

Universidad Privada del Norte

Fundamentos de Programación



Curso: Técnicas de programación orientada a objetos.

Docente: Martin Eduardo Torrez Rodriguez

Tema: Guía de Primeros pasos: GitHub

Integrantes:

- Mena Mendoza Andres Eduardo – N00429618

Lima, octubre del 2025

Contenido

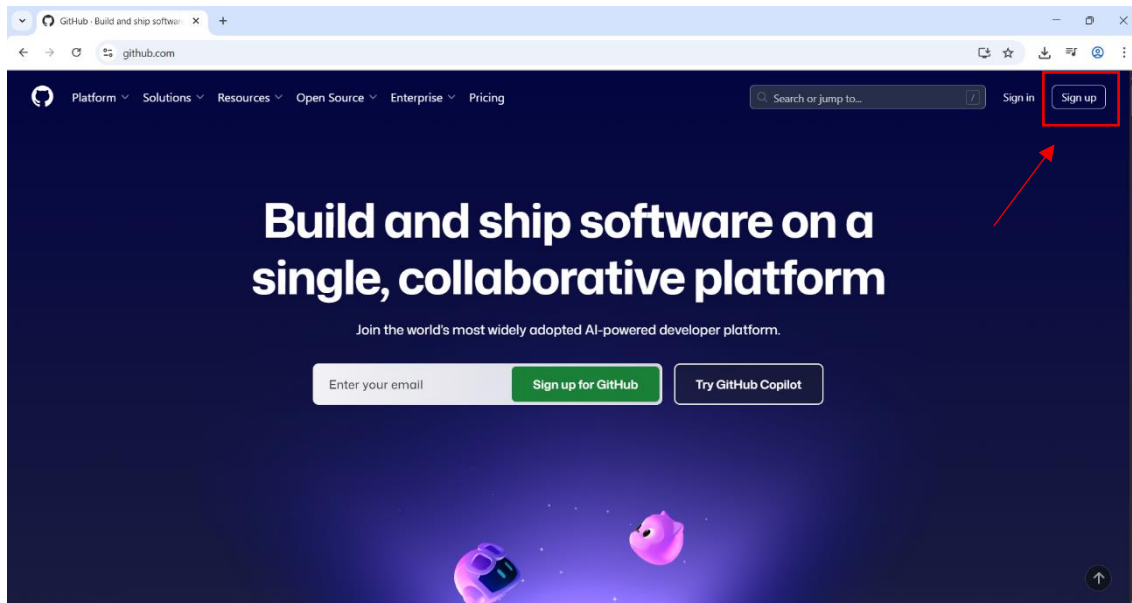
Contenido	2
<i>Guía de Primeros pasos: GitHub</i>	3
Empezaremos con crear una cuenta en GitHub:	3
Creación de un repositorio local	5
Clonar un repositorio de GitHub a la máquina local	6
Navegación básica por el repositorio	7
Primer commit	8
Realizar cambios en el código, Utilizar comandos git add y git commit	10
Visualizar historial de commits	10
Manejo de branches y merges	11
Crear y cambiar entre ramas (branches) utilizando comandos git	14
Branch y git checkout	14
Fusión de branches	14
Cuenta y repositorio en los que se trabajó	14

