



# Introducción al manejador de versiones Git

# ¿Qué es Git?

Git es un popular sistema de control de versiones.

Se utiliza para:

- Seguimiento de cambios de código
- Seguimiento de quién realizó cambios
- Colaboración en la codificación



Introducción



# ¿Qué hace Git?

- Gestiona proyectos con repositorios
- Clonar un proyecto para trabajar en una copia local
- Controle y realice un seguimiento de los cambios con Staging y Commit
- Branch y Merge para permitir el trabajo en Diferentes partes y versiones de un proyecto
- Extraer la versión más reciente del proyecto en una copia local
- Enviar actualizaciones locales al proyecto principal



# Trabajar con Git

- Inicializar Git en una carpeta, convirtiéndola en un repositorio
- Git ahora crea una carpeta oculta para realizar un seguimiento de los cambios en esa carpeta
- Cuando un archivo se cambia, se agrega o se elimina, se considera modificado
- Seleccione los archivos modificados que desea almacenar provisionalmente
- Se confirman los archivos preconfigurados, que solicita a Git que almacene una instantánea permanente de los archivos
- Git te permite ver el historial completo de cada confirmación.
- Puedes volver a cualquier confirmación anterior.
- Git no almacena una copia separada de cada archivo en cada confirmación, pero realiza un seguimiento de los cambios realizados en cada confirmación!



# Instalar Git

- Acceder al siguiente link: <https://www.git-scm.com/> :



Primeros pasos

The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--distributed-is-the-new-centralized". A search bar is on the right. The main content area describes Git as a free and open source distributed version control system, highlighting its speed, efficiency, ease of learning, and performance. It also lists features like cheap local branching, staging areas, and multiple workflows. To the right of the text is a diagram illustrating the distributed nature of Git with multiple repositories and their connections. Below the main text are four navigation links: "About" (advantages of Git), "Documentation" (command reference, Pro Git book, videos), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list, chat). On the right side of the page, there is a section for the "Latest source Release" (2.47.0) with a "Download for Windows" button.




# Instalar Git

- Dar click en Downloads:



Primeros pasos

staging areas, and multiple workflows.

	<b>About</b> The advantages of Git compared to other source control systems.		<b>Documentation</b> Command reference pages, Pro Git book content, videos and other material.
	<b>Downloads</b> GUI clients and binary releases for all major platforms.		<b>Community</b> Get involved! Bug reporting, mailing list, chat, development and more.



Primeros pasos

# Instalar Git

- Seleccionar la primer opción de descarga que ofrece Git:

## Download for Windows

[Click here to download](#) the latest (2.47.0(2)) 64-bit version of Git for recent maintained build. It was released 19 days ago, on 2024-10-22.

### Other Git for Windows downloads

#### Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

#### Portable ("thumbdrive edition")

[32-bit Git for Windows Portable.](#)

# Configurar Git

- Comprobamos que Git este correctamente instalado:

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~  
$ git --version  
git version 2.47.0.windows.2
```





Configuración

# Configurar Git

- Registrarnos con nuestro username y correo, para que Git sepa quienes somos:

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~  
$ git config --global user.name "EdgarGtz25"  
  
tuesp@DESKTOP-6KH5F3U MINGW64 ~  
$ git config --global user.email "johanito777@gmail.com"
```

# Configurar Git

- Activamos el coloreado de salida, al usar los comandos.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~  
$ git config --global color.ui true
```



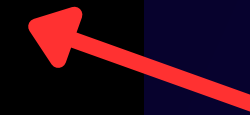


Configuración

# Configurar Git

- Git detecta una diferencia en los saltos de línea entre tu sistema y la configuración de Git para el repositorio. Git utiliza saltos de línea "LF" (Line Feed) por defecto, que es estándar en sistemas UNIX/Linux. En cambio, Windows usa "CRLF" (Carriage Return + Line Feed) para los saltos de línea.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~  
$ git config --global core.autocrlf true
```




Git te está informando que convertirá los saltos de línea de LF a CRLF la próxima vez que procese el archivo en Windows.



# Creación de nuestro espacio de trabajo

- Crear en alguna ruta de nuestro equipo una carpeta

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~  
$ cd Documents/  
  
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents  
$ mkdir repo  
  
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents  
$ ls  
IISExpress/  'My Web Sites'/  repo/
```






# Creación de nuestro espacio de trabajo

- Acceder a nuestra carpeta creada

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents
$ cd repo

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo
$ ls
```





# Repositorios en git

- El término **repositorio** de Git hace referencia al lugar donde se almacena el historial de cambios realizados en un proyecto, así como las diferentes versiones del mismo.



Mi primer  
repositorio





Mi primer  
repositorio

# Inicializar Git

- Para crear nuestro primer repositorio en git, usar el comando :

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents
$ cd repo

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo
$ git init
Initialized empty Git repository in C:/Users/tuesp/Documents/repo/.git/

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ ls

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ |
```




Mi primer  
repositorio

# Inicializar Git

- Git nos deja una carpeta oculta llamada *.git*, con las configuraciones correspondientes

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ ls -a
./  ../  .git/
```





Mi primer  
repositorio

# Git status

- Creamos un archivo .txt y vemos si esta dentro del repositorio:

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ echo "" > nombre_archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ ls
nombre_archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nombre_archivo.txt

nothing added to commit but untracked files present (use "git add" to t
ck)
```

# Git status

- Modificamos el archivo y volvemos a verificar el estado del repositorio:



Mi primer  
repositorio

```
nombre_archivo: Blo... MINGW64:/c/Users/tuesp/Documents/repo
Archivo Edición Form... tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
hola $ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nombre_archivo.txt

nothing added to commit but untracked files present (use "git add" to track)

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nombre_archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
```





Mi primer  
repositorio

# Git add

- Nos permite añadir archivos al área de **Stage**, que es una zona intermedia donde se preparan los cambios que queremos incluir en nuestro próximo commit.

```
nombre_archivo: Bloc
Archivo  Edición  Form
hola

MINGW64:/c/Users/tuesp/Documents/repo
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  nombre_archivo.txt

nothing added to commit but untracked files present (use "git add" to track)

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  nombre_archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
```

A red arrow points to the text "nothing added to commit but untracked files present (use 'git add' to track)" in the second terminal output.



Mi primer  
repositorio

# Git add

- Hacemos el git add con el nombre del archivo

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git add nombre_archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

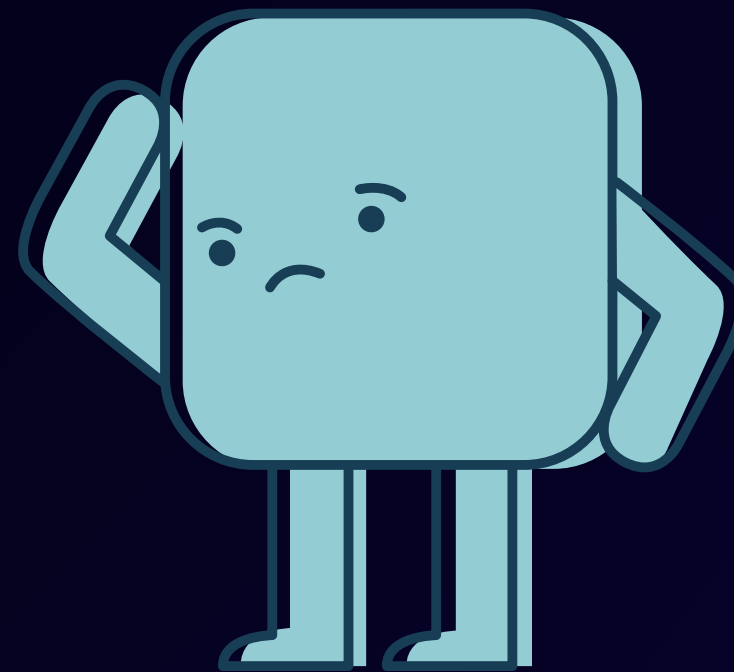
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   nombre_archivo.txt
```

