



El empleo  
es de todos

Mintrabajo

# Análisis y Desarrollo de Software



@SENAcomunica

[www.sena.edu.co](http://www.sena.edu.co)

## ¿Qué es GIT?

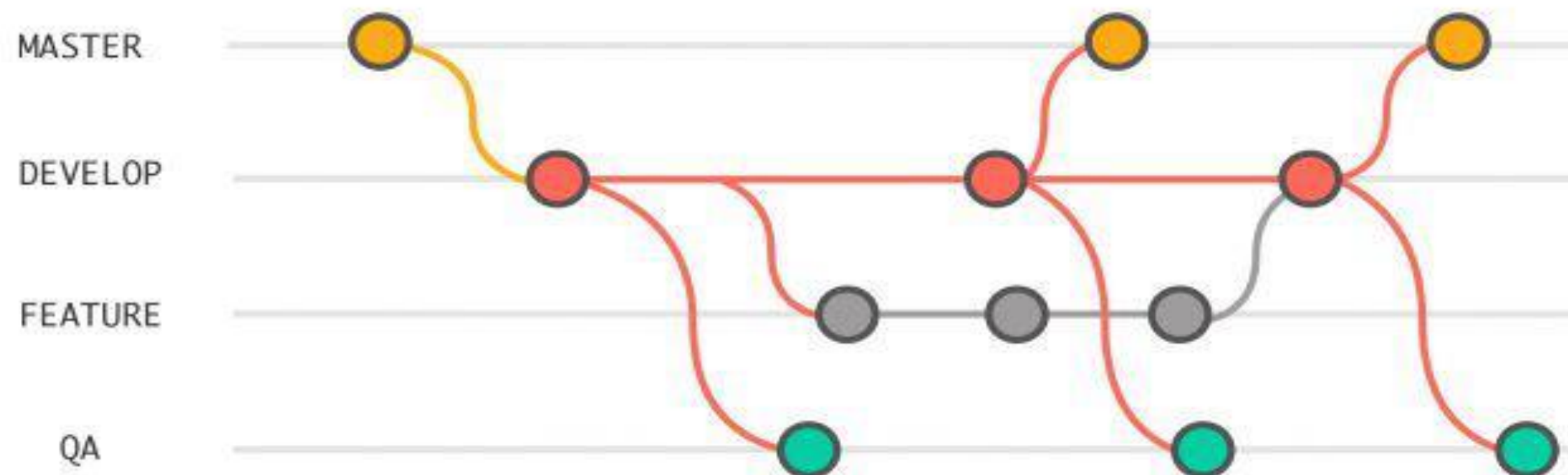
**Git es un sistema de control de versiones distribuido de código abierto y gratuito que permite por medio de repositorios, tener total control del recorrido histórico a lo largo de los proyectos.**



# ¿Qué lo hace especial?

La característica de Git que lo hace realmente interesante es su modelo de ramificación.

- Cambio de contextos sin fricciones.
- Código basado en roles.
- Flujo de trabajo basado en funciones.
- Experimentación desechable.





# Antes de GIT

Seguramente que muchos hemos experimentado la necesidad de solucionar la creación de múltiples archivos que hacen referencia a la misma información, además de la necesidad de trabajos en equipos, podamos acceder a la misma información, sin crear múltiples copias que dificultan luego la organización y control de trazabilidad del trabajo.



# Estados de GIT

Lo interesante de git es que nos ayuda:

- Llevar registro de lo que trabajamos.
- Coordinar el trabajo en equipo.
- Además Saber:
  - ¿Dónde ocurren los cambios?
  - ¿Cuándo ocurren?
  - ¿Quién los aplica?



# Desacoplemos la imagen

- Mi proyecto (Directorio o carpeta), solo existe en mi computador
  - **git init**
- Mi proyecto, se encuentra a la espera para irse al repositorio local.
  - **git add <file name>**
  - **git add .**
- Mi proyecto, se aloja en un repositorio local
  - **git commit -m "Descripción del commit"**

**working directory**

**staging area**

**repository**



Working Directory



Staging Area



Estoy listo

Reposit



Soy un repo



Rama main



commit inicial

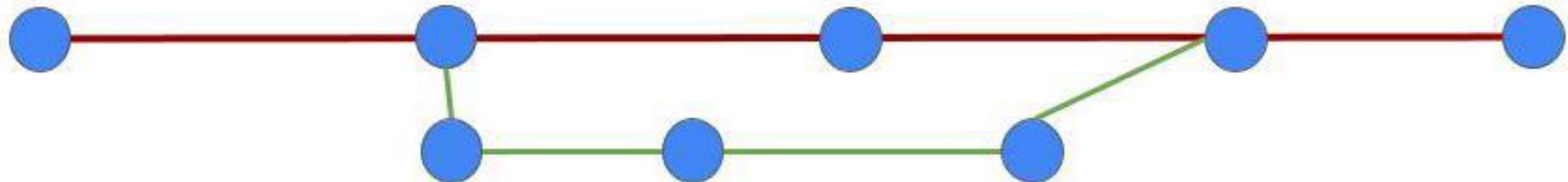
commit

commit

commit

commit

**NOTA:** Puedo crear varias ramas, y trabajar sobre el mismo proyecto, sin que esto interfiera, pues cada rama es independiente que más adelante podré fusionar si lo deseo con la rama principal.



