

## Proyecto de Software Actividad 2 - Git

La finalidad de esta actividad es resumir algunos de los conceptos básicos para configurar y comenzar a utilizar **Git** correctamente.

Pueden consultar la documentación y el libro del sitio oficial: <https://git-scm.com/> y <https://git-scm.com/book/es/v1> .

Las urls del servidor de la cátedra se centralizan junto con las presentaciones de git y gitlab dadas en: <https://www.proyecto2019.linti.unlp.edu.ar/>

### Actividad

1. Instalar Git: <https://git-scm.com/book/es/v1/Empezando-Instalando-Git> (para quienes traigan su notebook).
2. Configure su identidad globalmente utilizando el comando **git config**. Además del usuario y mail, ¿qué otras cosas pueden configurarse con este comando? ¿Qué cambios se produjeron?
3. Clonar el repositorio público (para ver en gitlab <https://gitlab.catedras.linti.unlp.edu.ar/proyecto2019-actividad/actividad-2>)
  - HTTP: <https://gitlab.catedras.linti.unlp.edu.ar/proyecto2019-actividad/actividad-2.git>
  - SSH: [git@gitlab.catedras.linti.unlp.edu.ar:proyecto2019-actividad/actividad-2.git](https://gitlab.catedras.linti.unlp.edu.ar/proyecto2019-actividad/actividad-2.git)

Investigue el contenido de la carpeta .git/.

4. Dentro del repositorio clonado anteriormente, genere una carpeta con su nombre y apellido respetando el siguiente formato **apellido\_nombre**.
5. Agregue contenido (archivos, carpetas nuevas) a su carpeta del repositorio recién creada utilizando **git add nombre\_de\_archivo\_o\_carpeta**.
6. Genere la primera versión dentro del nuevo repositorio utilizando el comando **git commit**.
7. Modifique algún archivo y vea esos cambios con el comando **git diff**, agregue estos cambios y realice un nuevo commit.
8. Antes de subir los cambios al repositorio remoto, ejecute el comando **git pull**, para obtener los cambios realizados por el resto de los/as alumnos/as.
9. Utilice el comando **git log** para ver los commits recién agregados por el resto de los/as compañeros/as.
10. Suba los cambios al repositorio remoto ejecutando el comando **git push origin master**.
11. Genere un nuevo branch **git checkout -b fix-bug-XXX**
12. Utilice el comando **git branch** para saber en qué branch se encuentra trabajando.
13. Agregue contenido (archivos, carpetas nuevas) a su carpeta del repositorio recién creada utilizando **git add nombre\_de\_archivo\_o\_carpeta**.
14. Genere la primera versión dentro del nuevo branch utilizando el comando **git commit**.

15. Realice el merge entre master y **fix-bug-XXX**. Para esto cambie de branch a master ejecutando **git checkout master**.
16. Realice el merge ejecutando **git merge fix-bug-XXX**
17. Elimine el branch generado en el paso 11 ejecutando el comando **git branch -d fix-bug-XXX**

Adicionalmente pude completar el siguiente tutorial: <https://learngitbranching.js.org/> que guiará en el uso de los comando básicos de Git.

Guías:

<https://guides.github.com/introduction/git-handbook/#basic-git>

<https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>