Nombre:

Pamela Blanco

Matrícula:

20231668

Carrera:

Desarrollo de software

Materia:

Programacion III

Nombre del docente:

Kelyn Tejada Belliard

Tema:

Tarea 3:

Fecha

03/04/2025

Indice

1. ¿Qué es Git?

Git es un sistema de control de versiones de código fuente. Es una herramienta que se utiliza para llevar un registro de los cambios en el código fuente de un proyecto y permitir a varios desarrolladores trabajar en el mismo proyecto de manera simultánea.

Con Git, es posible crear un repositorio que contenga el código fuente de un proyecto y llevar un registro de todos los cambios realizados en el repositorio. Los desarrolladores pueden colaborar en el mismo proyecto al hacer «commits» (guardar cambios) en el repositorio y luego «pushear» (enviar) estos cambios a un servidor centralizado para que puedan ser utilizados por otros desarrolladores.

Git es una herramienta muy popular en la comunidad de desarrollo de software debido a su flexibilidad, velocidad y capacidad para manejar grandes proyectos. También es ampliamente utilizado en proyectos de código abierto, ya que permite a cualquier persona contribuir al proyecto y hacer seguimiento de los cambios realizados en el código fuente.

2. ¿Para qué sirve el comando git init?

El comando git init sirve para crear un nuevo repositorio de Git. Se usa para configurar un nuevo repositorio o convertir un proyecto existente en uno de Git.

Se usa una sola vez durante la configuración inicial de un repositorio nuevo. Se puede usar para convertir un proyecto existente y sin versión en un repositorio de Git .

3. ¿Qué es una rama en Git?

Un branch o rama en Git es una línea de desarrollo independiente que se crea a partir de una rama principal, generalmente llamada master. Esta rama nueva puede ser utilizada para realizar cambios, experimentar con nuevas funcionalidades o corregir errores sin afectar la rama principal.

4. ¿Cómo saber en cuál rama estoy trabajando?

Para saber en qué rama estás trabajando en Git, puedes usar estos comandos:

Comando principal:

git branch

- Este comando muestra todas las ramas locales y marca con un * la rama en la que te encuentras.

Otra opción más detallada:

git status

- Este comando te muestra información sobre el estado del repositorio, incluyendo la rama actual, si hay cambios sin confirmar, archivos en staging, etc.

Mostrar la rama en la terminal (opcional):
Si quieres que tu terminal siempre muestre la rama en la que estás, puedes usar este comando en Git Bash o en la terminal de Linux/Mac:

git rev-parse --abbrev-ref HEAD

- Este comando devuelve únicamente el nombre de la rama actual.

5. ¿Quién creó Git?

Git es un sistema de control de versiones (VCS) distribuido que se ha convertido en el estándar de facto para el desarrollo de software. Fue creado por Linus Torvalds, el creador del kernel de Linux, en 2005. Git se caracteriza por su eficiencia, confiabilidad y compatibilidad con grandes proyectos de código fuente.

6. ¿Cuáles son los comandos esenciales de Git?

Algunos comandos esenciales de Git son:

- git init → Inicializa un repositorio Git en una carpeta.
- git clone <URL> → Clona un repositorio remoto en tu máquina.
- git status → Muestra el estado del repositorio.
- git add <archivo> → Agrega archivos al área de preparación (staging).
- git commit -m "mensaje" → Guarda los cambios en el historial con un mensaje.
- git push → Envía los cambios al repositorio remoto.
- git pull → Descarga y fusiona cambios del repositorio remoto.
- git branch → Lista las ramas disponibles.
- git checkout <rama> → Cambia a otra rama.
- git merge <rama> → Fusiona una rama con la actual.

7. ¿Qué es Git Flow?

Git Flow es un flujo de trabajo que define una estructura clara para el desarrollo con ramas en Git. Introduce ramas específicas para cada etapa del desarrollo, como:

- main (o master): Contiene la versión estable en producción.
- develop: Rama principal donde se integran las nuevas funcionalidades antes de pasar a producción.
- feature: Ramas para desarrollar nuevas características.
- release: Ramas para preparar versiones antes de fusionarlas en main.
- hotfix: Ramas para corregir errores críticos en producción.

Se usa con una extensión de Git llamada git-flow para gestionar las ramas de manera más estructurada.

8. ¿Qué es el desarrollo basado en trunk (Trunk Based Development)?

El Trunk Based Development (TBD) es una estrategia de desarrollo en la que los desarrolladores trabajan directamente en una rama principal (trunk, que suele ser main o master) en lugar de usar múltiples ramas de larga duración. Se enfoca en hacer commits frecuentes y pequeños, minimizando la necesidad de fusiones complejas. Es ideal para integración continua (CI/CD) y facilita la entrega rápida de software.

Bibliografía:

1. Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress. Recuperado de: https://git-scm.com/book/en/v2
2. Git Documentation. (n.d.). *Git Reference Manual*. Recuperado de: https://git-scm.com/doc
3. Atlassian. (n.d.). *Git Flow: A Successful Branching Model*. Recuperado de: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow
4. Vincent Driessen. (2010). *A Successful Git Branching Model*. Recuperado de: https://nvie.com/posts/a-successful-git-branching-model/
5. Loeliger, J., & McCullough, M. (2012). *Version Control with Git* (2nd ed.). O'Reilly Media.