Guía de Primeros pasos: GitHub

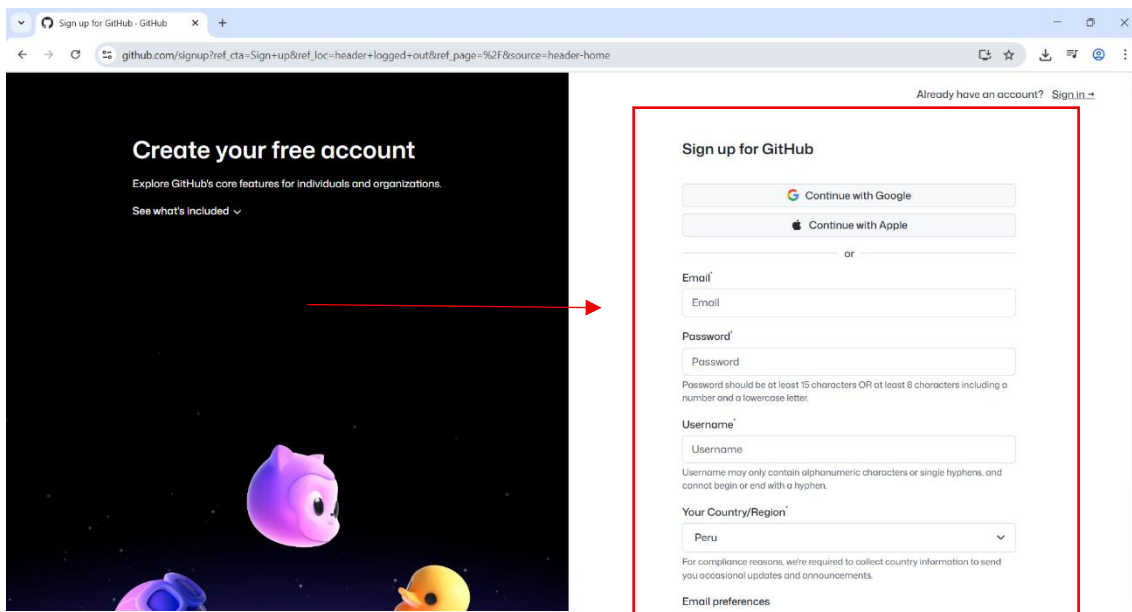
3

Empezaremos con crear una cuenta en GitHub:

1.- Iremos a la Url: Github.com y nos crearemos una cuenta como si fuera una red social.

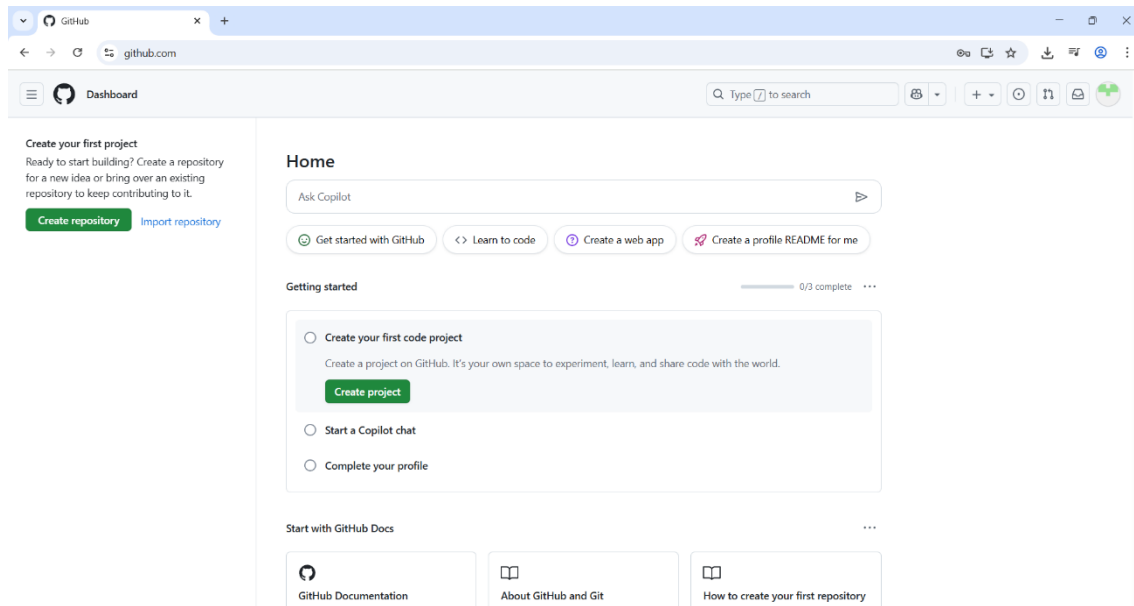


2.- Completamos nuestros datos y confirmamos el código que nos llegará al correo.



3.- Luego de eso ya podremos acceder a nuestro Dashboard y podremos empezar a utilizar las funciones de GitHub