# Práctica I

- Creen dos archivos nuevos y agréguenlos al área de stage



Mi primer  
repositorio





Mi primer repositorio

# Resultado esperado

- Creen dos archivos nuevos y agréguenlos al área de stage

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   nombre_archivo.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo2.txt
        archivo3.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git add archivo2.txt archivo3.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo2.txt
        new file:   archivo3.txt
        new file:   nombre_archivo.txt
```

1

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git rm --cached archivo2.txt archivo3.txt
rm 'archivo2.txt'
rm 'archivo3.txt'

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   nombre_archivo.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo2.txt
        archivo3.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git add .

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo2.txt
        new file:   archivo3.txt
        new file:   nombre_archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$
```

2



Mi primer  
repositorio

# Git commit

- Una vez que tenemos los cambios preparados en el área de Stage, ejecutaremos el comando `git commit` para **crear la fotografía y guardar los cambios en el repositorio**.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo2.txt
        new file:   archivo3.txt
        new file:   nombre_archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git commit -m "Cree repositorio y cree 3 archivos de texto sin contenido"
[master (root-commit) af01afc] Cree repositorio y cree 3 archivos de texto sin contenido
3 files changed, 1 insertion(+)
 create mode 100644 archivo2.txt
 create mode 100644 archivo3.txt
 create mode 100644 nombre_archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master
nothing to commit, working tree clean
```

`git commit -m "Message"`





Mi primer  
repositorio

# Git log

- Para visualizar las fotografías realizadas en un repositorio de Git, se utiliza el comando `git log`. Al ejecutarlo, Git muestra una lista con todos los commits realizados en el repositorio, incluyendo el **hash** único que identifica a cada uno de ellos.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git log
commit ef01afc92c5332c09f5e58f1c73746ae17737cf9 (HEAD -> master)
Author: EdgarGtz25 <johanito777@gmail.com>
Date:   Tue Nov 12 20:12:39 2024 -0600

    Cree repositorio y cree 3 archivos de texto sin contenido

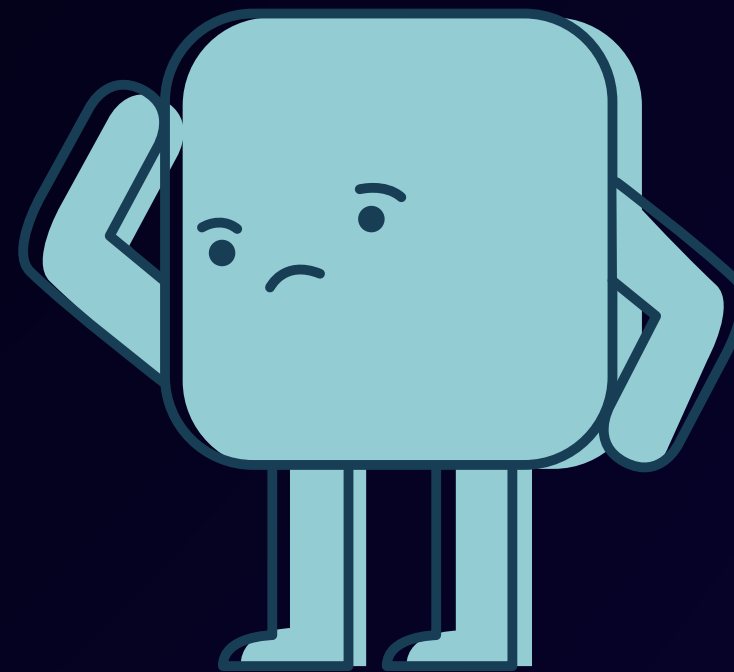
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git log --oneline
ef01afc (HEAD -> master) Cree repositorio y cree 3 archivos de texto sin
contenido
```

# Práctica II

- Presten atención a las indicaciones



Mi primer  
repositorio





Mi primer  
repositorio

# Git checkout

- Este comando nos permite situarnos en un punto específico del historial de commits o de un archivo.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))
$ git log
commit 0a183bdba9226cebf36a10bafc464e5c847d3d48 (HEAD, master)
Author: EdgarGtz25 <johanito777@gmail.com>
Date: Tue Nov 12 22:33:56 2024 -0600

    Segundos cambios

commit ef01afc92c5332c09f5e58f1c73746ae17737cf9
Author: EdgarGtz25 <johanito777@gmail.com>
Date: Tue Nov 12 20:12:39 2024 -0600

    Cree repositorio y cree 3 archivos de texto sin contenido

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))
$ git checkout ef
error: pathspec 'ef' did not match any file(s) known to git

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))
$ git checkout ef01
Previous HEAD position was 0a183bd Segundos cambios
HEAD is now at ef01afc Cree repositorio y cree 3 archivos de texto sin contenido

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((ef01afc...))
$ git log
commit ef01afc92c5332c09f5e58f1c73746ae17737cf9 (HEAD)
Author: EdgarGtz25 <johanito777@gmail.com>
Date: Tue Nov 12 20:12:39 2024 -0600

    Cree repositorio y cree 3 archivos de texto sin contenido
```

Se deben tomar como  
mínimo los 4  
primeros caracteres  
del hash



Mi primer  
repositorio

# Git checkout

- Además, nos llevará al estado previo de dicho archivo, correspondiente a la última fotografía tomada en la rama actual.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((ef01afc...))
$ git checkout 0a1
error: pathspec '0a1' did not match any file(s) known to git

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((ef01afc...))
$ git checkout 0a18
Previous HEAD position was ef01afc Cree repositorio y cree 3 archivos
HEAD is now at 0a183bd Segundos cambios

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))
$ git log
commit 0a183bdba9226cebf36a10bafc464e5c847d3d48 (HEAD, master)
Author: EdgarGtz25 <johanito777@gmail.com>
Date: Tue Nov 12 22:33:56 2024 -0600

    Segundos cambios

commit ef01afc92c5332c09f5e58f1c73746ae17737cf9
Author: EdgarGtz25 <johanito777@gmail.com>
Date: Tue Nov 12 20:12:39 2024 -0600

    Cree repositorio y cree 3 archivos de texto sin contenido

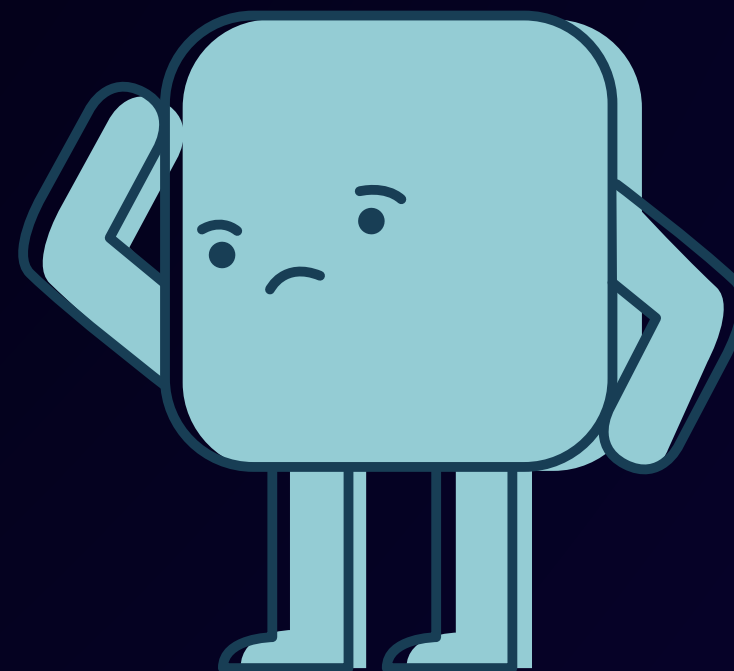
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))
$ |
```

# Práctica III

- Presten atención a las indicaciones



Mi primer  
repositorio







Mi primer  
repositorio

# Git checkout

- Estamos posicionados sobre un commit, no sobre la rama

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))  
$ git status  
HEAD detached at 0a183bd  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   index.html.txt  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    index.txt  
  
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))  
$ git add .  
  
