Ejemplo de uso de Git y Github

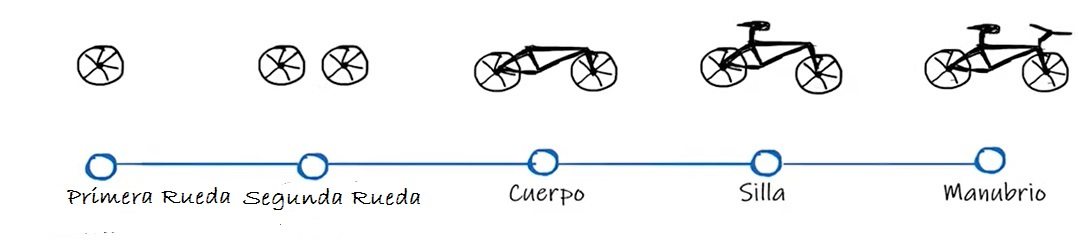
# PRIMERA PARTE

**Commits:** son los puntos en el tiempo.

**Historial de cambios**: un conjunto de puntos formando una línea.

**Log:** es una serie de puntos ordenados cronológicamente donde cada uno de estos puntos representa un avance con respecto al anterior.

**Staging area (área de ensayo)**: es un lugar donde se coloca las partes del progreso que queres guardar.



**1) Creamos la carpeta bici**

**2) Inicializamos git en la terminal bash de VSC:**

git init

**3) Configuro el usuario del proyecto**

git config --global user.name "Juan"

git config --global user.email "juan@gmail.com"

Para verificar la info ingresada:

git config --global user.email

Informa juan@gmail.com.

**4) Agregamos a la carpeta bici el archivo primera\_rueda.txt y hacemos**

git status

Indica que archivos se agregaron, modificaron o eliminaron.

**5) Movemos el cambio hecho al staging area**

git add primera\_rueda.txt

Luego con git status vemos que el archivo está en verde (porque está en el staging area) sino, estaría en rojo.

**6) Agregamos segunda\_rueda.txt**

Si hiciera un commit, solo se guardaría el primer archivo (porque está en el staging area).

git commit -m "primera\_rueda"

Luego, al hacer

git status

Solo vería los 2 últimos archivos y no el primero ya que paso del staging area al commit.

**7) Para ver el historial de commits**

git log

Muestra el código hash, autor, fecha y mensaje del commit.

Para ver un commit de una forma más compacta:

git log --oneline

o

git log --oneline --decorate

Al agregar más commits, estos se muestran del más reciente al más antiguo (orden cronológico).

**8) Si agrego texto a primera\_rueda.txt y hago**

git diff

Veo lo que se agregó en ese archivo.

**9) Paso al staging area este archivo primera\_rueda.txt (con las líneas agregadas)**

git add primera\_rueda.txt

**10) Para quitar un archivo del staging area**

git restore --staged primera\_rueda.txt

***Nota***: para agregar todos los archivos al staging area

git add .

**11) Para moverse entre commits**

git checkout <hash\_commit>

Y vuelvo a un momento anterior del proyecto.

**12) Para volver al último commit**

git ckeckout master

**13) Para ver diferencias entre 2 commits**

git diff <hash1\_commit> <hash2\_commit>

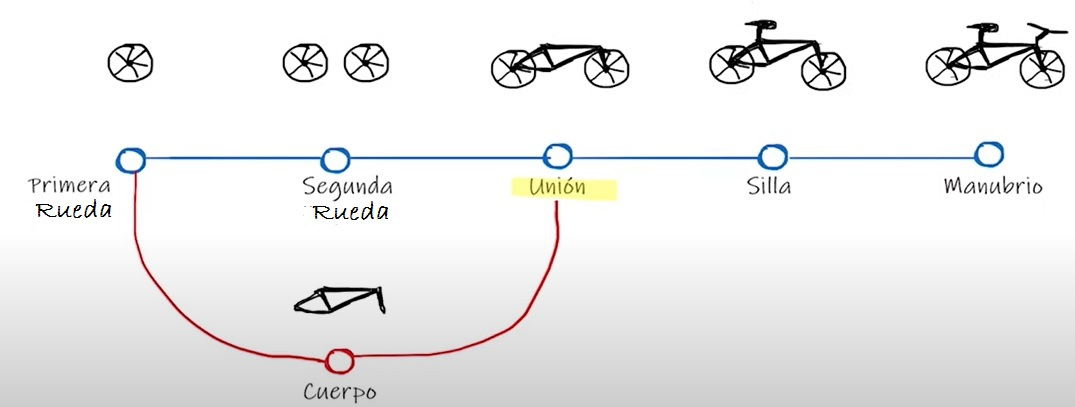
**14) Si modifique un archivo que no commitie y lo quiero volver a su estado anterior sin usar ctrl+z hago:**

git restore primera\_rueda.txt

o a todos los archivos a los que se les hizo cambios:

git restore .

# RAMAS o BRANCHS



**15) Para saber en qué rama estoy ahora parado**

git branch

**16) Para crear una nueva rama**

git branch <nombre\_rama>

Si vuelvo a escribir

git branch

Ahora aparecen ambas ramas

**17) Para moverme entre ramas**

git checkout segunda

o

git switch segunda

**18) Ahora hago cambios en segunda\_rueda.txt y lo guardo en un commit**

git add segunda\_rueda.txt

git commit -m "agregar línea en segunda rama"

veo los cambios con

git log --oneline

Allí veo que la rama master llega hasta determinado historial de cambios y luego sigue la rama segunda con el último commit.

**Unir ramas:** es llevar los commits de una rama a la otra rama

**19) Me paro en la rama master ya que quiero llevar los cambios de la rama segunda a la rama master**

git switch master

git merge segunda

Se abre el editor de commits bean (podría ser nano también) para editar un mensaje en el merge, el cual genera un nuevo commit

**20) Para ver en forma grafica**

git log --oneline --graph

# GITHUB

Crear cuenta en Github y crear repo

**Para ver los remotos que tengo**

git remote

**Para agregar un nuevo remoto**

git remote add origin <url>

**Para subir los cambios del repo local al repo remoto**

git push origin master

***Nota:*** *origin* es el remoto y *master* es la rama del repo local q deseo subir al remoto

Github pedirá credenciales a VSC

Luego, hacer refresh en Github y ver como subió al remoto mi repo local

**Ahora vamos a hacer cambios en el remoto y bajarlos al local.**

Para eso, agrego el archivo readme al repo remoto y hacemos

git pull origin master

y vemos en el repo local como se agregó el archivo readme.md