	<p style="text-align: center;">UNIVERSIDAD DON BOSCO ESCUELA DE COMPUTACIÓN</p>
<p style="text-align: center;">CICLO II</p>	<p style="text-align: center;">GUIA DE LABORATORIO #1 Desarrollo de Aplicaciones con Frameworks Introducción a Git</p>

I. OBJETIVOS.

- Que el estudiante conozca más sobre Git y GitHub Desktop.
- Que el estudiante se familiarice con el uso de Git para utilizar software de control de versiones.

II. INTRODUCCIÓN.

¿Qué es GIT?

Git es un sistema de control de versiones desarrollado para el seguimiento, control e historial de cambios en diferentes archivos, especialmente en los productos o desarrollos de software o relacionados con TI, pues proporciona un alto control sobre los cambios o actualizaciones en los equipos de trabajo.

El seguimiento de cambios de Git es una de las principales razones de uso puesto que permite resolver conflictos dentro de los desarrollos, deshacer cambios importantes, combinar el código de manera controlada entre otros.

Otra de las características de Git es que permite la creación de ramas que permiten dividir el flujo de trabajo entre varios equipos y llevar en paralelo el desarrollo de diferentes funcionalidades de manera simultánea.

¿Qué es Github?

GitHub es una plataforma basada en la nube diseñada para almacenar, gestionar y colaborar en proyectos utilizando Git como sistema de control de versiones. Facilita el trabajo en equipo mediante herramientas como repositorios remotos, revisión de cambios y fusión controlada del código.

¿Qué es Github Desktop?

GitHub Desktop es una aplicación de escritorio que simplifica el uso de Git y GitHub mediante una interfaz gráfica amigable. Está diseñada para gestionar repositorios sin necesidad de usar la línea de comandos, permitiendo realizar acciones como clonar repositorios, realizar commits, crear ramas, resolver conflictos y sincronizar cambios con GitHub de manera intuitiva.

Como se muestra en la Figura 1, las ramas de Git permiten un flujo de trabajo paralelo eficiente.



Figura 1. Diagrama sobre ramas en Git. Adaptado de "Git Basics: Branches", por Git SCM, s.f., Git SCM Documentation, <https://git-scm.com/images/about/branches@2x.png>.

Como se ilustra en la Figura 2, Git facilita la colaboración a través de un flujo de trabajo distribuido.