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo ((0a183bd...))  
$ git status  
HEAD detached at 0a183bd  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   index.html.txt  
    new file:   index.txt
```

# Git checkout

- Regresamos a la rama master



Mi primer  
repositorio

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)  
$ git checkout master  
Already on 'master'
```



Mi primer  
repositorio

# Git checkout

- Uso de `log --graph` para ver la lista de commits, con el flujo de su branch.

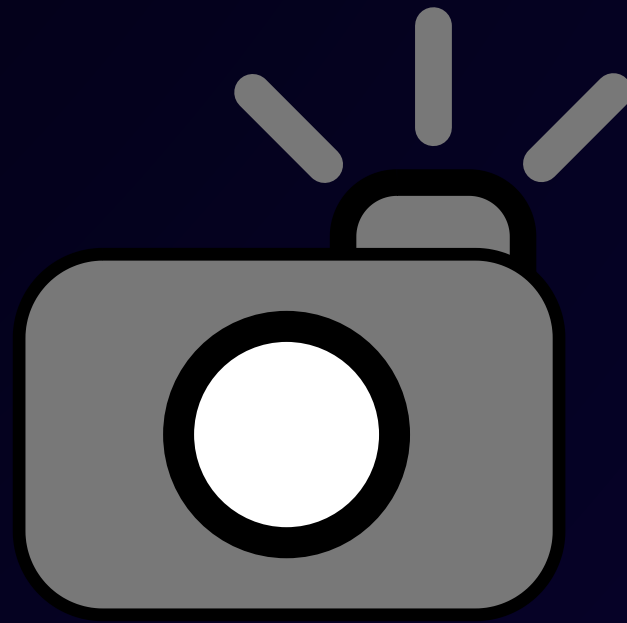
```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git log --graph
* commit f1b55e02eb3445ab8b28fb0b88fce87813880229 (HEAD -> master)
| Author: EdgarGtz25 <johanito777@gmail.com>
| Date:   Wed Nov 13 14:59:17 2024 -0600
|
|     --Agregue dos archivos nuevos--
|
| * commit 0a183bdba9226cebf36a10bafc464e5c847d3d48
| | Author: EdgarGtz25 <johanito777@gmail.com>
| | Date:   Tue Nov 12 22:33:56 2024 -0600
| |
| |     Segundos cambios
| |
| * commit ef01afc92c5332c09f5e58f1c73746ae17737cf9
| | Author: EdgarGtz25 <johanito777@gmail.com>
| | Date:   Tue Nov 12 20:12:39 2024 -0600
| |
| |     Cree repositorio y cree 3 archivos de texto sin contenido
```



Mi primer  
repositorio

# Git reset

- Si deseamos volver a la última fotografía completa tomada, podemos escribir git reset. Al lanzar este comando se nos informará de que se perderán los cambios en los archivos que no forman parte de un commit. Hecho esto, recuperaremos el contenido original del último punto de guardado de la rama.

