



Bootcamp

Desarrollo Web Full Stack

Nivel Básico
David Ricardo Rivera Arbeláez

UT TALENTOTECH

Módulo 4

Control de versiones con Git

- Introducción a Control de versiones
- Git (Repositorios, commits, branches)
- Trabajando con Git (push, pull, merge)
- Repositorios remotos (GitHub, pull request, issues)

Introducción a Control de versiones

El control de versiones es una práctica fundamental para cualquier desarrollador web, ya sea principiante o experimentado. Git es uno de los sistemas de control de versiones más populares y utilizados, y ofrece una amplia gama de funciones para rastrear, gestionar y colaborar en proyectos de software.

¿Qué es Git?

- Es un sistema de control de versiones distribuido (DVCS).
- Esto significa que cada desarrollador tiene una copia completa del historial del proyecto en su máquina local.
- Esto permite trabajar sin conexión a internet y facilita la colaboración en proyectos.

¿Por qué usar Git?

- Registro de cambios.
- Colaboración.
- Resolución de conflictos.
- Integración continua.

Conceptos básicos de Git

- Repositorio.
- Commit.
- Branch.
- Merge.
- Remote.

Comandos básicos de Git

- git init.
- git add.
- git commit.
- git branch.
- git checkout.
- git merge.
- git push.
- git pull.

Crear nuevo repositorio desde la línea de comandos

```
echo "# demo" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin URL REPOSITORIO  
git push -u origin main
```


Subir un repositorio existente desde la línea de comandos

```
git remote add origin URL REPOSITORIO  
git branch -M main  
git push -u origin main
```

Taller práctico de Git

1. Realizar cambios → Cada desarrollador realiza cambios en el código de su copia local.
2. Agregar cambios → Los desarrolladores usan el comando **git add** para agregar los cambios al área de preparación.
3. Crear un commit → Los desarrolladores crean commits con mensajes descriptivos que expliquen los cambios realizados

Taller práctico de Git

1. Subir cambios al remote → Los desarrolladores usan el comando **git push** para enviar sus cambios al remote, por ejemplo, a un repositorio GitHub.
2. Descargar cambios del remote → Otros desarrolladores pueden usar el comando **git pull** para descargar los cambios del remote y actualizar sus copias locales.
3. Crear branches → Si un desarrollador desea trabajar en una nueva característica sin afectar el código principal del proyecto, puede crear un nuevo Branch.
4. Fusionar cambios → Una vez que la nueva característica está completa, el desarrollador puede fusionar los cambios del Branch con el código principal.

Control de versiones con Git

Recursos de apoyo

- [1] [Sitio web oficial y guía oficial de Git](#)
- [2] [Guía oficial de Git](#)
- [3] [Curso de Git y GitHub en español](#)



¡Gracias!