

Guía Básica Git & Github

Instalación, inicio proyectos, creación y aplicación de cambios

Creada por: Mariel Sofía Gutiérrez Zapién
MARIELSGTZZ@GMAIL.COM

CONTENIDO DE LA GUÍA

Índice con hipervínculos

[1.Instalar git](#)

[2.Crear cuenta de GitHub](#)

[3.Crear un repositorio](#)

[4.Clonar un repositorio](#)

[5.Inicializar un proyecto dentro del repositorio](#)

a.`git add .`

b.`git commit -m “”`

c.`git push`

i. [Credenciales para autentificación](#)

[6.Cambios a repositorio](#)

a.[Crear una rama](#)

b.[Realizar cambios](#)

[7.Pull Requests](#)

a.[Crear](#)

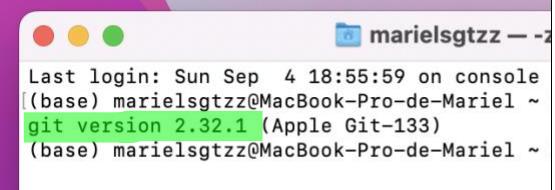
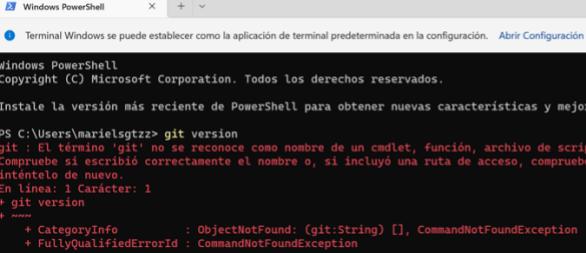
b.[Revisar](#)

c.[Aprobar](#)

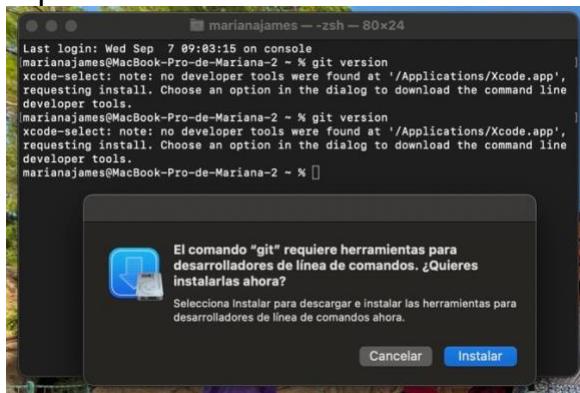
I. Instalar git

a. Abrir terminal

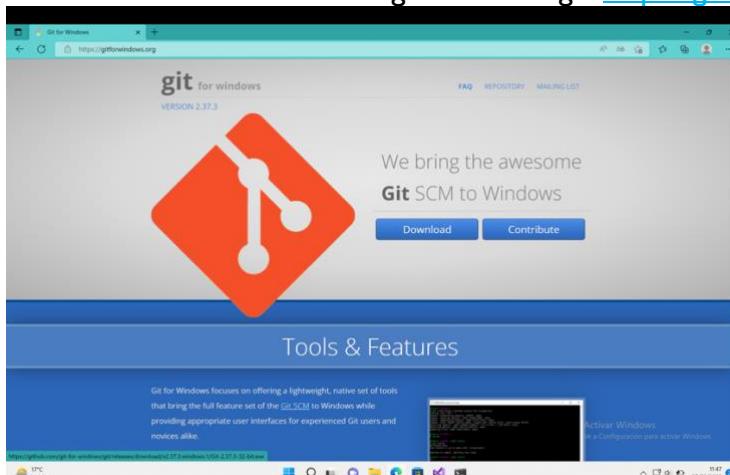
b. Para checar si git está instalado: `git version`

Si sí, va a decir qué versión se tiene	Si no, no va a reconocer el comando
 <pre>Last login: Sun Sep 4 18:55:59 on console (base) marielsgtzz@MacBook-Pro-de-Mariel ~ git version 2.33.1 (Apple Git-133) (base) marielsgtzz@MacBook-Pro-de-Mariel ~</pre>	 <pre>Windows PowerShell Copyright (C) Microsoft Corporation. Todos los derechos reservados. Instale la versión más reciente de PowerShell para obtener nuevas características y mejor PS C:\Users\marielsgtzz> git version git : El término 'git' no se reconoce como nombre de un cmdlet, función, archivo de scrip Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe inténtelo de nuevo. En línea: 1 Carácter: 1 + git version + ~~~ + CategoryInfo : ObjectNotFound: (git:String) [], CommandNotFoundException + FullyQualifiedErrorId : CommandNotFoundException</pre>

c. Para instalar en mac con darle click al botón de instalar cuando aparezca en el paso anterior



Para instalar en Windows dirigirse a esta liga: <https://gitforwindows.org/>



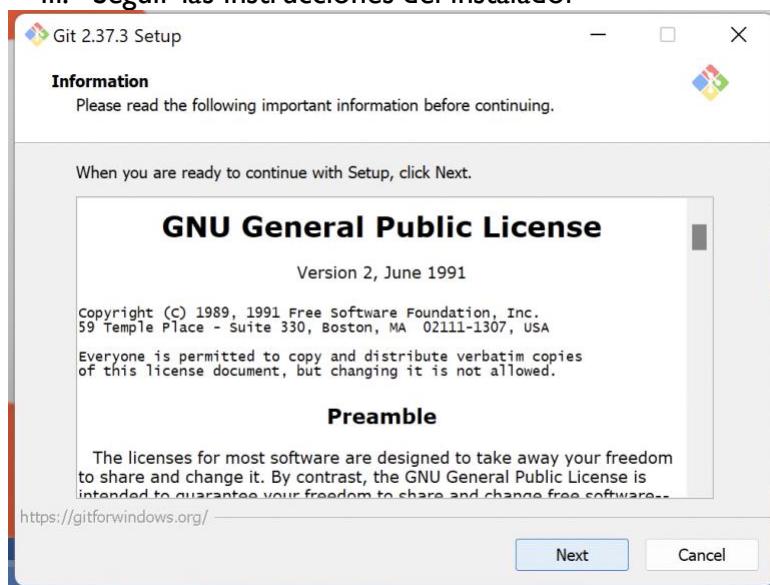
- i. Darle click a descargar (se descargará un archivo .exe)
- ii. Hay que ejecutar el archivo ejecutable (.dmg / .exe) y permitirle que realice cambios

Descargas



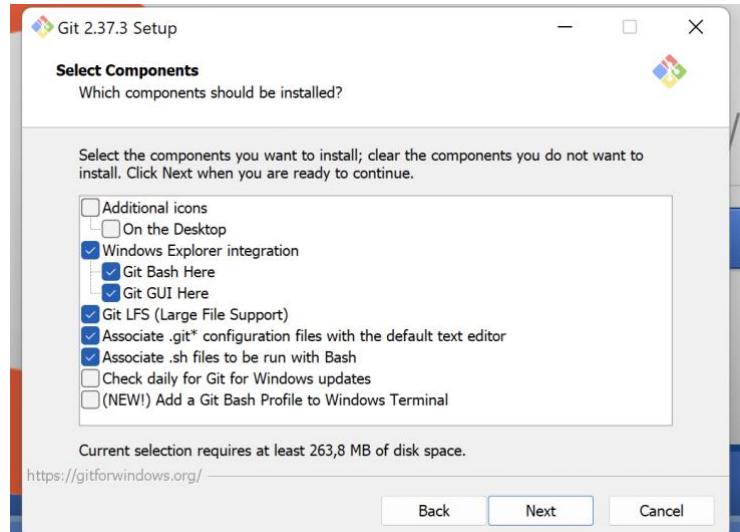
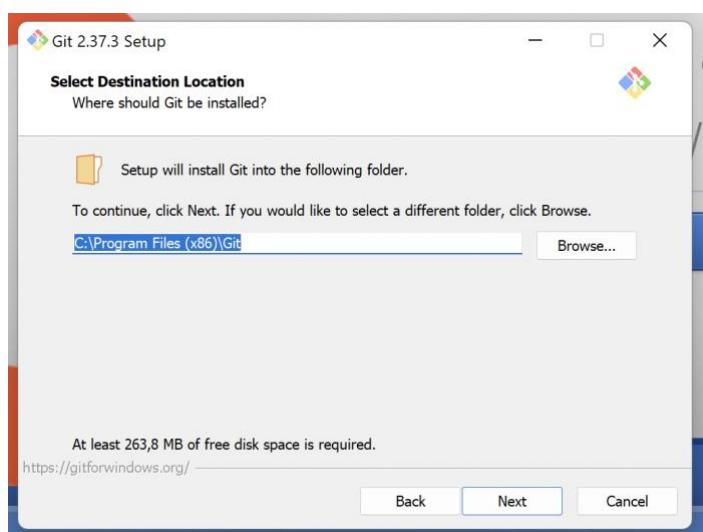
[Ver más](#)