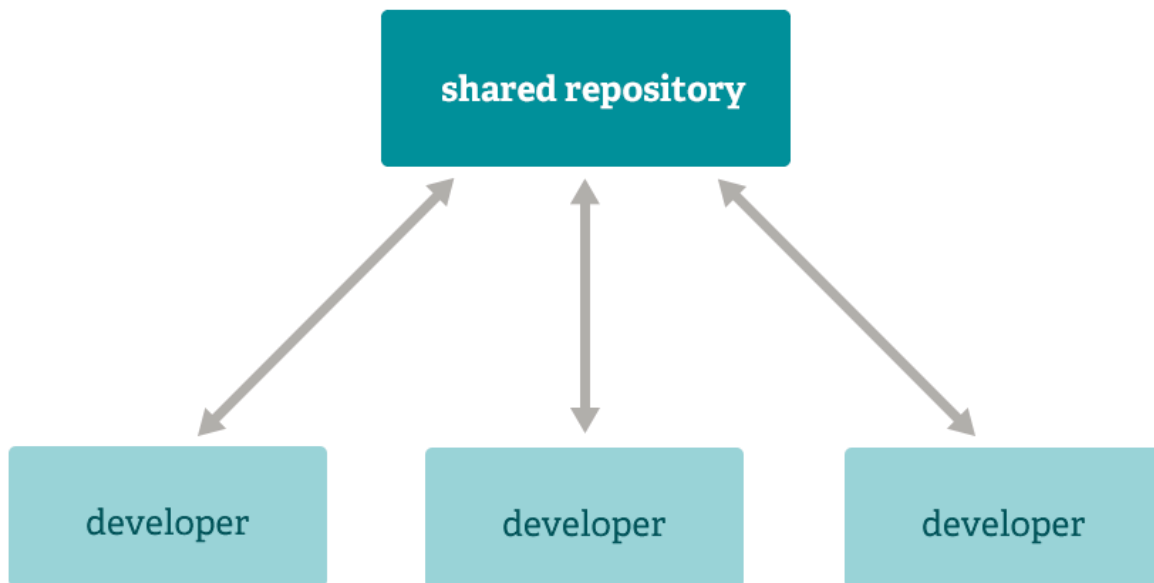


Figura 2. Flujo de trabajo distribuido en Git. Adaptado de "Git Basics: Workflow", por Git SCM, s.f., Git SCM Documentation, <https://git-scm.com/images/about/workflow-a@2x.png>.

III. Procedimiento.

Deberemos instalar Git en nuestros equipos y esto lo encontraremos en el sitio oficial de Git ([Git - Downloading Package](#)).

Download for Windows

[Click here to download](#) the latest (2.47.1(2)) 64-bit version of **Git for Windows**. This is the most recent maintained build. It was released **10 days ago**, on 2025-01-14.

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.](#)

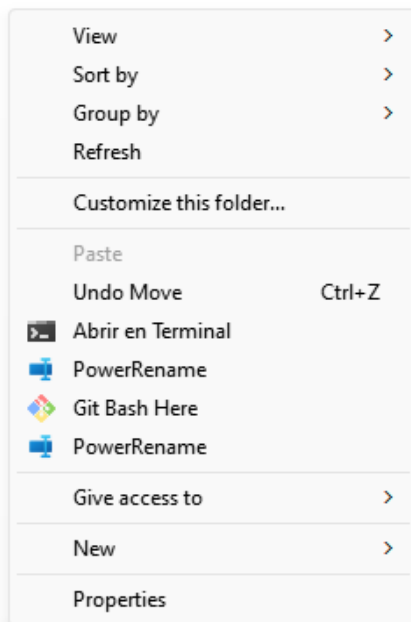
[64-bit Git for Windows Portable.](#)

Using winget tool

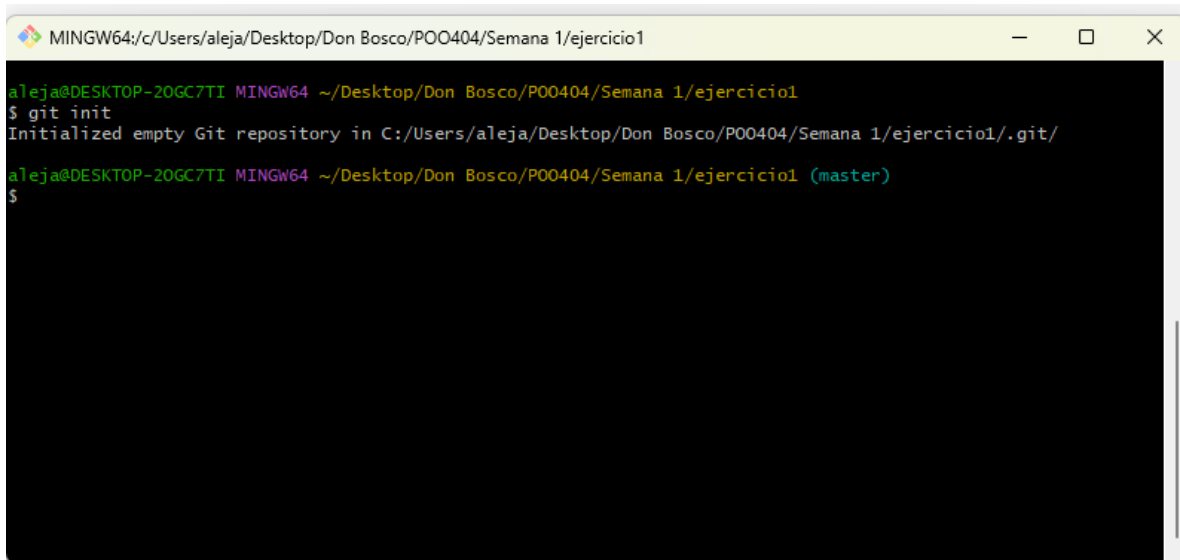
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

Una vez instalado Git iremos a crear una carpeta en **Documents** o **Desktop**, la llamaremos POO-{nuestro-carnet} y daremos click derecho dentro de la carpeta creada para abrir una consola de Git.

