



Git Version Control Workflow

GIT: CASOS DE USO.



por Ing. Ramiro Nogales



Situación 0: Primer día de clase

0.a. El estudiante crea su cuenta en GitHub

1. El estudiante debe ir a [GitHub](#) y hacer clic en **Sign up**.
2. Completar el formulario con un nombre de usuario, correo electrónico y contraseña. Por favor colocar un nombre de usuario pequeño, máximo de 8 a 10 caracteres, sin usar espacio en blanco.
3. Verificar la cuenta a través del correo electrónico enviado por GitHub.

Comentario: Esto es necesario para que el estudiante tenga su propio espacio en GitHub.

0.b. El estudiante clona el repositorio del docente en el computador del aula

El estudiante clona el repositorio del docente usando:

- git clone https://github.com/GBCURSOS/FSB1_152.git

Comentario: Esto descarga el repositorio del docente al computador del aula como base para trabajar.

0.c. El estudiante sube el repositorio en el computador del aula al repositorio de su cuenta

1. **Crear un repositorio vacío en la cuenta del estudiante:**
 - El estudiante va a GitHub, hace clic en **New repository** y lo nombra por ejemplo: el estudiante se llama Ramiro Nogales, entonces: FSB152_RN. Esto es un ejemplo, NO tiene por que hacerse igual.
 - No inicializar con README ni otros archivos.
2. **Cambiar la URL del repositorio remoto (origin) para apuntar al repositorio del estudiante:**
 - git remote set-url origin https://github.com/RNOGALES/FSB152_RN.git
3. **Verificar la nueva URL:**
 - git remote -v
4. **Subir los archivos iniciales al repositorio del estudiante:** git push -u origin main

Comentario: Esto carga los archivos del repositorio del docente, que están en el computador del aula, al repositorio personal del estudiante.

Situación 1: En la casa

1.a. El estudiante clona o descarga el repositorio de su cuenta en su computador personal

1. Clonar el repositorio del estudiante:

- git clone https://github.com/RNOGALES/FSB152_RN.git
- cd FSB152_RN

Comentario: Esto descarga el repositorio del estudiante al computador personal.

1.b. El estudiante modifica sus datos en su computador

El estudiante realiza cambios locales en los archivos. Por ejemplo:

- Editar un archivo HTML.
- Agregar un archivo con su nombre.

1.c. El estudiante sincroniza los datos de su computador al repositorio de su cuenta

1. Añadir los cambios:

- git add .

2. Hacer commit:

- git commit -m "Descripción de los cambios realizados"

3. Subir los cambios al repositorio del estudiante:

- git push origin main

Comentario: Esto actualiza el repositorio del estudiante con los cambios realizados en el computador personal.

Situación 2: De nuevo en el aula

Paso 1: El estudiante clona el repositorio del docente

Si el estudiante aún no ha clonado el repositorio del docente, debe hacerlo con este comando:

- git https://github.com/GBCURSOS/FSB1_152.git
- cd FBS1_152

Comentario: Esto descarga el repositorio del docente al computador del aula por primera vez.

Paso 2: El estudiante sincroniza el repositorio del docente (ya clonado, es decir, ya existe en el computador del aula)

Si el estudiante ya clonó previamente el repositorio del docente, puede actualizarlo para obtener los últimos cambios:

- cd FSB1_152
- git pull origin main

Comentario: Esto asegura que el repositorio del docente esté actualizado con los cambios más recientes.

Paso 3: El estudiante clona el repositorio del estudiante

Si el estudiante aún no ha clonado su propio repositorio, debe hacerlo con este comando:

- git clone https://github.com/RNOGALES/FBS152_RN.git
- cd repositorio-estudiante

Comentario: Esto descarga el repositorio personal del estudiante al computador del aula por primera vez.

Paso 4: El estudiante sincroniza el repositorio del estudiante (ya clonado, es decir, el repositorio del estudiante, ya está en el computador del aula)

Si el estudiante ya clonó previamente su repositorio, puede actualizarlo para obtener los últimos cambios:

- cd FSB152_RN
- git pull origin main

Comentario: Esto asegura que el repositorio del estudiante esté actualizado con los cambios más recientes.

