

Title Comandos de Git.

Keyword	Topic
① Repositorio	Clone, ~
② Clonar	El comando git clone es esencial para crear una copia local completa de un repositorio que se encuentra en un servidor remoto. Permite a los desarrolladores obtener no solo los archivos actuales de un proyecto, sino también todo el historial de commits, ramas y cambios realizados en ese repositorio. Este proceso es útil cuando se desea colaborar en proyectos existentes o extraer proyectos de código abierto. Al clonar, se debe proporcionar la URL del repositorio remoto, y el sistema lo descargará en un nuevo directorio local. Además, se puede clonar el repositorio completo o seleccionar una sola rama o subconjunto del historial, utilizando las opciones adicionales del comando.
③ Remoto	
④ Local	
⑤ Copia	
⑥ Ramas	
⑦ Commits	
Questions	
① ¿Es posible clonar solo una rama específica de un repositorio?	
② ¿Qué más cosas se pueden realizar con este comando?	Es una herramienta clave para que varios colaboradores puedan trabajar simultáneamente en sus propias copias locales del mismo proyecto sin interferir en el trabajo de otros, manteniendo el historial y los cambios sincronizados.

Summary: El comando git clone permite copiar un repositorio remoto en una ubicación local, trayendo archivos, historial de cambios y configuraciones, facilitando la colaboración en proyectos compartidos.

Title Comandos de Git.

Keyword	Topic Push ~
① Push	
② Cambios	Git Push se utiliza para enviar los cambios realizados en el repositorio local a un repositorio remoto. Este comando es clave para la colaboración en proyectos, ya que permite compartir actualizaciones y sincronizar el trabajo de múltiples desarrolladores. Funciona enviando commits
③ Confirmados	
④ Repositorio	
⑤ Local	
⑥ Remoto	
⑦ Sincronización	
⑧ Colaboración	
⑨ Autenticación	
⑩ Conflictos.	
Questions	
⑪ ¿Cómo se pueden proteger las ramas remotas de sobrescripciones accidentales?	Los cambios que hayan sido confirmados (commit) y no incluye archivos no versionados o que aún no han sido añadidos al control de versiones. Además, se puede especificar tanto la rama remota como la local, y es posible que se soliciten credenciales de autenticación para completar el proceso.
⑫ ¿Qué implicaciones tiene forzar un Push?	El comando también ofrece opciones avanzadas, como forzar el Push, lo que sobrescribe el historial remoto, pero esta acción debe realizarse con cautela para evitar conflictos o pérdida de datos en el repositorio compartido.

Summary: Envía los cambios confirmados en el repositorio local a uno remoto, actualizando la rama correspondiente y permitiendo compartir avances en un proyecto colaborativo. Es un comando fundamental para compartir los progresos de un proyecto con otros colaboradores.

Title Comandos de Git.

Keyword	Topic
① Commit	Commit ~
② Confirmar	Git commit se utiliza para confirmar los cambios en el repositorio local, creando una instantánea de los archivos que han sido modificados o añadidos desde el último commit. Este comando es clave en el flujo de trabajo de Git, ya que permite registrar un estado estable del proyecto en un momento determinado. Al ejecutar commit, se puede incluir un mensaje descriptivo que resume los cambios realizados, lo que facilita el seguimiento y la colaboración entre equipos.
③ Cambios	
④ Respaldo	
⑤ Local	
⑥ Mensaje	
⑦ Instantánea	
⑧ Histórico	
Questions	
① ¿Cómo se pueden incluir archivos no añadidos a un commit?	Es importante destacar que solo los archivos previamente añadidos al área de preparación serán parte del commit, mientras que los archivos sin seguimiento o no añadidos no se incluirán. Un buen uso de los commits ayuda a mantener un historial limpio y organizado, permitiendo referirse a versiones anteriores en caso de ser necesario.
② ¿Cómo se gestiona el orden de los commits al trabajar en múltiples ramas?	Además, se pueden usar opciones adicionales como -amend, que permite modificar el último commit, o -m, que facilita agregar un mensaje en una línea sin dejar un editor de texto.

Summary: Confirma los cambios en el repositorio local, registrando una instantánea del estado del proyecto junto a un mensaje descriptivo de los cambios realizados.

Title Comandos de Git.

Keyword	Topic	Branch ~
① Ramas ② Gestión ③ Crear ④ Eliminar ⑤ Listar ⑥ Activa ⑦ Asterisco ⑧ Flujo de trabajo		<p>Git branch se utiliza para gestionar y listar ramas en un repositorio de git. Las ramas permiten a los desarrolladores trabajar en diferentes características, correcciones o experimentos de forma apartada del código principal. Al usar branch, se puede crear una nueva rama, borrar una existente o renombrar una rama actual. El comando también muestra una lista de todas las ramas desplazables, destacando la rama activa con un asterisco.</p>
Questions		<p>La funcionalidad de branch es esencial para la organización del flujo de trabajo en proyectos colaborativos. Permitiendo a los equipos gestionar el desarrollo de manera más eficiente. Para cambiar entre ramas, se combina con checkout, mientras que para fusionar cambios, se utilizan comandos adicionales como merge. Además, el comando branch tiene opciones como -d para eliminar ramas de manera segura y -a para listar todas las ramas, tanto locales como remotas.</p>

Summary: Gestiona las ramas en un repositorio, permitiendo crear, eliminar y listar ramas, facilitando el trabajo en diferentes características y correcciones de manera aislada.

Title Comandos de Git.

Keyword	Topic
① Checkout	Checkout ~
② Cambiar	Git checkout se usa para cambiar de rama en un repositorio. Git o hace restaurar archivos en el área de trabajo a un estado anterior. Permite a los desarrolladores combinar entre diferentes ramas, facilitando el trabajo en diversas características o correcciones de forma independiente. Al ejecutar git checkout nombre-de-rama, Git cambia el contexto del área de trabajo a la rama seleccionada, permitiendo ver y modificar el código de esa rama. También se puede usar para deshacer cambios en archivos no comprometidos o hacer restaurar archivos a un estado previo usando git checkout -- nombre-de-archivo.
③ Ramas	
④ Restaurar	
⑤ Modificar	
⑥ Deshacer	
⑦ Área de trabajo	
⑧ Estado anterior	
⑨ Archivos	
Questions	
① ¿Cómo se puede crear y cambiar a una nueva rama en un solo comando?	Este comando es crucial para la gestión eficiente del flujo de trabajo, especialmente cuando se trabaja en proyectos con múltiples ramas. Además, checkout puede ser combinado con otros comandos como git branch para gestionar ramas y con git commit para guardar los cambios realizados en la rama actual.
② ¿Qué sucede si se hace checkout en una rama con cambios no comprometidos?	

Summary: Permite cambiar entre ramas en un repositorio y restaurar archivos a estados anteriores, facilitando la gestión de diferentes aspectos del desarrollo y el trabajo en varias ramas.