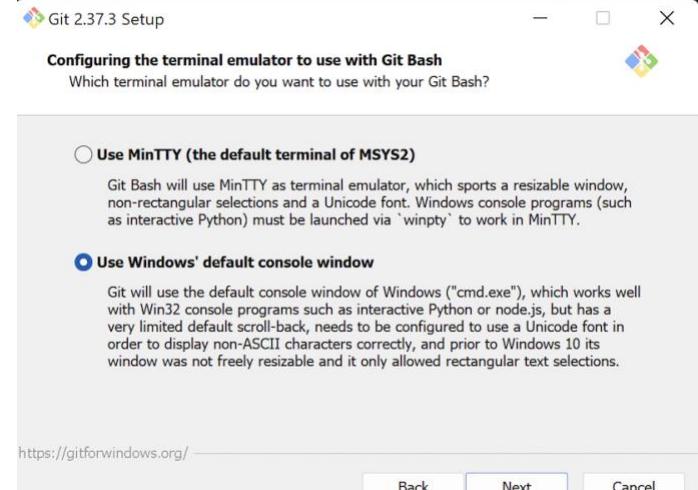
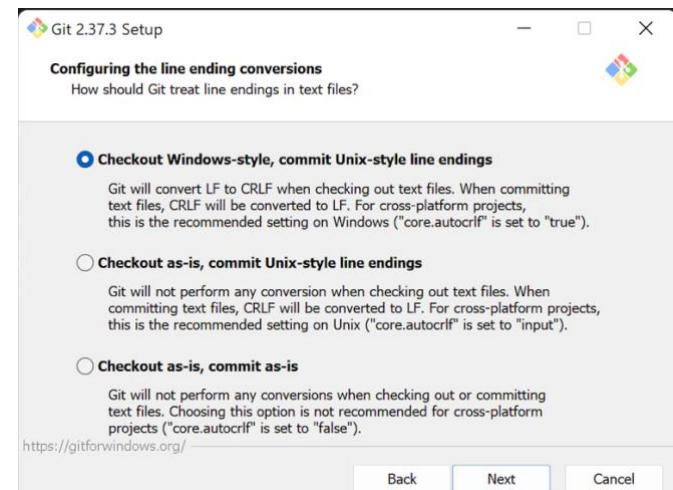
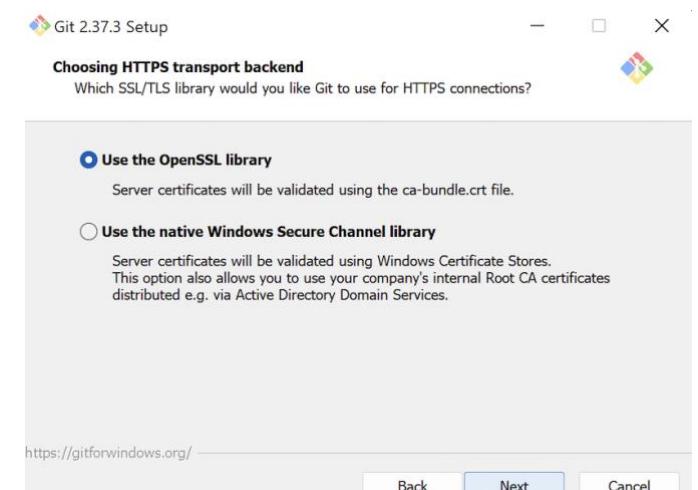
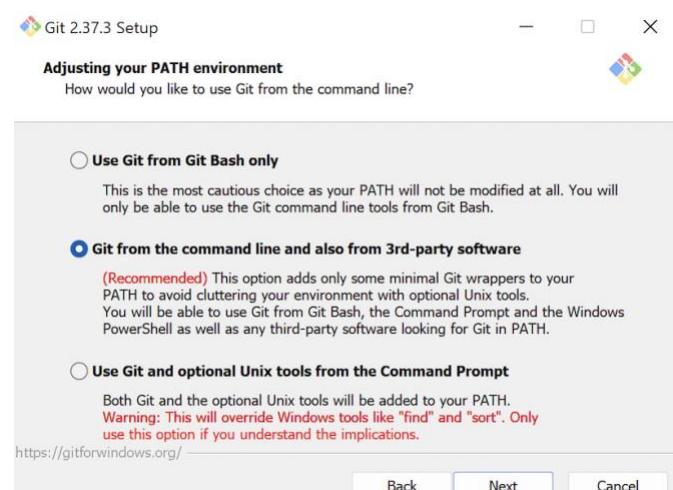
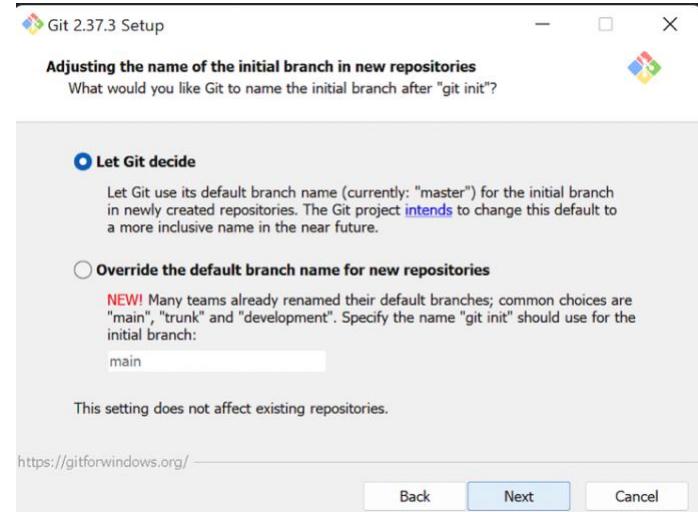
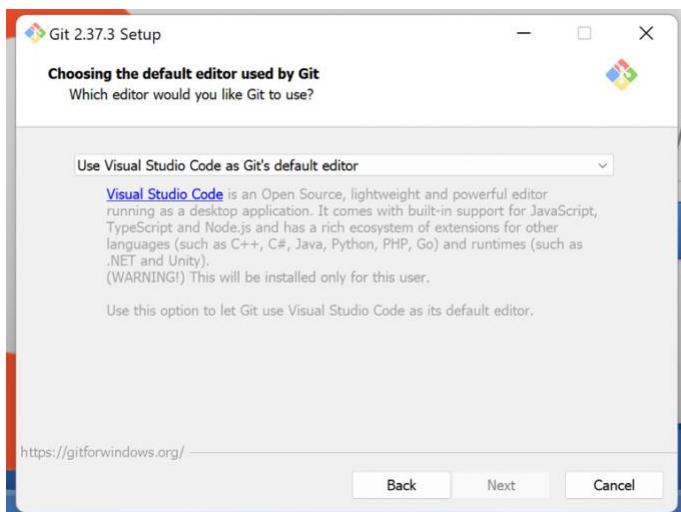
iii. Seguir las instrucciones del instalador

