

# GITHUB

## REPOSITORIO GLOBAL (REMOTO)

### ¿Qué es GitHub?

GitHub es un sitio web (nube) en donde podemos almacenar los archivos y proyectos de programación de manera gratuita. En GitHub podemos tener la cantidad de proyectos que queramos, en donde a cada proyecto le corresponderá un repositorio.

### Datos:

1. Desde octubre de 2020, Github cambió el nombre de la rama principal **master** para los repositorios nuevos. Ahora se llama **main**.
  2. Para Git, al repositorio remoto se le llama **origin**
- Crear repositorio desde GitHub
    - Repositories ---> New
    - Elegimos un nombre para el repositorio ---> Repository name ---> Create repository
    - GitHub nos da un link y con él clonamos el repositorio en nuestra máquina
      - `git clone linkDelRepo`
      - También lo podemos usar para clonar el repositorio de alguien que nos haya invitado a participar en su proyecto
        - Abrir terminal en la ubicación en donde queramos clonar el proyecto
        - Copiar URL del repositorio que queramos clonar
        - `git clone URL`
    - Apuntar al repositorio remoto desde el local
      - `git remote add origin http://...`
    - Para verificar que todo quedó bien se escribe el comando:
      - `git remote -v`
  - Enviar cambios al repositorio remoto
    - `git push origin main`
  - Descargar los cambios que existen en el repositorio remoto
    - `git pull origin main`

## Conectar repositorio local con repositorio remoto

- Tener creado previamente un repositorio en GitHub
- Ir a la ubicación del mismo y copiar la URL
- Escribir el comando **`git remote add origin URL`**

- Para verificar que el paso anterior se ejecutó correctamente, correr el comando `git remote -v` . Deberíamos ver en la terminal la palabra origin seguida de la URL

### **Detectando y resolviendo conflictos (al trabajar varios en un mismo archivo)**

- Si una persona sube a github sus cambios antes que yo, y yo intento subir los míos inmediatamente después, me aparecerá un mensaje diciendo que hay conflictos.
- Tendré que escribir `git pull origin main` para traer los cambios de mi compañero
- El editor de código nos mostrará el/los conflicto/s que haya/n en nuestro proyecto (la primera parte es nuestro código y la segunda la de mi compañero)
- Podemos resolver el conflicto de tres modos:
  - Dejamos nuestro código, versión local
  - Dejamos el código que viene de GitHub
  - Unimos ambos
- Una vez que Git termine de fusionar todos los cambios, la consola nos devolverá un mensaje diciendo "Merge branch "master"....". Deberemos agregar un commit que explique por qué ese merge fue necesario
  - Para salir de la consola sin escribir el mensaje, tendremos que apretar:
    - `:q`
    - `ctrl + x`
- Maneras de evitar conflictos:
  - Mantener los commit pequeños y subir al repositorio frecuentemente
  - Utilizar ramas para trabajar en paralelo a la versión principal del proyecto

**.gitignore:** Archivo que se crea con ese nombre y ahí se pone la lista de directorios, archivos, etc., que no queremos que se trackeen.