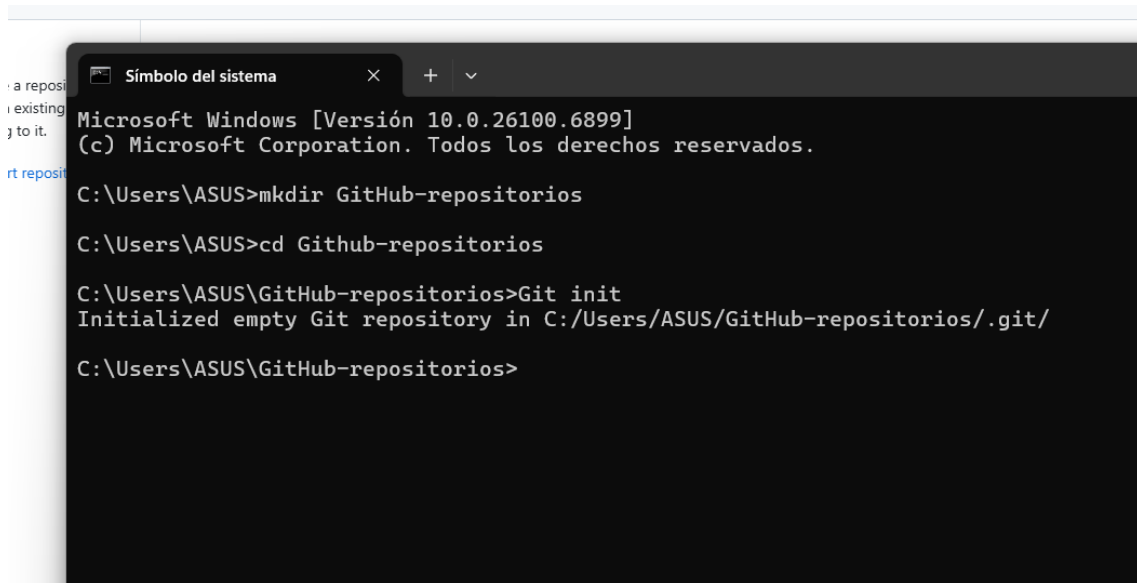
4



Creación de un repositorio local

5

1.- Abriremos la Consola (Cmd), nos dirigiremos a la carpeta donde queramos crear el nuevo repositorio y crearemos el repositorio mediante el comando “Git init”



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ASUS>mkdir GitHub-repositorios

C:\Users\ASUS>cd GitHub-repositorios

C:\Users\ASUS\GitHub-repositorios>Git init
Initialized empty Git repository in C:/Users/ASUS/GitHub-repositorios/.git/

C:\Users\ASUS\GitHub-repositorios>
```

Nota:

El comando “mkdir” crea una nueva carpeta en el directorio actual, en este caso creamos la carpeta “GitHub-repositorios”

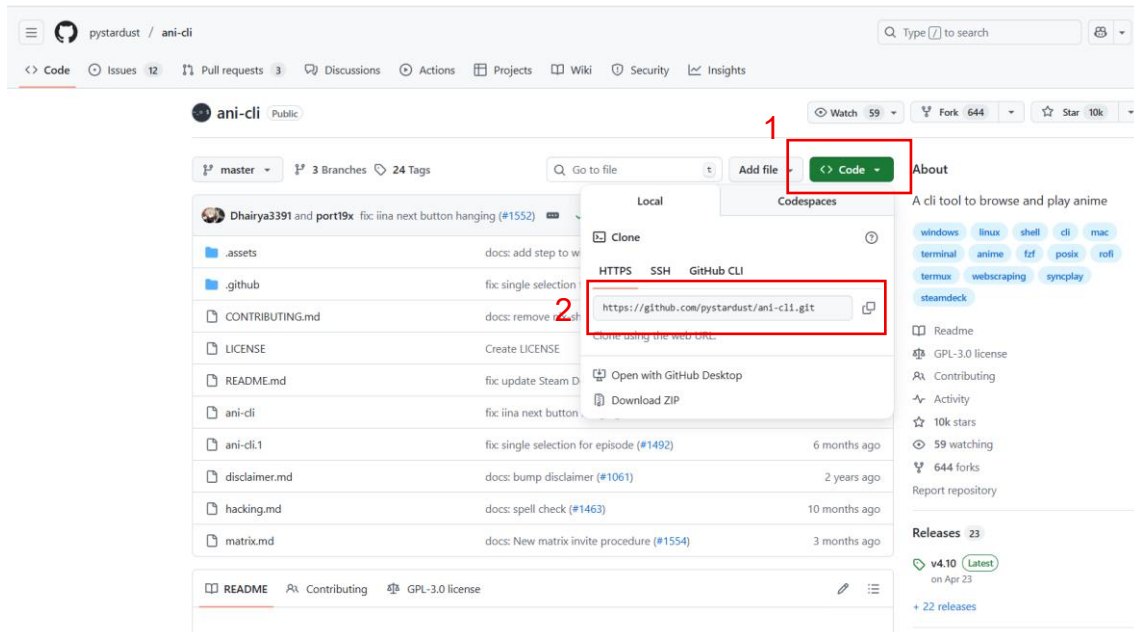
El comando “cd” se utiliza para cambiar de directorio, aca lo utilizamos para abrir el directorio que acabamos de crear llamado “GitHub-repositorios”

El comando “git init” se utiliza para crear el repositorio en el directorio actual.

Clonar un repositorio de GitHub a la máquina local

6

1.- Buscamos un repositorio que nos interese en GitHub y seleccionamos en Code (1), y copiamos la Url (2).



2.- Abrimos la consola (CMD), nos dirigimos al directorio donde queremos clonar el repositorio y con el comando “git clone + (url copiado)” procedemos a clonar el repositorio.

```

Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ASUS>mkdir GitHub-repositorios

C:\Users\ASUS>cd GitHub-repositorios

C:\Users\ASUS\GitHub-repositorios>Git init
Initialized empty Git repository in C:/Users/ASUS/GitHub-repositorios/.git/

C:\Users\ASUS\GitHub-repositorios>git clone https://github.com/pystardust/ani-cli.git
Cloning into 'ani-cli'...
remote: Enumerating objects: 2585, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 2585 (delta 22), reused 10 (delta 10), pack-reused 2558 (from 3)
Receiving objects: 100% (2585/2585), 11.72 MiB | 964.00 KiB/s, done.
Resolving deltas: 100% (1577/1577), done.

