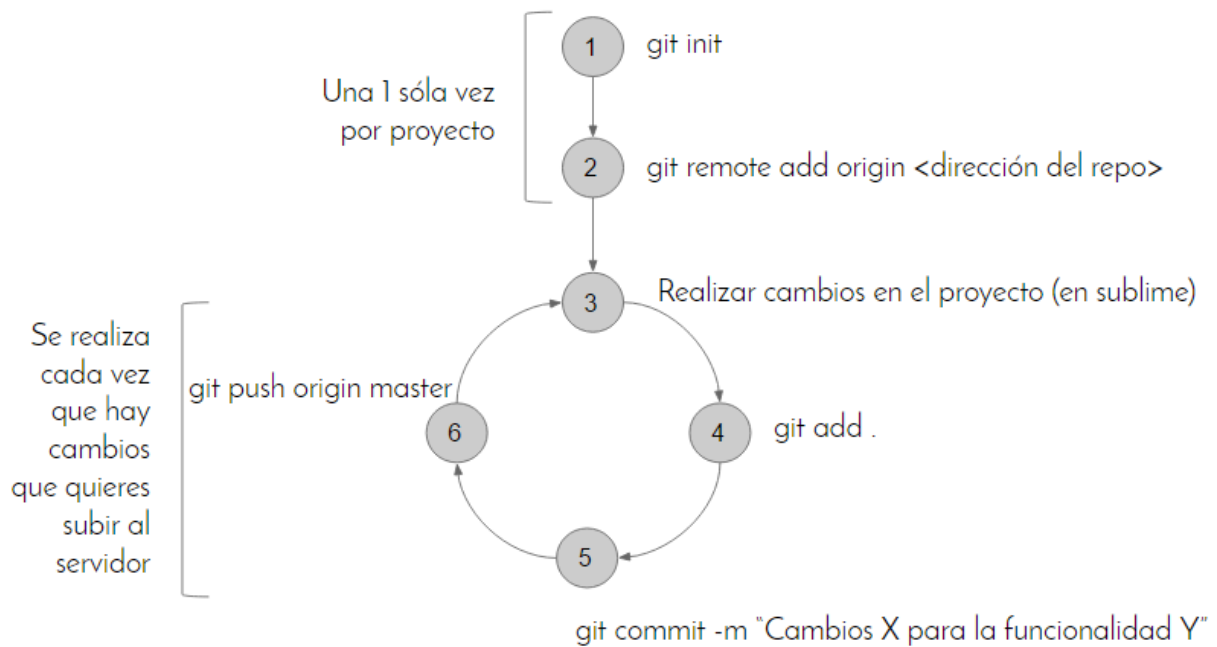




<https://styde.net/curso-de-git/>

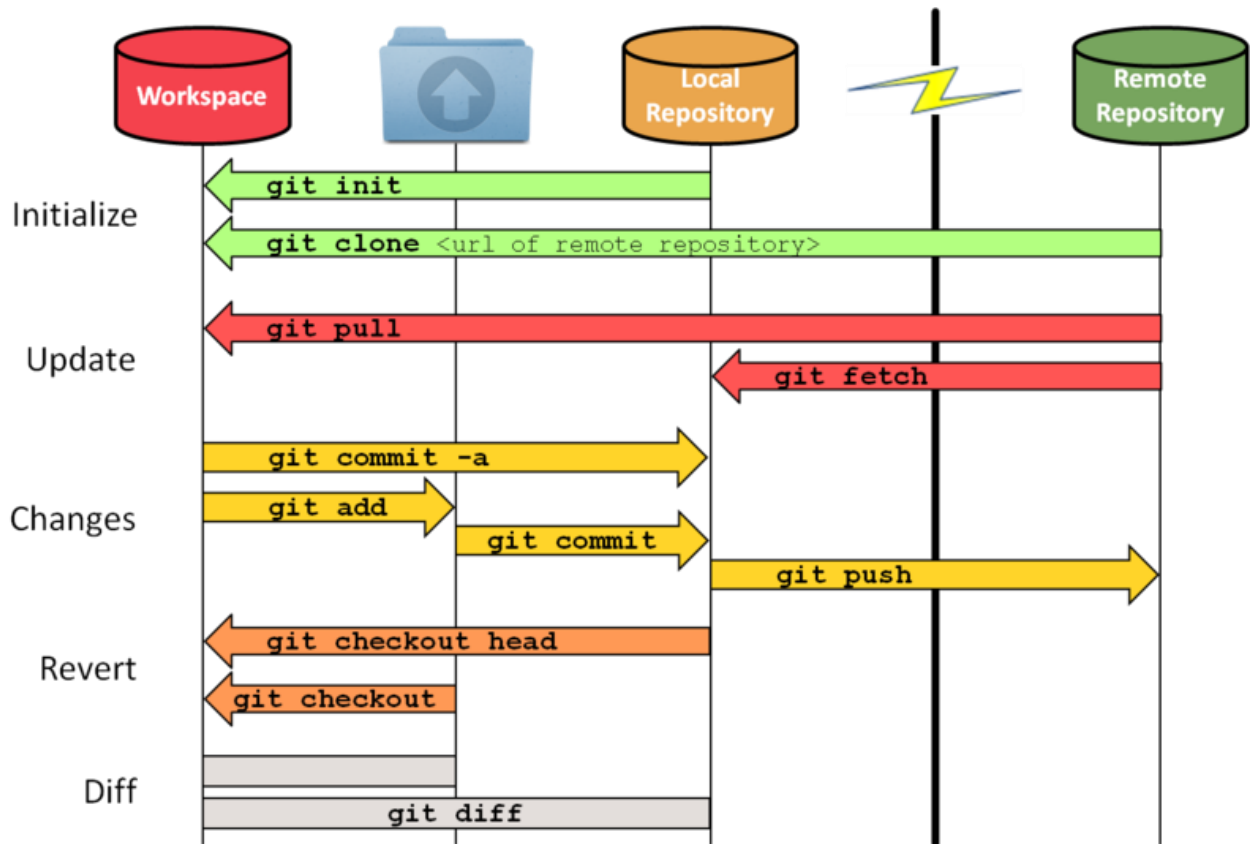
## Ciclo de Vida de un proyecto GIT



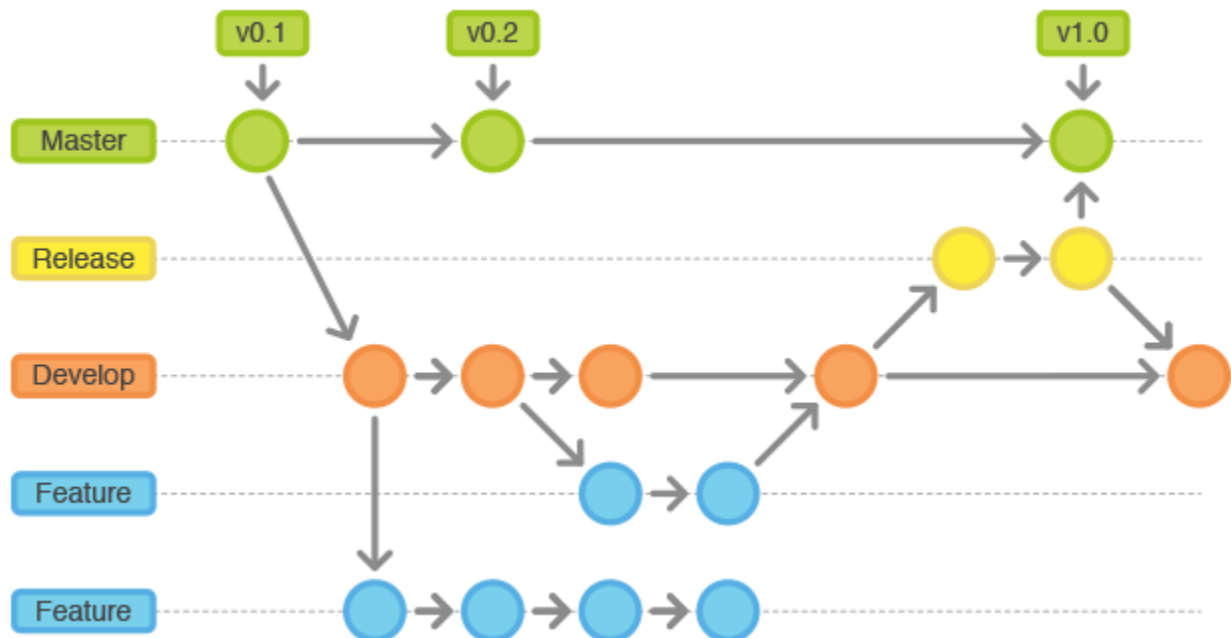
## Ciclo de Vida de un Archivo GIT



## Interacción GIT con Repositorio Remoto



## Esquema de Branches e Historial de Commits



## Clonar un nuevo proyecto

Comando	Descripción
<code>mkdir nombre_directorio</code>	Crear un nuevo directorio para el proyecto
<code>git init</code>	Inicializa control de versiones GIT
<code>git remote add origin https://github.com/usuario/repositorio.git</code>	Agrega el repositorio GIT al proyecto
<code>git pull origin master</code>	Descargar archivos del repositorio en la rama <b>master</b>

## Crear un nuevo proyecto

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

## Crear nuevas ramas

Comando	Descripción
<code>git checkout nombre_rama</code>	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
<code>git status</code>	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.
<code>git diff master branchB</code>	Ver las diferencias entre dos árboles. Esto podría ser la diferencia entre tu entorno de trabajo y tu área de ensayo
<code>git log --oneline   cat</code>	Muestra los últimos commits efectuados
<code>git diff eec242d 98d1360</code>	Muestra la diferencia entre 2 Commits

## Confirmar cambios

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

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