# Lista de comandos en GIT



Comando	Descripción
<code>git config --global user.name "User Name"</code>	Configurar nombre de usuario de manera global en mi máquina.
<code>git config --global user.email email@gmail.com</code>	Configurar el email de manera global
<code>git init</code>	iniciar repositorios
<code>git config -- list</code>	Conocer los valores de configuración global
<code>git add path(ruta)</code> <code>git add .</code>	Permite que git le haga seguimiento al archivo agregado agrega todos los archivos que existen.
<code>git commit -m "Mensaje de guardado"</code>	Guardar modificaciones en repositorio.
<code>git commit -am "mensaje de guardado"</code>	Agrega y guarda las modificaciones(Sólo es válido si se ha realizado por lo menos una vez un <b>add</b> ).
<code>git checkout -b "branch name"</code>	Crear rama y ubicarse en ella
<code>git checkout branch name</code>	Ubicarse desde la rama deseada



# Lista de comandos en GIT



Comando	Descripción
<code>git status</code>	ver el estado actual de mis archivos
<code>git merge branch name</code>	El comando git merge permite tomar las líneas independientes de desarrollo creadas por git branch e integrarlas en una sola rama.
<code>git branch</code>	Ver ramas creadas en mi repositorio
<code>git diff</code>	git diff es un comando multiusos de Git que, cuando se ejecuta, lleva a cabo una función para establecer las diferencias en los orígenes de datos de Git.
<code>git diff branch1..other-feature-branch</code>	Comparar dos ramas.
<code>git mv OLD-FILENAME NEW-FILENAME</code>	renombrar archivo.
<code>git reset --soft HEAD~1</code>	revertir el ultimo commit y manteniendo los cambios
<code>git reset --hard HEAD~1</code>	revertir el ultimo commit y borrar los cambios
<code>git log --pretty=format:"%h - %an, %ar : %s"</code>	Ubicarse desde la rama deseada

# Lista de comandos en GIT



Comando	Descripción
git branch -M main	Cambia el nombre de la rama
git remote add origin <url de repo>	Agregar repositorio remoto
git remote remove origin	borrar conexión remota
git push -u origin <nombre rama>	Empujar los cambios al repo remoto.
git fetch	descarga commits, archivos y referencias de un repositorio remoto a tu repositorio local.
git pull	se emplea para extraer y descargar contenido desde un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido. Antes de hacer push siempre hacer pull es buena practica
git revert id_del_commit	revertir un commit ya publicado
git rm --cached <nombre archivo.ext>	eliminar un archivo del staging area y dejar de rastrearlo
git reset <nombre archivo.ext>	sacar un archivo del staging area
git branch -d nombrerama	eliminar una rama que ya ha sido fusionada

# Lista de comandos en GIT



Comando	Descripción
git checkout numero_commit	Revertir a un commit en específico
git reflog	devuelve un listado con los movimientos por los que ha pasado el repositorio
git commit --amend -m "nuevo mensaje para el último commit"	Añadir archivos al último commit, luego agrego con <b>git add “archivos faltantes”</b> , por último <b>git commit --amend --no-edit</b>
git reset HEAD <archivo>	Mover un archivo del staging area al workspace
git ckeckout -- <archivo>	Suponemos que hemos realizado algunos cambios sobre un archivo llamado archivo.txt pero queremos deshacerlos y que el archivo vuelva a tener el contenido con el que se guardó en el último commit en el repositorio.
git rm –cached nomarchivo	Elimina un archivo del stage
git show nomarchivo	Cambios ocurrido en un archivo, en los dos últimos commit
git commit –am “message”	Agrega los cambios y haces commit, esto es solo en los archivos que ya han sido agregados
git tag –a vx.x –m “mensaje” hash	Para crear un punto específico de la versión

# Lista de comandos en GIT



Comando	Descripción
git push origin --tags	Enviar al repositorio los tags creados
git tag -d nomtag	Borrar tags
git push origin :refs/tags/nomtag	Borrar tags en el repo remoto
gitk	Historia y estructura de ramas
git log --all --graph --decorate --oneline	Historia y estructura de ramas
alias arbol ="git log --all --graph --decorate --oneline "	Crear alias de lo anteriores
git tag	Ver listado de tags

<https://education.github.com/git-cheat-sheet-education.pdf>



# Referencias bibliograficas

<https://git-scm.com/book/es/v2/GitHub-Creaci%C3%B3n-y-configuraci%C3%B3n-de-la-cuenta>



**GRACIAS**

Línea de atención al ciudadano: 018000 910270  
Línea de atención al empresario: 018000 910682



@SENAcomunica

[www.sena.edu.co](http://www.sena.edu.co)