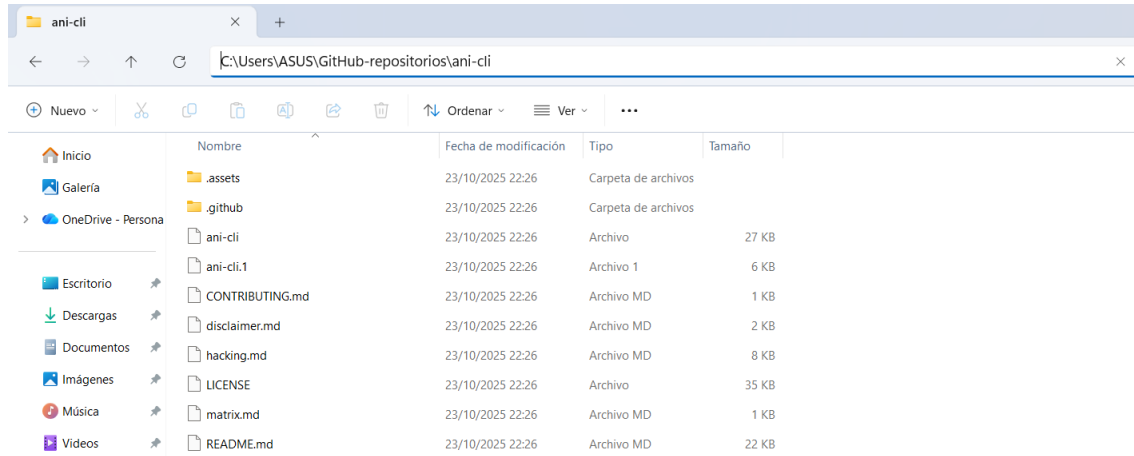
C:\Users\ASUS\GitHub-repositorios>
    
```

Nota: el comando “git clone + (url del repositorio a clonar)” clona el repositorio en el directorio actual.

Navegación básica por el repositorio

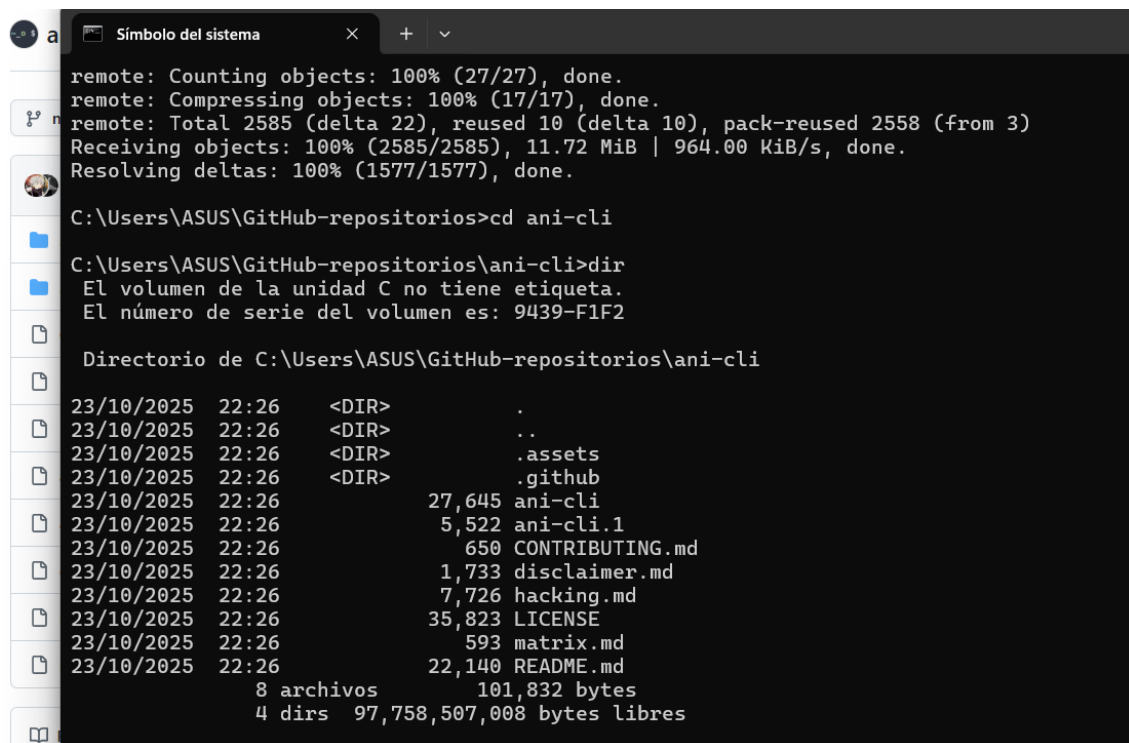


Podemos utilizar el explorador de archivos tradicional para navegar por el repositorio clonado.



-- O --

Podemos usar la consola (CMD) y mediante comandos navegar en el repositorio.



Nota:

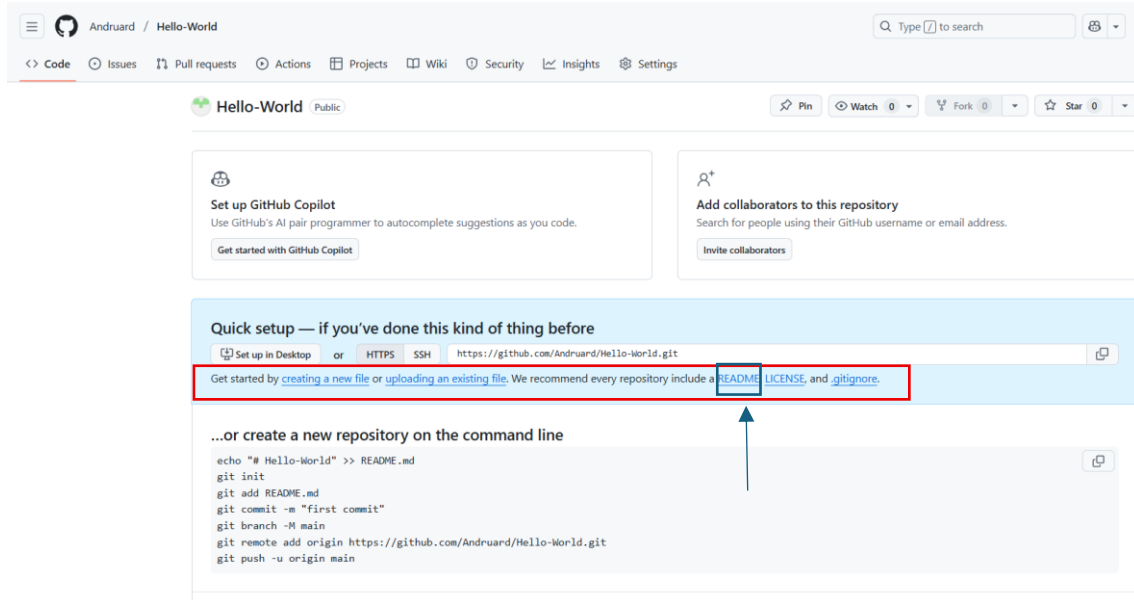
El comando “dir” muestra el contenido del directorio actual en este caso se usa dentro del directorio de “ani-cli” repositorio clonado recientemente.