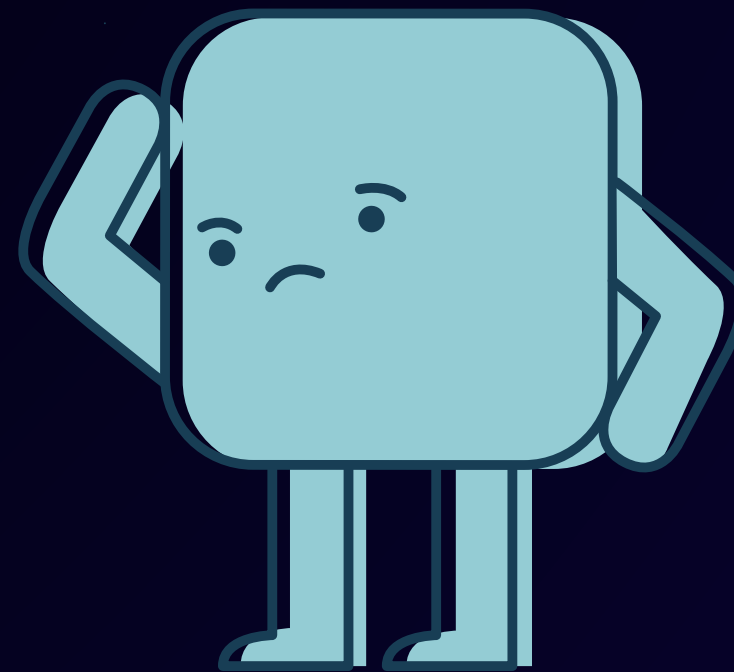


# Práctica IV

- Presten atención a las indicaciones

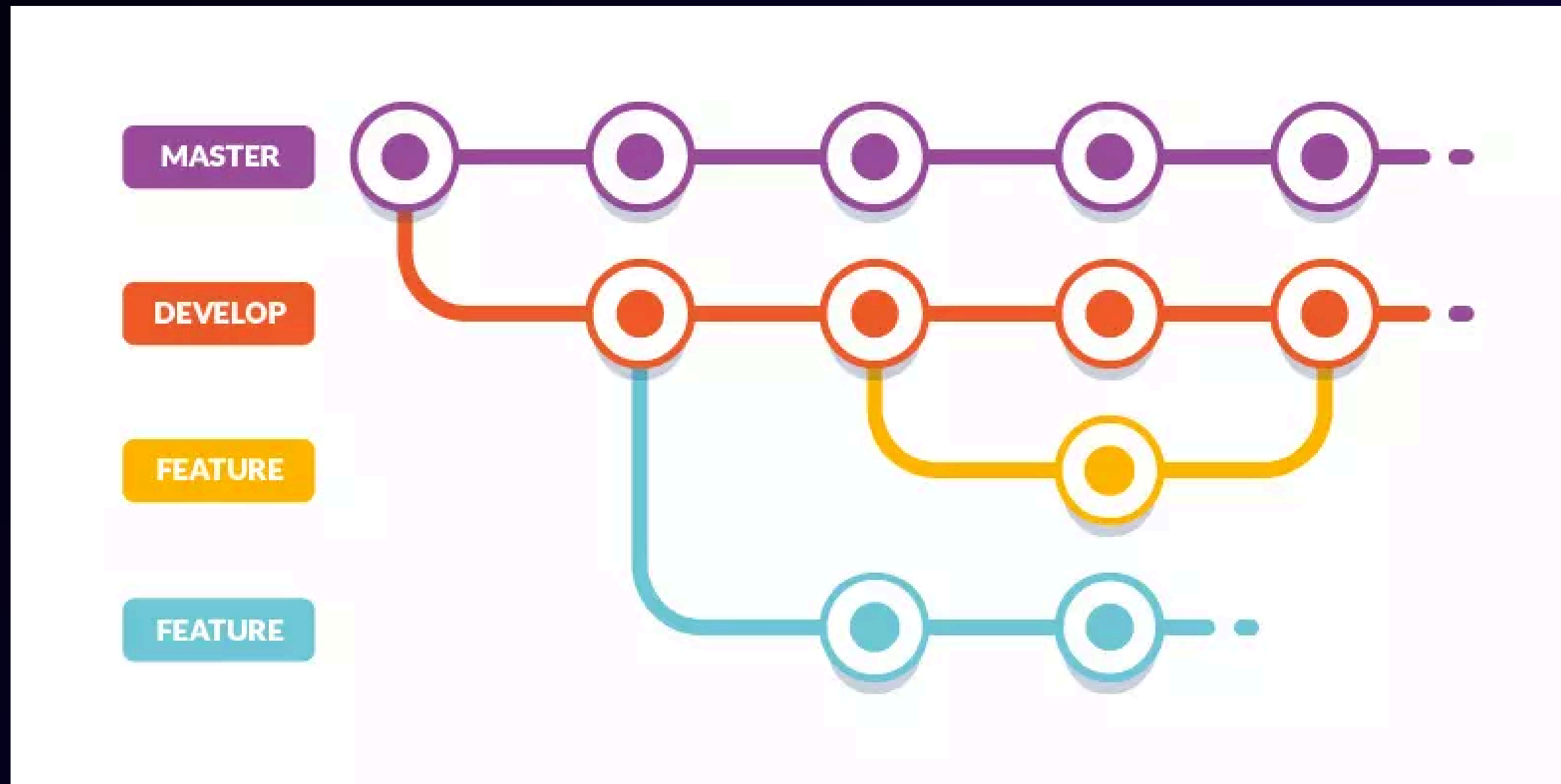


Mi primer  
repositorio



# Git branches

- Las ramas en Git permiten a los equipos trabajar en diferentes flujos de trabajo de manera independiente sin afectar la rama principal.





# Git branches

- Para crear una rama en git ocupamos *git branch*. Seguido a este para cambiarnos entre ramas ocupamos *git switch*.

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status -s

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git branch desarrollo

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git switch desarrollo
Switched to branch 'desarrollo'

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ |
```





# Git branches

- Podemos hacer cambios sin afectar a la rama principal **master**

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ git status
On branch desarrollo
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo de desarrollo.txt

nothing added to commit but untracked files present (use "git add" to track)

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ git add .

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ git commit -m "Agregue archivo a desarrollo"
[desarrollo d0367fa] Agregue archivo a desarrollo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo de desarrollo.txt
```



# Git branches

- Podemos ver el flujo de ramas con `git log --graph`

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ git log --graph
* commit d0367faf2f32812ddb6fcd2ea69af08e04bfd0f3 (HEAD -> desarrollo)
| Author: EdgarGtz25 <johanito777@gmail.com>
| Date:   Wed Nov 13 21:23:40 2024 -0600
|
|     Agregue archivo a desarrollo
|
* commit f1b55e02eb3445ab8b28fb0b88fce87813880229 (master)
| Author: EdgarGtz25 <johanito777@gmail.com>
| Date:   Wed Nov 13 14:59:17 2024 -0600
|
|     --Agregue dos archivos nuevos--
|
* commit 0a183bdba9226cebf36a10bafc464e5c847d3d48
| Author: EdgarGtz25 <johanito777@gmail.com>
| Date:   Tue Nov 12 22:33:56 2024 -0600
|
|     Segundos cambios
|
* commit ef01afc92c5332c09f5e58f1c73746ae17737cf9
| Author: EdgarGtz25 <johanito777@gmail.com>
| Date:   Tue Nov 12 20:12:39 2024 -0600
|
|     Cree repositorio y cree 3 archivos de texto sin contenido
|
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ |
```



# Git merge

- Para fusionar cambios de una rama con otra, ocupamos *git branch*

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (desarrollo)
$ git switch master
Switched to branch 'master'

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git merge desarrollo
Updating f1b55e0..d0367fa
Fast-forward
 archivo de desarrollo.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archivo de desarrollo.txt

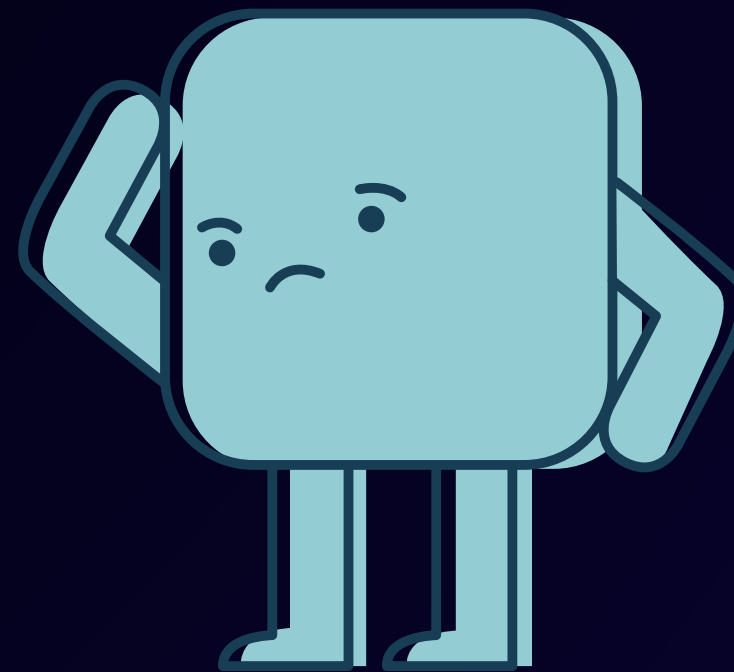
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo (master)
$ git status
On branch master
nothing to commit, working tree clean
```

# Práctica V

- Presten atención a las indicaciones



Mi primer  
repositorio





# GitHub

- GitHub es una plataforma de alojamiento de código que utiliza el sistema de control de versiones Git. GitHub nos permite subir nuestro código a un **servidor remoto** para que otras personas puedan interactuar con él.

