

INTRODUCCION A LA INFORMATICA

Lista de comandos en Git

CONFIGURACIÓN BÁSICA

Configurar los nombras que salen en los commits

```
git config --global user.name "Beatriz"
```

Configurar Email

```
git config --global user.email betty@gmail.com
```

INICIANDO REPOSITORIO

Iniciamos Git en la carpeta donde está el proyecto

```
git init
```

Clonamos el repositorio de Github

```
git clone <url>
```

Añadimos todos los archivos para el commit

```
git add .
```

Hacemos el primer commit

```
git commit -m "Texto que identifique por que se hizo el commit"
```

subimos al repositorio

```
git push origin master
```

GIT CLONE

Clonamos el repositorio de github

```
git clone <url>
```

GIT ADD

Añadimos todos los archivos para el commit

```
git add .
```

Añadimos el archivo para el commit

```
git add <archivo>
```

Añadimos todos los archivos para el commit omitiendo los nuevos

```
git add --all
```

Añadimos todos los archivos con la extensión especificada

```
git add *.txt
```

Añadimos todos los archivos dentro de un directorio y de una extensión específica

```
git add docs/*.txt
```

Añadimos todos los archivos dentro de un directorios

```
git add docs/
```

GIT COMMIT

Cargar en el HEAD los cambios realizados

```
git commit -m "Texto que identifique por que se hizo el commit"
```

Agregar y Cargar en el HEAD los cambios realizados

```
git commit -a -m "Texto que identifique por que se hizo el commit"
```

De haber conflictos los muestra

```
git commit -a
```

GIT PUSH Y PULL

Subimos al repositorio

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

Bajar del repositorio

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

GIT REMOTE

Agregar repositorio remoto

`git remote add origin <url>`

Cambiar de remote

`git remote set-url origin <url>`

Remover repositorio

`git remote rm <name/origin>`

Muestra lista repositorios

`git commit -m "Texto que identifique por que se hizo el commit"`

subimos al repositorio

`git remote -v`

Muestra los branches remotos

`git remote show origin`

Limpiar todos los branches eliminados

`git remote prune origin`

GIT BRANCH

Crea un branch

`git branch <nameBranch>`

Lista los branches

`git branch`

Comando -d elimina el branch y lo une al master

`git branch -d <nameBranch>`

Elimina sin preguntar

`git branch -D <nameBranch>`

GIT REBASE

Los rebase se usan cuando trabajamos con branches esto hace que los branches se pongan al día con el master sin afectar al mismo

Une el branch actual con el master, esto no se puede ver como un merge

`git rebase`

Cuando se produce un conflicto nos da las siguientes opciones:

cuando resolvemos los conflictos --continue continua la secuencia del rebase donde se pauso

`git rebase --continue`

GIT REBASE

Omite el conflicto y sigue su camino

`git rebase --skip`

Devuelve todo al principio del rebase

`git rebase --abort`

Para hacer un rebase a un branch en específico

`git rebase <nameBranch>`

OTROS COMANDOS

Lista un estado actual del repositorio con lista de archivos modificados o agregados

`git status`

Quita del HEAD un archivo y le pone el estado de no trabajado

`git checkout -- <file>`

Crea un branch en base a uno online

`git checkout -b newlocalbranchname origin/branch-name`

Busca los cambios nuevos y actualiza el repositorio

`git pull origin <nameBranch>`

Cambiar de branch

`git checkout <nameBranch/tagname>`

Une el branch actual con el especificado

`git merge <nameBranch>`

Verifica cambios en el repositorio online con el local

`git fetch`

Borrar un archivo del repositorio

`git rm <archivo>`