Primer commit

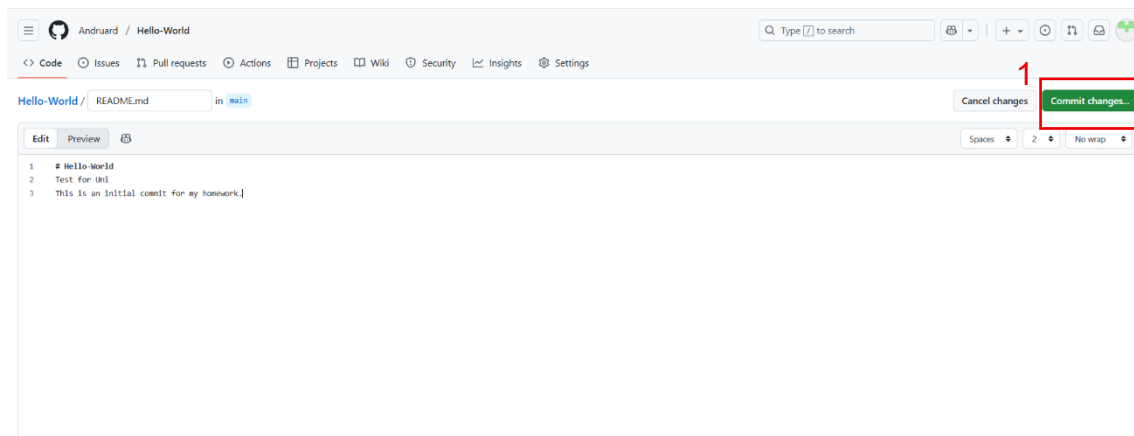
8

Para nuestro primer commit podríamos agregar un readme a un repositorio recién creado, en este caso he creado un repositorio llamado “Hello-World”.

1.- Le daremos al botón de “Readme” es una de las opciones que GitHub nos da para crear nuestro primer archivo en el repositorio.

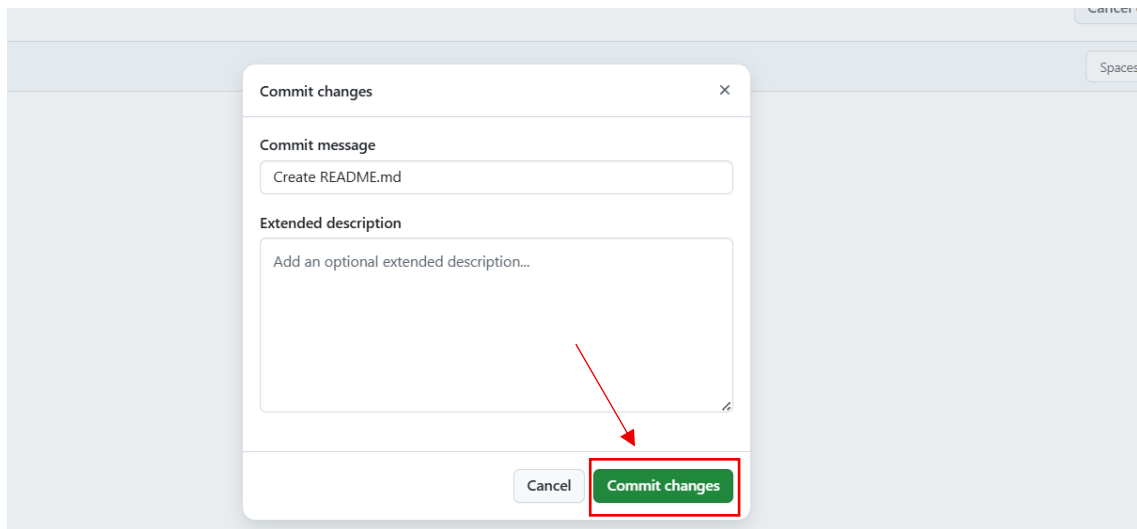


2.- Agregamos texto para nuestro “Readme” y luego le damos a “commit changes” (1).

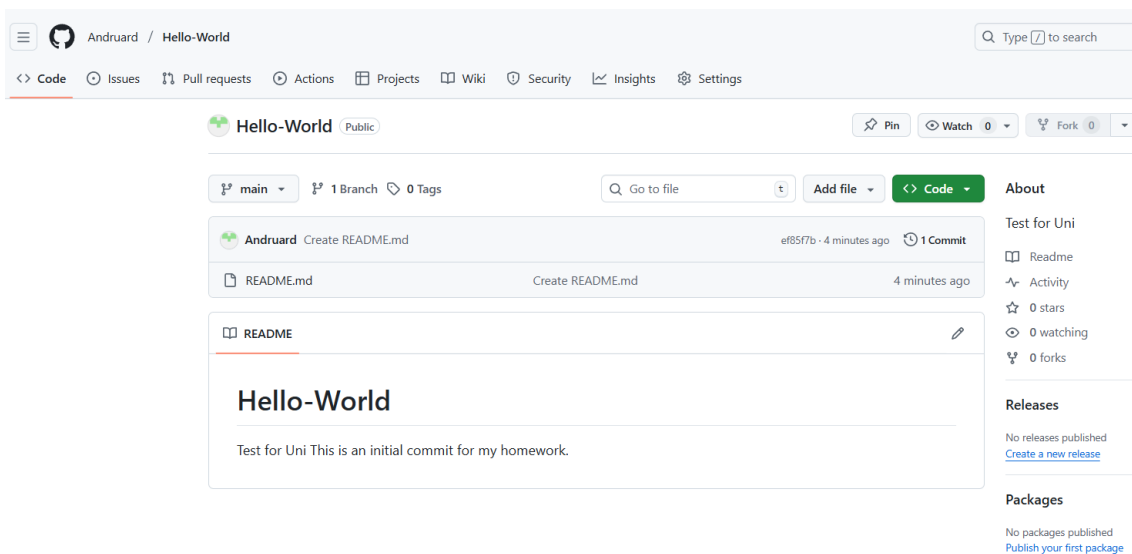