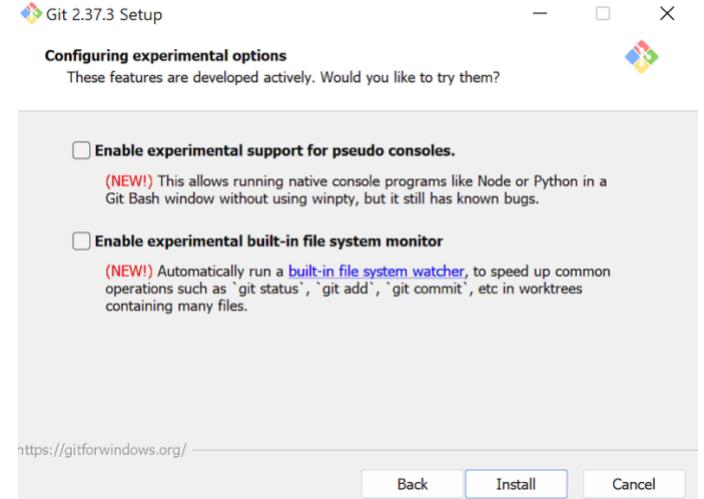
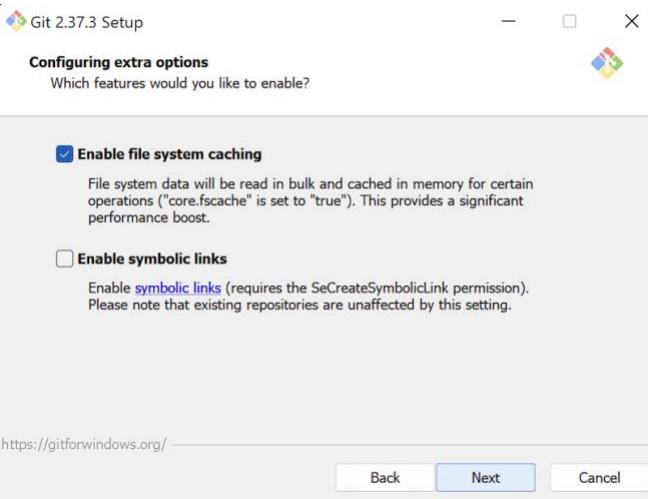
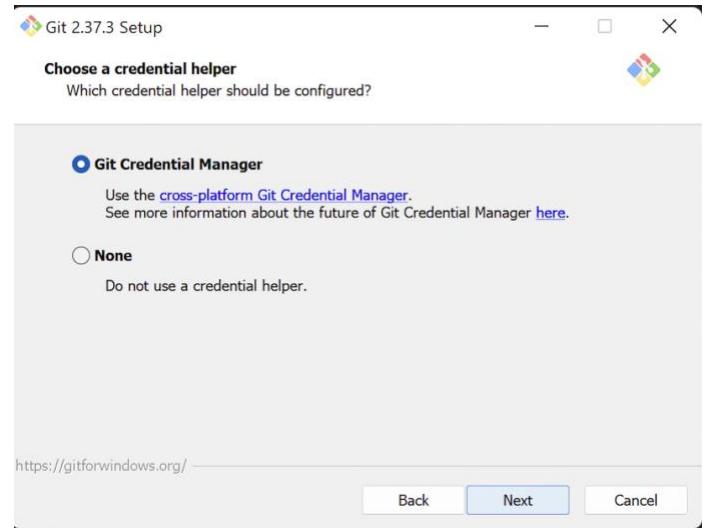
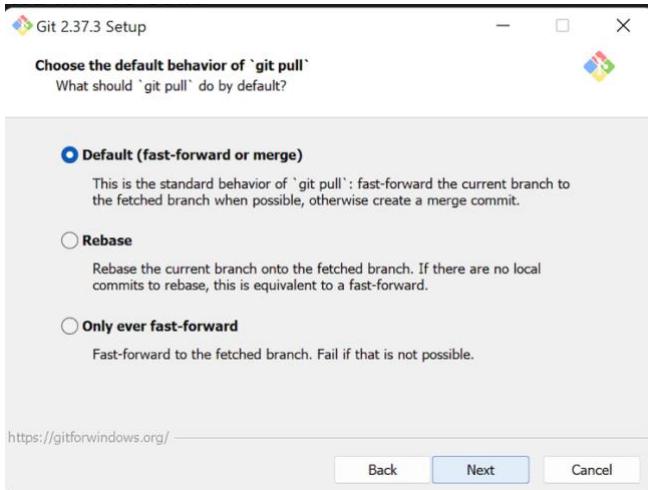


iv. Configuración recomendada

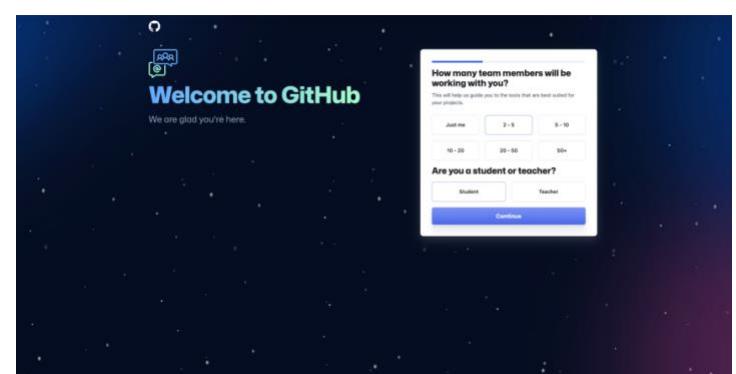
Decir dónde queremos que se instale (recomendable que sea en C:)







2. Crearse una cuenta de github <https://github.com/> (recomendable que sea con el correo institucional para que puedan tener el github para estudiantes)



The tools you need to build what you want.

Soup to nuts, GitHub has it all.

What specific features are you interested in using?

Select all that apply so we can point you to the right GitHub plan.

- Collaborative coding
Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.
- Automation and CI/CD
Actions, Packages, API, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.
- Security
Private repos, SFA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependable alerts, and more.
- Enterprise security
SAML, LDAP, IP allow list, GitHub Connect, and Audit log API.
- Client Apps
GitHub Mobile, GitHub CLI, and GitHub Desktop.
- Project Management
Projects, Labels, Milestones, Issues, Unified Contribution Graph, Org activity graph, Org dependency insights, Repo insights, Wiki, and GitHub Insights.
- Team Administration
Organizations, Invitations, Team sync, Custom roles, Domain verification, Audit Log API, Reps creation restriction, and Notification restriction.
- Community
GitHub Marketplace, GitHub Sponsors, GitHub Skills, and Electron.

Continue

Learn to ship software like a pro.

GitHub gives students free access to the best developer tools so they can learn by doing.

Free	Get additional student benefits
<ul style="list-style-type: none"> Unlimited public/private repositories 2,000 CI/CD minutes/month 500MB of Packages storage Community support 	GitHub Pro <ul style="list-style-type: none"> Protect your branches Draft pull requests Pages and Wikis 3,000 CI/CD minutes/month 2GB of Packages storage Web-based support
GitHub Student Developer Pack <ul style="list-style-type: none"> Free access to the industry's best developer tools Hundreds of offers, including DigitalOcean, Microsoft Azure, Heroku, MongoDB, Datadog, Twilio, and Stripe. 	
GitHub Campus Expert training <ul style="list-style-type: none"> Enrich your college technical community Learn the skills to build diverse tech communities on campus with training, mentorship, and support from GitHub. 	
Continue for free	
Apply for your GitHub student benefits	

Use your computer's camera to take a picture now.

Take a picture

Already have an image? We accept .jpg or .png files under 10MB.

Upload an image





Uploading...
Captura de Pantalla 2022-09-07 a la(s)
12.22.37.png
66%

What e-mail address do you use for school? *

Note: Selecting a school-issued email address gives you the best chance of a speedy review.

mgutierrez12@itam.mx Instituto Tecnológico Autónomo de México (ITAM) (Mexico Autonomous Institute of Technology)

+ Add an email address

What is the name of your school? *

Note: If your school is not listed, then enter the full school name and continue. You will be asked to provide further information about your school on the next page.

Instituto Tecnológico Autónomo de México (ITAM)

We chose this school based on your email. If this isn't your school, please use a different email.

How do you plan to use GitHub? *

For school and learning purposes.

Please note, your request cannot be edited once it has been submitted, so please verify your details for accuracy before sending them to us.

Continue

Recordatorio que git != github

Git	GitHub
Es el responsable del control de versiones (terminal)	Sitio web

3. Crear un repositorio (recomiendo habilitar la creación del README)

The screenshot shows the GitHub homepage with several features visible:

- Create your first project:** Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.
- Recent activity:** When you take actions across GitHub, we'll provide links to that activity here.
- The home for all developers — including you.** Welcome to your personal dashboard, where you can find an introduction to how GitHub works, tools to help you build software, and help merging your first lines of code.
- Start writing code:** Start a new repository, Create your profile README, Contribute to an existing repository, Find repos that need your help >.
- Use tools of the trade:** Write code in your web browser, Install a powerful code editor, Set up your local dev environment.
- Get started on GitHub:** About version control and Git, The GitHub Flow.
- Universe 2022:** Let's build from here, Join the global developer event for cloud, security, community, and AI. Register today and get a 20% off early bird discount. Register now.
- GitHub Copilot:** Get suggestions for lines of code and entire functions in real-time. Learn more about Copilot.
- Welcome to GitHub Global Campus!** Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community. Go to Global Campus.
- Latest changes:** 18 minutes ago: Link existing branches to an issue, 5 hours ago: Better suggested pull request description from commit message.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner * **Repository name ***

Great repository names are short and memorable. Need inspiration? How about [fluffy-spork](#)?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file This is where you can write a long description for your project. [Learn more](#).

Add .gitignore Choose which files not to track from a list of templates. [Learn more](#).

Choose a license A license tells others what they can and can't do with your code. [Learn more](#).

This will set `main` as the default branch. Change the default name in your [settings](#).

Info: You are creating a public repository in your personal account.

