



## Introduction to GIT

Git es un sistema de control de versiones distribuido, diseñado para gestionar proyectos de desarrollo de software con velocidad y eficiencia. Es una herramienta que se utiliza para llevar un registro de los cambios en el código fuente de un proyecto y permitir a varios desarrolladores trabajar en el mismo proyecto de manera simultánea.



## Basic commands in GIT

**Git clone:** Comando para descargarte el código fuente

`git clone <https://link-con-nombre-del-repositorio>`

**Git branch:** Comando para crearlas, listarlas y eliminarlas.

`git branch <nombre-de-la-rama>`

**Git checkout:** Para cambiarte de una rama a otra. También lo podemos usar para chequear archivos y commits.

`git checkout <nombre-de-la-rama>`

**Git status:** Nos da toda la información necesaria sobre la rama actual.

`git status`

**Git add:** El comando para incluir los cambios del o de los archivos en el siguiente commit.

`git add <archivo>`

**Git commit:** Para guardar nuestros cambios (quizás después de una tarea o asunto específico).

`git commit -m "mensaje de confirmación"`

**Git push:** Después de haber confirmado tus cambios.

`git push <nombre-remoto> <nombre-de-tu-rama>`

**Git pull:** El comando se utiliza para recibir actualizaciones del repositorio remoto.

`git pull <nombre-remoto>`

**Git revert:** A veces, necesitaremos deshacer los cambios que hemos hecho.

`git revert 3321844`

**Git merge:** Git merge básicamente integra las características de tu rama con todos los commits realizados a las ramas dev (o master).

`git merge <nombre-de-la-rama>`



## Working with repositories on GitHub (Branches, Merge, Conflicts)

### Acerca de los repositorios

Un repositorio es el elemento más básico de GitHub. Es un lugar donde puedes almacenar el código, los archivos y el historial de revisiones de cada archivo. Los repositorios pueden contar con múltiples colaboradores y pueden ser públicos como privados.

Puedes ser propietario de repositorios individualmente o puedes compartir la propiedad de los repositorios con otras personas en una organización.

En cualquier caso, los permisos administran el acceso a los repositorios.





### Terminología de los repositorios

Término	Definición
Sucursal	Una versión paralela del código contenido en el repositorio, pero que no afecta a la rama principal.
Clonar	Para descargar una copia completa de los datos de un repositorio de GitHub.com, incluidas todas las versiones de cada archivo y carpeta.
Bifurcar	Un nuevo repositorio que comparte la configuración de visibilidad y código con el repositorio «ascendente» original.
Solicitud de incorporación de cambios	Una solicitud para combinar los cambios de una rama en otra.
Merge	Para aplicar los cambios de una rama y en otra.
Control remoto	Un repositorio almacenado en GitHub, no en el equipo.
Ascendente	La rama de un repositorio original que se ha bifurcado o clonado. La rama correspondiente de la rama clonada o bifurcada se denomina «descendente».

### Ramas (Branches)

Las ramas en GitHub permiten a los desarrolladores trabajar en diferentes características o correcciones de errores de manera aislada. La rama principal se llama main o master. Las ramas adicionales se crean para desarrollar nuevas funcionalidades o corregir errores sin afectar la rama principal.

- Crear una rama:

```
git branch nombre-de-la-rama
git checkout nombre-de-la-rama
```

### Fusión (Merge)

La fusión es el proceso de combinar los cambios de una rama con otra. Generalmente, se fusionan las ramas de características con la rama principal una vez que el desarrollo y las pruebas han concluido.

- Fusionar una rama con main:

```
git checkout main
git merge nombre-de-la-rama
```





Nombre: Gina Belen Molina Gutiérrez

Fecha: 1 Semana

Tema: Git and GitHub

## Conflictos (Conflicts)

Los conflictos ocurren cuando dos ramas tienen cambios en las mismas líneas de código o en áreas que no pueden combinarse automáticamente. Git no puede resolver estos conflictos por sí mismo y requiere intervención manual.

- Resolver conflictos:
  - a. Git marcará los archivos en conflicto después de un intento de fusión.
  - b. Abre los archivos conflictivos y busca las marcas de conflicto (<<<<<<, =====, >>>>>>).
  - c. Edita el archivo para resolver los conflictos manualmente.
  - d. Después de resolver los conflictos, añade los archivos al área de preparación:  
`git add nombre-del-archivo`
  - e. Finaliza la fusión:  
`git commit -m "nota"`

## Advanced commands (Pull Request, Fork, Rebase, Stach, Clean, Cherry, pick, etc.)

### Pull Request (PR):

Un Pull Request es una solicitud para que los cambios realizados en una rama sean revisados y fusionados con otra rama (generalmente main). Es una herramienta fundamental para la colaboración en GitHub.

### Fork:

Un Fork es una copia de un repositorio en tu propia cuenta de GitHub. Esto permite hacer cambios libremente sin afectar el repositorio original.

```
git clone https://github.com/tu-usuario/nombre-del-repositorio.git
```

### Rebase:

Rebase se usa para integrar cambios de una rama en otra, reescribiendo el historial de commits. Es útil para mantener un historial de commits más limpio.

- Rebase una rama con main

```
git checkout nombre-de-la-rama
git rebase main
```

### Stash

Stash guarda temporalmente los cambios no confirmados para que puedas trabajar en otra cosa y luego recuperarlos.

- Guardar cambios en stash:

```
git stash
```

- Ver stashes disponibles:

```
git stash list
```

### Clean

Clean elimina archivos no rastreados en el repositorio de trabajo.

- Eliminar archivos no rastreados:

```
git clean -f
```

### Cherry-pick

Cherry-pick aplica un commit específico de una rama en otra.

- Aplicar un commit específico:

```
git cherry-pick <commit-hash>
```

