

## TEXT CLASS REVIEW

### TEMAS A TRATAR EN LA CUE

- Que es GIT
- Importancia de un sistema de control de versiones
- Comandos básicos para listar archivos o directorios

Muchas veces estamos desarrollando un programa, un informe o cualquier archivo que requiere ir avanzando poco a poco. En estas ocasiones, es bastante común generar distintas copias del mismo archivo:

- Archivo\_final
- Archivo\_final\_2.0
- Archivo\_final\_ultima\_version

Así es como surge la necesidad de tener registro del avance y desarrollo del trabajo que estamos realizando. Al momento de trabajar en la producción de un software, también se vuelve fundamental llevar un historial de él y un registro de todas esas versiones.

Normalmente todos los desarrolladores que se encuentran involucrados en el desarrollo de un proyecto van agregando cosas, realizando correcciones, nuevas funcionalidades y, si tenemos un historial de todo aquello, podemos ver que se ha hecho en un proyecto y quien lo ha hecho.



*Ilustración 1 Logo GIT*

## SISTEMA DE CONTROL DE VERSIONES GIT

GIT es un sistema de control de versiones diseñado por [Linus Torvalds](#) (creador de Linux) que nos ayuda a organizar el código de trabajo, el historial y evolución de este. En otras palabras, es un sistema que controla todas las versiones que existan de nuestro código, permitiéndonos navegar entre las diferentes versiones que han existido, crear ramas, que serán lugares donde podremos trabajar sin alterar la versión estable actual y que, cuando se encuentre terminada una nueva funcionalidad, nos permite incorporarla a la versión estable sin riesgos.

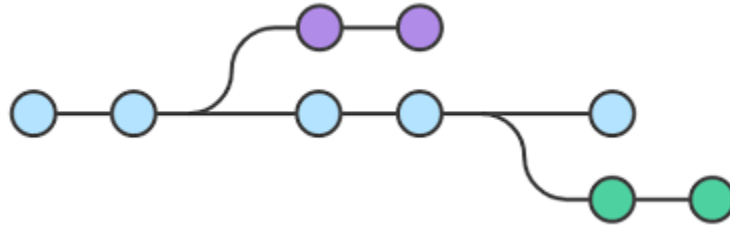
Además del control de las distintas versiones, GIT nos permite, a los distintos miembros de un equipo de trabajo, sincronizar las labores, registrar quien realizo alguna actualización y tener una copia del proyecto para poder manipularlo sin riesgo de manipular la versión estable.

Algunas de las características más importantes de GIT son:

- Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
- Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes.
- Re-almacenamiento periódico en paquetes.

## TERMINOLOGÍAS BÁSICAS

- **Commit:** es el estado de un proyecto en un determinado momento de la historia del mismo. Generalmente va acompañado de un mensaje descriptivo con los cambios que contiene.
- **Repositorio:** es la carpeta principal donde se encuentran almacenados los archivos que componen el proyecto.
- **Rama (Branch):** una rama es una línea alterna del tiempo, en la historia de nuestro repositorio. De esta forma no afectamos la versión estable del proyecto que es la rama principal. Por defecto, la rama principal es la rama máster.



*Ilustración 2 Concepto de Ramas*

Para conocer más de GIT podemos dirigirnos a su página oficial <https://git-scm.com/>

## OTROS SISTEMAS DE CONTROL DE VERSIONES

Para conceptos de nuestro aprendizaje nos centraremos en conocer GIT, sin embargo, es necesario saber que no es el único sistema de control de versiones que existe y que, si bien GIT es el sistema más utilizado, existen otras posibilidades, entre ellos:

**Mercurial:** Sistema de control de versiones gratuito y libre, parecido a GIT. Su página oficial es <https://mercurial.selenic.com/>

**SVN:** Apache Subversion es un sistema de control de versiones open source basado en un repositorio cuyo funcionamiento se asemeja a un sistema de ficheros. Su página web oficial es <http://subversion.apache.org/>

**CVS:** Concurrent Versions System es otro sistema de control de versiones del cual puedes saber más en la página <https://www.nongnu.org/cvs/>