Paso 5: Realiza modificaciones

El estudiante realiza cambios locales en los archivos. Por ejemplo:

- Editar un archivo HTML.
- Agregar un archivo con su nombre.

Comentario: Este es el momento donde el estudiante trabaja en los archivos para completar tareas o realizar actualizaciones.

Paso 6: El estudiante sincroniza el repositorio del estudiante

Una vez que el estudiante haya realizado sus cambios, debe subirlos al repositorio personal:

1. **Añadir los cambios:**
 - git add .
2. **Hacer commit:**
 - git commit -m "Descripción de los cambios realizados"
3. **Subir los cambios al repositorio del estudiante:**
 - git push origin main

Comentario: Esto actualiza el repositorio del estudiante con los cambios realizados en el computador del aula.

Resumen de comandos para la Situación 2

Aquí están todos los comandos juntos para referencia rápida:

Clonar repositorio del docente (si no está clonado)

- git clone <https://github.com/cuenta-docente/repositorio-docente.git>
- cd repositorio-docente

Sincronizar repositorio del docente (si ya está clonado)

- git pull origin main

Clonar repositorio del estudiante (si no está clonado)

- git clone <https://github.com/cuenta-estudiante/repositorio-estudiante.git>
- cd repositorio-estudiante

Sincronizar repositorio del estudiante (si ya está clonado)

- git pull origin main

Realizar cambios locales (editar archivos)

Añadir, commitear y subir cambios al repositorio del estudiante

- git add .
- git commit -m "Descripción de los cambios realizados"
- git push origin main

Situación 3: Caso extra

3.a. El estudiante actualiza del repositorio docente al repositorio del estudiante directamente

1. **Añadir el repositorio del docente como un segundo remoto:** `git remote add docente`
https://github.com/GBCURSOS/FSB1_152.git
2. **Verificar los remotos configurados:**
 - `git remote -v`
3. **Traer los cambios del repositorio del docente:**
 - `git pull docente main`
4. **Subir los cambios al repositorio del estudiante:**
 - `git push origin main`

Comentario: Esto sincroniza los cambios del repositorio del docente con el repositorio del estudiante.

Situación 4: Error en un desarrollo

Si el estudiante realizó un cambio que no funcionó (identificado con el mensaje "commit 10") y quiere **recuperar el código desde un commit anterior** (en este caso, el commit con el mensaje "cambio 09"), hay varias formas de hacerlo en Git.

Opción 1: Usar git checkout para recuperar archivos específicos

Si el estudiante solo quiere recuperar algunos archivos del commit anterior ("cambio 09"), puede usar git checkout para restaurar esos archivos sin afectar todo el historial.

Pasos:

1. **Identificar el hash (número único de identificación de la versión) del commit "cambio 09":** Esto mostrará el historial de commits. Busca el commit con el mensaje "cambio 09" y se copia su hash (un identificador único, ej.: a1b2c3d4).
 - `git log`
2. **Recuperar los archivos del commit "cambio 09":** El `--` indica que se deben recuperar todos los archivos del commit especificado.
 - `git checkout a1b2c3d4 -- .`
3. **Hacer commit de los cambios recuperados:**
 - `git add .`
 - `git commit -m "Revertir a cambio 09"`

Comentario: Esto restaura los archivos al estado del commit "cambio 09" y crea un nuevo commit con los cambios recuperados.

Recuperar un archivo específico con git checkout

Si el estudiante quiere recuperar un archivo específico desde un commit anterior (por ejemplo, el commit "cambio 09"), puede hacerlo siguiendo estos pasos:

Paso 1: Identificar el hash del commit

Primero, el estudiante debe encontrar el hash del commit desde el que quiere recuperar el archivo. Puede usar el comando:

- `git log`

Esto mostrará el historial de commits. Busca el commit con el mensaje "cambio 09" y copia su hash (ej.: a1b2c3d4).