3.- Colocamos un “commit message” donde se colocho lo que realiza el commit, en este caso nuestro commit crea un readme entonces “Create README.md” y luego le damos a “Commit changes”

9



4.- ahora tenemos un archivo Readme en nuestro repositorio.



Realizar cambios en el código, Utilizar comandos git add y git commit

10

1.- Se procede a usar el comando “git add” y “git commit” para agregar el archivo README.md al repositorio local

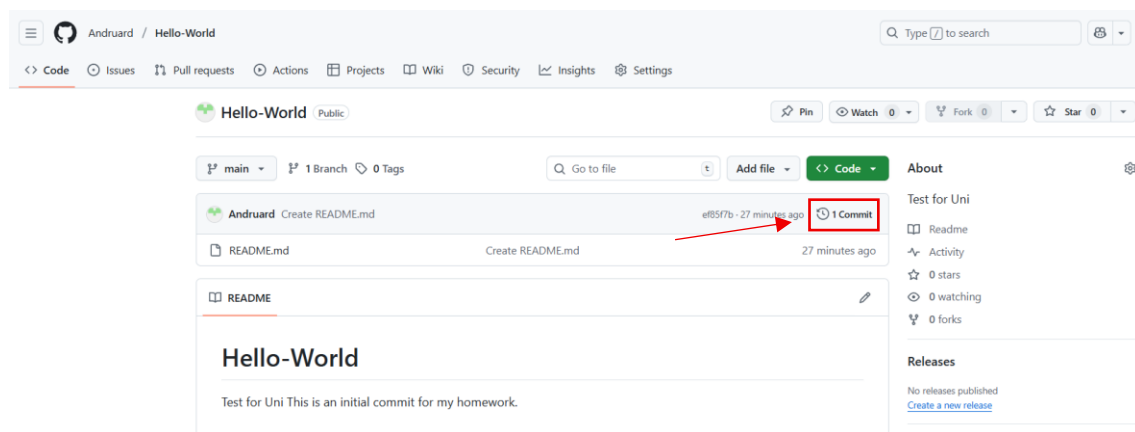
```
ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ git commit -m "Create README.md"
[main (root-commit) f579f3b] Create README.md
1 file changed, 3 insertions(+)
create mode 100644 README.md