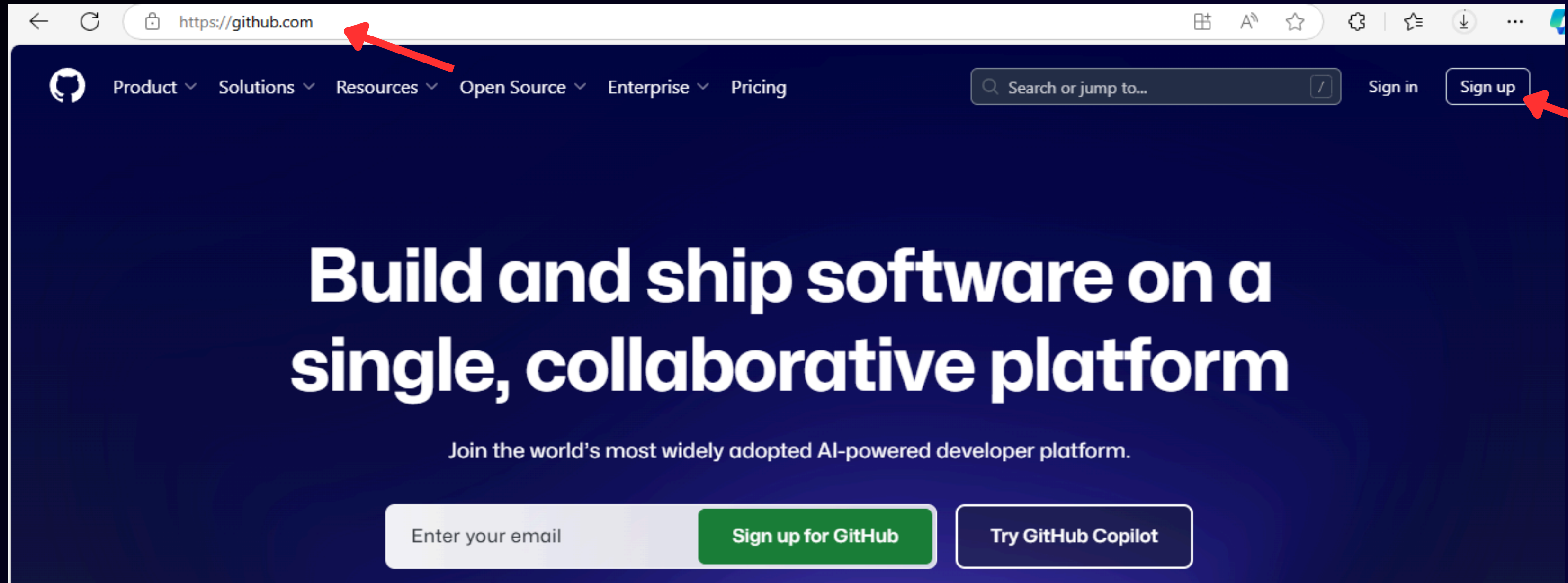




# GitHub VS Git

- Con Git hemos estado trabajando de manera local. Todo el flujo de trabajo que hemos visto hasta ahora se ha llevado a cabo en nuestro propio ordenador.
- Dicho esto, **¿qué es GitHub?** GitHub es una plataforma que emplea Git en la nube, de forma remota. Nos permite subir nuestro código a un servidor remoto para que otras personas (o nosotros mismos) puedan interactuar con él desde cualquier parte.

