

Manual del Líder Técnico

para trabajo con versiones en equipos de proyectos finales

Como veníamos adelantando en el Instructivo Git general, el Líder Técnico (de ahora en más LT) tendrá además de su rol como desarrollador, tareas específicas a realizar antes de comenzar el trabajo con versiones, así como también responsabilidades extra de verificación de la calidad del código.

Como desarrolladores deberán...

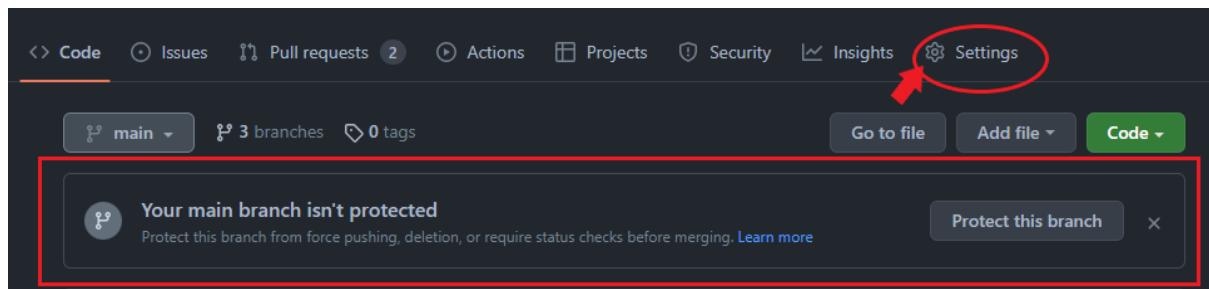
- Cumplir con los estándares de calidad definido por el grupo.
- Realizar los procedimientos diarios como el resto de sus compañeros y compañeras.

Como LT deberán:

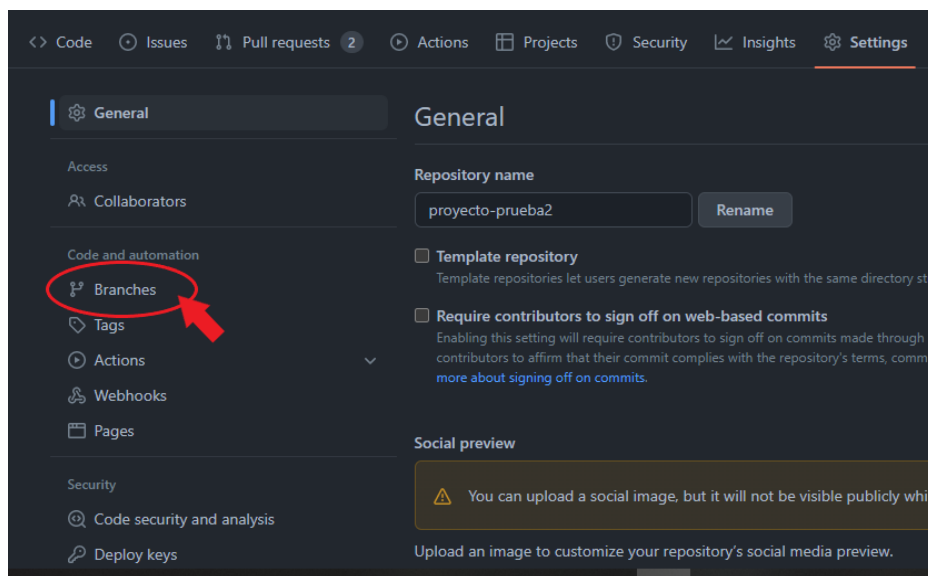
- Asegurarse de que las tareas de sus compañeros estén completas antes de aceptar un pull request.
- Asegurarse que cada versión que se añada a las ramas principales estén funcionando correctamente y sin errores.
- **Pasar a main la versión final que mostrarán en la demo**, antes de la ceremonia. El link es un entregable más.
- Asegurarse de que las entregas cumplan con el formato predefinido.
- Aceptar o rechazar pull requests.
- Solicitar cambios en los códigos de los compañeros cuando sean necesarios.
- Hacer cambios necesarios a los códigos de los compañeros antes de hacer un merge en la rama principal siempre que:
 - Se le haya dado oportunidad al compañero o compañera, o estén en tiempo límite antes de una presentación demo y esta tarea esté dentro de los requerimientos.
 - Mientras la corrección tome poco tiempo o se disponga de ese tiempo por haber concluido las propias. En caso de que no, también puede asignar la tarea de modificación a otro compañero o solicitar que asista al dueño del código con los cambios necesarios.

