, ejecutamos los mismos comandos que vimos hasta ahora: git add, git commit, git status y git log. Pero cuando queramos subir esos cambios, debemos utilizar git push con el nombre de la rama en que estamos posicionados

al hacer eso, no se asusten, no se eliminan los archivos, solamente dejass de hacerles seguimiento con GIT