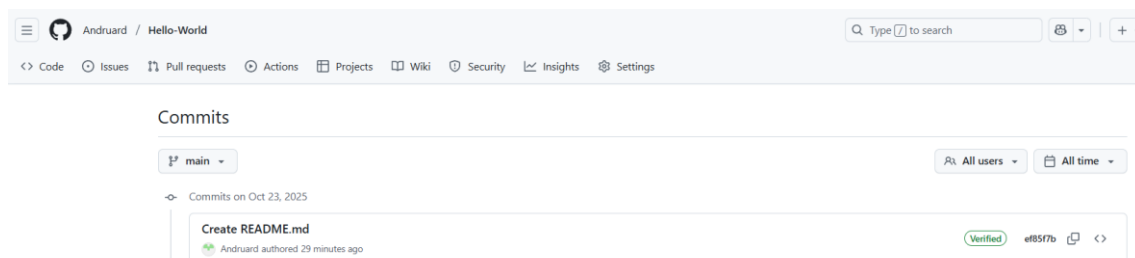
ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ |
```

Visualizar historial de commits

1.- Iremos a la página principal del repositorio y seleccionaremos la parte donde dice “commits”.



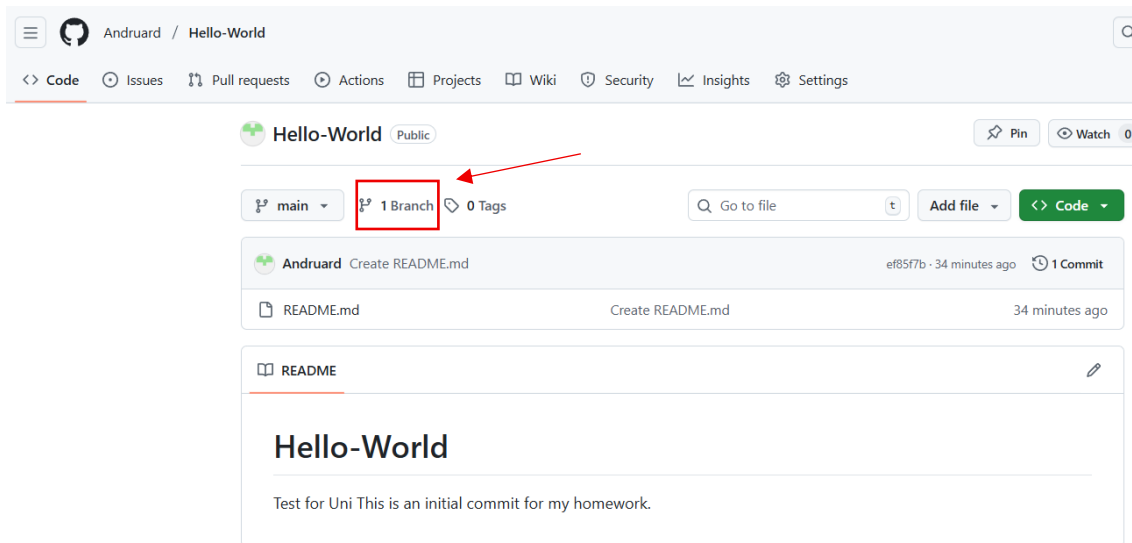
2.- Aquí podemos ver el historial de commits.



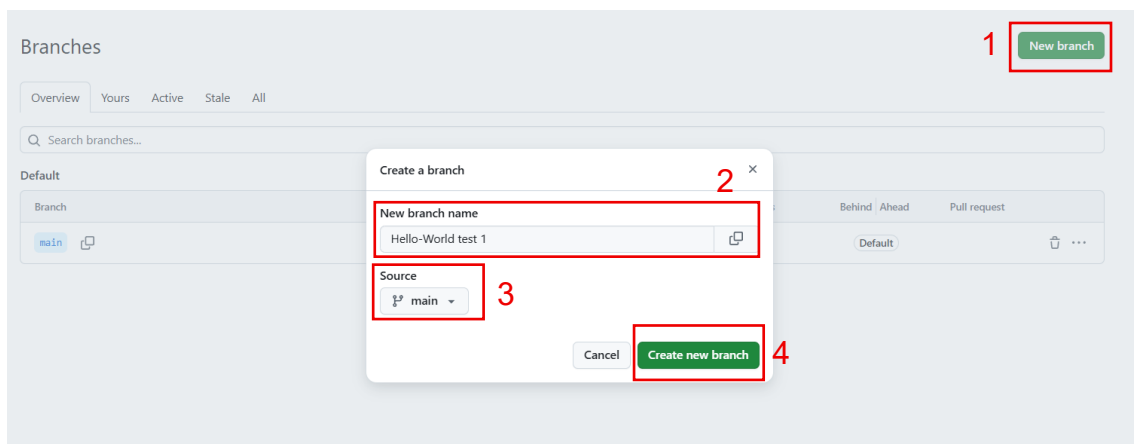
Manejo de branches y merges

11

1.- para crear una Branch debemos de ir a la opción de “Branches” en la pagina principal del repositorio deseado.



2.- Le damos a New Branch (1) y luego colocamos el nombre (2) y seleccionamos de donde queremos que derive en este caso será Main (3), por ultimo le damos a “Create new Branch” (4).



3.- luego de tener nuestra Branch ahora haremos un “pull request” para seguir con un merge y modificar el Main.

12

Branches

Overview Yours Active Stale All

Search branches...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	44 minutes ago				Default

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
Hello-world-test-1	1 minute ago		0	1	

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
Hello-world-test-1	1 minute ago		0	1	

New pull request

- Activity
- View rules
- Rename branch

4.- Colocamos un titulo y creamos el pull request

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about d](#)

base: main ← compare: Hello-World-test-1 ✓ Able to merge. These branches can be automatically merged.

Add a title

Update README.md

Add a description

Write Preview

Add your description here...

Markdown is supported Paste, drop, or click to add files

Create pull request

5.- luego nos dirigimos a la pestaña de “Pull requests” (1) y aquí podremos hacer merge (2).

13

The screenshot shows the GitHub interface for a pull request titled "Update README.md #1". The repository is "Andruard / Hello-World". The "Pull requests" tab is selected and highlighted with a red box and a red number "1". Below the title, it says "Andruard wants to merge 1 commit into main from Hello-World-test-1". The pull request details show a conversation with a comment from "Andruard" stating "No description provided." and a commit "Update README.md" which is verified. A green box with a checkmark indicates "No conflicts with base branch" and "Merging can be performed automatically." Below this, a green button labeled "Merge pull request" is highlighted with a red box and a red number "2". At the bottom right, there is a link "Still in progress? Convert to draft".