Create repository

4. Clonar el repositorio

- En la sección de code picarle al botón verde y copiar la liga https

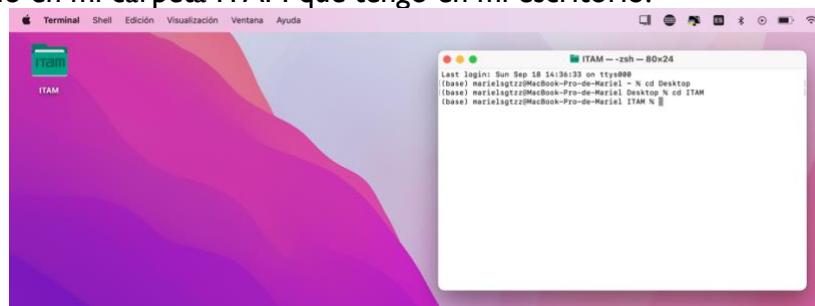
b. Abrir la terminal y definir en qué parte se quiere la carpeta/repositorio
Cuando escriben **ls** en la terminal les muestran los archivos y carpetas que hay en esa dirección.

```
Last login: Thu Sep  8 14:45:53 on ttys000
(base) marielsgtzz@MacBook-Pro-de-Mariel ~ % ls
ALTER          NetBeansProjects
Applications   Parallels
Applications (Parallels) Pictures
Creative Cloud Files Postman
Desktop        Projects
Documents      Public
Downloads     eclipse
Library        node_modules
Microsoft      package-lock.json
Movies         package.json
Music          yarn.lock
(base) marielsgtzz@MacBook-Pro-de-Mariel ~ %
```

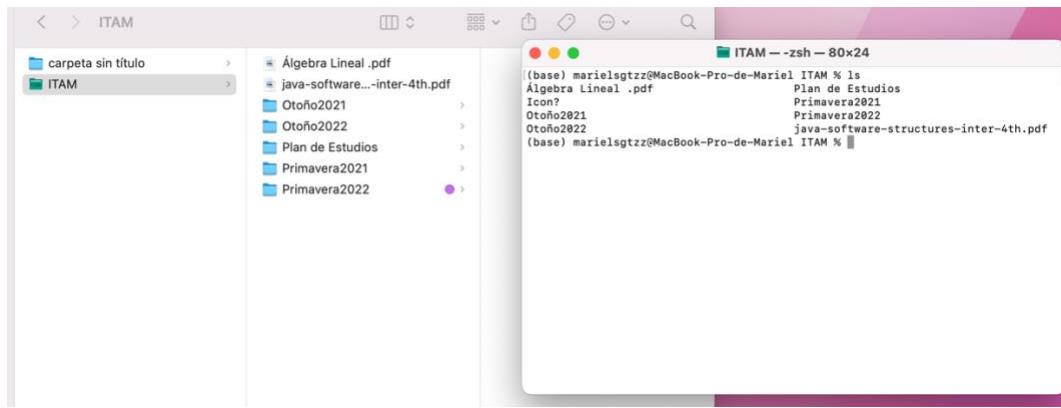
Para moverse a una de esas carpetas escriben **cd elNombreDeLaCarpeta**. Por ejemplo: **cd Desktop**. Cuando fue exitoso el movimiento, cambia la dirección actual que se muestra en la terminal.

```
Last login: Sun Sep 18 14:36:33 on ttys000
(base) marielsgtzz@MacBook-Pro-de-Mariel ~ % cd Desktop
(base) marielsgtzz@MacBook-Pro-de-Mariel Desktop %
```

Si queremos nuestro repositorio local dentro de una carpeta que está en Desktop se vuelve a hacer lo mismo, en este caso **cd ITAM** porque yo quiero mi repositorio en mi carpeta ITAM que tengo en mi escritorio.



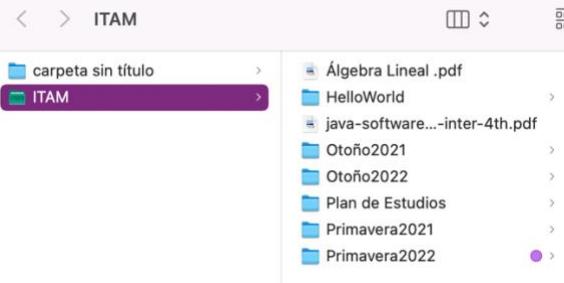
Nuevamente, si escribo **ls** puedo ver qué archivos tengo en ese directorio, en este caso:



- c. Cuando ya se está en el directorio donde se quiere el repositorio, hay que pegar la liga del paso 4a y escribir `git clone <url>`

```
(base) marielsgtzz@MacBook-Pro-de-Mariel ITAM % git clone https://github.com/MarielGuZa/HelloWorld.git
Cloning into 'HelloWorld'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
(base) marielsgtzz@MacBook-Pro-de-Mariel ITAM %
```

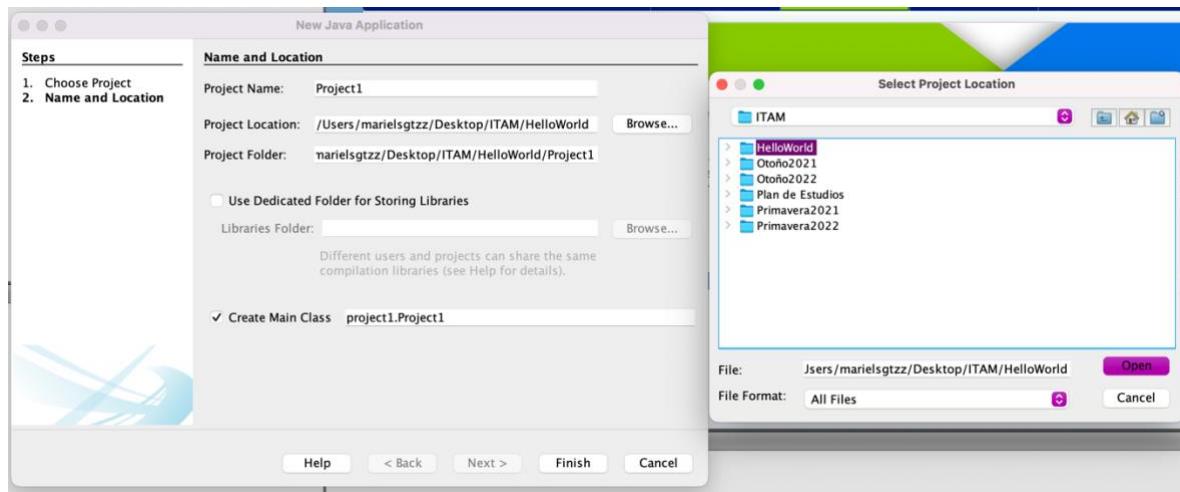
Con esto se crea la carpeta donde está el repositorio local.



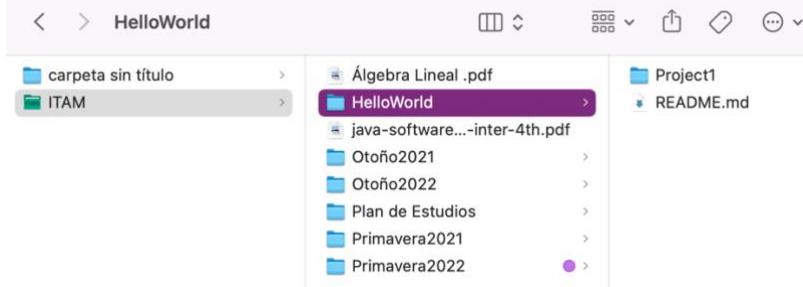
5. Poner su proyecto en esa carpeta

Ejemplo con Netbeans.

Si es nuevo proyecto, verificar que la 'Project Location' sea la carpeta que se creó en el paso 4c.



Ya está en
el
repositorio
local



Falta
tenerlo
también en
el público

MarielGuZa / HelloWorld Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main · 1 branch · 0 tags Go to file Add file · Code · About

MarielGuZa Initial commit c9e2231 6 minutes ago 1 commit

README.md initial commit 6 minutes ago

README.md

>HelloWorld

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

6. En la terminal movernos al repositorio

Nos quedamos en Desktop/ITAM pero el repositorio está en Desktop/ITAM/HelloWorld, entonces hay que movernos con la terminal dentro del repositorio (en este caso usaremos `cd HelloWorld`)

