Sistema de Control de Versiones Git Hooks en Ubuntu

Gabriel Carbajal Carrasco

2024-09-03

1. Introducción a los Git Hooks

Los Git Hooks son scripts personalizados que se ejecutan automáticamente en respuesta a eventos en un repositorio de Git. Puedes usarlos para validar código antes de hacer commits, ejecutar pruebas automáticas antes de realizar un push, o enviar notificaciones después de ciertos eventos.

2. Ubicación de los Hooks en un Repositorio

Cada repositorio de Git tiene una carpeta llamada hooks dentro del directorio .git que contiene ejemplos de hooks predefinidos. Estos ejemplos están desactivados por defecto (los archivos terminan en .sample).

Para activar un hook, simplemente elimina la extensión .sample y edita el archivo según tus necesidades.

```
gabriel@gabriel-VirtualBox:~$ cd New_Repositorio/.git/hooks
gabriel@gabriel-VirtualBox:~/New_Repositorio/.git/hooks$ ls
applypatch-msg.sample      pre-commit.sample          pre-receive.sample
commit-msg.sample          pre-merge-commit.sample    push-to-checkout.sample
fsmonitor-watchman.sample  prepare-commit-msg.sample  sendemail-validate.sample
post-update.sample         pre-push.sample            update.sample
pre-applypatch.sample      pre-rebase.sample
```

3. Creación de un Hook Pre-Commit

Un hook pre-commit se ejecuta antes de que se complete un commit. Puede usarse para revisar el código o prevenir commits que no cumplen con ciertas reglas.

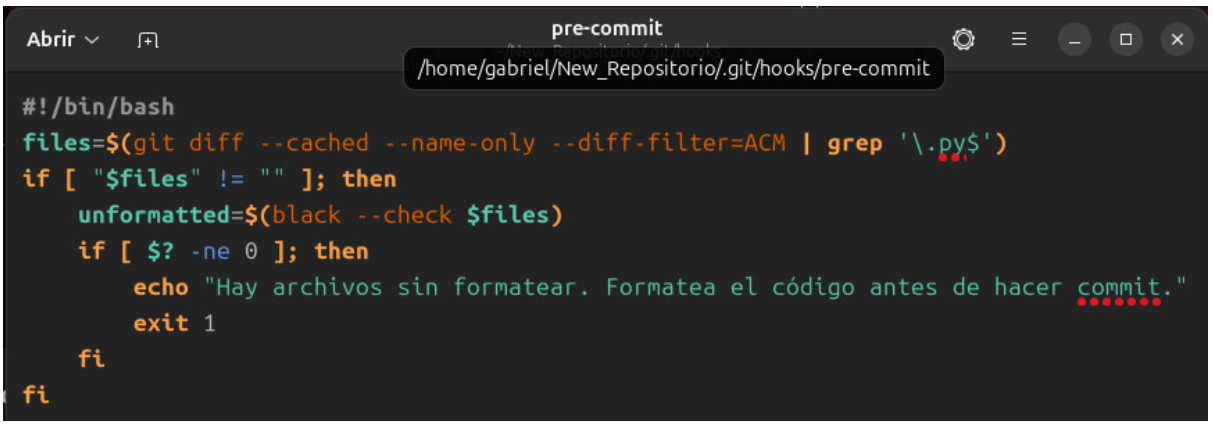
3.1 Crear el Hook

Crea un archivo llamado pre-commit en la carpeta .git/hooks/ y hazlo ejecutable.

```
gabriel@gabriel-VirtualBox:~/New_Repositorio/.git/hooks$ touch pre-commit
gabriel@gabriel-VirtualBox:~/New_Repositorio/.git/hooks$ chmod +x pre-commit
```

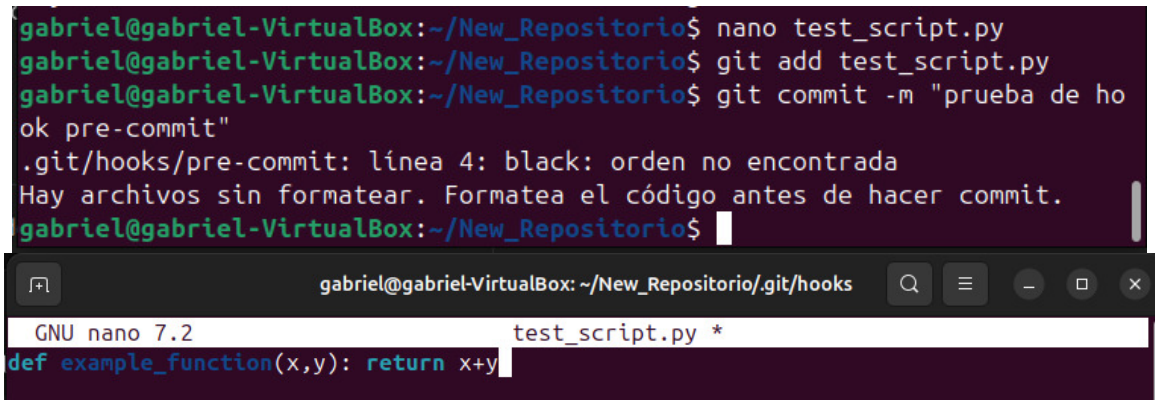
3.2 Escribir el Script del Hook

Por ejemplo, este script impide que se realicen commits si hay archivos con código Python que no están formateados según black.



