**Git+Gihub**

**Todo un sistema de control de versiones de cero**

**Tema: Nuestro primer repositorio.**

**Documentación:** [**https://git-scm.com/**](https://git-scm.com/)

[**https://github.com/**](https://github.com/)

Al momento de trabajar con los comandos de Git con Windows se aconseja que trabaje con la PowerShell o el Git Bash y en Linux o OSX se aconseja que se trabaje con la terminal integrada del sistema operativo…

**Comando:** **git config ––global init.defaultBranch <Nombre de la rama>**

**Descripción:** Indicarle a Git cual será el nuevo nombre para la rama principal.

**Comando:** **git init**

**Descripción:** Inicializa un repositorio. **Nota:** Al momento de inicializar un repositorio, Git crea una carpeta con el nombre de **“.git”** de forma oculta. Este directorio nosotros nunca lo debemos de tocar, si nosotros borramos este directorio, se borraría toda la línea del tiempo del seguimiento de los cambios de nuestros archivos.

* Si nosotros volvemos a ejecutar el comando **“git init”** en el mismo repositorio, este se reinicia, pero no borra los commits realizados anteriormente.

**Comando:** **git status**

**Descripción:** Nos puede dar información sobre lo siguiente:

* Los commits.
* La rama en la que nosotros nos encontramos trabajando.
* Los archivos y directorios a los cuales no se les está dando un seguimiento **(Untracked files).** Si algunos de estos archivos se borran, a Git simplemente no le importan, porque no se han agregado con algún commit.
* Los archivos y directorios que están listos para realizarles un commit y que aún no se les hace un commit **(Changes to be committed).**
* Si no hay ningún archivo o directorio para realizarle un commit.

**Comando:** **git add .**

**Descripción:** Se toman todos los últimos cambios de los archivos de nuestro repositorio, los prepara y los sube al stage.

**Si por accidente nosotros subimos un archivo al stage y queremos eliminarlo, debemos utilizar lo siguiente.**

**Comando:** **git reset <Nombre del archivo>**

**Descripción:** Remover un archivo de stage que anteriormente se subió con el comando de **“git add .”**

**Comando:** **git commit ––message “primer commit” || git commit –m “primer commit”**

**Descripción:** Se agrega al repositorio local una FOTOGRAFÍA (copia de todos los cambios) de los archivos adheridos con el comando **“git add .”**

* **Cada fotografía (cada commit) que nosotros realizamos, nos permite realizar muchas cosas, como viajes en la línea del tiempo en cualquier punto en específico del repositorio.**
* Hacer commits frecuentemente, es algo que se acostumbra a realizar, usualmente se hace un commit cada vez que nosotros terminamos alguna funcionalidad importante o algo que vale la pena que se necesite que se quede con la actualización.

**Nota - CRLF**

**Nota de actualización.**

**Algunos en la próxima clase han presentado este problema con el CRLF, no es nada serio, es básicamente una interpretación de un carácter. Simplemente ejecuten este comando si presentan el error: git config core.autocrlf true**

**Una foto del posible error:**