7. `git add .`

8. `git commit -m "Inicializacion proyecto"`

Si nos quedamos con duda de en qué condiciones está el proyecto escribimos `git status` y la terminal nos da un mensaje de qué es lo que sigue. Cuando hay errores este comando es muy útil para googlear soluciones.

```
● ● ● HelloWorld — zsh — 80x24
[(base) marielsgtzz@MacBook-Pro-de-Mariel ITAM % cd HelloWorld]
[(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git add .]
[(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git commit -m "Inicializaci]
on proyecto"
[main b85bd59] Inicializacion proyecto
 8 files changed, 1986 insertions(+)
  create mode 100644 .gitignore
  create mode 100644 Project1/build.xml
  create mode 100644 Project1/manifest.mf
  create mode 100644 Project1/nbproject/build-impl.xml
  create mode 100644 Project1/nbproject/genfiles.properties
  create mode 100644 Project1/nbproject/project.properties
  create mode 100644 Project1/nbproject/project.xml
  create mode 100644 Project1/src/project1/Project1.java
[(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git status]
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

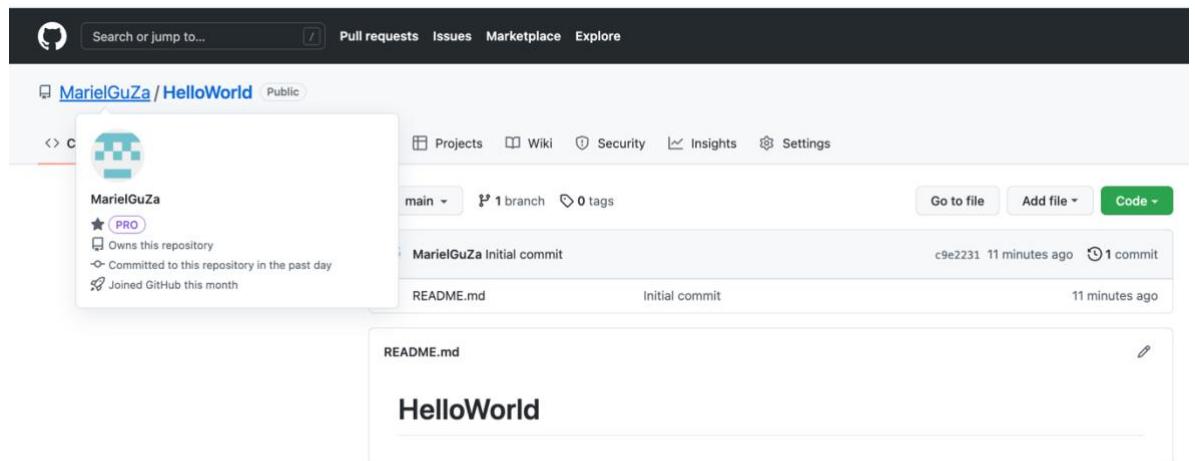
9. `git push`

Si es la primera vez que se usa esta cuenta de GitHub, al querer realizar el primer push nos van a pedir credenciales.

```
● ● ● HelloWorld — git-remote-https · git push — 80x24
[(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git push]
Username for 'https://github.com':
```

a) Ingresar usuario

El usuario no es el email de la cuenta, pero el username que elegimos



MarielGuZa / HelloWorld Public

MarielGuZa

★ PRO

Owns this repository

Committed to this repository in the past day

Joined GitHub this month

main 1 branch 0 tags

MarielGuZa Initial commit c9e2231 11 minutes ago 1 commit

README.md Initial commit 11 minutes ago

README.md

HelloWorld

b) Ingresar contraseña

De forma similar, la contraseña no es la de la cuenta, es una token que tenemos que generar



```
HelloWorld — git-remote-https ▷ git push — 80x24
```

```
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git push
```

```
Username for 'https://github.com': MarielGuZa
```

```
Password for 'https://MarielGuZa@github.com': ?
```

i. Ir a ajustes en el perfil

The screenshot shows a GitHub repository page for 'MarielGuZa / HelloWorld'. On the right side, the user's profile sidebar is open, showing the 'Settings' option under 'Feature preview' which is highlighted in blue. Other options include 'Signed in as MarielGuZa', 'Your profile', 'Your repositories', 'Your codespaces', 'Your projects', 'Your stars', 'Your gists', 'Help', and 'Sign out'.

ii. Ir a developer settings

The screenshot shows the 'Public profile' section of the GitHub settings. The left sidebar lists various settings categories like 'Public profile', 'Account', 'Appearance', etc. The main area contains fields for 'Name', 'Profile picture' (with a placeholder image), 'Public email' (disabled), 'Bio', 'URL', 'Twitter username', 'Company', and 'Location'. A note at the bottom states that all fields are optional and can be deleted. A green 'Update profile' button is at the bottom left. Below the profile section is a 'Contributions & Activity' section with a note about making the profile private.

iii. Personal Access tokens

The screenshot shows the 'Personal access tokens' section of the developer settings. The left sidebar has tabs for 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens', with 'Personal access tokens' selected. The main area displays a table of generated tokens with columns for 'Token name', 'Scopes', 'Created', 'Last used', and 'Actions' (with 'Revoke' and 'Edit' buttons). A note at the bottom explains that personal access tokens function like OAuth tokens and can be used for Git over HTTPS or API authentication. Buttons for 'Generate new token' and 'Revoke all' are at the top right.

iv. Generar una nueva token

Como es para ustedes, que en este caso están como administradores, le asignan todos los permisos a la token.

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

HelloWorld

What's this token for?

Expiration *

30 days The token will expire on Tue, Oct 18 2022

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input checked="" type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input checked="" type="checkbox"/> notifications	Access notifications

v. Darle click a ‘Generar token’

<input checked="" type="checkbox"/> admin:enterprise	Full control of enterprises
<input type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner groups
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input checked="" type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input checked="" type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate token

Cancel

vi. **Copiar la token que se generó y luego pegarla en la terminal en donde se pidió (paso 9b)**

```
>HelloWorld — zsh — 80x24
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git push
Username for 'https://github.com': MarielGuZa
>Password for 'https://MarielGuZa@github.com':
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (14/14), 15.63 KiB | 7.81 MiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MarielGuZa>HelloWorld.git
  c9e2231..b85bd59 main -> main
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld %
```

Después de darle `ctrl+v` o `command+v` en la terminal para pegar la contraseña, no se va a ver texto, pero sí se pegó. Si se hizo todo adecuadamente aparecerá un mensaje como el de arriba, si no probablemente aparecerá un error 403 que tiene que ver con la autorización, revisen los pasos e intenten nuevamente de ser ese el caso.

Si la autorización fue exitosa se ejecutará el push (marcado por azul), el tiempo que esto tarde depende de cuántos archivos van en el cambio.

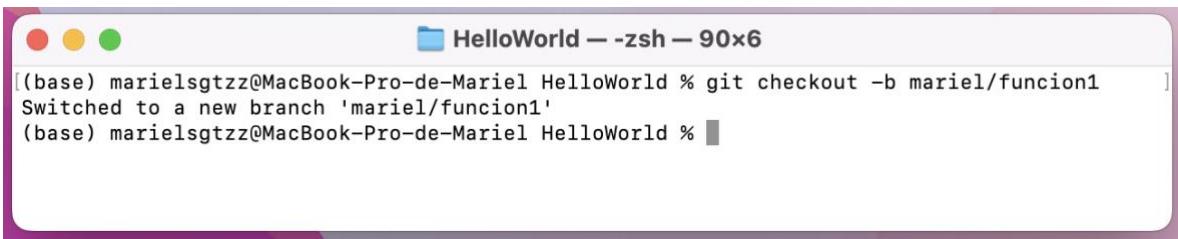
Si ahora checamos el repositorio en GitHub ya se pueden ver los cambios.

The screenshot shows a GitHub repository page for 'HelloWorld' owned by 'MarielGuZa'. The repository has 1 main branch, 0 tags, and 2 commits. The latest commit was made 17 minutes ago. The repository has 0 stars, 1 watching, and 0 forks. It has no releases published. The code is written in Java, with 100% coverage. The README.md file contains the text 'HelloWorld'.

10. Crear una Branch para continuar haciendo cambios.

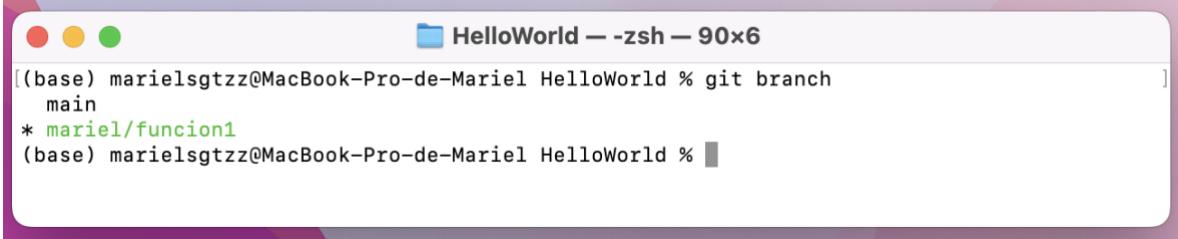
Para realizar cambios posteriores es recomendable que sigan el mecanismo de branches (aka ramas).

Para crear una puede ser `git Branch nombreDeLaRama` pero con el comando de `git checkout -b nombreDeLaRama` creamos una rama y nos cambiamos a ella.