3.3 Probar el Hook

Intenta hacer un commit con un archivo Python que no esté formateado correctamente. El hook evitará el commit hasta que el archivo esté formateado. Instalando Python, pip y black previamente.

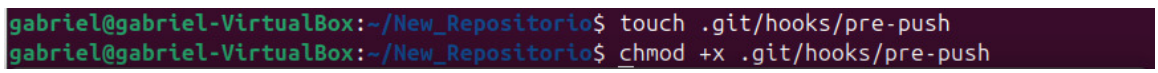


4. Creación de un Hook Pre-Push

Un hook pre-push se ejecuta antes de enviar los cambios a un repositorio remoto. Puedes usarlo para ejecutar pruebas automáticas o asegurarte de que el código pase ciertas validaciones.

4.1 Crear el Hook

Crea un archivo llamado pre-push en la carpeta .git/hooks/ y hazlo ejecutable.

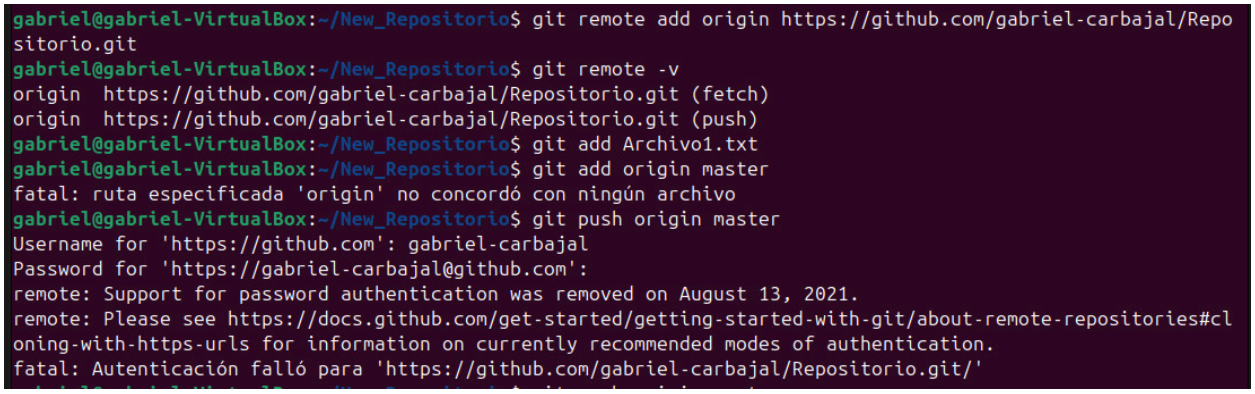


4.2 Escribir el Script del Hook

Este ejemplo de script ejecuta pruebas unitarias antes de hacer un push:

4.3 Probar el Hook

Intenta hacer un push. Si las pruebas fallan, el hook evitar´a que los cambios se env´ıen al servidor remoto.

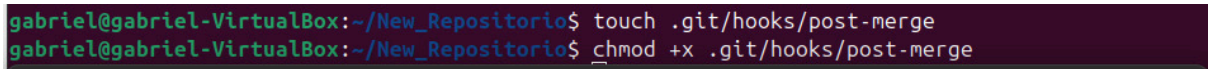


5. Creación de un Hook Post-Merge

Un hook post-merge se ejecuta después de fusionar una rama. Puedes usarlo para ejecutar scripts que configuran el entorno de desarrollo o realicen acciones posteriores a la fusión.

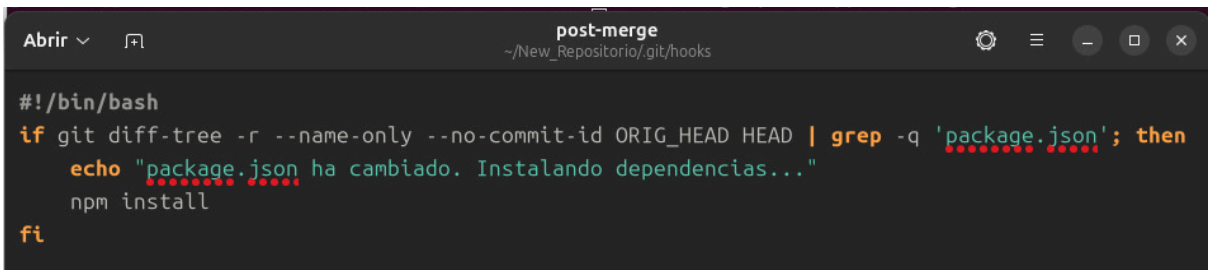
5.1 Crear el Hook

Crea un archivo llamado post-merge en la carpeta .git/hooks/ y hazlo ejecutable.



5.2 Escribir el Script del Hook

Por ejemplo, este script reinstala las dependencias del proyecto si el archivo package.json cambio durante la fusión:



5.3 Probar el Hook

Fusiona una rama que haya cambiado el archivo package.json. El hook ejecutará automáticamente npm install.