Practicas de GIT

Estados de un archivo: Untracked, Unmodified, Modified y Staged

Estados de Git: Working Directory, Staging Area y LocalRepo

1)

**Consigna**: Después de clonar un repositorio, creas un nuevo archivo llamado "nuevo\_documento.txt" y lo añades al área de preparación. Luego, realizas cambios en el archivo "viejo\_documento.txt" y lo añades al área de preparación también. Finalmente, decides eliminar el archivo "borrar.txt" del directorio local y ejecutas git rm borrar.txt. ¿En qué estado y sección de Git se encuentran "nuevo\_documento.txt", "viejo\_documento.txt" y "borrar.txt" respectivamente después de estos pasos?

**Respuesta y Justificación:**

"nuevo\_documento.txt"; Staged y Staging Area (estaría en el estado Staged y en la sección Staging Area porque se agregó al área de preparación con git add.)

"viejo\_documento.txt"; Staged y Staging Area ("viejo\_documento.txt" estaría en el estado Modified y en la sección Working Directory porque se realizaron cambios en el archivo después de haber sido rastreado por Git.)

"borrar.txt”; Untracked y Working Directory ("borrar.txt" estaría en el estado Untracked y en la sección Working Directory porque fue eliminado del directorio local pero aún no se ha añadido al área de preparación ni al repositorio.)

2)

**Consigna:** Clonas un repositorio y realizas varios cambios en el archivo "codigo.py". Añades estos cambios al área de preparación y realizas un commit. Luego, decides deshacer los últimos cambios que realizaste en "codigo.py". ¿Cómo puedes revertir estos cambios para que "codigo.py" vuelva a su estado anterior al último commit?

**Respuesta y Justificación:**

"codigo.py"; Unmodified y Working Directory y RepoLocal (Después de realizar un commit en Git, los archivos que han sido confirmados y no han tenido modificaciones posteriores estarán en el estado "Unmodified" en el "Working Directory" y también estarán presentes en el "Repositorio local", reflejando la última versión confirmada del archivo.)

3)

**Consigna:** Clonas un repositorio y creas dos archivos nuevos llamados "archivo1.txt" y "archivo2.txt". Luego, modificas "archivo1.txt" y lo añades al área de preparación. Después, decides eliminar "archivo2.txt" del directorio local y ejecutas git rm archivo2.txt. ¿En qué estado y sección de Git se encuentran "archivo1.txt" y "archivo2.txt" respectivamente después de estos pasos?

**Respuesta y Justificación:**

"archivo1.txt"; Staged y Staged Area y Working Directory ("archivo1.txt" estaría en el estado Staged y en la sección Staging Area porque se agregó al área de preparación con git add archivo1.txt.)

"archivo2.txt"; Untracked y Working Directory ("archivo2.txt" estaría en el estado Untracked y en la sección Working Directory porque fue eliminado del directorio local pero aún no se ha añadido al área de preparación ni al repositorio local.)

4)

**Consigna:** Clonas un repositorio y creas un archivo "informacion.txt". Después, modificas este archivo y ejecutas git add informacion.txt. Luego decides revertir los cambios en "informacion.txt" para que vuelva a su estado anterior al último commit. ¿Qué comandos utilizarías y en qué estado y sección quedaría el archivo después de estos pasos?

**Respuesta y Justificación:**

"informacion.txt”; Untracked y Working Directory (Después de modificar y agregar el archivo "informacion.txt" al área de preparación con git add informacion.txt, el archivo estaría en el estado Staged y en la sección Staging Area., pero cuando reinvertiste los cambios y restaurar el archivo a su estado anterior al último commit, "informacion.txt" volvería al estado Untracked y en la sección Working Directory)

5)

**Consigna:** Comienzas clonando un repositorio y abriendo el archivo "documento.pdf". Después, decides cambiar algunas palabras, haces un "add" seguido de un "commit" y revisas el estado con "status". Te das cuenta de que faltan tildes y procedes a corregirlas, realizando otro "commit" y luego un "git add".

Luego, creas un nuevo archivo llamado "documento2.pdf" y le aplicas un "git add". Sin embargo, decides revertir los cambios en "documento.pdf". Posteriormente, haces un commit con el mensaje "Enviando los archivos 'documento2.pdf'" y haces un push.