```
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git checkout -b mariel/funcion1
Switched to a new branch 'mariel/funcion1'
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld %
```

Para checar que ramas se tienen y en cual se está se usa el comando **git branch**



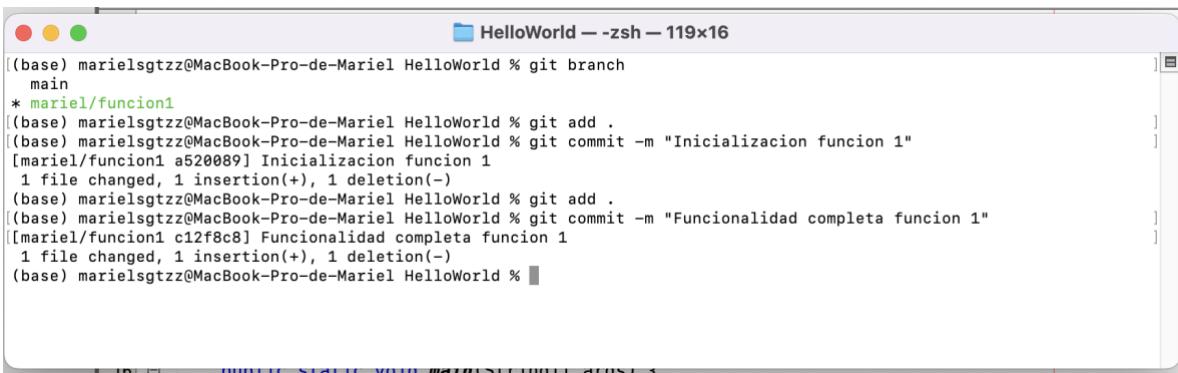
```
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git branch
  main
* mariel/funcion1
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld %
```

Para movernos a una rama ya existente usamos **git checkout nombreDeLaRama**.

II. Realizar nuevos cambios

Ya que se está en una nueva rama hacer cambios → git add. → git commit -m "msg"

*En un push pueden haber varios commits



```
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git branch
  main
* mariel/funcion1
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git commit -m "Incializacion funcion 1"
[mariel/funcion1 a520089] Incializacion funcion 1
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git commit -m "Funcionalidad completa funcion 1"
[mariel/funcion1 c12f8c8] Funcionalidad completa funcion 1
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld %
```

En este caso se hicieron dos commits, si queremos ver la lista de commits que se han hecho se escribe **git log**, el cual muestra los commits empezando por el más reciente.

El código que viene después de commit que está **marcado en amarillo** es el **commit hash** el cual se puede usar para hacer referencia a ese commit en particular.

```

(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld % git log
commit c12f8c8d62d15c48c49f0d4c29eed15d5f9a2459 (HEAD -> mariel/funcion1)
Author: MarielGuza <mgutii1220itam.mx>
Date: Mon Sep 19 21:04:31 2022 -0500

    Funcionalidad completa funcion 1

commit a520089e92f217275b33a3cc9282d356089e122d
Author: MarielGuza <mgutii1220itam.mx>
Date: Mon Sep 19 21:03:52 2022 -0500

    Inicializacion funcion 1

commit b85bd59309e083533ad72589dd6af5b4b11b99d (origin/main, origin/HEAD, main)
Author: MarielGuza <mgutii1220itam.mx>
Date: Sun Sep 18 14:57:58 2022 -0500

    Inicializacion proyecto

commit c9e2231bcc038e6c4d857d1352a96bd853e2f395
Author: MarielGuza <113052643+MarielGuza@users.noreply.github.com>
Date: Sun Sep 18 14:48:59 2022 -0500

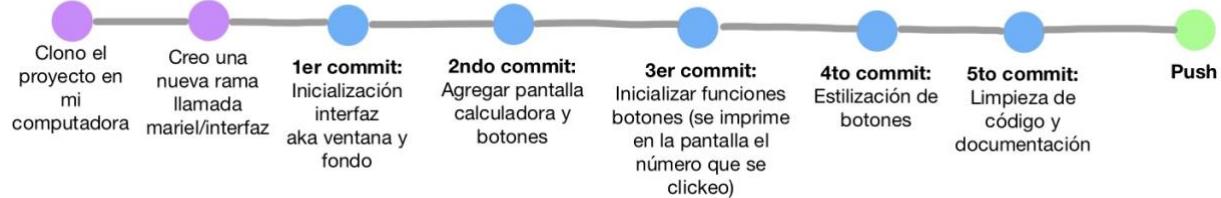
    Initial commit
(base) marielsgtzz@MacBook-Pro-de-Mariel HelloWorld %

```

Una **buenas prácticas** es que se haga commit cada vez que se haga un cambio que resulte en código funcional y push cuando la función/parte del proyecto ya esté “completo” y/o se quiera mostrar el avance a alguien más del equipo (ojo, se hace push a la rama personal, no al main).

Ejemplo, digamos que estoy trabajando en el proyecto calculadora, particularmente en agregar la interfaz gráfica mientras otra compañera trabaja en la función para pasar de notación infija a posfija.

Mi proceso de trabajo se vería algo así:



Este proceso se vería algo así en la terminal:

```
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git checkout -b mariel/interfaz
Switched to a new branch 'mariel/interfaz'
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git commit -m "Inicializacion interfaz aka ventana y fondo"
[mariel/interfaz 9108224] Inicializacion interfaz aka ventana y fondo
 2 files changed, 23 insertions(+), 1 deletion(-)
  create mode 100644 Calculadora/src/calculadora/InterfazCalculadora.java
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git commit -m "Agregar pantalla y botones"
[mariel/interfaz ee952f5] Agregar pantalla y botones
 1 file changed, 24 insertions(+)
  create mode 100644 Calculadora/src/calculadora/FuncionesInterfaz.java
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git commit -m "Inicializar botones funciones (imprimir numero)"
[mariel/interfaz 9829992] Inicializar botones funciones (imprimir numero)
 1 file changed, 5 insertions(+)
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git commit -m "Estilización botones"
[mariel/interfaz a02ad4d] Estilización botones
 1 file changed, 17 insertions(+)
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git add .
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git commit -m "Limpieza codigo y documentacion"
[mariel/interfaz eda2ae2] Limpieza codigo y documentacion
 2 files changed, 3 insertions(+), 2 deletions(-)
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git push
fatal: The current branch mariel/interfaz has no upstream branch.
To push the current branch and set the remote as upstream, use

