

¿Cuál es el propósito de utilizar "git stash clear" en Git? *

1/1

- ☒ Eliminar permanentemente todos los cambios guardados en el stash.
- ☐ Restaurar los cambios guardados en el stash.
- ☐ Guardar los cambios en un stash con un mensaje descriptivo.

¿Cuál es el comando utilizado para ver el historial de commits en Git? *

1/1

- ☐ git status
- ☐ git checkout
- ☒ git log

¿Cuál es el propósito principal de utilizar el comando "git restore" en Git? *

1/1

- ☐ Deshacer el último commit y eliminar permanentemente los cambios realizados en los archivos
- ☐ Mostrar los cambios realizados en los archivos desde el último commit.
- ☒ Restaurar archivos específicos a su estado anterior sin afectar otros cambios o commits.

¿Qué comando se utiliza para mover archivos desde el Working Directory al Staging Area en Git? *1/1

- ☐ git commit
- ☒ git add
- ☐ git status



¿Qué se debe hacer después de crear/modificar el archivo .gitignore? *

1/1

- ☐ Fusionar los cambios en la rama principal del repositorio.
- ☒ Publicar los cambios y revisar qué sucede con los archivos ignorados
- ☐ Eliminar todos los archivos mencionados en el .gitignore.

¿Qué se entiende por el Repository en Git? *

1/1

- ☐ Es el lugar donde se hacen los cambios en los archivos.
- ☒ Es el lugar donde se almacenan los commits realizados.
- ☐ Es el lugar donde se preparan los cambios para un commit.

¿Qué es un repositorio en Git? *

1/1

- ☐ Una herramienta para hacer copias de seguridad de los archivos de un proyecto.
- ☒ Una ubicación para almacenar los archivos de un proyecto junto con su historial de versiones.
- ☐ Un archivo de registro de cambios en un proyecto.



¿Cuál es la precaución que debes tener en cuenta al utilizar el comando "git reset --hard" en Git? * 1/1

- ☐ El comando "git reset --hard" no afecta los commits y solo deshace los cambios en los archivos.
- ☐ Los commits eliminados se pueden recuperar fácilmente utilizando el comando "git restore".
- ☒ Los commits eliminados se pierden permanentemente y no se pueden recuperar fácilmente.

¿Cuál de las siguientes afirmaciones describe mejor a Git? *

1/1

- ☒ Un sistema de control de versiones distribuido.
- ☐ Una herramienta de diseño gráfico.
- ☐ Un lenguaje de programación.

¿Qué es una rama en Git? *

1/1

- ☐ Un registro de los cambios realizados en los archivos en un momento dado.
- ☒ Una bifurcación en el desarrollo del proyecto que permite trabajar en paralelo sin afectar la rama principal.
- ☐ Una copia de seguridad de todos los archivos en un repositorio.



¿Qué comando se utiliza para descargar cambios de un repositorio remoto y fusionarlos automáticamente con un repositorio local? *1/1

- ☐ git push
- ☒ git pull
- ☐ git fetch

¿Cuál es el objetivo principal del Git Workflow? *

1/1

- ☐ Permitir el desarrollo continuo del proyecto mediante la creación y fusión de ramas.
- ☐ Asegurar que todos los miembros del equipo se adapten rápidamente al flujo de trabajo.
- ☒ Todas las opciones son correctas.

¿Cómo se verifica que los cambios aplicados con "git cherry-pick" están presentes en la rama actual?

*0/1

- ☒ Utilizando el comando "git log" para revisar el historial de cambios.
- ☐ Inspeccionando los archivos modificados y comprobando que los cambios deseados estén presentes.
- ☐ Todas las opciones son correctas.

Respuesta correcta

- ☒ Todas las opciones son correctas.



¿Qué comando se utiliza para descargar cambios de un repositorio remoto a un repositorio local sin fusionarlos automáticamente? *1/1

- ☐ git pull
- ☒ git fetch
- ☐ git push

¿Cuál es el propósito principal del archivo .gitignore en Git? *

1/1

- ☒ Ignorar archivos y directorios específicos en el repositorio.
- ☐ Restablecer todos los cambios a una versión anterior.
- ☐ Rastrear y guardar cambios en un repositorio.

¿Cuál de las siguientes afirmaciones es verdadera sobre el control de versiones con Git?

*1/1

- ☒ Ayuda a mantener un registro histórico de cambios realizados en un proyecto.
- ☐ Solo se utiliza en proyectos pequeños y no es adecuado para proyectos grandes.
- ☐ No permite revertir cambios a versiones anteriores de un proyecto.

¿Qué comando se utiliza para clonar un repositorio remoto en Git? *

1/1

- ☐ git pull
- ☒ git clone
- ☐ git commit



¿Qué sucede después de crear un Pull Request en GitHub? *

1/1

- ☒ El propietario del repositorio revisa los cambios y realiza comentarios.
- ☐ Los cambios se descargan automáticamente en el repositorio local.
- ☐ Los cambios se fusionan automáticamente en la rama principal del repositorio.

¿Cuál es el propósito principal del comando "git push"? *

1/1

- ☐ Descargar un repositorio remoto a la computadora local.
- ☐ Crear una nueva rama en el repositorio remoto.
- ☒ Enviar los cambios locales al repositorio remoto.

¿Qué hace Git cuando ocurre un conflicto de fusión? *

1/1

- ☐ Descarta los cambios conflictivos y continúa con la fusión de manera automática.
- ☐ Resuelve automáticamente los conflictos y realiza la fusión correctamente.
- ☒ Pausa el proceso de fusión y solicita al usuario que resuelva los conflictos manualmente.

