

Pasos para crear un proyecto local con repositorio Git y asociarle un repositorio remoto en Github:

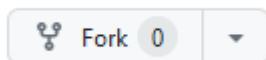
1. Crear un proyecto nuevo en VSC
2. Abrir una terminal (cmd) en VSC en la carpeta de ese proyecto
3. Iniciar repositorio local en esa carpeta. Se creará automáticamente una carpeta oculta .git donde guardará la información del repositorio local → `git init`
4. Crear un repositorio remoto en Github
5. Asociarle el repositorio remoto al repositorio local → `git remote add origin urlRepositorioRemoto`
6. Trabajar en local y cuando se quieran subir los cambios al repositorio remoto de GitHub:
 - a. `git add .`
 - b. `git commit -m "etiqueta de cambios realizados"`
 - c. `git push`

Para agregar “colaboradores” en un proyecto de github:

1. Desde el usuario inicial que ha subido o ha creado el proyecto en github, desde settings/collaborator se van agregando los otros usuarios que colaborarán en el proyecto
2. Estos colaboradores deberán aceptar la invitación de colaboración

(optativo) Para que cada usuario colaborador tenga una copia del repositorio original de github en su propia cuenta de github:

1. Desde su cuenta de github hay que realizar un fork



Para que l@ colaborador@s puedan trabajar en local con un repositorio que está inicialmente en GitHub

1. **Clonar el repositorio para copiarlo en local:** Crea una carpeta en tu disco y la pones en VSC, abres una terminal y te metes en la carpeta y desde ella clonas el repositorio remoto de github en el que colaboras

```
git clone <URL_del_repositorio_GitHub_original>
```

2. **Cámbiate a la carpeta del repositorio clonado:** Utiliza el comando `cd` en tu terminal para cambiar al directorio del repositorio clonado.

```
cd nombre_del_repositorio
```

Para que cada colaborador@ pueda realizar sus aportaciones al proyecto de forma independiente al resto de colaborador@s

1. **Crear una nueva rama en local:** Utiliza el comando `git branch` seguido del nombre de la nueva rama que deseas crear. Por ejemplo:

```
git branch nueva_rama
```

2. **Cambiar a la nueva rama:** Para empezar a trabajar en la nueva rama que has creado, necesitas cambiar a ella. Puedes hacerlo usando el comando `git checkout` seguido del nombre de la rama. Por ejemplo:

```
git checkout nueva_rama
```

O, si estás utilizando Git 2.23 o superior, puedes combinar estos dos pasos en uno solo usando el comando `git switch`:

```
git switch -c nueva_rama
```

3. **Hacer tus cambios locales:** Realiza los cambios necesarios en tus archivos como lo harías normalmente.
4. **Agregar y confirmar tus cambios:** Utiliza los comandos `git add` y `git commit` para agregar y confirmar tus cambios en la nueva rama. Por ejemplo:

```
git add .
```

```
git commit -m "Mensaje de confirmación"
```


5. **Subir la nueva rama a GitHub:** Utiliza el comando `git push` seguido de los nombres de las ramas local y remota. Por ejemplo:

```
git push -u origin nueva_rama
```


6. Esto creará la nueva rama en el repositorio remoto en GitHub y subirá tus cambios a esa rama.

FIN DEL PROYECTO: Desde GitHub se unirá cada aportación de cada colaborador@ (su rama) con el proyecto ppal (rama main)

- Desde Github, el usuario propietario del repositorio donde se creó el proyecto puede ver su rama y la de los otros colaboradores
- Si quiere que los cambios realizados en su rama se unan a la rama ppal o main, debe realizar un pull-request:
 - a. En el caso de que varios colaborado@s hayan hecho cambios en algún fichero común, se mostrarían dichas aportaciones de modo que se confirme cómo ha de quedar el fichero que ha sido modificado por varios colaboradores
 - b. En el caso de no haber aportaciones en ficheros comunes, se puede confirmar el pull-request, y ya tendremos nuestros cambios en la rama ppal del proyecto.

 usuario2 had recent pushes 6 minutes ago

[Compare & pull request](#)

 rama1 had recent pushes 5 minutes ago

[Compare & pull request](#)



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view