

ACTIVIDAD 8. Desarrollo Full stack

5/02/2025

¿Qué es GIT?

GIT es un sistema de control de versiones distribuido y de código abierto que permite a los desarrolladores gestionar los cambios en sus proyectos de software. Su principal ventaja es que facilita el trabajo colaborativo al permitir que múltiples personas modifiquen archivos de forma simultánea sin riesgo de perder información. Con GIT, se puede llevar un seguimiento detallado de cada modificación, facilitando revertir cambios si es necesario.

The image shows the text 'git init' in a stylized, italicized font, representing the command used to initialize a new Git repository.

¿Cómo iniciar un proyecto con GIT?

Para comenzar a utilizar GIT en un proyecto, el primer paso es abrir una terminal y dirigirse a la carpeta donde se encuentra el proyecto o, si aún no existe, crear un nuevo directorio. Luego, se debe ejecutar el comando:

- Este comando creará un directorio oculto llamado **.git**, el cual almacenará toda la información necesaria para gestionar las versiones del proyecto.

¿Qué es un merge en GIT?

El comando merge en GIT se usa para combinar los cambios de una rama con otra. Su función principal es unificar el trabajo realizado en diferentes ramas dentro del proyecto, asegurando que las modificaciones de ambas partes se mantengan en el historial de cambios. Es muy útil en entornos colaborativos donde se crean ramas temporales para el desarrollo de nuevas funciones y posteriormente se integran en la rama principal.

Comandos esenciales en GIT

1. Configuración inicial

Antes de comenzar a trabajar con GIT, es necesario establecer el nombre de usuario y el correo electrónico con los siguientes comandos:

```
git config --global user.name "TuNombre"
```

```
git config --global user.email "tuemail@example.com"
```

```
git config --list
```



2. Inicializar un repositorio

Para iniciar un nuevo repositorio en GIT:

```
git init
```

3. Agregar y confirmar cambios

Para ver el estado del repositorio y agregar archivos al área de preparación:

```
git status
```

```
git add nombre-del-archivo
```

```
git add . # Agrega todos los archivos modificados
```

```
git commit -m "Mensaje descriptivo del cambio"
```

4. Consultar el historial de cambios

```
git log
```

```
git log --oneline
```

```
git diff # Muestra diferencias entre la versión actual y la última confirmada
```

```
git diff --staged # Muestra diferencias de los archivos preparados para commit
```



5. Manejo de ramas

Las ramas permiten trabajar en diferentes versiones del código sin afectar la principal:

git branch # Lista todas las ramas disponibles

git branch nombre-rama # Crea una nueva rama

git checkout nombre-rama # Cambia a una rama existente

git checkout -b nombre-rama # Crea y cambia a una nueva rama en un solo paso

git merge otra-rama # Fusiona otra-rama con la rama actual

git branch -d nombre-rama # Elimina una rama

6. Trabajar con repositorios remotos

Para conectar el proyecto con un repositorio remoto y sincronizar cambios:

git remote add origin url-del-repositorio

git remote -v # Verifica los repositorios remotos configurados

git push -u origin main # Sube cambios a la rama principal del remoto

git pull origin main # Descarga cambios desde el repositorio remoto

git clone url-del-repositorio # Clona un repositorio en la



7. Uso de etiquetas en GIT

Las etiquetas permiten marcar versiones importantes del proyecto:

git tag # Muestra todas las etiquetas creadas

git tag -a v1.0 -m "Versión 1.0" # Crea una etiqueta con un mensaje

git push origin --tags # Sube las etiquetas al repositorio remoto

Diferencias entre GIT, GitHub, GitHub Desktop y GitKraken

- **GIT:** Es la herramienta base, funciona en la terminal y permite gestionar versiones de código localmente.
- **GitHub:** Es una plataforma en línea que permite almacenar repositorios de GIT y facilita el trabajo colaborativo.
- **GitHub Desktop:** Proporciona una interfaz gráfica sencilla para gestionar repositorios de GIT sin necesidad de usar la terminal.
- **GitKraken:** Es una aplicación avanzada con interfaz gráfica que permite gestionar múltiples repositorios en una sola ventana, ideal para quienes buscan mayor facilidad visual en el manejo de GIT.

