

Git & GitHub

Advanced Commands (Pull Request, Fork, Rebase)

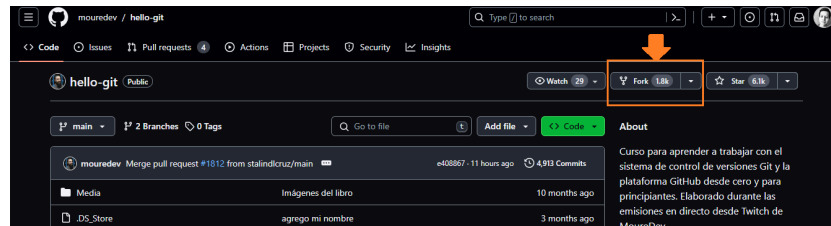
PULL REQUEST (SOLICITUD DE INCORPORACIÓN DE CAMBIOS)

Un Pull Request es una función de GitHub que permite al equipo de trabajo solicitar la revisión y aprobación de sus cambios antes de fusionarlos en la rama principal de desarrollo, denominada "master" o "main".

Estos cambios se proponen en una rama, lo cual garantiza que la rama predeterminada contenga únicamente trabajo finalizado y aprobado.

Al crear un Pull Request se genera una conversación en la que los demás miembros pueden seguir y comentar los cambios propuestos en el código del repositorio. Esto permite detectar cualquier error o problema potencial antes de integrarlos en la rama principal, lo que ayuda a mantener el proyecto más limpio y estable.

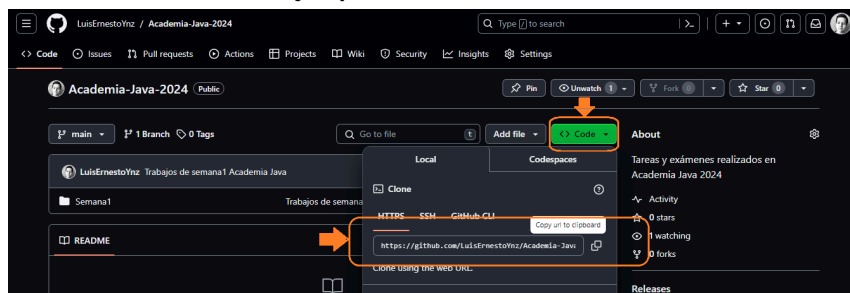
1. Realizamos un fork del repositorio de GitHub haciendo un clic en el botón **fork** de la parte superior de la página. Esto creará una instancia del repositorio completo en tu cuenta.



2. Clona el repositorio

Una vez que el repositorio esté en tu cuenta, clónalo en tu computadora para trabajar localmente.

Para clonarlo, hacer clic en el botón **"Code"** y **copia el link**.



3. Abre la terminal de Git en tu PC y ejecuta el siguiente comando. Esto clonará el repositorio localmente.

\$ git clone [DIRECCIÓN HTTPS]

4. Ahora hemos configurado una copia de la rama maestra desde el repositorio principal del proyecto en línea.

5. En la terminal de Git en tu PC, vamos al repositorio clonado ejecutando el siguiente comando:

\$ cd [NOMBRE DEL REPOSITORIO]

6. Crea una rama. Es una buena práctica crear una rama (branch) nueva cuando trabajas con repositorios. El nombre de la rama debe ser breve y debe reflejar el trabajo que estamos haciendo.

Ahora crea una rama usando el comando git checkout:

\$ git checkout -b [Nombre de la Rama]

7. Realizar cambios esenciales al proyecto. Agregar esos cambios a la rama recién creada (Área de preparación), usando el comando git add:

\$ git add .

8. Guardar los cambios (confirmarlos) utilizando el comando git commit:

\$ git commit -m "Comentario referente a la confirmación"

9. Envíar los cambios a GitHub. Para enviar los cambios a GitHub, debemos identificar el nombre del repositorio remoto.

\$ git remote

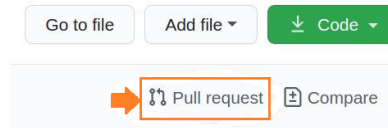
Para este repositorio el nombre es "origin".