# Hacer una cuenta en GitHub o loguearse





# Colocar sus datos



Welcome to GitHub!  
Let's begin the adventure

Enter your email\*

✓ repeh65107@edectus.com

Create a password\*

✓ calixto.2024

Enter a username\*

→ EdgarGtz-G1z|

Continue



# Verificamos la cuenta



Welcome to GitHub!  
Let's begin the adventure

**Verify your account**

## Protección de su cuenta

Solucione este rompecabezas para que sepamos que es una persona real

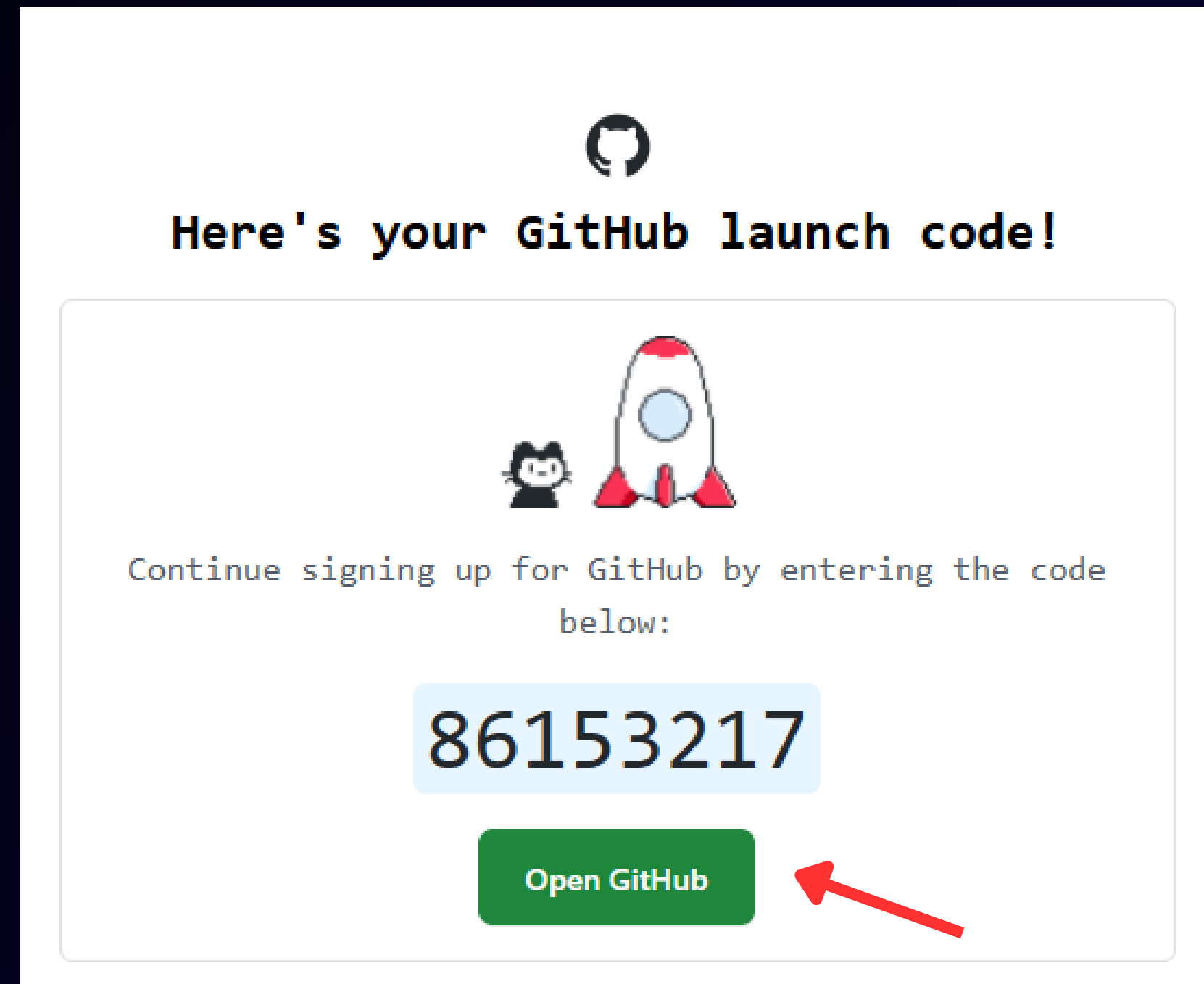
Verificar

6361808061e61a042.5947979301

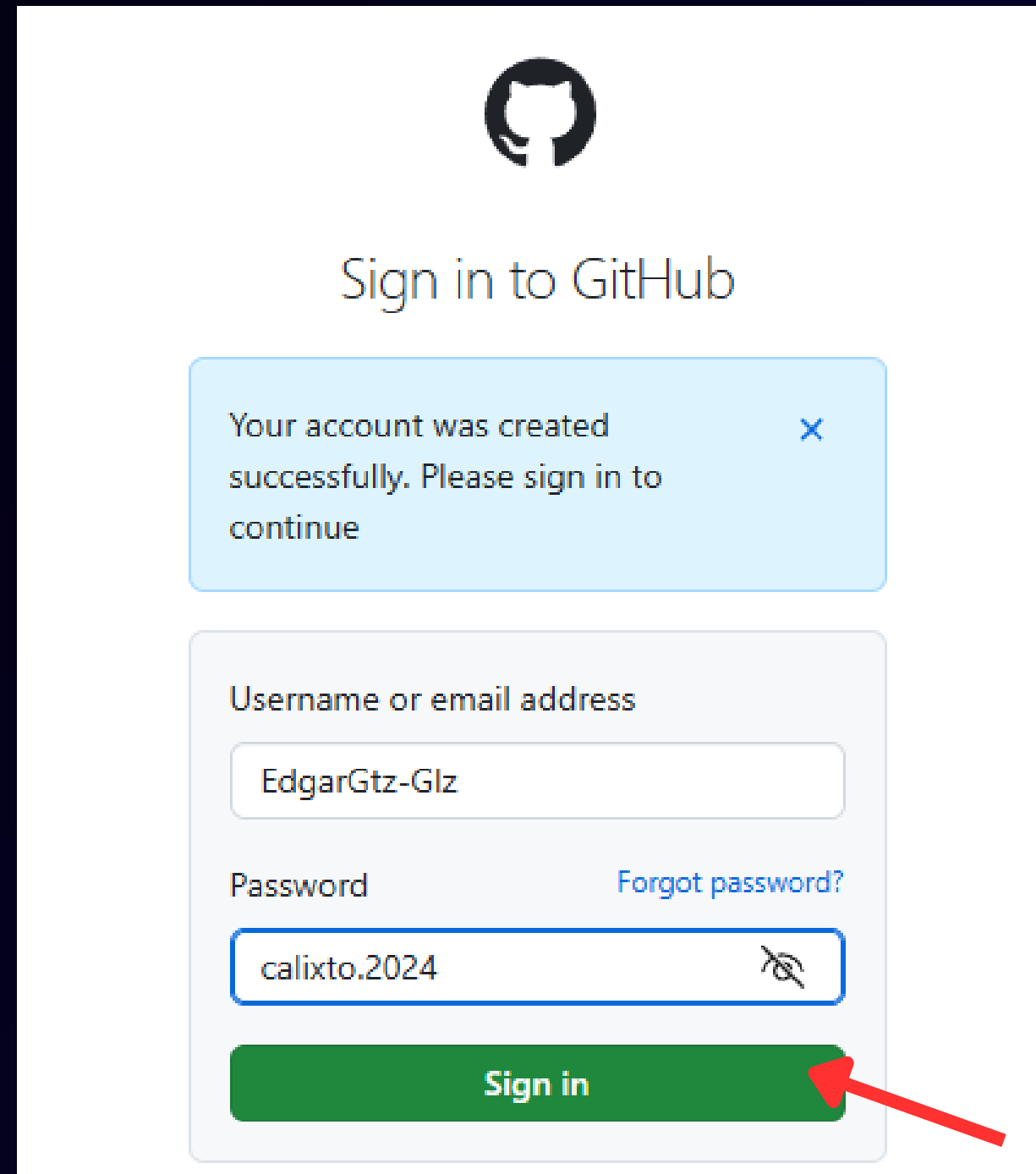





# Colocar el código de seguridad en su correo



# Iniciamos sesión en GitHub

A screenshot of the GitHub sign-in page. At the top is the GitHub logo. Below it is the text "Sign in to GitHub". A light blue notification box contains the message "Your account was created successfully. Please sign in to continue" with a close button (X). Below this is a form with two input fields: "Username or email address" containing "EdgarGtz-Glz" and "Password" containing "calixto.2024". There is a "Forgot password?" link next to the password field. At the bottom is a green "Sign in" button, which is highlighted by a red arrow pointing to it from the right.




Sign in to GitHub

Your account was created successfully. Please sign in to continue [×](#)

Username or email address

EdgarGtz-Glz

Password [Forgot password?](#)

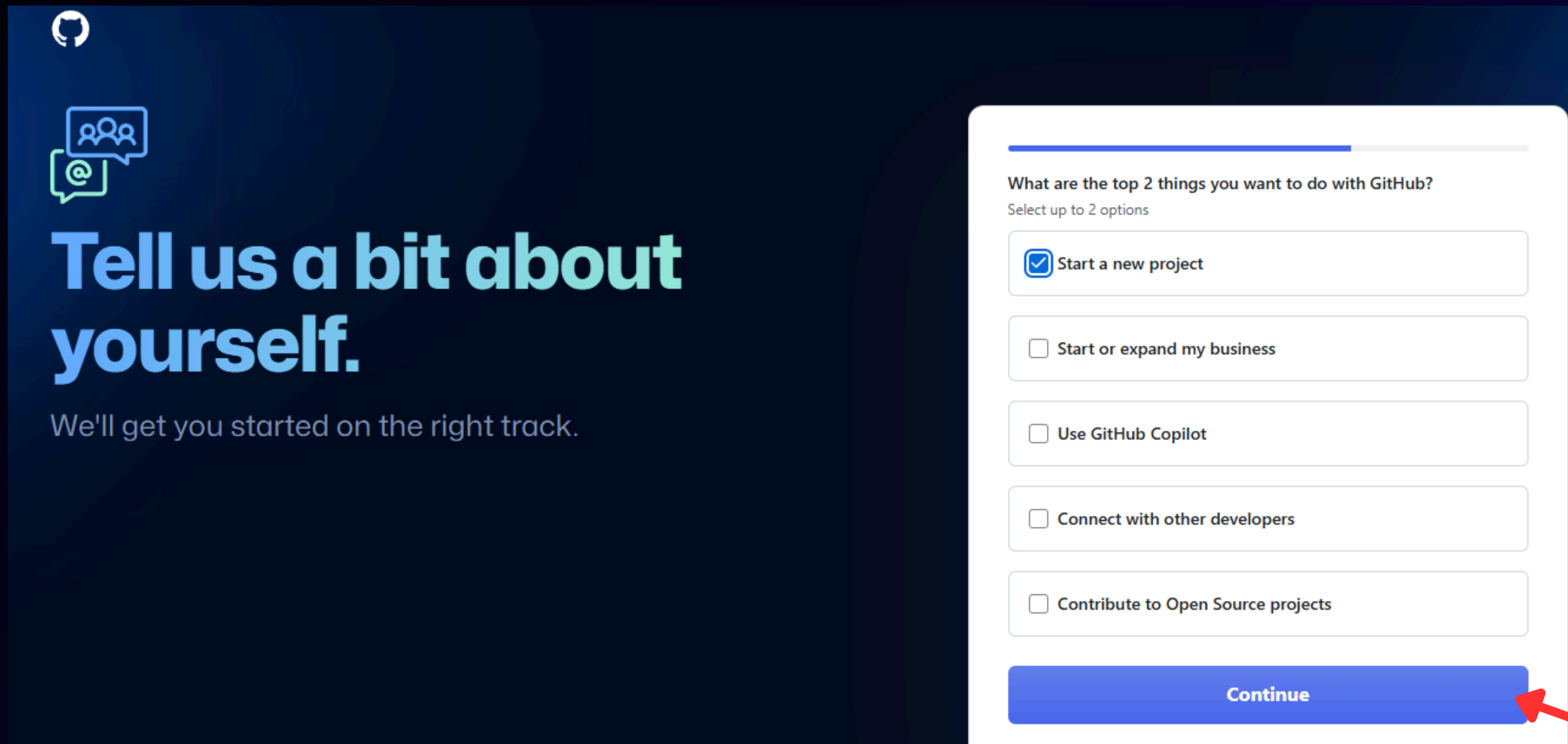
calixto.2024 


**Sign in**


# Welcome to GitHub

A screenshot of the GitHub welcome screen. On the left, there's a dark blue background with the GitHub logo at the top left, followed by an icon of three people in speech bubbles. Below this, the text "Welcome to GitHub" is written in a large, bold, light blue font, and "We are glad you're here." is written in a smaller, lighter blue font. On the right, there's a white registration form with a blue progress bar at the top. The form contains two sections: "How would you describe yourself?" with four radio button options (Student, Teacher, Working developer, Other) and "How many team members will be working with you?" with six radio button options (Just me, 2-5, 5-10, 10-20, 20-50, 50+). At the bottom of the form is a blue "Continue" button, which is pointed to by a red arrow from the right.