  git push --set-upstream origin mariel/interfaz

(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git push --set-upstream origin mariel/interfaz
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 8 threads
Compressing objects: 100% (27/27), done.
Writing objects: 100% (32/32), 2.75 KiB | 2.75 MiB/s, done.
Total 32 (delta 14), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (14/14), completed with 2 local objects.
remote:
remote: Create a pull request for 'mariel/interfaz' on GitHub by visiting:
remote:   https://github.com/MarielGuZa/ProyectoCalculadora/pull/new/mariel/interfaz
remote:
To https://github.com/MarielGuZa/ProyectoCalculadora.git
 * [new branch]  mariel/interfaz -> mariel/interfaz
Branch 'mariel/interfaz' set up to track remote branch 'mariel/interfaz' from 'origin'.
```

En github la rama main se vería así:

mariel/interfaz had recent pushes 2 minutes ago

Compare & pull request

main ▾ 2 branches 0 tags

Go to file Add file ▾ Code ▾

MarielGuZa	Incializacion proyecto	f2ffaa2 7 minutes ago	2 commits
Calculadora	Incializacion proyecto	7 minutes ago	
.gitignore	Incializacion proyecto	7 minutes ago	
README.md	Initial commit	8 minutes ago	

README.md

ProyectoCalculadora

Abajo del botón de code vemos que dice que solo hay dos commits, eso es porque estamos viendo la rama de main (podemos cambiar cuál estamos viendo en la parte circulada en morado)

This branch is 5 commits ahead of main.

	MarielGuZa Limpieza codigo y documentacion	eda2ae2 5 minutes ago	7 commits
📄	Calculadora	Limpieza codigo y documentacion	5 minutes ago
📄	.gitignore	Inicializacion proyecto	9 minutes ago
📄	README.md	Initial commit	11 minutes ago

README.md

ProyectoCalculadora

La rama en la que estábamos trabajando

Si le damos click a la parte que dice commits (circulado en morado) podemos ver el detalle de los commits:

- Commits on Sep 20, 2022
 - Limpieza codigo y documentacion
MarielGuZa committed 6 minutes ago
 - Estilización botones
MarielGuZa committed 7 minutes ago
 - Inicializar botones funciones (imprimir numero)
MarielGuZa committed 7 minutes ago
 - Agregar pantalla y botones
MarielGuZa committed 8 minutes ago
 - Inicialización interfaz aka ventana y fondo
MarielGuZa committed 9 minutes ago
 - Inicialización proyecto
MarielGuZa committed 10 minutes ago
 - Initial commit
MarielGuZa committed 11 minutes ago

Y si le damos click a un commit en específico podemos ver detalles sobre él.

Ya que como todavía no hacemos **merge**, el código que acabamos de empujar a github solo aparecerá en la rama correspondiente.

I2. Crear una pull request (PR)

Por cuestiones de limpieza y preservar código funcional en los repositorios es que se recomienda crear ramas nuevas y trabajar en ellas. Una vez que se cumplió el propósito de la rama, que en el ejemplo anterior implica que se terminó la fase inicial de la interfaz,

buscamos unir el código de la rama al repositorio principal, lo que implica hacer un **merge**.

Para hacer un merge se necesita crear una pull request (conocida coloquialmente como PR) para que el resto del equipo pueda revisar, comentar y aprobar/rechazar los cambios.

Si vemos nuevamente la imagen anterior:

El mensaje enmarcado por el rectángulo morado nos está indicando que hay cambios en esta rama que no se encuentran en la rama principal, para resolver esto tenemos que hacer un **pull request**.

This screenshot shows a GitHub repository interface. At the top, there's a yellow header bar with the text "mariel/interfaz had recent pushes 5 minutes ago" and a green "Compare & pull request" button. Below the header, there are navigation buttons for "Code", "Issues", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". The "Code" button is highlighted with a purple rectangle. The main content area shows a list of commits from "MarielGuZa":

Commit	Message	Time
eda2ae2	Limpieza codigo y documentacion	5 minutes ago
Calculadora	Limpieza codigo y documentacion	5 minutes ago
.gitignore	Inicializacion proyecto	9 minutes ago
README.md	Initial commit	11 minutes ago

Below the commit list, there's a file editor for "README.md" with the title "ProyectoCalculadora".

Podemos hacer el pull request directamente desde el **botón** que nos muestra github (enmarcado en azul en la imagen anterior) o desde la pestaña marcada por 'Pull request'

This screenshot shows the same GitHub repository interface, but with the "Pull requests" tab selected (highlighted with a blue rectangle). The rest of the interface is identical to the previous screenshot, showing the commit history and the "ProyectoCalculadora" README file.

Crear una pull request se ve de esta forma:

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.

base: main ▾ compare: marel/interfaz ✓ Able to merge. These branches can be automatically merged.

Inicialización Interfaz Gráfica (1)

Leave a c Este PR agrega la interfaz gráfica al proyecto. Se agregaron los botones que se ven en una calculadora simple y también se agregó un label que sirve como pantalla en donde se muestran los resultados de las operaciones realizadas. Actualmente los botones tienen una funcionalidad básica en donde se imprime el número al que se dio click, esta operatividad se agregó para inicializar las funciones de los botones, en estos lugares se llamarían otros métodos para darle funcionalidad a la calculadora como tal.

Attach files by dragging & dropping, selecting or pasting them.

Create pull request (5)

① Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers (3)
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use Closing keywords in the description to automatically close issues

Helpful resources
[GitHub Community Guidelines](#)

⑥ Commits on Sep 20, 2022

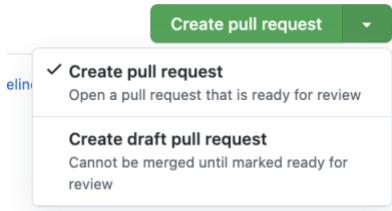
- o 5 commits
- 3 files changed
- 1 contributor

Commit	Author	Date	File Changes	SHA	Actions
Inicializacion interfaz aka ventana y fondo	MarielGuZa	committed 30 minutes ago		9108224	🔗 ⚡
Agregar pantalla y botones	MarielGuZa	committed 29 minutes ago		ee952f5	🔗 ⚡
Inicializar botones funciones (imprimir numero)	MarielGuZa	committed 28 minutes ago		9829992	🔗 ⚡
Estilización botones	MarielGuZa	committed 28 minutes ago		a02ad4d	🔗 ⚡
Limpieza codigo y documentacion	MarielGuZa	committed 27 minutes ago		eda2ae2	🔗 ⚡

Showing 3 changed files with 70 additions and 1 deletion.

Split Unified

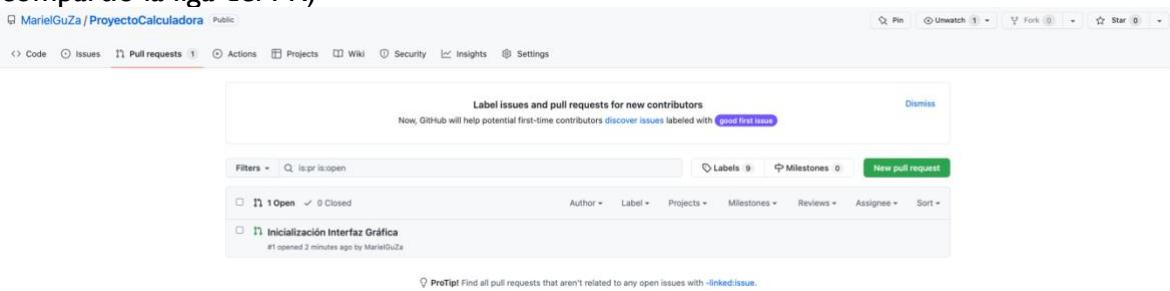
Componente de pull request	Descripción
1	El título de la pull request (PR), debe de ser representativo de la funcionalidad que brindan los cambios que se buscan incluir en la main branch.
2	La intención es que aquí el desarrollador elabore más respecto a los cambios que se realizaron, de implicar cambios <u>gráficos</u> es recomendable que también se incluyan fotos de dichos cambios.
3	Si se agregan reviewers, les llegará un correo cuando se cree la PR para que la revisen. Las cuestiones de quien y cuantas personas necesitan revisar un PR para aprobarlo es una cuestión de permisos que se pueden cambiar en cada proyecto (lo ideal es que

	mínimo 1 persona que no trabajó directamente en los cambios necesite revisar y aprobar la PR para que se pueda hacer merge, cuando son equipos chicos es recomendable que todos los integrantes revisen los cambios)
4	Muestra que rama se quiere incluir en el main.
5	Si se le da click en la flecha se presentan dos opciones:  <p>Si se crea la PR como draft es guardarla como borrador en lo que se agregan detalles de los cambios para después publicarla para que pueda ser posteriormente revisada y 'mergeada'</p>
6	Detalles de los cambios que se buscan aprobar.

¡¡Importante!! Una vez que se cree una pull request es una buena práctica no seguir modificando dicha rama a menos que sea para atender comentarios que se realizaron al PR. Los commits a los que se le hagan push (si es que sigue siendo la misma rama con la que se generó el PR) aparecerán en la pull request en GitHub aunque se hagan posteriormente a la creación del PR.

I 3. Revisar una pull request (PR)

Una vez que se creó una pull request ya aparece en la lista de PRs en la ventana de 'Pull requests' y se puede revisar si se le da click al título (también se puede accesar si se compartió la liga del PR)



En la pestaña de 'Conversation' del PR se puede ver el resumen de este y los comentarios que se han hecho al respecto (Aquí se pueden agregar directamente comentarios generales)

Si lo que se quiere es comentar cuestiones más particulares, como sobre una parte específica del código cambiado, entonces se usa la ventana de 'Files changed'.

Initialización Interfaz Gráfica #1

MarielGuZa wants to merge 5 commits into main from mariel/interfaz

Conversation 6 Commits 5 Checks 0 Files changed 3

Changes from all commits ▾ File filter ▾ Conversations ▾

Filter changed files

Viewed

Calculadora/src/calculadora/Calculadora.java