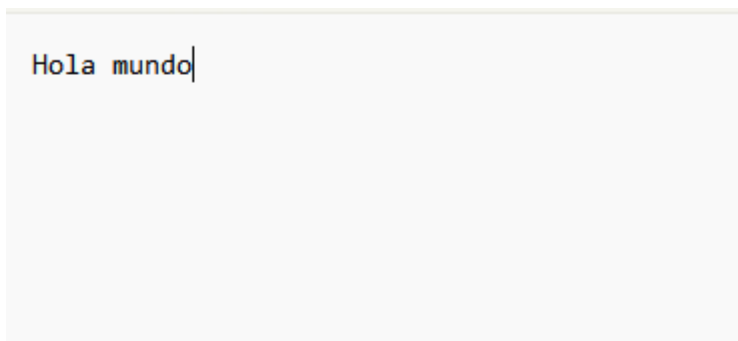


E iniciaremos un repositorio de Git con el comando **git init**.

A screenshot of a terminal window with a yellow title bar. The title bar text is "MINGW64:/c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1". The terminal has a black background with green and white text. The text shows the user "aleja@DESKTOP-20GC7TI" in a "MINGW64" shell at the directory "~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1". They entered the command "\$ git init", and the output was "Initialized empty Git repository in C:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1/.git/". The prompt then changed to "aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)" followed by a new "\$" prompt.

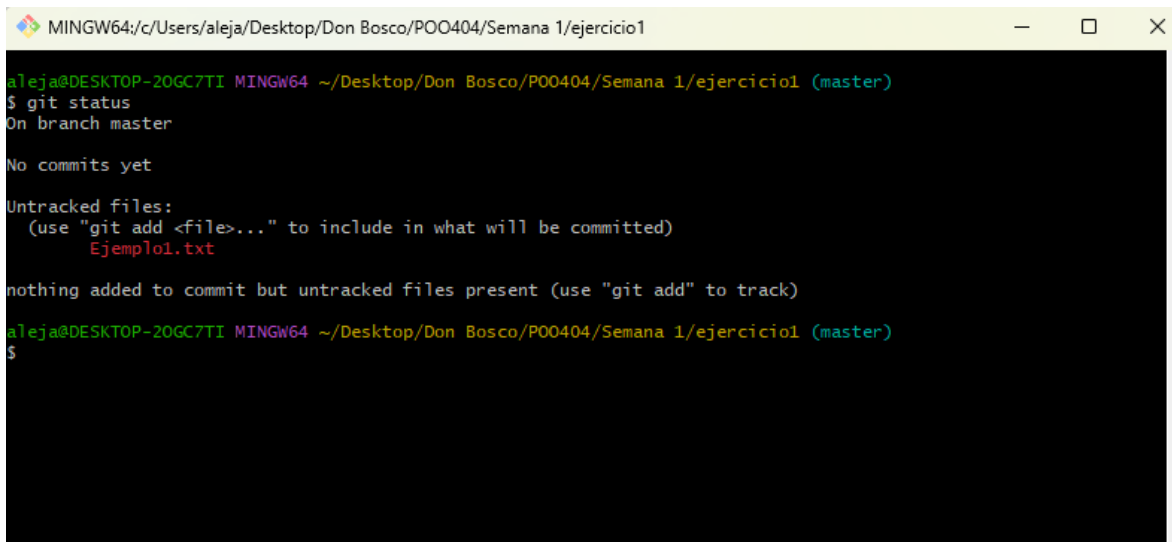
```
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1
$ git init
Initialized empty Git repository in C:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1/.git/
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$
```

Crearemos un archivo llamado **Ejemplo1** dentro de la carpeta que creamos y escribiremos dentro **Hola mundo**.

