

Fundamentos Esenciales de Git y GitHub

Cuestionario de comprensión:

- 1) ¿Cuál es el principal beneficio de usar la bifurcación (forking) en un VCS como Git?

El principal beneficio de usar bifurcación (forking) en un sistema de control de versiones como Git es que permite crear una copia completa e independiente de un repositorio, sobre la cual puedes trabajar libremente sin afectar el proyecto original.

Ventajas clave:

- Permite experimentar, modificar o mejorar el código sin riesgo para el repositorio principal.
- Facilita la colaboración en proyectos open source, ya que los cambios pueden enviarse luego mediante un pull request.
- Da control total sobre la copia del proyecto (historial, ramas, configuraciones, etc.).

La bifurcación permite contribuir o desarrollar de forma aislada y segura, manteniendo la integridad del repositorio original.

- 2) ¿Por qué es importante configurar la identidad global (user.name y user.email) antes de realizar cualquier acción en Git?
 - Identifica al autor de cada cambio.
Cada *commit* que hagas en Git incluye información sobre quién lo realizó. Esto permite saber qué persona hizo qué cambio, algo esencial para la trazabilidad y colaboración en equipos.
 - Garantiza la integridad del historial.
Git usa esta información para mantener un historial confiable y transparente, lo que facilita auditorías, revisiones y la resolución de conflictos.

- Evita problemas al sincronizar con repositorios remotos.
Plataformas como GitHub o GitLab usan el correo electrónico configurado para asociar tus commits con tu cuenta. Si no está bien configurado, tus contribuciones pueden aparecer sin autor o no vincularse a tu perfil.

configurar tu identidad global asegura que tus contribuciones sean correctamente registradas, reconocidas y rastreables dentro del proyecto.

- 3) ¿Qué término se usa para describir a Git, indicando que cada usuario tiene una copia completa del historial del proyecto?
 - Esto significa que cada usuario tiene una copia completa del repositorio, incluyendo todo el historial de cambios, las ramas y las versiones del proyecto.
 - En consecuencia, los desarrolladores pueden trabajar sin conexión, realizar *commits* y ver el historial localmente, sincronizando los cambios con otros solo cuando sea necesario.