Comando	Descripción
<code>git add nombre_archivo</code>	Agregar el(los) archivo(s) al control de versiones GIT

## Descartar los cambios pendientes en archivo(s)

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

## Integrar cambios al proyecto

Comando	Descripción
<code>git add *.*</code>	Agregar los archivos al control de versiones GIT o después de hacer cambios al código
<code>git commit -m "descripción"</code>	Confirmar los cambios con un commit

<code>git pull origin master</code>	Descargar archivos del repositorio en la rama <b>master</b> para prevenir conflictos de código
<b>-- Si surgen conflictos al hacer merge se deben resolver --</b>	
<code>git push -u origin master</code>	Subir cambios confirmados al repositorio en la rama <b>master</b>

## 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:

```
<!DOCTYPE HTML>
<html>
  <head>
<<<<<<< HEAD
    <title>Nuevo Titulo</title>
=====
    <title>Nuevo Titulo para la web</title>
>>>>>>> contenido
  </head>
  <body>
    <p>Contenido de la web</p>
    <p>Nuevo párrafo de la página</p>
  </body>
</html>
```

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
<code>git commit -a</code>	Confirmar los cambios de la resolución del conflicto
<code>git merge --abort</code>	Cancelar Merge

## Deshacer un Commit

Comando	Descripción
<code>git revert nombre_del_commit</code>	Hace un nuevo commit haciendo los cambios contrarios al commit seleccionado