

## Introducción a Git

### ¿Qué es Git?

Git es un sistema de control de versiones distribuido que permite gestionar el historial de cambios en un proyecto. Con Git, múltiples colaboradores pueden trabajar en el mismo proyecto simultáneamente y mantener un registro de las modificaciones realizadas en el código.

### Conceptos clave:







**Repositorio (Repository):** Un repositorio es un lugar donde se almacena el código de un proyecto junto con su historial de versiones. Puede ser local en tu máquina o remoto en un servidor.

**Commit:** Un commit es un conjunto de cambios que se registran en el historial de versiones. Cada commit tiene un mensaje que describe los cambios realizados.

**Branch (Rama):** Un branch es una línea independiente de desarrollo. Puedes tener varias ramas en un repositorio, lo que permite trabajar en características o correcciones de errores sin afectar la rama principal (normalmente llamada master o main).





**Merge:** La operación de merge combina los cambios de una rama en otra. Por lo general, se realiza para fusionar cambios de una rama secundaria de vuelta a la rama principal.

## Comandos básicos en Git


-  **git init:** Inicia un nuevo repositorio Git en tu directorio local.
-  **git clone URL:** Clona un repositorio remoto en tu máquina.
-  **git add archivo(s):** Añade cambios al área de preparación (staging area) para el próximo commit.
-  **git commit -m "mensaje":** Registra los cambios añadidos en el área de preparación en el historial de versiones con un mensaje descriptivo.
-  **git status:** Muestra el estado actual del repositorio, incluyendo archivos modificados y archivos en el área de preparación.
-  **git pull:** Actualiza tu rama local con los cambios más recientes desde la rama remota configurada. Combina automáticamente los cambios descargados en tu rama local.

## Trabajando con repositorios en Git (Branches, Merge, Conflicts)

### Branches (Ramificación):

-  **git branch:** Lista todas las ramas en el repositorio.
-  **git branch nombre\_rama:** Crea una nueva rama con el nombre especificado.
-  **git checkout nombre\_rama:** Cambia a la rama especificada.
-  **git merge nombre\_rama:** Fusiona los cambios de la rama especificada en la rama actual.

### Merge (Fusión):

-  **git merge nombre\_rama:** Fusiona los cambios de la rama especificada en la rama actual.

## Conflictos:

- ✚ Durante un merge, si dos ramas modifican la misma parte de un archivo, puede ocurrir un conflicto.
- ✚ **git pull:** Actualiza el repositorio local con los cambios remotos.
- ✚ **git mergetool:** Abre una herramienta de fusión para resolver conflictos.
- ✚ Después de resolver conflictos, ejecutar git commit para finalizar la fusión.

## Trabajando con Repositorios en Git:

### Branches (Ramificación):

Crear una nueva rama:

**git branch nombre\_rama:** Crea una nueva rama con el nombre especificado para desarrollar nuevas funcionalidades o realizar cambios sin afectar la rama principal.

Cambiar a una rama existente:

**git checkout nombre\_rama:** Cambia al contenido de la rama especificada, permitiéndote trabajar en una línea de desarrollo separada.

Crear y cambiar a una nueva rama de manera combinada:

**git checkout -b nueva\_rama:** Crea y cambia a una nueva rama en un solo comando.

Listar todas las ramas:

**git branch:** Muestra una lista de todas las ramas en tu repositorio, resaltando la rama actual con un asterisco.

### Merge (Fusión):

Fusionar una rama en la rama actual:

**git merge nombre\_rama:** Combina los cambios de la rama especificada en la rama actual, comúnmente usado para incorporar nuevas funcionalidades o correcciones de errores a la rama principal.

Eliminar una rama después de la fusión:

**git branch -d nombre\_rama:** Elimina una rama secundaria después de fusionar sus cambios en la rama principal.

## Conflictos:

Resolver conflictos durante la fusión:

Ejecutar git pull puede llevar a conflictos, para resolver el conflicto se deben seguir los siguientes pasos:

1. Abrir el archivo conflictivo.
2. Modificar para conservar las partes deseadas.
3. Marcar el conflicto como resuelto con git add.
4. Completar la fusión con git merge --continue.

### Etiquetas (Tags):

Crear una etiqueta ligera:

**git tag nombre\_etiqueta:** Crea una etiqueta para marcar un punto específico en la historia del proyecto, como una versión estable.

Crear una etiqueta anotada:

**git tag -a nombre\_etiqueta -m "Mensaje de la etiqueta":** Crea una etiqueta anotada con un mensaje descriptivo.

Listar etiquetas:

**git tag:** Muestra una lista de todas las etiquetas en el repositorio.

Colaboración:

Copiar un repositorio remoto:

**git clone URL:** Clona un repositorio remoto en tu máquina, proporcionando una copia local para trabajar.

Enviar cambios al repositorio remoto:

**git push origin nombre\_rama:** Envía tus cambios locales al repositorio remoto en la rama especificada.