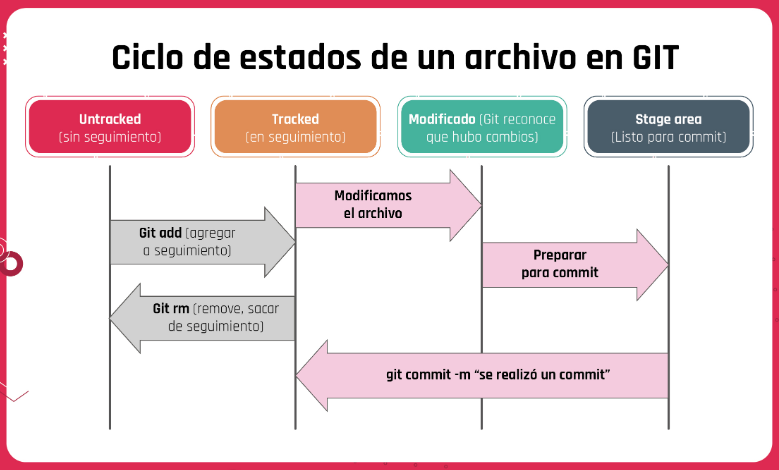
GIT BASH



En Git el ciclo de vida del archivo es:

1. Creación del archivo, por ejemplo **index.html** (**aquí el archivo está en estado untracked**).
2. **Cambiamos el estado** del archivo a **seguimiento** a través del comando **git add index.html** (especificando nombre del archivo index.html para este caso). Si el archivo ya no será modificado, pasar al paso 6.
3. Si el archivo es modificado, **deja de estar en seguimiento y pasa a un estado de modificado**, ya que Git detecta que hubo cambios.
4. Guardamos el archivo con **ctrl+g** (o ctrl + s) y luego escribimos **git add index.html** (especificando nombre del archivo index.html para este caso), el archivo pasa al **stage area**.
5. El archivo está listo para realizar el **commit**, que es cuando se genera el punto histórico, a través del comando **git commit**.

Para especificar que git no realice más seguimiento a un archivo, usamos el comando **git rm --cached** donde si bien el archivo como tal no se ha eliminado, Git procede a ignorarlo y pasa a un estado de **untracked**.

CONECTAR REPOSITORIOS LOCALES CON REMOTOS

Conectar el repositorio local con el remoto con Git HUB (Espacio en nube para crear repositorios remotos)

$ git remote add origin + URL del repositorio remoto

CREAR UN REPOSITORIO LOCAL

Primero indicar credenciales de usuario

Desde la terminal, ejecutar comandos:

$git config --global user.name "mi-usuario"

$git config --global user.email "mimail@mail.com"

posicionarse en la carpeta que se quiere hacer un repositorio

Ejecutar el comando $ git init

(este crea un repositorio local en la carpeta de archivos, inicialmente se crea vacio, luego se debe indicar qué archivos se desea guardar)

para colocar archivos en el repositorio, por ej archivo funciones.js:

$ git add funciones.js / o $ git add . (para agregar todos los archivos)

confirmar archivos:

para confirmar la adhesion de archivos al repositorio ejecutar el comando

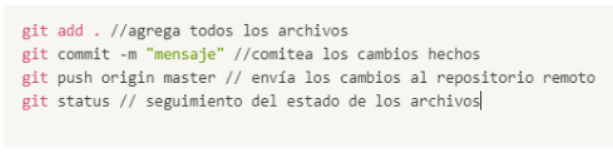
$git commit -m "subo proyectos al repositorio"

Los commits generan un punto cronologico en el tiempo sobre un archivo, va guardando todas las versiones de ese archivo con sus modificaciones

Por ultimo para mandarlos a git hub

$git push origin main

De este modo se verían los archivos subidos en GIT HUB



BAJAR (CLONAR) PROYECTOS DE GIT HUB EN EL REPOSITORIO LOCAL (PC)

$git clone + URL del repositorio remoto

DESCARGAR CAMBIOS (ACTUALIZAR PROYECTOS) SUBIDOS A GIT HUB

$git pull origin main 🡨 actualiza cambios y descarga archivos nuevos en caso que existan

GIT LOG 🡪 Explora el historial del repositorio

RAMAS (BRANCH)

Las ramas permiten trabajar en versiones paralelas para evitar subir cambios no validados o código dudoso a la rama principal.

RAMA MASTER EN GIT BASH

Para cambiar la rama master a la rama main

$ git config --global init.defaultBranch main