



PRESENTACIÓN

Nombre y Apellido:

Naroly Tolentino

Matricula:

20231785

Docente:

Kelyn Belliard

Materia:

Programación III

Tema:

Asignación individual III

Fecha:

2/4/2025

Contents

¿Qué es Git?	3
¿Para qué sirve el comando git init?	3
¿Qué es una rama en Git?	3
¿Cómo saber en cuál rama estoy trabajando?	3
¿Quién creó Git?	3
¿Cuáles son los comandos esenciales de Git?	3
¿Qué es Git Flow?	4
¿Qué es el desarrollo basado en trunk?	4
Bibliography	4

1. ¿Qué es Git?

Es un sistema de control de versiones distribuido que permite a varias personas trabajar en un mismo proyecto de forma colaborativa. Fue diseñado para ser rápido, eficiente y flexible, especialmente en proyectos de desarrollo de software. Con Git, puedes guardar el historial de cambios de tus archivos, volver a las versiones anteriores si algo sale mal y coordinar el trabajo en equipo.

2. ¿Para qué sirve el comando git init?

Se usa para empezar un proyecto con Git desde cero. Cuando lo ejecutas en una carpeta, crea un repositorio nuevo en ese lugar. Básicamente inicializa el control de versiones colocando una subcarpeta oculta llamada .git, donde se guarda toda la información del historial, las ramas y los cambios.

3. ¿Qué es una rama en Git?

Es como una línea paralela de desarrollo dentro de un proyecto. Es como si divides el trabajo en caminos separados para probar cosas nuevas o hacer cambios sin afectar el código principal.

4. ¿Cómo saber en cuál rama estoy trabajando?

Usando el comando git branch. Esto muestra una lista de todas las ramas que existen en tu repositorio, y la rama en la que estás trabajando aparece marcada con un asterico (*). También puedes usar git status, que te dice el nombre de la rama actual al inicio del mensaje.

5. ¿Quién creó Git?

Git fue creado por Linus Torvalds, un programador famoso por también haber creado el sistema operativo Linux.

6. ¿Cuáles son los comandos esenciales de Git?

- git init: para empezar un repositorio nuevo
- git clone: para copiar un repositorio que ya existe desde un servidor
- git add: para agregar cambios de archivos al área de preparación antes de guardarlos.
- git commit: guarda los cambios preparados con un mensaje que explica qué hiciste.
- git push: envía tus cambios al repositorio remoto

- git pull: trae los cambios del repositorio remoto a tu computadora
- git branch: crea, lista o maneja ramas
- git checkout: cambia entre ramas o restaura archivos
- git merge: une los cambios de una rama a otra

7. ¿Qué es Git Flow?

Es una forma organizada de usar ramas en Git para manejar proyectos. Es como un plan que te dice cómo nombrar y usar las ramas para que todo quede claro.

8. ¿Qué es el desarrollo basado en trunk?

Es otro estilo de trabajar con Git, mas simple que Git Flow. Aquí, todos trabajan directamente en una sola rama principal, llamada trunk o a veces main. En lugar de hacer muchas ramas largas, haces cambios pequeños y los integras rápido al trunk con commits frecuentes.

Bibliography

Atlassian. (2025, Abril 2). *Gitflow Workflow*. Retrieved from Atlassian:

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

Community, S. E. (2025, Abril 2). *What is the difference between Trunk-Based Development and Gitflow?*

Retrieved from Software Engineering Stack Exchange:

<https://softwareengineering.stackexchange.com/questions/442910/what-is-the-difference-between-trunk-based-development-and-gitflow>

Development, T. B. (2025, Abril 2). *Trunk-Based Development*. Retrieved from Trunk Based

Development: <https://trunkbaseddevelopment.com/>

Documentation, G. (2025, Abril 2). *Introducción a Git*. Retrieved from Git Documentation: [https://git-](https://git-scm.com/doc)

[scm.com/doc](https://git-scm.com/doc)