Academia Java by Xideral

Examen Tercera Semana

Asael Marcial Grajales

Tabla de contenido

GIT Advanced commands

Pull Request

Permiten comentarles a otros acerca de los cambios que has insertado en una rama de un repositorio en GitHub. Una vez que se abre un pull request, puedes debatir y revisar los posibles cambios con los colaboradores y agregar confirmaciones de seguimiento antes de que los cambios se fusionen en la rama base.

Después de inicializar una solicitud de extracción, verás una página de revisión que muestra una descripción general de alto nivel de los cambios entre tu rama (la rama de comparación) y la rama base del repositorio. Puede agregar un resumen de los cambios propuestos, revisar los cambios realizados por las confirmaciones, agregar etiquetas, hitos y usuarios asignados, y @mention a equipos o colaboradores individuales.

Otros colaboradores pueden revisar tus cambios propuestos, agregar comentarios de revisión, contribuir con el debate sobre la solicitud de extracción e incluso agregar confirmaciones a la solicitud de extracción.

Una vez que estás conforme con los cambios propuestos, se puede fusionar la solicitud de extracción.

Puedes vincular una solicitud de incorporación de cambios a una incidencia para mostrar que una corrección está en curso y para cerrar automáticamente la incidencia cuando un usuario fusione mediante combinación dicha solicitud.

Acerca de las solicitudes de incorporación de cambios - GitHub Docs

Fork

La función de fork en Git se define como una operación usual en el sistema de Git que se encarga de la creación de una copia de un repositorio en la cuenta de usuario.

Cabe destacar que este repositorio copiado será igual al repositorio desde el que se realiza el fork en Git. A pesar de esto, cuando se cree la copia, cada repositorio se ubicará en espacios diferentes y tendrán la posibilidad de evolucionar de forma diferente, de acuerdo a las acciones del usuario en cada uno de estos recursos de Git.

Dentro de las características más importantes de la opción de fork en Git podemos encontrar que, al realizar este proceso, se puede experimentar o modificar el repositorio copiado de manera libre, sin que esto implique alguna afectación al repositorio y al proyecto original.

Además, después de realizar un fork en Git, se obtendrán como resultado dos repositorios iguales, pero que tienen URLs diferentes.

Gracias a las funciones y características de un fork en Git, este suele implementarse con el objetivo de proponer modificaciones o cambios en el proyecto de un tercero que no permite la edición directa.

¿Qué es fork en Git? | KeepCoding Tech School

Rebase

El comando git rebase te permite cambiar fácilmente una serie de confirmaciones, modificando el historial de tu repositorio. Puedes reordenar, editar o combinar confirmaciones.

Normalmente, se usaría git rebase para lo siguiente:

- Editar mensajes de confirmación previos.
- Combinar varias confirmaciones en una.
- Eliminar o revertir confirmaciones que ya no son necesarias.

Para cambiar de base todas las confirmaciones entre otra rama y el estado de rama actual, puedes ingresar el siguiente comando en tu Shell:

$ git rebase --interactive OTHER-BRANCH-NAME

Para cambiar de base las últimas confirmaciones en tu rama actual, puedes ingresar el siguiente comando en tu shell:

$ git rebase --interactive HEAD~7

Hay seis comandos disponibles mientras se cambia la base:

1. **Pick:** Reordenar los comandos pick cambia el orden de las confirmaciones cuando la fusión mediante cambio de base está en curso. Si eliges no incluir una confirmación, debes eliminar la línea completa.

2. **Reword:** El comando reword es similar a pick, pero después de usarlo, la fusión mediante cambio de base se pausará y le dará la oportunidad de modificar el mensaje de confirmación.
3. **Edit:** Si elige realizar edit en una confirmación, se le dará la oportunidad de modificar la confirmación, lo que significa que puede agregar o cambiar la confirmación por completo. También puedes realizar más confirmaciones antes de continuar con el cambio de base.
4. **Squash:** Este comando te permite combinar dos o más confirmaciones en una única confirmación.
5. **Fixup:** Esto es similar a squash, pero la confirmación que se va a combinar tiene su mensaje descartado. La confirmación simplemente se fusiona en la confirmación de arriba y el mensaje de la confirmación anterior se usa para describir ambos cambios.
6. **Exec:** Permite ejecutar comandos shell de forma arbitraria con una confirmación.