A screenshot of a text editor with a light gray background. The text "Hola mundo" is written in a monospaced font, and a vertical cursor is positioned at the end of the text.

```
Hola mundo|
```

Ahora utilizaremos el comando **git status** para ver los archivos nuevos o modificados dentro de nuestro repositorio.

A terminal window titled 'MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1'. The prompt is 'aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)'. The user enters '\$ git status'. The output shows 'On branch master', 'No commits yet', and 'Untracked files: (use "git add <file>..." to include in what will be committed) Ejemplo1.txt'. A message follows: 'nothing added to commit but untracked files present (use "git add" to track)'. The prompt returns to '\$'.

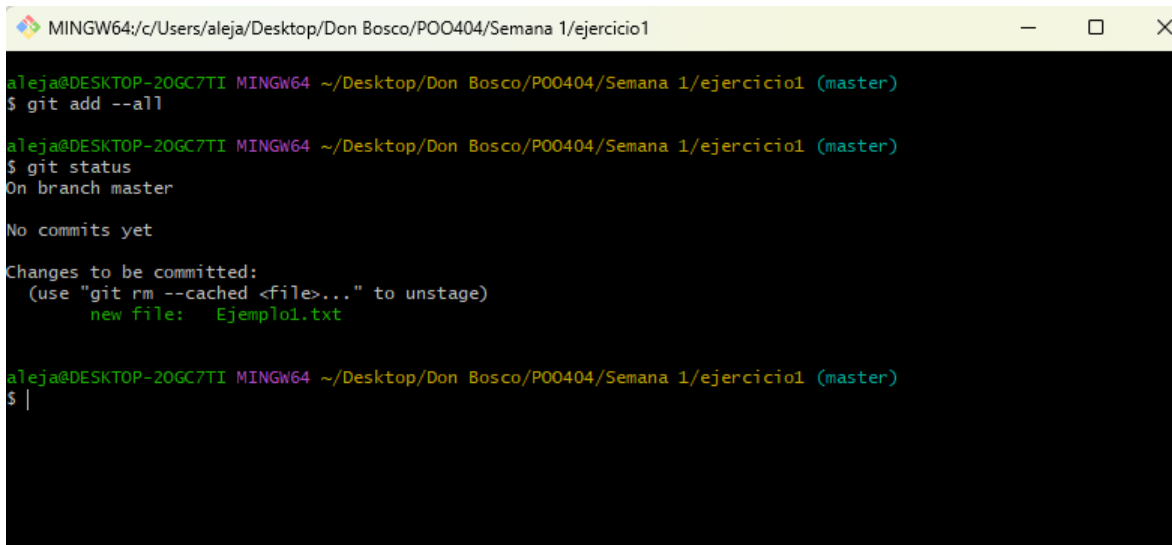
```
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Ejemplo1.txt

nothing added to commit but untracked files present (use "git add" to track)
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$
```

Para guardar los cambios utilizaremos el comando **git add --all**.

A terminal window titled 'MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1'. The prompt is 'aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)'. The user enters '\$ git add --all'. The prompt returns to '\$'. The user enters '\$ git status'. The output shows 'On branch master', 'No commits yet', and 'Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: Ejemplo1.txt'. The prompt returns to '\$'.

```
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git add --all
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Ejemplo1.txt

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ |
```

Utilizaremos el comando **git commit -m**"agregando archivo Ejemplo1".

```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git commit -m"Agregando archivo Ejemplo1"
[master (root-commit) 0723daa] Agregando archivo Ejemplo1
1 file changed, 1 insertion(+)
 create mode 100644 Ejemplo1.txt

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git status
On branch master
nothing to commit, working tree clean

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$
```

