Taller

1. Responda los siguientes enunciados
   1. Explica brevemente qué es el control de versiones y por qué es importante en el desarrollo de software.
   2. Enumera al menos tres comandos de Git que se utilizan comúnmente al iniciar un nuevo repositorio.
   3. ¿Por qué es importante configurar tu nombre de usuario y dirección de correo electrónico en Git?
   4. ¿Para qué se usan los comandos git clone, git add, git commit, y git push?
   5. Explica qué es una rama (branch) en Git y cómo se utiliza para el desarrollo de software en equipo.

Respuestas:

a. El control de versiones protege el código fuente tanto de las catástrofes como del deterioro casual de los errores humanos y las consecuencias accidentales, y es importante porque sincroniza las versiones y garantiza que los cambios no entren en conflicto con los cambios de otros usuarios.

b. git init, git add y git commit -m "mensaje"

c. Lo primero que se debe hacer al instalar Git es establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque los "commits" de Git usan esta información, y es introducida de manera inmutable en los commits que envías.

d. Git add: Mueve los cambios del directorio de trabajo al área del entorno de ensayo. Así puedes preparar una instantánea antes de confirmar en el historial oficial.

Git clone: Crea una copia de un repositorio de Git existente. La clonación es la forma más habitual de que los desarrolladores obtengan una copia de trabajo de un repositorio central.

Git commit: Confirma la instantánea preparada en el historial del proyecto. En combinación con git add, define el flujo de trabajo básico de todos los usuarios de Git.

e.  Es una versión del código del proyecto sobre el que estás trabajando. Estas ramas ayudan a mantener el orden en el control de versiones y manipular el código de forma segura.

Las ramas en Git son esenciales para el desarrollo de software en equipo, ya que permiten a los desarrolladores trabajar en diferentes características o correcciones de forma aislada. Al crear una rama, cada miembro del equipo puede realizar cambios sin afectar el código principal, lo que minimiza conflictos. Esto facilita la colaboración, ya que cada desarrollador puede compartir su progreso mediante pull requests, permitiendo revisiones y pruebas antes de integrar los cambios.