

### 3.2.1 Actividad 1

Realiza un informe donde expliques los siguientes comandos de git (realiza un ejemplo en tu computador y toma pantallazo).

Rta: los comandos de git

El comando add se realiza `$ git add` . tres opciones del comando `git-add`: `-u`, `-A` y `-n`. Las dos primeras nos permiten actualizar en el índice directamente ficheros borrados sin necesidad de ejecutar el comando `git-rm`. La última opción, `-n`, simula lo que hará el comando `git-add` sin realmente actualizar el índice. Los tres son opciones útiles que de haberlas conocido cuando empecé con git hace años me habrían ahorrado unos cuantos comandos.

El comando `init` para crear un nuevo repositorio, usa el comando `git init`. `git init` es un comando que se utiliza una sola vez durante la configuración inicial de un repositorio nuevo. Al ejecutar este comando, se creará un nuevo subdirectorio `.git` en tu directorio de trabajo actual. También se creará una nueva rama principal.

El comando `git status` muestra el estado del directorio de trabajo y del área del entorno de ensayo. Permite ver los cambios que se han preparado, los que no y los archivos en los que Git no va a realizar el seguimiento. El resultado del estado no muestra ninguna información relativa al historial del proyecto. Para ello, debes usar `git log`. Enumera los archivos que se han preparado, los que están sin preparar y los archivos sin seguimiento.

El comando `git status` es un comando relativamente sencillo. Simplemente, muestra lo que ha ocurrido con los comandos `git add` y `git commit`. Los mensajes de estado también incluyen información importante para preparar archivos o deshacer su preparación. A continuación, se indica el resultado de ejemplo que muestra las tres categorías principales de una llamada de `git status`.

El comando `git commit` captura una instantánea de los cambios preparados en ese momento del proyecto. Las instantáneas confirmadas pueden considerarse como versiones "seguras" de un proyecto: Git no las cambiará nunca a no ser que se lo pidas expresamente. Antes de ejecutar `git commit`, se utiliza el comando `git add` para pasar o "preparar" los cambios en el proyecto que se almacenarán en una confirmación. Estos dos comandos, `git commit` y `git add`, son dos de los que se utilizan más frecuentemente.

El comando `git push` se usa para cargar contenido del repositorio local a un repositorio remoto. El envío es la forma de transferir confirmaciones desde tu repositorio local a un repositorio remoto. Es el equivalente a `git fetch`, pero mientras que al recuperar se importan las confirmaciones a ramas locales, al enviar estas se exportan a ramas remotas. Las ramas remotas se configuran

mediante el comando `git remote`. Los envíos pueden sobrescribir los cambios, por lo que se debe tener cuidado a la hora de realizarlos. Estos problemas se describen a continuación.

El comando `git clone` se usa para crear una copia o clonar un repositorio remoto. Se utiliza `git clone` con la URL de un repositorio. Git es compatible con varios protocolos de red y sus formatos de URL correspondientes. En este ejemplo, usaremos el protocolo Git SSH. Las URL Git SSH siguen la siguiente estructura:

del comando `git branch` y una exposición general del modelo de creación de ramas de Git. La creación de ramas es una función disponible en la mayoría de los sistemas de control de versiones modernos. La creación de ramas en otros sistemas de control de versiones puede tanto llevar mucho tiempo como ocupar mucho espacio de almacenamiento.

`git merge` combinará varias secuencias de confirmaciones en un historial unificado. En los casos de uso más frecuentes, `git merge` se utiliza para combinar dos ramas. Los ejemplos siguientes del presente documento se centrarán en este patrón de fusión de ramas. En estos casos, `git merge` toma dos punteros de confirmación.

**Nota: buenos días profesor en este punto no puedo poner imágenes ya que mi computador se dañó y estoy haciendo el trabajo desde mi celular gracias por la comprensión.**