Crearemos una nueva rama que luego combinaremos con nuestra rama master y así agregar de forma controlada nuestros cambios. Utilizaremos el comando **git branch rama1**.

```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git branch rama1

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git branches
git: 'branches' is not a git command. See 'git --help'.

The most similar command is
    branch

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git branch
* master
  rama1

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ |
```

Para movernos a esa nueva rama utilizaremos el comando **git checkout rama1**.

```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git checkout rama1
Switched to branch 'rama1'

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (rama1)
$ git branch
* master
  rama1

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (rama1)
$ |
```

Ahora modificaremos nuestro archivo **Ejemplo1** y agregaremos nuevos datos.

Hola mundo

Soy el ingeniero Alejandro Meléndez|

Ahora guardamos los cambios agregados en la **rama1**.

```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (rama1)
$ git status
On branch rama1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Ejemplo1.txt

no changes added to commit (use "git add" and/or "git commit -a")
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (rama1)
$ git add --all

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (rama1)
$ git status
On branch rama1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Ejemplo1.txt

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (rama1)
$
```


Y hacemos un **commit** para crear la versión de nuestro rama.

```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (ramal)
$ git commit -m"Agregada data al archivo"
[ramal 3b763a4] Agregada data al archivo
1 file changed, 3 insertions(+), 1 deletion(-)

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (ramal)
$ git status
On branch ramal
nothing to commit, working tree clean

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (ramal)
$
```

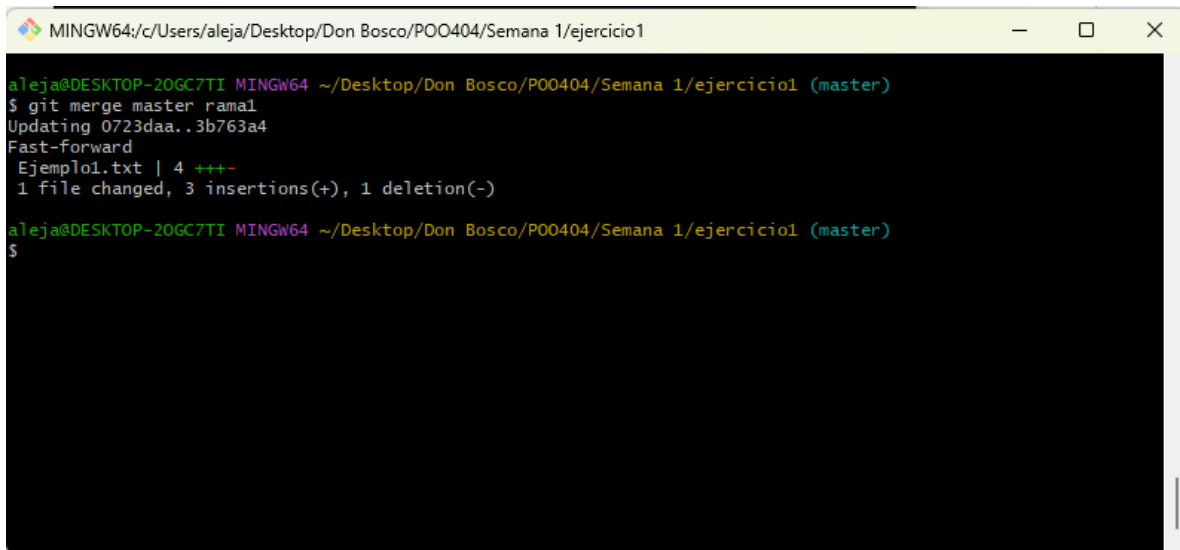
Para combinar las ramas primero deberemos movernos a la rama **master**.

```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (ramal)
$ git checkout master
Switched to branch 'master'

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git branch
* master
  ramal

aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ |
```