6.- Luego de hacer Merge la Branch queda inútil ya que fue agregada a la Main y puede ser borrada sin problemas.

The screenshot shows the same GitHub pull request after it has been successfully merged. The status at the top now says "Merged" in a purple box. The pull request details show the same conversation and commit. Below the commit, a message states "Andruard merged commit 7e332f3 into main now". At the bottom, a purple box with a checkmark indicates "Pull request successfully merged and closed" and "You're all set — the Hello-World-test-1 branch can be safely deleted." A button labeled "Delete branch" is visible next to this message.

Crear y cambiar entre ramas (branches) utilizando comandos git Branch y git checkout

14

1.- podemos usar el comando “git branch + (nombre de la rama)” para crear una nueva rama y el comando “git checkout + (nombre de la rama)” para cambiar a esta.

```
ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (hwtest1)
$ git checkout main
Switched to branch 'main'
Your branch and 'origin/main' have diverged,
and have 1 and 3 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with yours)

ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ git branch Hello-world-test1

ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ git checkout Hello-world-test1
Switched to branch 'Hello-world-test1'

ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (Hello-world-test1)
$
```

Fusión de branches

1.-podemos usar el comando “git merge + (el nombre de la rama a fusionar)” para fusionar ramas.

```
ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ git checkout main
D      README.md
Already on 'main'
Your branch and 'origin/main' have diverged,
and have 1 and 3 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with yours)

ASUS@DESKTOP-AH8CNTQ MINGW64 ~/GitHub-repositorios/Hello-World (main)
$ git merge Hello-world-test1
Already up to date.
```

Cuenta y repositorio en los que se trabajó

Github: <https://github.com/Andruard>

Repositorio: <https://github.com/Andruard/Hello-World>