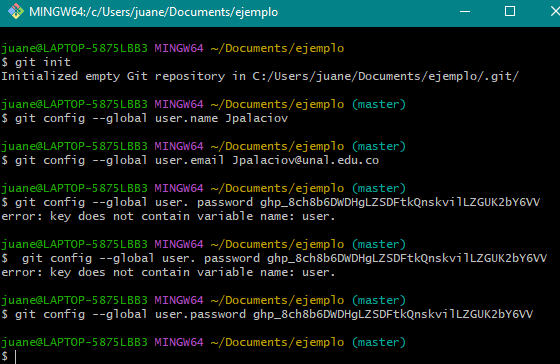
CONTROL DE VERSIONES

Un sistema de control de versiones permite GUARDAR datos o si se comete un error se podrá regresar en el tiempo, también se podrá trabajar de manera colaborativa e incluso en distintas plataformas como Word y Excel, so solamente en código. Otra gran ventaja es que se podrán crear distintas ramas y fusiones entre las mismas, tiene un servicio de almacenamiento autónomo y permite agilidad en las distintas metodologías.

Ejemplos:

* GitHub
* CVS
* Mercurial

Configuración git:



Comandos Básicos:

Una rama es una área o lugar específico de trabajo, se hace para dividir tareas, viene por defecto como *main, se hacen de acuerdo a la tarea a realizar*

* Git init: creará un nuevo repositorio local GIT
* Git clone: se usa para copiar un repositorio
* Git add: agrega archivos que se han cambiado junto con los archivos a ser confirmados
* Git commit: crea cometarios acerca de los cambios realizados
* Git config: establece una configuración especifica del usuario
* Git status: muestra la lista de los archivos que se han cambiado junto con los archivos confirmados
* Git push: permite subir los cambios definidos en el commit
* Git checkout: puedes ser usado para navegar entre ramas
* Git remote: permite ver todos los repositorios remotos
* Git branch: lista, crea o borra ramas (git branch, gitbranchNombreRama, git branch –d nombreRama)
* Git pull: fusiona los cambios que se han hecho en el repositorio (git pull)
* Git merge: se fusiona una rama con otra rama activa
* Git log: se usa para ver el historial del repositorio (git log)
* Git show head: muestra el estado actual del repositorio (git show head)
* Git reset: se usa para borrar datos del repositorio