Luego de identificar el nombre podemos enviar en forma segura los cambios a GitHub.

git push origin [Nombre de la Rama]

10. Crea un pull request

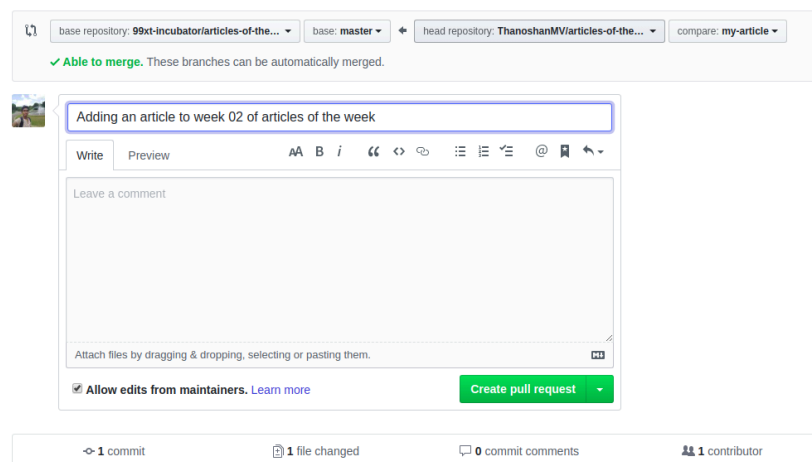
Dirigete a tu repositorio en GitHub y verás un botón llamado **"Pull request"**, haz clic en él.



Proporcione todos los detalles necesarios de lo que ha realizado (se pueden referenciar problemas usando "#"). Una vez hecho esto, se envía el pull request.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



Si el pull request enviado es aceptado se recibirá un mail.

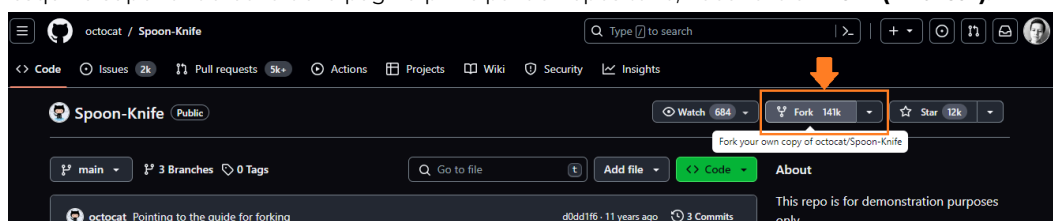
FORK (BIFURCAR)

Una bifurcación es un nuevo repositorio que comparte la configuración de visibilidad y código con el repositorio original.

La bifurcación de un repositorio es similar a la duplicación de un repositorio, con las siguientes diferencias.

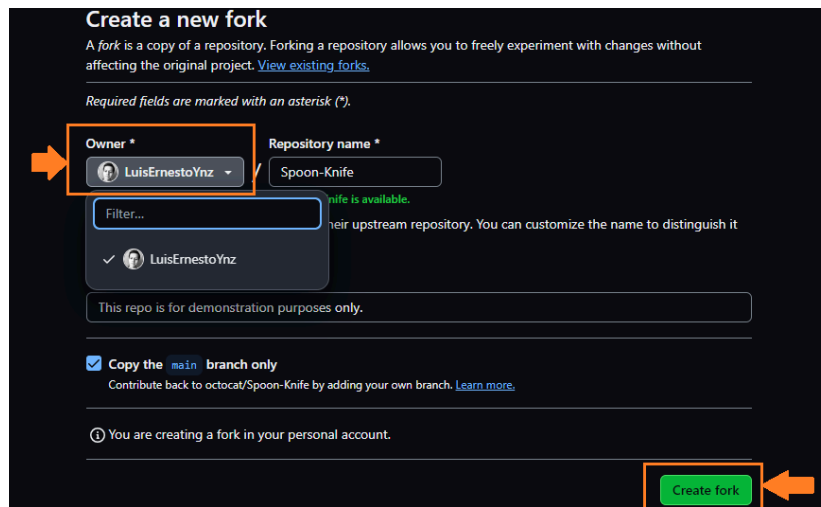
- Puedes usar una solicitud de incorporación de cambios para sugerir cambios de la bifurcación al repositorio ascendente.
- Puedes llevar cambios desde tu repositorio ascendente a tu bifurcación local sincronizando tu bifurcación con el repositorio ascendente.
- Las bifurcaciones tienen sus propios miembros, ramas, etiquetas, directivas, problemas, solicitudes de incorporación de cambios, discusiones, acciones, proyectos y wikis.
- Las bifurcaciones heredan las restricciones de sus repositorios ascendentes. Por ejemplo, las reglas de protección de ramas no se pueden pasar si el repositorio ascendente pertenece a una organización en un plan GitHub Free.