# ¿Quiénes somos?

A screenshot of the GitHub onboarding form. The form is white and stands out against a dark blue background. It contains a progress bar at the top, a question about top 2 things to do with GitHub, a list of five options with checkboxes, and a "Continue" button at the bottom. A red arrow points to the "Continue" button.





## Tell us a bit about yourself.

We'll get you started on the right track.

What are the top 2 things you want to do with GitHub?  
Select up to 2 options

- ☒ Start a new project
- ☐ Start or expand my business
- ☐ Use GitHub Copilot
- ☐ Connect with other developers
- ☐ Contribute to Open Source projects

[Continue](#)

# Elige el plan con el que trabajaremos GitHub



## Learn to ship software like a pro.

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

### Free

- > Unlimited public/private repositories
- > 2,000 CI/CD minutes/month  
Free for public repositories
- > 500MB of Packages storage  
Free for public repositories
- > 120 core-hours of Codespaces compute

### Get additional student benefits

#### GitHub Pro



#### Protect your branches

Ensure that collaborators on your repository cannot make irrevocable changes to branches.



Draft pull requests



Continue for free










# Dashboard de GitHub



 Dashboard

Search Type / to search



### 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.

Create repository Import repository

## Home

Send feedback Filter

<> Start writing code

### Start a new repository for EdgarGtz-Glz

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name \*

☐ Public  
Anyone on the internet can see this repository

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

Create a new repository



### Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

EdgarGtz-Glz / README.md Create



```
1 - 🙋 Hi, I'm @EdgarGtz-Glz
2 - 👀 I'm interested in ...
3 - 📖 I'm currently learning ...
4 - ❤️ I'm looking to collaborate on ...
5 - 📧 How to reach me ...
6 - 🗨️ Pronouns: ...
7 - ⚡ Fun fact: ...
8
```

### Explore repositories

 renovatebot / renovate 



Home of the Renovate CLI: Cross-platform Dependency Automation by Mend.io

☆ 17.6k TypeScript

 supabase / supabase 

The open source Firebase alternative. Supabase gives you a dedicated Postgres database to build your web, mobile, and AI applications.

☆ 73.7k TypeScript

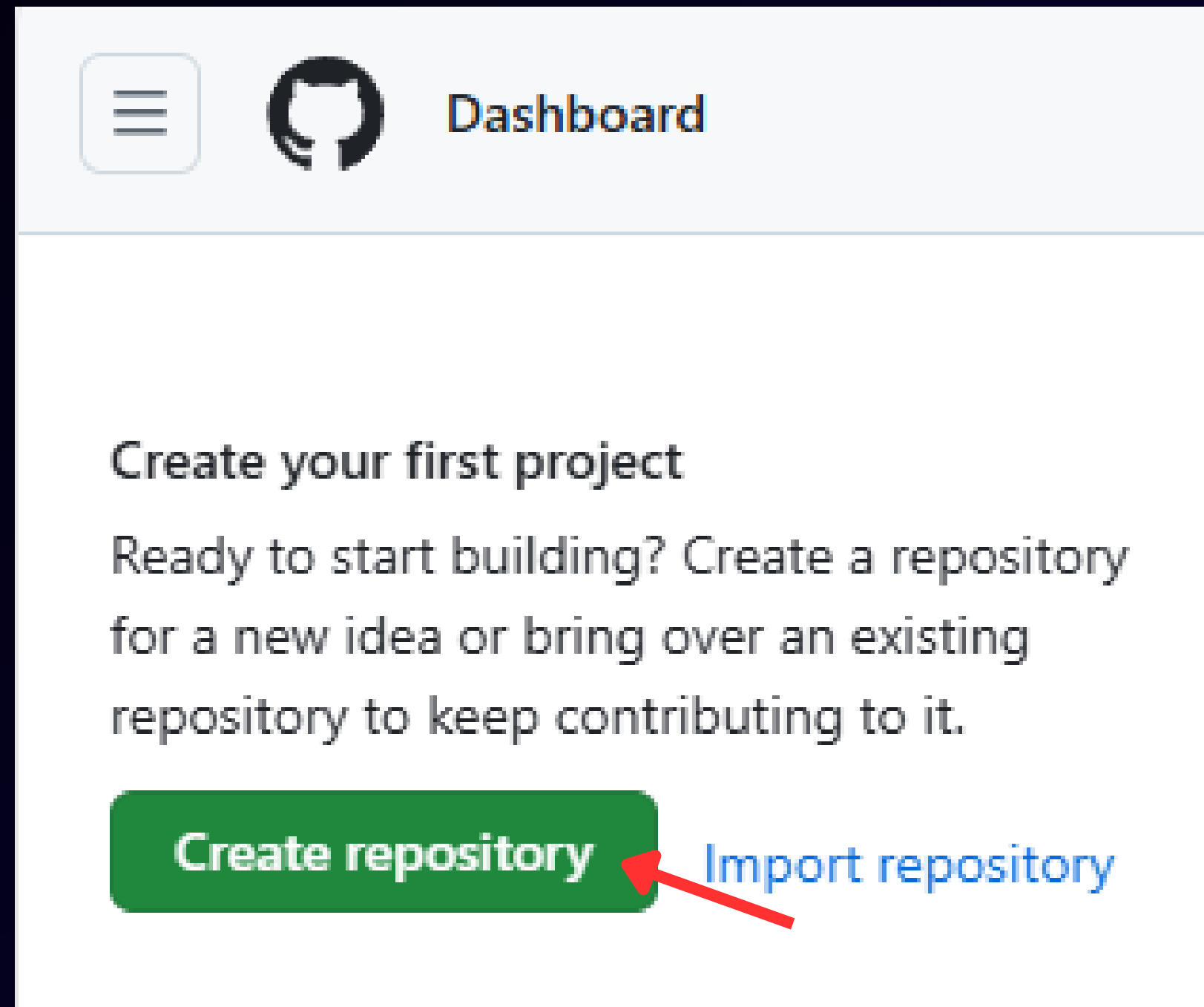
 getsentry / sentry-python 

The official Python SDK for Sentry.io

☆ 1.9k Python

[Explore more →](#)

# Creamos nuestro primer repositorio en GitHub (remoto)



# ¿Cómo sera el repositorio remoto?




## Create a new repository

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

*Required fields are marked with an asterisk (\*).*


Owner \*

Repository name \*

 EdgarGtz-Glz

 / 


miPrimerRepoRemoto


 miPrimerRepoRemoto is available.

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-waffle](#) ?