Configuraciones iniciales

Luego de hacer el primer commit podemos hacer la configuración de protección de ramas. En la imagen siguiente vemos el mensaje que nos indica que nuestra rama principal *main* no está protegida. Vamos a hacer una configuración básica para impedir *merge* directos. Para eso primero nos dirigiremos a *Settings*:



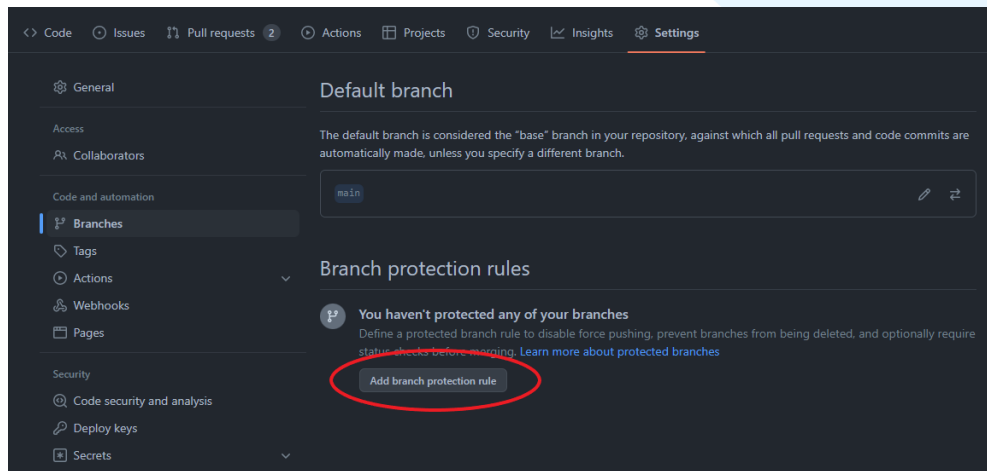
Luego seleccionaremos *Branches*.



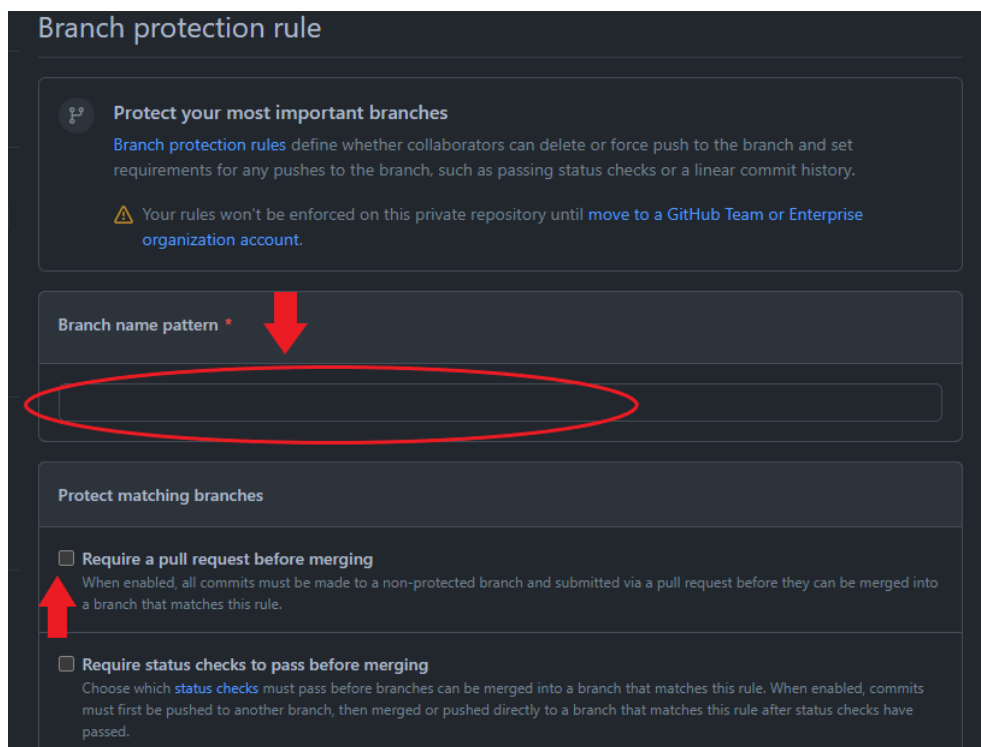
Dentro del menú agregaremos una regla de protección seleccionando el botón *Add branch protection rule*.



Proyecto Final Integrador



En el campo de abajo pondremos el nombre de la rama a proteger y luego tildaremos *Require a pull request before merging*.



Se nos desplegará un menú. Deberemos seleccionar las dos opciones de la imagen.



Proyecto Final Integrador

Protect matching branches

- ☒ **Require a pull request before merging**
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.
- ☒ **Require approvals**
When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.
Required number of approvals before merging: 1 ▼
- ☐ **Dismiss stale pull request approvals when new commits are pushed**
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.
- ☒ **Require review from Code Owners**
Require an approved review in pull requests including files with a designated code owner.
- ☐ **Require approval of the most recent push**
The most recent push must be approved by someone other than the last pusher.
- ☐ **Require status checks to pass before merging**
Choose which **status checks** must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

