

# GIT



por **Gabriel Hernández Cardoso**

## ¿Qué es GIT?

Es un sistema de control de versiones.

## ¿Qué beneficios tiene GIT?

- Tiene un historial de versiones
- Colaboración
- Sistema distribuido
- Seguridad

## Conceptos claves:

- **Repositorio** : Es un tipo de almacenamiento digital centralizado que los desarrolladores utilizan para realizar y administrar cambios en el código fuente de una aplicación.
- **Commits** : Es la idea de conformar un conjunto de cambios provisionales de forma permanente.
- **Ramas** : Es una versión del código del proyecto sobre el que estás trabajando. Estas ramas ayudan a mantener el orden en el control de versiones y manipular el código de forma segura.
- **Staging** : Es donde se "cosechan" los cambios que formarán parte de la próxima confirmación.
- **Merge** : La fusión es la forma que tiene Git de volver a unir un historial bifurcado
- **Pull Request** : Es la situación donde un colaborador pide que revisen sus cambios antes de hacer merge a una rama, normalmente master.

## Como instalar GIT

Para instalar Git, dirígete a la línea de comandos y ejecuta el siguiente comando: `sudo dnf install git-all`.

Una vez que se haya completado la salida del comando, puedes verificar la instalación escribiendo: `git version`.

## Como instalar SourceTree

Para esto simplemente nos iremos a la página oficial de SourceTree, nos descargamos el instalador y lo configuraremos ajustándose a nuestras necesidades para tenerlo a nuestro gusto.

## Comandos básicos de GIT

- **git init** : Inicializa un nuevo repositorio, se usa al comenzar un nuevo proyecto para que git lo controle.
- **git clone** : Este clona un repositorio, para utilizarlo tenemos que poner en el cmd `git clone <url-del-repo>`, este lo que hace es descargar un repositorio en el ordenador
- **git add** : Añade archivos en el área de preparación, el staging, lo que hace es que prepara los archivos para ser confirmados en el commit, al utilizar tenemos dos opciones, '`git add <archivo>`' para añadir algo en específico, o bien, `git add .`, para añadir todas las modificaciones.
- **git commit** : Crea un commit en el estado de preparación, su uso sería `git commit -m "mensaje del commit"`, lo que hace que guarda los cambios hechos con el `git add`, se necesita un mensaje para comentar los cambios hechos.
- **git status** : Muestra el estado del repositorio, nos informa sobre los archivos modificados, los no rastreados y los que están listos para ser confirmados.
- **git log** : Comando que nos muestra el historial de commits, nos muestra id, autor, fecha y mensaje.
- **git diff** : Muestra los cambios entre archivos, mostrando las diferencias entre los archivos modificados pero no añadidos.
- **git branch** : Con este, si lo ponemos solo, podemos ver en que rama nos encontramos, y si le ponemos algún nombre, lo que hace es crear una nueva rama.
- **git checkout** : Lo podemos utilizar para cambiar de rama, aunque también podemos restaurar un archivo a su último estado.
- **git merge** : Este fusiona ramas, su uso es `git merge <nombre-de-la-rama>`, obtiene los objetos del repositorio remoto y los fusiona con la rama local.
- **git pull** : Descarga y fusiona cambios desde el repositorio remoto.
- `git push` : Envía commits al repositorio remoto.
- **git remote** : Gestiona los repositorios remotos, podemos usar `git remote -v`, que nos muestra las URL de los remotos configurados, o también podemos usar `git remote add <nombre> <url>` para añadir un dispositivo remoto.
- **git fetch** : Este descarga cambios del repositorio remoto sin actualizar, es decir que no lo fusiona con nuestra rama local, solo descarga los cambios de una rama remota.

- **git stash** : Guarda temporalmente los cambios no confirmados, para que podamos cambiar de rama sin perder cambios.

### **Como crear un nuevo repositorio en git hub**

Para esto simplemente, nos iremos a nuestra cuenta de git hub, en el inicio, en la esquina superior derecha hay un símbolo más, en el que si pulsamos, nos da la opción de crear un nuevo repositorio.

### **Como subir algo a un repositorio de git hub**

Para este simplemente solo tendremos que hacer esto que nos dice el mismo git :

```
git init
```

```
git add README.md
```

```
git commit -m "primer commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/chmpagnegaby/Desarrollo-web-en-entorno-servidores.git
```

```
git push -u origin main
```