

Video 1: Control de Versiones con Git y GitHub

¿Qué es Git?

Git es la herramienta de control de versiones más utilizada por programadores. Su función es clara: gestiona versiones de archivos de forma eficaz, algo vital en proyectos colaborativos.

¿Quién creó Git y por qué es tan relevante?

El creador de Git es Linus Torvalds, el mismo desarrollador detrás del núcleo de Linux, quien creó esta herramienta para resolver sus propias necesidades de control de versiones.

¿Cómo funciona Git en tu entorno de trabajo?

Git se usa en la terminal o en editores como Visual Studio Code, utilizando comandos clave como commit, pull, push, merge, entre otros.

¿Qué aprenderás en este curso de Git?

En este curso, aprenderás a:

- Configurar Git en tu computadora y crear repositorios locales.
- Realizar cambios, crear ramas, fusionarlas y gestionar el flujo de trabajo en equipo.
- Configurar un repositorio remoto en GitHub para colaborar en proyectos.
- Revisar y fusionar aportes de otros programadores, resolver conflictos y entender un flujo de trabajo profesional.

Video 2: Fundamentos de Git

Comandos de Git

Comando para verificar la versión de git en el computador: **git --versión**

Comando para inicializar git: **git init**

Comando para ajustar el nombre de la rama principal por defecto: **git config --global init.defaultBranch main**

Comando para asegurar el nombre de la rama: **git branch -m main**

Comando para revisión de ayudas con el nombre de los demás comandos: **git -help**

Comando básico para configurar el usuario de la instalación de git: **git config --global user.name "Nombre de usuario"**

Comando básico para configurar el correo de la instalación de git: **git config --global user.email "Dirección de email"**

Comando para verificar los cambios hechos en la instalación de git: **git config --list**

Comandos adicionales de la terminal

Comando para limpiar la terminal: clear

Comando para crear una carpeta: mkdir <nombre de la carpeta>

Mover entre carpetas: cd <nombre de directorio>

Video 3: Comandos Básico 2 y Flujo de Trabajo

Comandos de Git

Comando para revisar el estado de cambios en el repositorio: **git status**

Comando para cancelar el envío de un archivo antes de hacer un commit: **git rm --cached <nombre del archivo>**

Comando para confirmar los cambios que están en Staging: **git commit -m "Mensaje"**

Comandos adicionales de la terminal

Comando para revisar los archivos ocultos de Windows: **ls -a**

Conceptos:

Staging: El estado intermedio que están los archivos antes de realizar un commit.

Comandos imagen

```
### 📁 Inicialización
git init                # Inicializa un nuevo repositorio Git
git clone <repo-url>    # Clona un repositorio desde una URL

### 🛠️ Desarrollo diario
git status              # Muestra el estado de los cambios
git add <archivo>       # Añade cambios al área de preparación
git commit -m "Mensaje" # Confirma los cambios con un mensaje

### 🌲 Gestión de ramas
git branch              # Lista las ramas
git branch <nombre-rama> # Crea una nueva rama
git switch <nombre-rama> # Cambia a una rama
git branch -d <nombre-rama> # Elimina una rama

### 💚 Integración y colaboración
git merge <rama>        # Fusiona los cambios de una rama
git remote add <nombre> <url> # Añade un repositorio remoto
git push <remoto> <rama>    # Envía los cambios a un repositorio remoto
git pull <remoto> <rama>    # Obtiene y fusiona cambios desde un repo remoto

### 🔄 Recuperación y limpieza
git fetch               # Recupera cambios sin fusionar
git reset --hard HEAD   # Desecha todos los cambios locales
git revert <hash-commit> # Revierte los cambios de un commit

### ✨ Avanzado y utilidades
git diff <a> <b>         # Compara dos commits, ramas o archivos
git show <hash>         # Muestra detalles de un commit
git stash               # Guarda cambios temporales y limpia el working tree
git stash pop           # Restaura los cambios guardados del stash
git cherry-pick <hash>  # Aplica un commit específico en la rama actual
git rebase <base>       # Reaplica commits sobre otra base
```

Video 4: Gestión de ramas en Git: creación, fusión y eliminación eficiente

Comandos de Git

Comando para consultar en que ramas te encuentras: **git branch**

Comando para moverse entre ramas: **git branch switch <nombre de la rama>**

Comando para unir la rama al principal: **git merge <nombre de la rama a fusionar>**

Comando para eliminar la rama fusionada: **git branch -D <nombre de la rama>**

Video 5: Git Reset vs Git Revert: Manejo de Historial y Corrección de Errores