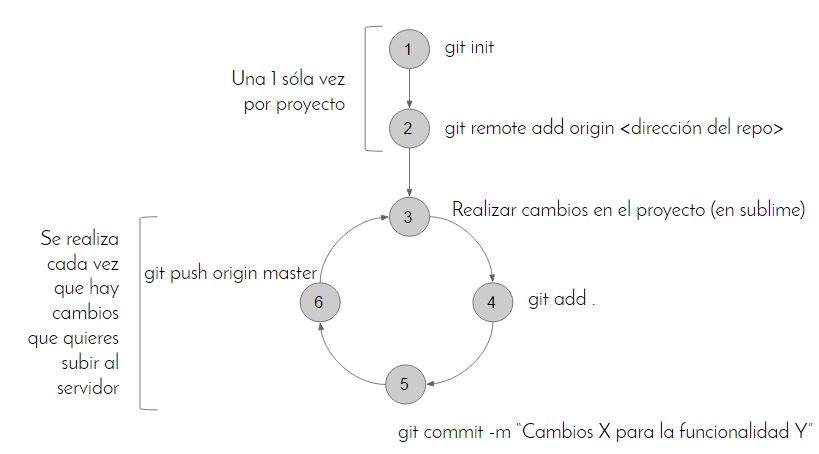
# Resultado de imagen para git

# <https://styde.net/curso-de-git/>

### **Ciclo de Vida de un proyecto GIT**



### **Ciclo de Vida de un Archivo GIT**



### **Resultado de imagen para ciclo workflow gitInteracción GIT con Repositorio Remoto**

### **Resultado de imagen para ciclo workflow gitEsquema de Branches e Historial de Commits**

### **Clonar un nuevo proyecto**

|  |  |
| --- | --- |
| Comando | Descripción |
| mkdir nombre\_directorio | Crear un nuevo directorio para el proyecto |
| git init | Inicializa control de versiones GIT |
| git remote add origin https://github.com/usuario/repositorio.git | Agrega el repositorio GIT al proyecto |
| git pull origin master | Descargar archivos del repositorio en la rama **master** |

### **Crear un nuevo proyecto**

|  |  |
| --- | --- |
| Comando | Descripción |
| mkdir nombre\_directorio | Crear un nuevo directorio para el proyecto |
| git init | Inicializa control de versiones GIT |
| Copiar o crear archivos del proyecto | Copiar o crear archivos y directorios del proyecto |
| Crear archivo “.gitignore” | Agregar los archivos o directorios que deben ser **excluidos** del control de versiones GIT |
| git add \*.\* | Agregar todos los archivos al control de versiones GIT |
| git remote add origin https://github.com/usuario/repositorio.git | Agrega el repositorio GIT al proyecto |
| git commit -m "first commit" | Confirmar los cambios con un commit |
| git push -u origin master | Subir cambios confirmados al repositorio en la rama **master** |

### **Crear nuevas ramas**

|  |  |
| --- | --- |
| Comando | Descripción |
| git checkout nombre\_rama | Crea una rama a partir de la rama actual y cambia a esa rama. Si la rama ya existe sólo cambia a dicha rama. |

### **Estatus en git**

|  |  |
| --- | --- |
| Comando | Descripción |
| git status | Mostrará los diferentes estados de los archivos en tu directorio de trabajo y área de ensayo. Qué archivos están modificados y sin seguimiento y cuáles con seguimiento, pero no confirmados aún. |
| git diff master branchB | Ver las diferencias entre dos árboles. Esto podría ser la diferencia entre tu entorno de trabajo y tu área de ensayo |
| git log –oneline | cat | Muestra los últimos commits efectuados |
| git diff eec242d 98d1360 | Muestra la diferencia entre 2 Commits |

### **Confirmar cambios**

|  |  |
| --- | --- |
| Comando | Descripción |
| git commit -m "descripción" | Crea una rama a partir de la rama actual y cambia a esa rama. Si la rama ya existe sólo cambia a dicha rama. |
| -m “descripcion” | Pasar mensaje de confirmación |
| -a | Saltarse el paso git add |

### **Agregar archivo(s) al control de versiones GIT**

|  |  |
| --- | --- |
| Comando | Descripción |
| git add nombre\_archivo | Agregar el(los) archivo(s) al control de versiones GIT |

### **Descartar los cambios pendientes en archivo(s)**

|  |  |
| --- | --- |
| Comando | Descripción |
| git checkout -- ruta-archivo | Descarta todo cambio realizado en el archivo en el área de trabajo, es decir, que éste queda restablecido hasta HEAD (el último commit realizado). |

### **Quitar archivos de control de versiones GIT**

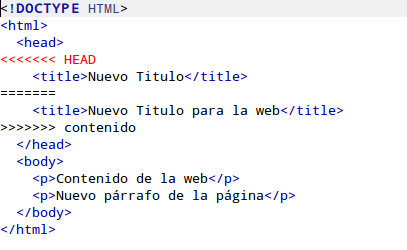
|  |  |
| --- | --- |
| Comando | Descripción |
| git reset ruta-archivo | Quita el archivo del control de versiones GIT sin eliminarlo físicamente. |
| git rm | se utiliza para eliminar archivos del área de ensayo y el directorio de trabajo para Git. Es similar a git add en que pone en escena una eliminación de un archivo para la próxima confirmación. |

### **Integrar cambios al proyecto**

|  |  |
| --- | --- |
| Comando | Descripción |
| git add \*.\* | Agregar los archivos al control de versiones GIT o después de hacer cambios al código |
| git commit -m "descripción" | Confirmar los cambios con un commit |
| git pull origin master | Descargar archivos del repositorio en la rama **master** para prevenir conflictos de código |
| -- Si surgen conflictos al hacer merge se deben resolver -- | |
| git push -u origin master | Subir cambios confirmados al repositorio en la rama **master** |

### **Resolviendo conflictos**

Algunas veces la unión de dos ramas no resulta tan bien, sino que ocurre un conflicto, esto cuando los commits de la rama a fusionar y la rama actual modifican la misma parte en un archivo en particular y git no puede decidir cuál versión elegir, y te avisa que tú debes resolverlo. Por ejemplo:



donde tenemos que elegir entre lo que está entre <<<<<<< HEAD y =======que es contenido que tenemos en la rama donde estamos haciendo el merge (master) o entre ======= y >>>>>>> contenido donde están los cambios hechos en la rama que queremos unir (contenido).

Para ello arreglamos el archivo con los cambios elegidos, guardamos, agregamos y hacemos commit de los cambios:

### **Después de resolver el Conflicto**

|  |  |
| --- | --- |
| Comando | Descripción |
| git commit -a | Confirmar los cambios de la resolución del conflicto |
| git merge --abort | Cancelar Merge |

### **Deshacer un Commit**

|  |  |
| --- | --- |
| Comando | Descripción |
| git revert nombre\_del\_commit | Hace un nuevo commit haciendo los cambios contrarios al commit seleccionado |