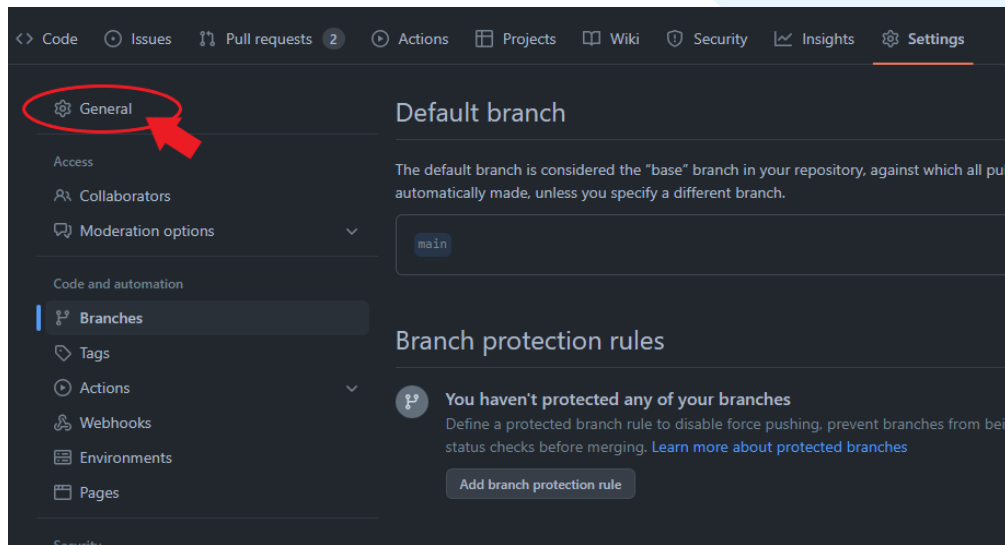
Con la primera estaremos indicando el número de aprobaciones que deberá tener el código antes de poder hacer un *merge*. Con la segunda, que requerirá de la aprobación del dueño del código, que debería ser el LT.

Cambio de propiedad del repositorio

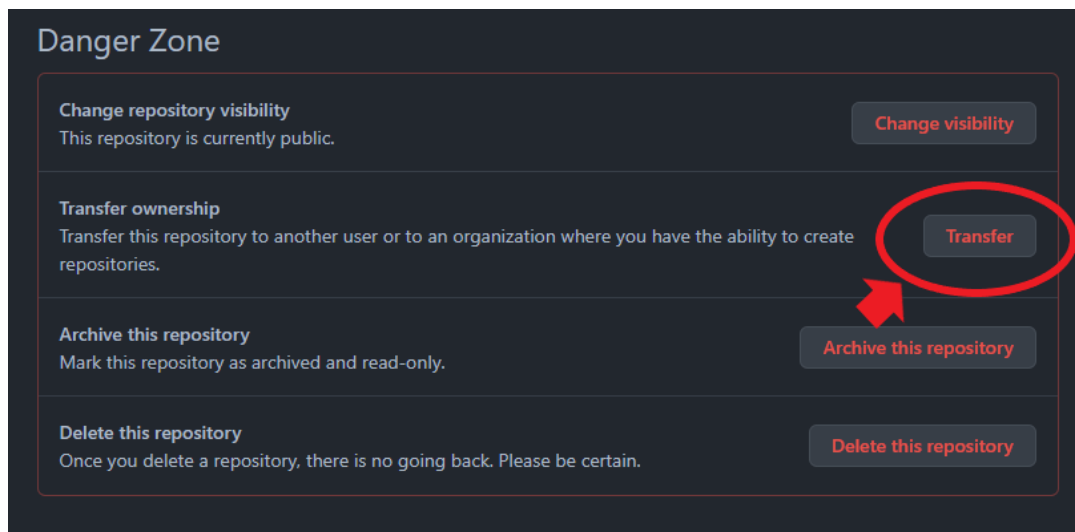
En caso de que no hayan seguido las indicaciones de roles y el LT no sea el dueño del repositorio podemos transferir la propiedad ingresando a *Settings > General* y desplazándonos hasta el fondo de la página donde encontraremos opciones en rojo.



Proyecto Final Integrador



Deberemos seleccionar el botón *Transfer*.



Se nos desplegará una ventana. En la primera opción escribiremos el nombre de usuario del LT en GitHub. En la segunda opción deberemos copiar el URI del proyecto (que tendrá el formato *nombre-usuario/nombre-proyecto*) para indicarle a GitHub que somos humanos y que estamos conscientes del cambio que estamos por realizar. Al presionar el botón en rojo, el LT será el dueño del código y quien podrá aceptar o rechazar solicitudes de *merge* a ramas principales.





Proyecto Final Integrador

Limit how many branches and tags can be updated in a single push (Beta)
Pushes will be rejected if they attempt to update more than this. [Learn more](#) about this setting, and see [this help page](#).

Transfer repository

⚠ To understand admin access, teams, issue assignments, and redirects after a repository is transferred, see [Transferring a repository](#) in GitHub Help.

Transferring may be delayed until the new owner approves the transfer.

New owner's GitHub username or organization name

Type `intisolar/proyecto-prueba2` to confirm.

[I understand, transfer this repository.](#)

Security Status Docs Contact GitHub Pricing API Training