```
@@ -14,7 +14,18 @@ public class Calculadora {  
    // ...  
    // param args the command line arguments  
    public static void main(String[] args) {  
        // TODO code application logic here  
        System.out.println("Hello World");  
    }  
}
```

Viewed

Calculadora/src/calculadora/FuncionesInterfaz.java

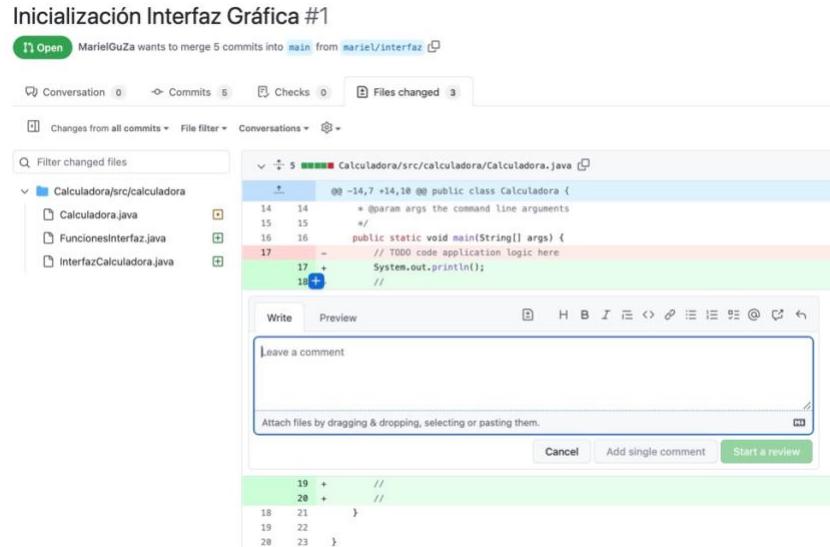
```
... @@ -8,0 +1,27 @@  
+ /*  
+ * Click nbfs://nhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  
+ * Click nbfs://nhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  
+ */  
+ package calculadora;  
+  
+ //  
+ //  
+ //  
+ // Author: marielguzg  
+ //  
+ //  
+ // Right-click to change the license information or choose this  
+ // file in the Source Explorer.  
+ // Right-click to edit this file.  
+ //
```

Viewed

En la barra lateral muestra la estructura del proyecto y en qué parte de esta se encuentran los archivos cambiados. Los símbolos a la derecha del nombre del archivo indican si el archivo se cambió, agregó o eliminó.

En la derecha podemos ver y copiar el código, el estatus es hasta el último commit incluido en el PR. Las líneas agregadas y/o cambiadas aparecen marcadas en verde y las eliminadas en rojo.

Si se quiere comentar una línea en específico hay que posicionarse sobre ella y darle click al símbolo de más que aparece.



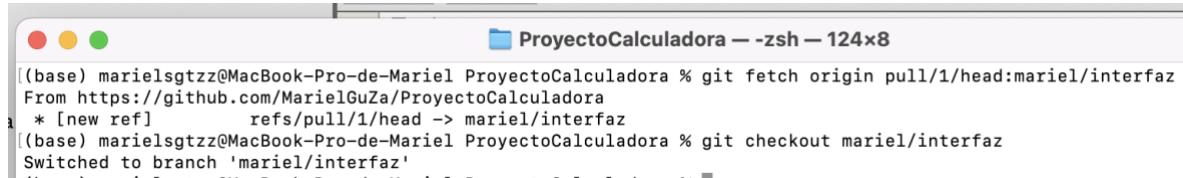
Si no basta visualmente analizar el código para confirmar que todo sigue funcionando, es oportuno correr el código de forma local.

Pensemos ahora que somo el reviewer y queremos ejecutar el código directamente desde NetBeans.

A screenshot of a GitHub repository page for "MarielGuZa / ProyectoCalculadora". The top navigation bar includes "Code", "Issues", "Pull requests" (with a value of 1), "Actions", "Projects", "Wiki", "Security", "Insights", "Settings", and "CodeQL". The "Pull requests" tab is highlighted. Below the navigation, the title "Inicialización Interfaz Gráfica #1" is shown, along with a green "Open" button and the text "MarielGuZa wants to merge 5 commits into main from mariel/interfaz".

1. Ver el ID de la pull request, en este caso 1. (circulado en morado)
2. Ver la rama en la que se trabajó. (circulado en azul)
3. En la terminal escribir `git fetch origin pull/1/head:branchName`
¡¡Importante!! Si estabas trabajando en otra rama haciendo otros cambios, es importante que les hagas commit antes de querer ver otros cambios, esto para evitar que tu trabajo se pierda.
4. Cambiarte a la rama donde están los cambios

Esto se vería así:



```
ProyectoCalculadora -- zsh -- 124x8
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git fetch origin pull/1/head:mariel/interfaz
From https://github.com/MarielGuZa/ProyectoCalculadora
 * [new ref]      refs/pull/1/head -> mariel/interfaz
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git checkout mariel/interfaz
Switched to branch 'mariel/interfaz'
```

Ahora el repositorio local será un reflejo de la rama en el momento en que se hizo el fetch y se podrá correr el código que se incluyó en el PR.

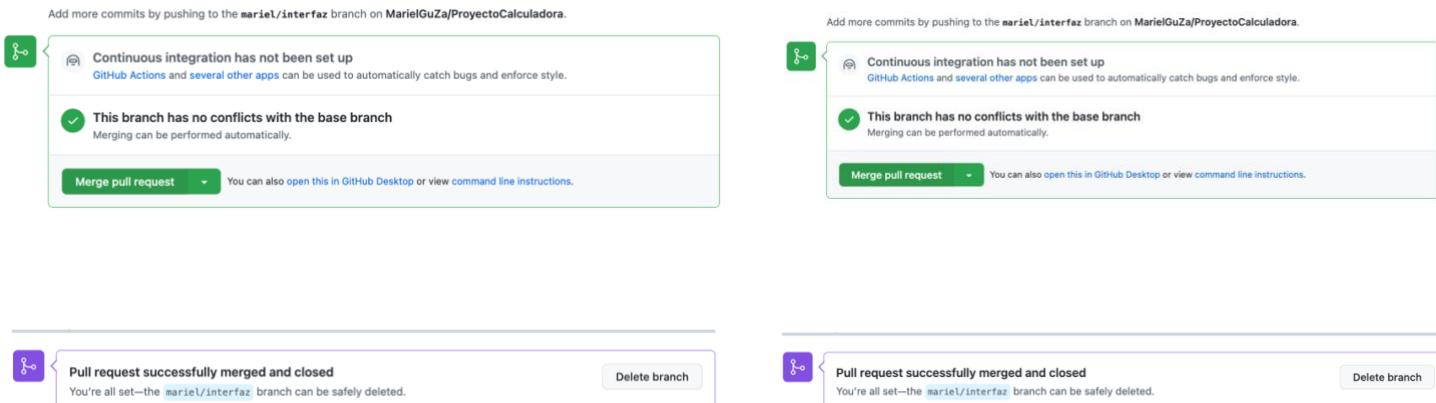
Alguien diferente a quien creó el PR puede realizar y empujar cambios (una buena práctica y cortesía es preguntarle a la dueña de la rama antes de realizar algún cambio). Los cambios que se realicen sobre la rama actualizan el PR.

Si quisiera revisar la rama y alguien realizó cambios sobre ella desde la última vez que yo la modifiqué sería necesario que me moviera a esa rama y que ejecute `git pull`, lo cual actualiza la rama en mi repositorio local según como se encuentra en GitHub. Si hay cambios que realicé que no se han guardado (add o commit) aparecerá un mensaje de error.

14. Aprobar una pull request (PR)

¡¡Importante!! Si se aprobó una pull request, pero la desarrolladora realizó más cambios antes de completar el PR, es necesario que se vuelva a aprobar la pull request.

Cuando las personas encargadas de revisar el código ya estén de acuerdo con los cambios ya se puede realizar el merge. La forma default de los PRs es que cualquier persona puede hacer merge en cualquier momento, una buena práctica sería establecer las configuraciones apropiadas al repositorio para asegurar su buen uso.



The screenshot shows the GitHub interface for merging a pull request. It displays two side-by-side merge buttons for a pull request from the 'mariel/interfaz' branch to the 'main' branch. Both buttons have the same text and appearance, indicating they perform the same function. The buttons are green with white text and are labeled 'Merge pull request'. Below each button is a link to 'GitHub Desktop' and 'command line instructions'.

Una vez que se hace merge, los commits del PR ya aparecerá en la rama a donde se unieron los cambios, en este caso main.

Commits on Sep 20, 2022

- Merge pull request #1 from MarielGuZa/mariel/interfaz ... Verified 75160e4
- Limpieza codigo y documentacion eda2ae2
- Estilización botones a02ad4d
- Inicializar botones funciones (imprimir numero) 9829992
- Agregar pantalla y botones ee952f5
- Inicializacion interfaz aka ventana y fondo 9108224
- Inicializacion proyecto f2ffaa2
- Initial commit 9df4b1f

Newer Older

Una **buenas prácticas** es que después de un merge se elimine la rama, esto para evitar posibles errores cuando se esté codificando después. Para realmente eliminar la branch se tiene que eliminar del repositorio global (el que vemos en GitHub) y en el local (la copia que tenemos en nuestra computadora).

- Para borrar en GitHub se puede hacer siguiendo los pasos que se muestran anteriormente, la opción aparece después de hacer el merge. También se puede desde la ventana de Code, después de darle click a branches (circulado en morado)

2 branches 0 tags

Commit	Message	Time
	MarielGuZa Merge pull request #1 from MarielGuZa/mariel/interfaz ...	75160e4 6 minutes ago
	Calculadora Limpieza codigo y documentacion	2 hours ago
	.gitignore Inicializacion proyecto	2 hours ago
	README.md Initial commit	2 hours ago

README.md

ProyectoCalculadora

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Search branches...

Overview Yours Active Stale All branches New branch

Default branch
main Updated 5 minutes ago by MarielGuZa Default

Your branches
mariel/ramaExtra Updated 5 minutes ago by MarielGuZa 0 0 Delete mariel/ramaExtra New pull request

Active branches
mariel/ramaExtra Updated 5 minutes ago by MarielGuZa 0 0 Delete mariel/ramaExtra New pull request

- Para borrar la rama localmente:

- a. No estar en esa rama, en caso de ser ese el caso moverse a alguna otra
- b. git branch -D nombreDeLaRama

```
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git branch  
  main  
* mariel/interfaz
```

```
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git branch -D mariel/interfaz  
Deleted branch mariel/interfaz (was eda2ae2).
```

```
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.
```

```
(base) marielsgtzz@MacBook-Pro-de-Mariel ProyectoCalculadora % git branch  
* main
```

Bibliografía:

<https://docs.github.com/es/get-started>