¿Qué hace el comando "git stash" en Git? *

1/1

- ☒ Guarda temporalmente los cambios realizados en el directorio de trabajo.
- ☐ Elimina permanentemente los cambios no deseados en un archivo.
- ☐ Crea una copia de seguridad de todos los archivos en el repositorio.



¿Cuál es el comando utilizado para ver el estado actual de un repositorio en Git? *1/1

- ☒ git status
- ☐ git merge
- ☐ git log

¿Qué es importante tener en cuenta sobre el uso del comando "git restore" en Git? *1/1

- ☒ "git restore" solo restaura los archivos modificados en el directorio de trabajo y no modifica el historial de commits.
- ☐ "git restore" afecta tanto los archivos en el área de trabajo como el historial de commits.
- ☐ "git restore" solo puede revertir cambios en archivos específicos, no en todo el directorio de trabajo.

¿Cuál es el comando utilizado para confirmar los cambios en Git y registrarlos en el historial de versiones? *1/1

- ☐ git push
- ☐ git branch
- ☒ git commit



¿Cuál es el propósito de utilizar "git cherry-pick" en Git? *

1/1

- ☒ Seleccionar y aplicar cambios específicos de un commit en otra rama sin fusionar todo el historial.
- ☐ Restaurar un commit eliminado previamente en el repositorio.
- ☐ Eliminar permanentemente un commit del historial del repositorio.

¿Qué comando se utiliza para ver la lista de cambios guardados en git stash? * 1/1

- ☐ git stash log
- ☐ git stash show
- ☒ git stash list

¿Cuál es el comando utilizado para ver todas las ramas en un repositorio Git? * 1/1

- ☐ git log
- ☒ git branch
- ☐ git status

¿Qué sección del comando "git status" muestra los cambios realizados en los archivos que no se han realizado commit? * 1/1

- ☐ "Untracked files"
- ☐ "Changes to be committed"
- ☒ "Changes not staged for commit"



¿Qué comando se utiliza para deshacer todos los cambios realizados en tus archivos y volver a un commit anterior en Git? *1/1

- ☐ git reset --hard HEAD1
- ☒ git reset --hard <commit-id>
- ☐ git reset --soft <commit-id>

¿Cuál es el comando utilizado para ver las ramas fusionadas en Git? *1/1

- ☐ git log
- ☒ git branch --merged
- ☐ git status

¿Qué información se necesita para utilizar el comando "git cherry-pick"? *1/1

- ☐ El mensaje descriptivo del commit que deseas aplicar.
- ☒ El ID del commit que contiene los cambios específicos que deseas aplicar.
- ☐ El nombre de la rama en la que se encuentra el commit que deseas aplicar.

¿Cuál es el propósito del Working Directory en Git? *1/1

- ☒ Es el lugar donde se hacen los cambios en los archivos.
- ☐ Es el lugar donde se preparan los cambios para un commit.
- ☐ Es el lugar donde se almacenan los commits realizados.



¿Qué se incluye en la carpeta creada por "git clone" en la computadora local? * 1/1

- ☒ Todos los archivos y la estructura del proyecto, incluyendo el historial de versiones y ramas.
- ☐ Solo los archivos más recientes del repositorio remoto.
- ☐ Solo los archivos que han sido confirmados mediante commit.

¿Qué comando se utiliza para crear un nuevo commit que deshace los cambios realizados en un commit anterior en Git? * 1/1

- ☐ git undo <commit-id>
- ☐ git commit --revert <commit-id>
- ☒ git revert <commit-id>

¿Qué son los conflictos de fusión ("merge conflicts") en Git? * 1/1

- ☐ Cambios que se realizan en el mismo archivo sin afectar el historial de versiones.
- ☒ Situaciones en las que se combinan archivos que tienen cambios conflictivos.
- ☐ Errores que ocurren durante el proceso de fusión en Git.



¿En qué casos sería útil utilizar "git revert" en Git? *

0/1

- ☒ Cuando necesitas deshacer los cambios realizados en el último commit y volver al estado anterior.
- ☐ Cuando quieres deshacer los cambios realizados en un commit específico sin eliminarlo por completo y sin afectar los commits posteriores.
- ☐ Cuando deseas eliminar permanentemente todos los commits anteriores y comenzar desde cero.

Respuesta correcta

- ☒ Cuando quieres deshacer los cambios realizados en un commit específico sin eliminarlo por completo y sin afectar los commits posteriores.

¿Qué comando se utiliza para fusionar una rama específica en la rama actual en Git? * 1/1

- ☒ git merge
- ☐ git pull
- ☐ git branch

¿Cuál es el comando utilizado para cambiar a una rama específica en Git? * 1/1

- ☐ git push
- ☐ git pull
- ☒ git switch



¿Qué se entiende por el Staging Area en Git? *

0/1

- ☒ Es el lugar donde se almacenan los commits realizados.
- ☐ Es el lugar donde se hacen los cambios en los archivos.
- ☐ Es el lugar donde se preparan los cambios para un commit.

Respuesta correcta

- ☒ Es el lugar donde se preparan los cambios para un commit.

¿Cuál es la diferencia principal entre "git revert" y "git reset" en Git? *

1/1

- ☐ "git revert" puede deshacer todos los cambios realizados en tus archivos y volver al estado anterior, mientras que "git reset" deshace los cambios realizados en el último commit.
- ☐ "git revert" y "git reset" son comandos equivalentes y se pueden utilizar indistintamente en cualquier situación.
- ☒ "git revert" crea un nuevo commit que deshace los cambios realizados en un commit anterior, mientras que "git reset" puede eliminar commits y modificar el historial del proyecto.

Este formulario se creó en Egg Cooperation.

Google Formularios

