

Clase 3

▼ Status	In progress
----------	-------------

Git

- Es un software de control de versiones.
- Sirve para realizar un back up de archivos y que siempre estén actualizados.
- Da la posibilidad de compartir nuestro trabajo con el resto del equipo.
- Mantiene las actualizaciones sobre el código fuente.
- Registra quien hizo cada modificación.
- Permite desarrollar diferentes versiones de un archivo sin tener que generar múltiples versiones.
- Se puede retroceder en los cambios y volver al punto en el tiempo deseado.

Los 3 estados de Git

1. Working directory: es donde se trabaja con los archivos, se modifica, se crean, se eliminan, etc. "Consigo a los personajes".
2. `git add`
3. Staging area: archivos en seguimiento por git, listos para ser agregados a un commit. "Personajes en el escenario".
4. `git commit`
5. Repositorio local: registro de los archivos local. "Le saco la foto a los personajes".

Para comprobar que Git esté bien instalado

`git --version` arrojará la versión de git instalada en el sistema

Repositorios

Es una carpeta oculta reflejo de la original.

Es un almacén de archivos en paquetes que se llaman **commits**.

Los commits son como fotos de nuestro proyecto en un determinado momento y que guardan una versión específica de esta a la que podemos volver si es necesario. Sirven para hacer un seguimiento de los cambios.

1. Locales: en mi equipo.
2. Remotos: en GitHub.

Crear repositorio local

Posarme sobre la carpeta en donde quiero crear el repositorio, hacer click derecho y poner abrir git aquí.

`git init` inicia el repositorio de esa carpeta. Al crearlo estará vacío. Luego debo agregar los archivos.

Al crear un repositorio dentro de la carpeta elegida se crea una carpeta oculta

Agregar tu identidad

Es el usuario y mail usados en GitHub.

`git config user.name maropra` ingresa el nombre de usuario

`git config user.email mopradolce@gmail.com` ingresa el mail

`git config --global user.name/email` configura el usuario permanentemente para no tener que ingresar cada vez que inicio un nuevo repositorio

`git config user.name/email` chequea el nombre de usuario o mail

Agregar archivos al repositorio

Debo estar dentro de una carpeta con un repositorio iniciado para que funcione.

`git add .` agrega todos los archivos de la carpeta al seguimiento de git.

`git add archivo.html` agrega solo el archivo indicado al seguimiento de git.

Se recomienda hacer ese comando cuando se hayan hecho todas las modificaciones necesarias.

Git solo hace seguimiento de los archivos, no de las carpetas, por lo que si hay una carpeta vacía no la registrará. Solo lo hará cuando tenga archivos dentro.

Estado de los archivos

`git status` informa sobre el estado de los archivos y su relación con el repositorio.

Los archivos dentro de esa carpeta que no hayan sido agregados aparecerán como "archivos sin seguimiento" en rojo.

Cuando un archivo sufre una modificación hay que agregarlo de nuevo porque git lo toma como un archivo nuevo, al agregarlo lo actualiza.

Confirmar archivos

Esto posiciona el archivo en el repositorio y se realiza con los commits.

Los **commits** son puntos en el tiempo a los que puedo volver de ser necesario, son un paquete de modificaciones.

Tienen una fecha, autor, mensaje y nro de id.

```
git commit -m "mensaje que represente la modificación hecha"
```

Estado del repositorio

`git log` muestra todos los commits hechos en el tiempo con el autor, el id y el mensaje.

Recuperar archivo

`git checkout --nombreArchivo.js` puede recuperar un archivo borrado en el commit anterior.

Dejar de seguir un archivo

`git rm --cached nombreArchivo.js` al ejecutar status aparecerá sin seguimiento

`git restore --staged nombreArchivo.js` saca un archivo de seguimiento, vuelve al working directory.