1. En la esquina superior derecha de la página principal del repositorio, hacer clic en **Fork (Bifurcar)**.



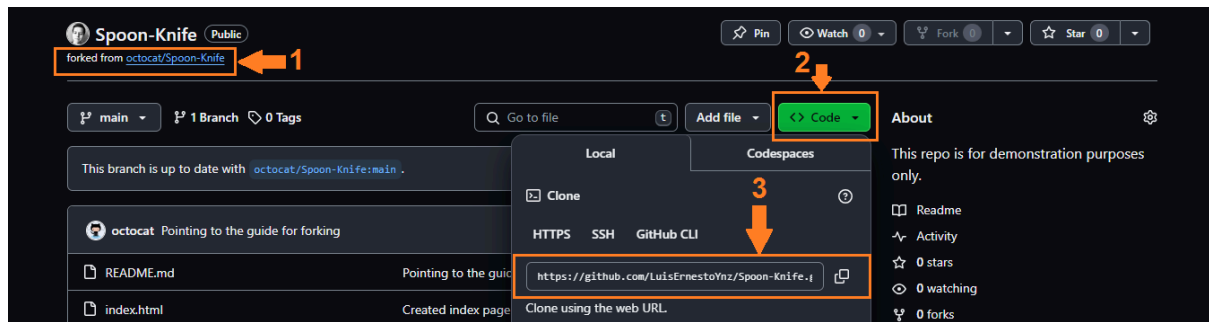
2. En "Owner" (Propietario), selecciona el menú desplegable y haz clic en un propietario del repositorio bifurcado.

3. De forma predeterminada, las bifurcaciones tienen el mismo nombre que sus repositorios ascendentes.
4. Para distinguir aún más la bifurcación, en el campo "Nombre del repositorio", escribe un nombre. *Opcional*
5. En el campo de "Descripción", escribe una descripción de la bifurcación. *Opcional*
6. Selecciona Copiar solo la rama DEFAULT. *Opcional*
7. Haz clic en **Crear bifurcación**.



Clonar el repositorio bifurcado

1. Para añadir los archivos de un repositorio bifurcado localmente en tu equipo. En *GitHub.com*, ve a tu bifurcación (se reconoce el repositorio bifurcado por la leyenda que tiene debajo del nombre del proyecto):
2. Hacer clic en **<> Code**.
3. Copia la dirección URL del repositorio.



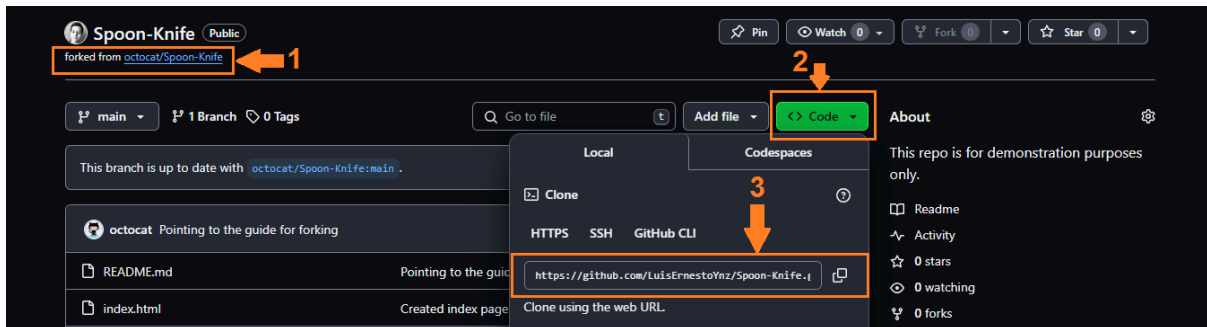
4. Abrir Git Bash.
5. Cambia el directorio de trabajo actual a la ubicación en donde quieres clonar el directorio.
6. Escriba el comando git clone seguido de la dirección URL que ha copiado antes. Tendrá este aspecto, con el nombre de usuario de GitHub en lugar de YOUR-USERNAME. Se creará el clon local.

```
$ git clone https://github.com/YOUR-USERNAME/Spoon-Knife
```

Configurar a Git para sincronizar tu bifurcación con el repositorio ascendente

Cuando bifurcas un proyecto para proponer cambios en el repositorio ascendente, puedes configurar Git para extraer cambios del repositorio ascendente en el clon local de tu bifurcación.