**Respuesta y Justificación:**

"documento.pdf"

Estado: Unmodified (Porque al hacerle un PUSH este coincidiría con el repo Remoto)

Sección de Git: Working Directory y LocalRepo(Al hacerle un add y posteriormente hacerle un commit)

"documento2.pdf"

Estado: Unmodified (Porque al hacerle un PUSH este coincidiría con el repo Remoto)

Sección de Git: Working Directory y LocalRepo(Al hacerle un add y posteriormente hacerle un commit)

6)

**Consigna:** Clonas un repositorio, creas un nuevo archivo "informacion.txt" en el directorio principal del repositorio. Luego, modificas este archivo y lo agregas al área de preparación con git add. Realizas un commit con el mensaje "Agregando información al archivo informacion.txt". Finalmente, ejecutas un push al repositorio remoto.

**Respuesta y Justificación:**

"informacion.txt"

Estado: Unmodified (Porque al hacerle un PUSH este coincidiría con el repo Remoto)

Sección de Git: Working Directory y LocalRepo(Al hacerle un add y posteriormente hacerle un commit)

7)

**Consigna:** Clonas un repositorio, eliminas un archivo llamado "documento.pdf" que existe en el repositorio. Luego, realizas un commit con el mensaje "Eliminando archivo documento.pdf".7)

**Respuesta y Justificación:**

"documento.pdf"

Estado: Untracked (Porque el archivo existía y fue eliminado por lo cual pasaría a ese estado)

Sección de Git: Working Directory y RepoLocal (Porque fue eliminado del Working Directory, y para borrarlo por completo se hizo un commit para poder borrarlo también en el RepoLocal. Así eliminándolo por completo)

8)

**Consigna:** Clonas un repo. Ahora tienes "configuracion.txt" en tu directorio de trabajo. Realizas modificaciones en este archivo y lo añades al área de preparación utilizando el comando. Luego, decides revertir los cambios en "configuracion.txt" ¿Qué comandos utilizarías y en qué estado y sección quedaría el archivo después de estos pasos?

**Respuesta y Justificación:**

Estado: Unmodified

Sección de Git: Working Directory y RepoLocal

9)

**Consigna:** Luego de clonar. Tienes un archivo llamado "datos.txt" en tu directorio de trabajo. Modificas este archivo y luego haces un “add”. A continuación, decides eliminar "datos.txt" utilizando el comando git rm datos.txt. En el que decides poner un add para luego hacer un “commit” ¿En qué estado y sección de Git se encuentra "datos.txt" después de estos pasos?

**Respuesta y Justificación:**

Estado: --------

Sección de Git: ----------

10)

**Consigna:** Clonas un repositorio. Luego, creas un nuevo archivo llamado "nuevo\_archivo.txt" en tu directorio de trabajo y haces un git add nuevo\_archivo.txt. Por último, confirmas los cambios utilizando el comando commit. ¿En qué estado y sección de Git se encuentra "nuevo\_archivo.txt" después de estos pasos?

**Respuesta y Justificación:**

Estado: Staged

Sección de Git: Working Directory y repositorio local

11)

**Consigna:** Clonas un repo, abro el archivo “hola.txt” y lo modifico luego hago un commit y un push origin main.

**Respuesta y Justificación:**

“hola.txt”

Estado: Modified

Sección de Git: Working Directory y repositorio local

12)

**Consigna:** Clono un repo, creo el archivo”hola.txt” y abro para solo leer el archivo “Chao.txt” que ya existía en el repo. Ejecuto un add “Chao.txt”

**Respuesta y Justificación:**

”hola.txt”

Estado: Untracked

Sección de Git: Working Directory

“Chao.txt”

Estado: Unmodified

Sección de Git: Working Directory y repositorio local

13)

**Consigna:** En un repositorio que tenía mi pc, ejecuto un git status y me dice que el archivo “hola.txt” está en estado Untracked, por lo que ejecuto un commit.

**Respuesta y Justificación:**

Estado: Untracked

Sección de Git: Working Directory

14)

**Consigna:** Clono un repo, modifico el archivo “hola.txt”, hago un commit, y luego un add.

**Respuesta y Justificación:**

Estado: Staged

Sección de Git: Working Directory y repositorio local y repositorio local