**Git.**

Es muy útil para compartir proyectos con distintas personas, tener varias versiones de un mismo proyecto y poder guardar proyectos en la nube. Es por esto que es una de las más requeridas por el mercado a la hora de trabajar con control de versiones y de forma colaborativa.

Como muchas de las herramientas que veremos a lo largo de la cursada, Git requiere de práctica y tiene muchos accesorios que agregan funcionalidades, pero en esta clase nos centraremos en sus cuestiones principales.

**Introducción a Git**

Una herramienta que tiene muchos de los beneficios que nos dan las que mencionamos anteriormente: trabajar en grupo de forma colaborativa, comunicar cambios, manejar distintas versiones. Como dijimos al principio del módulo, vamos a conocer una herramienta que nos va a servir durante toda la cursada.

**Instalación de Git**

git --version nos indica que versión de git tenemos instalada.

**Uso de GitHub y creación de repositorios remotos**

**Creación de repositorios locales**

Palabras clave

***Repositorio loca***l: es el que tiene todos los archivos (que hayas guardado en él) en la computadora.

***Commits***: son los paquetes que nos van a permitir ir haciendo un seguimiento de los cambios que vamos realizando, dado que cada uno de ellos tiene una timestamp, o fecha de creación, y un autor. Los commits van a ser nuestro historial de cambios que se fueron haciendo en el proyecto.

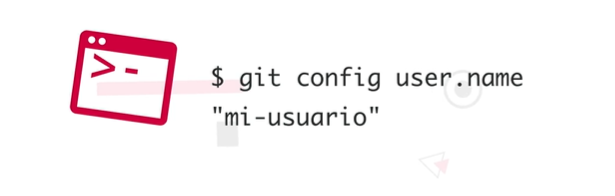
Estos son los comandos que vimos en el video:

***git init*** *// crea el repositorio*

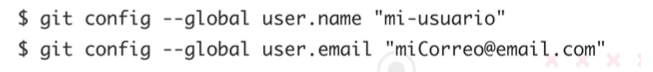
***git config user.name "nombre de usuario"*** *// agrega nuestra identidad*

***git config user.email "nombreUsuario@email.com"*** *// agrega nuestro email*

***git remote add origin https://github.com/DH/RepoRemoto*** *// apunta al repositorio remoto*







**Agregado de archivos al repositorio local**

Estos son los comandos que vimos en el video:

***git add .***”nombre”/ agrega solo este archivo al repositorio.

***git add .*** // agrega todos los archivos

***git status*** // seguimiento del estado de los archivos

**Confirmación de archivos en el repositorio local**

Cuando trabajamos con archivos, estamos acostumbrados a que estos se guarden en forma automática o, a pedirle al programa que lo guarde (el famoso Ctrl + S). Confirmar las modificaciones en Git es de suma importancia ya que nos permite establecer un punto de control. ¡Veamos cómo hacerlo!

Comandos

Estos son los comandos que utilizamos en el video para confirmar archivos:

***git add .*** // agrega los archivos

***git commit -m "mensaje"*** // comitea los cambios hechos

**Subida de archivos al repositorio remoto**

**Bajada de archivos desde un repositorio remoto**

**¿Cómo resolver conflictos?**