Acerca de la fusión mediante cambio de base de Git - GitHub Docs

Stash

El comando git stash almacena temporalmente (o guarda en un stash) los cambios que hayas efectuado en el código en el que estás trabajando para que puedas trabajar en otra cosa y, más tarde, regresar y volver a aplicar los cambios más tarde. Guardar los cambios en stashes resulta práctico si tienes que cambiar rápidamente de contexto y ponerte con otra cosa, pero estás en medio de un cambio en el código y no lo tienes todo listo para confirmar los cambios.

Esta funcionalidad es útil cuando has hecho cambios en una rama que no estás listo para realizarle commit, pero necesitas cambiar a otra rama.

Para guardar tus cambios en el stash, ejecuta el comando:

git stash save "mensaje opcional"

Esto guarda los cambios y revierte el directorio de trabajo a como se veía en tu último commit. Los cambios guardados están disponibles en cualquier rama de ese repositorio.

Para ver lo que hay en tu stash, ejecuta el comando:

git stash list

Para ver un resumen de los cambios que hiciste en el stash:

git stash show NOMBRE-DEL-STASH.

Para recuperar los cambios del stash y aplicarlos a la rama actual en la que estás, tienes dos opciones:

git stash apply NOMBRE-DEL-STASH aplica los cambios y deja una copia en el stash

git stash pop NOMBRE-DEL-STASH aplica los cambios y elimina los archivos del stash

Puede haber conflictos cuando se aplican los cambios.

Para borrar los cambios guardados en stash sin aplicarlos, ejecuta el comando:

git stash drop NOMBRE-DEL-STASH

Para limpiar todo del stash, ejecuta el comando:

git stash clear

git stash: Cómo guardar los cambios | Atlassian Git Tutorial

Git Stash Explicado: Cómo Almacenar Temporalmente los Cambios Locales en Git (freecodecamp.org)

Clean

El comando Git Clean se usa para limpiar los archivos sin seguimiento en el repositorio. Si queremos eliminar los archivos no deseados, podemos usar el comando de limpieza en Git. Cuando los desarrolladores quieren eliminar los archivos sin seguimiento en el repositorio de trabajo, este comando les resulta muy útil.

El uso del comando Git Clean es el siguiente:

- git clean -n: para ejecutar en seco.
- git clean -f: eliminación forzada de archivos.
- git clean -f -x: eliminar archivos gitignore.
- git clean -f -d: elimina los directorios sin seguimiento.

Por defecto, no eliminará:

- Los archivos gitignore.
- Nuevos directorios creados recientemente.
- Archivos de índice.
- Archivos de commit existentes.

Git - Clean - GeeksforGeeks

Cherry-pick

Git chery-pick se encarga de elegir uno o más commit o confirmaciones de cambios de una rama específica para luego aplicarla a otra rama. Esto quiere decir que los commit de la plataforma de Git pueden elegirse por referencia, para que añadirlos al HEAD actual de trabajo, que no es más que el commit en el que se encuentra posicionado el repositorio del usuario en cada momento.

Dentro de sus propiedades y características, es de fácil ejecución, ya que para utilizarlo, solo se necesita conocer el commit determinado que se desea aplicar en la rama.

El también se caracteriza por su utilidad para la rápida corrección de errores en el sistema, evitando que impacten sobre más usuarios. Al mismo tiempo, esta herramienta puede utilizarse para el trabajo independiente en un mismo proyecto de desarrollo, gracias a que permite ir avanzando en los procesos de manera sencilla y práctica.

Git cherry-pick también ofrece múltiples opciones que extienden sus funciones, como, por ejemplo:

- **-e o --edit:** Solicita un mensaje de commit o confirmación antes de llevar a cabo la ejecución del comando.
- **--no-commit:** En vez de realizar un commit nuevo, mueve el contenido al directorio de trabajo de la rama actual.

También se caracteriza por su gran potencia de ejecución, por lo que su uso no es siempre recomendable, y debe ser cuidadoso para evitar inconvenientes como commits duplicados.

¿Qué es Git cherry-pick? | KeepCoding Tech School