Paso 2: Recuperar el archivo específico

Una vez que tengas el hash del commit, usa el siguiente comando para recuperar el archivo deseado: `git checkout a1b2c3d4 -- ruta/al/archivo`

Por ejemplo:

- Si quieres recuperar el archivo `index.html`:
 - `git checkout a1b2c3d4 -- index.html`
- Si el archivo está en una carpeta, especifica la ruta completa:
 - `git checkout a1b2c3d4 -- carpeta/archivo.css`

Comentario: Esto restaurará el archivo al estado exacto en el que estaba en el commit "cambio 09". Los cambios realizados después de ese commit se perderán **solo para ese archivo**.

Paso 3: Confirmar los cambios (opcional)

Si deseas registrar la recuperación del archivo en el historial de Git, puedes hacer un nuevo commit:

- `git add ruta/al/archivo`
- `git commit -m "Recuperado archivo desde commit 'cambio 09'"`

Comentario: Esto no es obligatorio, pero es útil si quieres documentar que recuperaste un archivo específico.

Ejemplo completo

Supongamos que el estudiante modificó accidentalmente el archivo `style.css` y quiere recuperarlo desde el commit "cambio 09".

1. **Ver el historial de commits:** Encuentra el hash del commit "cambio 09" (ej.: `a1b2c3d4`).
 - `git log`
2. **Recuperar el archivo específico:**
 - `git checkout a1b2c3d4 -- style.css`
3. **Verificar el estado:** Verás que `style.css` ha sido restaurado al estado del commit "cambio 09".
 - `git status`
4. **Confirmar el cambio (opcional):**
 - `git add style.css`
 - `git commit -m "Recuperado style.css desde commit 'cambio 09'"`

Resumen de comandos

Aquí están los comandos clave para recuperar un archivo específico:

Ver el historial de commits

- `git log`

Recuperar un archivo específico desde un commit

- `git checkout a1b2c3d4 -- ruta/al/archivo`

(Opcional) Confirmar el cambio en el repositorio

- `git add ruta/al/archivo`
- `git commit -m "Recuperado archivo desde commit anterior"`

Opción 2: Usar git reset para retroceder al commit "cambio 09"

Si el estudiante quiere **eliminar completamente los cambios realizados después del commit "cambio 09"**, puede usar git reset. Hay dos tipos de reset:

a) Reset suave (--soft):

Este método retrocede el puntero del branch al commit "cambio 09", pero mantiene los cambios en el área de preparación (staging area).

- `git reset --soft a1b2c3d4`

Comentario: Los cambios realizados después de "cambio 09" estarán disponibles para editar o descartar.

b) Reset duro (--hard):

Este método elimina completamente todos los cambios realizados después del commit "cambio 09". **¡Cuidado! Esto es irreversible.**

- `git reset --hard a1b2c3d4`

Comentario: Este comando restaura el repositorio al estado exacto del commit "cambio 09", eliminando cualquier cambio posterior.

Opción 3: Usar git revert para deshacer el commit "cambio 10"

Si el estudiante no quiere eliminar commits ni modificar el historial, puede usar git revert para crear un nuevo commit que deshaga los cambios introducidos por "cambio 10".

Pasos:

1. **Identificar el hash del commit "cambio 10":** Copia el hash del commit "cambio 10" (ej.: e5f6g7h8).
 - git log
2. **Revertir el commit "cambio 10":**
 - git revert e5f6g7h8
3. **Confirmar el revert:** Git creará automáticamente un nuevo commit que deshace los cambios de "cambio 10".

Comentario: Esta opción es segura porque no modifica el historial existente. Es ideal si el repositorio ya ha sido compartido con otros.

Resumen de comandos

Aquí están los comandos para cada opción:

Recuperar archivos específicos (git checkout):

- `git log`
- `git checkout a1b2c3d4 -- .`
- `git add .`
- `git commit -m "Revertir a cambio 09"`

Retroceder al commit "cambio 09" (git reset):

- Reset suave:
 - `git reset --soft a1b2c3d4`
- Reset duro:
 - `git reset --hard a1b2c3d4`

Deshacer el commit "cambio 10" (git revert):

- `git log`
- `git revert e5f6g7h8`