1. En *GitHub.com*, ve a tu bifurcación (se reconoce el repositorio bifurcado por la leyenda que tiene debajo del nombre del proyecto):
2. Hacer clic en **<> Code**.
3. Copia la dirección URL del repositorio.



4. Abrir *Git Bash*.

5. Cambiar el directorio de la ubicación de la bifurcación que se clonó.

- Ir al directorio principal, escribir solo **cd** sin ningún otro texto.
- Generar una lista de los archivos y carpetas en su directorio actual, escribir **ls**.
- Ir a uno de los directorios mostrados, escribir **cd your_listed_directory**.
- Subir un directorio, escribir **cd ..**

6. Escribir `git remote -v` y presionar Entrar. Se mostrará el repositorio remoto configurado actualmente para la bifurcación.

```
$ git remote -v
```

7. Escribir `git remote add upstream` y pegar la dirección URL que se copió en el paso 3 y presionar Enter. Tendrá este aspecto:

```
$ git remote add upstream https://github.com/ORIGINAL_OWNER/Spoon-Knife.git
```

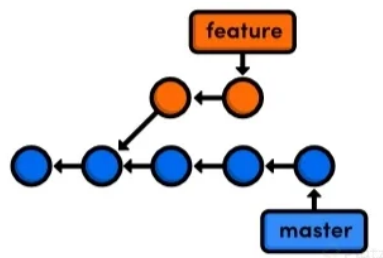
8. A fin de verificar el nuevo repositorio ascendente que has especificado para tu bifurcación, escribe `git remote -v` otra vez. Debería ver la dirección URL de la bifurcación como origen y la dirección URL del repositorio ascendente como upstream.

```
$ git remote -v
```

```
> origin https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)
> origin https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
> upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (fetch)
> upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (push)
```

REBASE

Rebase es el proceso de mover o combinar una secuencia de confirmaciones en una nueva confirmación base. El proceso general se puede visualizar de la siguiente manera.

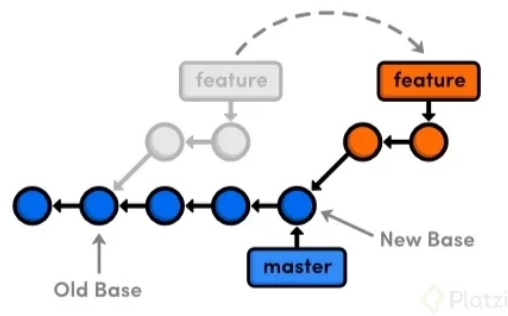


Créditos de la imagen a: Carlos Eduardo Díaz, Platzi

Para hacer un rebase en la rama `feature` de la rama `master`, se hace uso de los siguientes comandos:

```
$ git checkout feature
$ git rebase master
```

Esto trasplanta la rama `feature` desde su locación actual hacia la punta de la rama `master`:



Créditos de la imagen a: Carlos Eduardo Díaz, Platzi

Para fusionar la rama feature con la rama master

```
$ git checkout master  
$ git rebase feature
```

Nunca se debe reorganizar las confirmaciones una vez que se hayan enviado a un repositorio público. La reorganización sustituirá las confirmaciones antiguas por las nuevas y parecería que esa parte del historial de tu proyecto se hubiera desvanecido de repente.

Con rebase puedes recoger todos los cambios confirmados en una rama y ponerlos sobre otra. Cambiamos a la rama que queremos traer los cambios con el comando:

```
$ git checkout experiment
```

Aplicamos rebase para traer los cambios de la rama que queremos

```
$ git rebase master
```

REFERENCIAS:

PULL REQUEST

<https://platzi.com/clases/1557-git-github/19957-utilizando-pull-requests-en-github/#:~:text=918-Pull%20request%3A,a%20una%20rama%2C%20normalmente%20master.>

<https://www.freecodecamp.org/espanol/news/como-hacer-tu-primer-pull-request-en-github/>

FORK

<https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo>

REBASE Y MERGE (ESTE ÚLTIMO COMO DATO PARA FUTURAS CONSULTAS)

<https://www.freecodecamp.org/espanol/news/la-guia-definitiva-para-git-merge-y-git-rebase/>

<https://platzi.com/clases/1557-git-github/19975-git-rebase-reorganizando-el-trabajo-realizado/#:~:text=Rebase%20es%20el%20proceso%20de,visualizar%20de%20la%20siguiente%20manera.>

GIT

<https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>