Description (optional)

Haciendo uso de la tecnología GitHub, para alojar mi primer repositorio remoto :D

☒  Public

☐  Private

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

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

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

Add .gitignore

.gitignore template: None

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

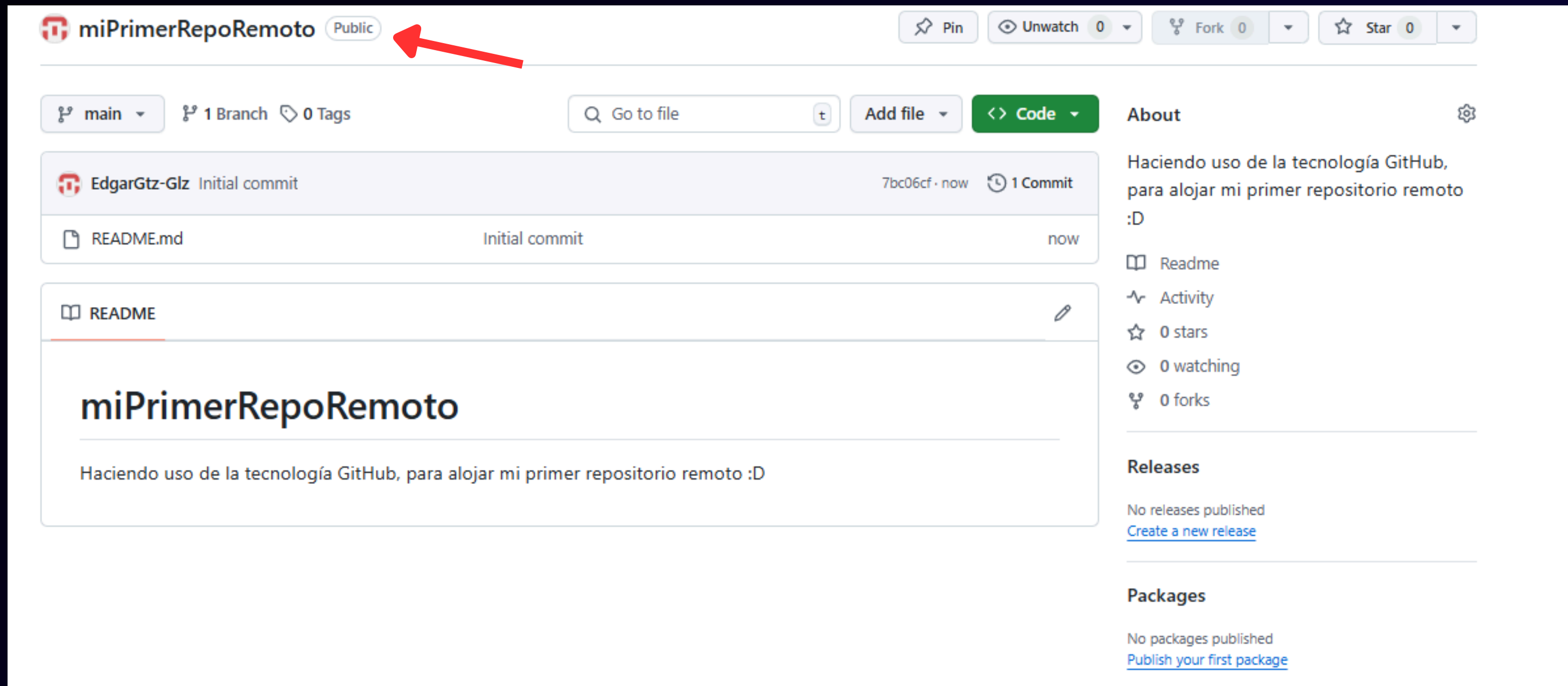
Choose a license

License: None

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

Create repository

# ¡Tenemos nuestro primer repositorio creado!

A screenshot of a newly created GitHub repository page. The repository is named "miPrimerRepoRemoto" and is set to "Public". A red arrow points to the "Public" label. The page shows the repository's initial commit by "EdgarGtz-Glz" with a commit hash of "7bc06cf" and the message "Initial commit". A file named "README.md" is listed. The repository description is "Haciendo uso de la tecnología GitHub, para alojar mi primer repositorio remoto :D". The right sidebar shows repository statistics: 0 stars, 0 watching, and 0 forks. There are also links to "Create a new release" and "Publish your first package".

miPrimerRepoRemoto Public

main 1 Branch 0 Tags

Go to file Add file Code

EdgarGtz-Glz Initial commit 7bc06cf · now 1 Commit

README.md Initial commit now

README

## miPrimerRepoRemoto

Haciendo uso de la tecnología GitHub, para alojar mi primer repositorio remoto :D

About

Haciendo uso de la tecnología GitHub, para alojar mi primer repositorio remoto :D

- Readme
- Activity
- 0 stars
- 0 watching
- 0 forks

Releases

No releases published  
[Create a new release](#)

Packages

No packages published  
[Publish your first package](#)

# Git clone



```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents
$ cd repo-remoto/

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto
$ ls

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto
$ git clone https://github.com/EdgarGtz25/miPrimerRepoRemoto.git
Cloning into 'miPrimerRepoRemoto'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto
$ ls
miPrimerRepoRemoto/

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto
$ cd miPrimerRepoRemoto/

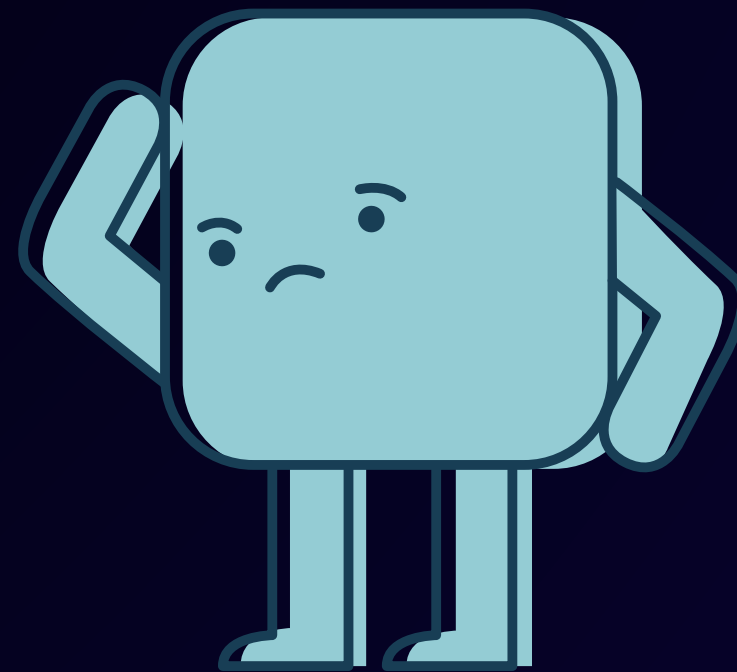
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ |
```

# Práctica V

- Creen un archivo y haganle commit



Mi primer  
repositorio





# Resultado esperado

```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ git status -s
?? archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ git add archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ git status -s
A  archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ git commit -m "Agregue un archivo a mi repo remoto"
[main 7d39099] Agregue un archivo a mi repo remoto
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo.txt

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ git status -s

tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$
```





# Git push


- El comando *git push* se usa para enviar (o "empujar") los cambios de tu repositorio local a un repositorio remoto



```
tuesp@DESKTOP-6KH5F3U MINGW64 ~/Documents/repo-remoto/miPrimerRepoRemoto (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/EdgarGtz25/miPrimerRepoRemoto.git
    7955fe4..7d39099  main -> main
```




# Git push



- Resultado en GitHub







 **miPrimerRepoRemoto** Public

 Pin  Unwatch 1

 **main**  **1 Branch**  **0 Tags**

 Add file  **Code**

 **EdgarGtz25** Agregue un archivo a mi repo remoto 7d39099 · 2 minutes ago  **2 Commits**

 README.md	Initial commit	17 hours ago
 archivo.txt	Agregue un archivo a mi repo remoto	2 minutes ago

# Git pull

- El comando git pull se utiliza para descargar y fusionar los cambios del repositorio remoto a tu repositorio local.