Y aquí utilizaremos el comando **git merge master rama1**.



```
MINGW64; c:/Users/aleja/Desktop/Don Bosco/POO404/Semana 1/ejercicio1
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$ git merge master rama1
Updating 0723daa..3b763a4
Fast-forward
 Ejemplo1.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
aleja@DESKTOP-20GC7TI MINGW64 ~/Desktop/Don Bosco/POO404/Semana 1/ejercicio1 (master)
$
```

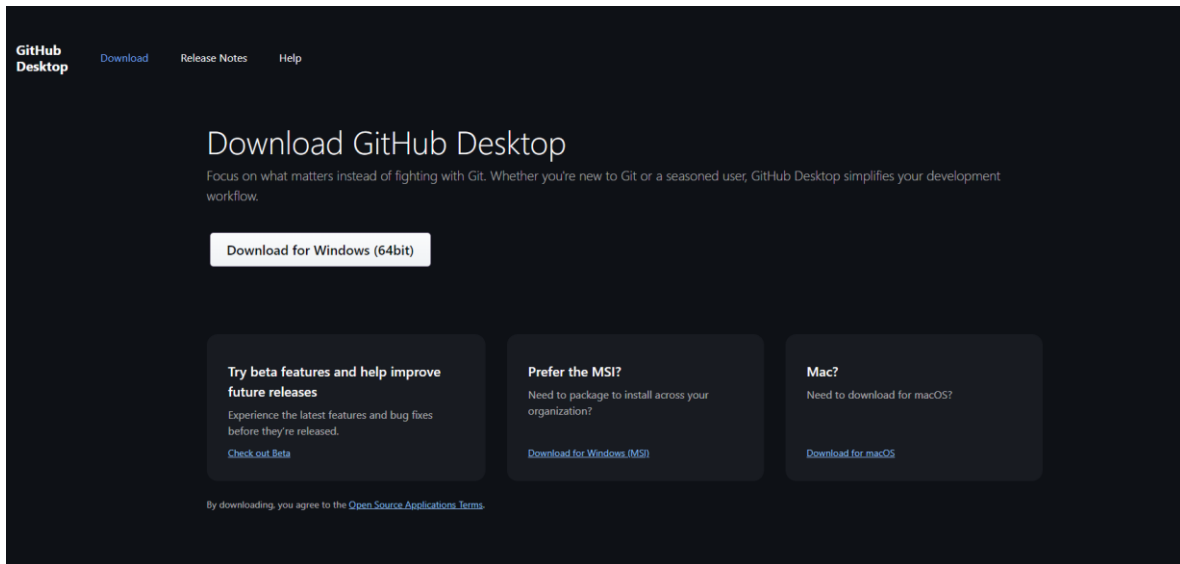
Explicación de comandos utilizados.

- **git init:** Inicializa un nuevo repositorio Git en el directorio actual, permitiendo rastrear cambios en los archivos.
- **git status:** Muestra el estado del repositorio, indicando archivos nuevos, modificados o eliminados.
- **git add --all:** Agrega todos los archivos nuevos o modificados al área de preparación para el próximo commit.
- **git commit -m "mensaje":** Guarda los cambios preparados en el historial del repositorio junto con un mensaje descriptivo.
- **git branch <nombre-de-la-rama>:** Crea una nueva rama a partir del estado actual del repositorio.
- **git checkout <nombre-de-la-rama>:** Cambia el contexto de trabajo a la rama especificada.
- **git merge <rama>:** Fusiona los cambios de una rama en la rama actual, integrando su contenido.

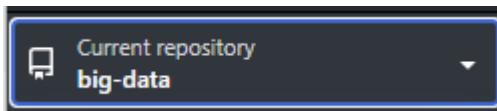
Ejercicio 2:

Utilizar Github Desktop

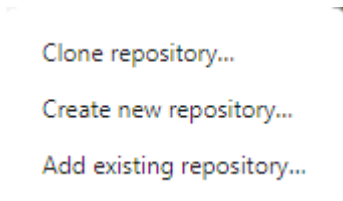
Lo primero que haremos será descargar la aplicación de Github Desktop ([Download GitHub Desktop | GitHub Desktop](#)).



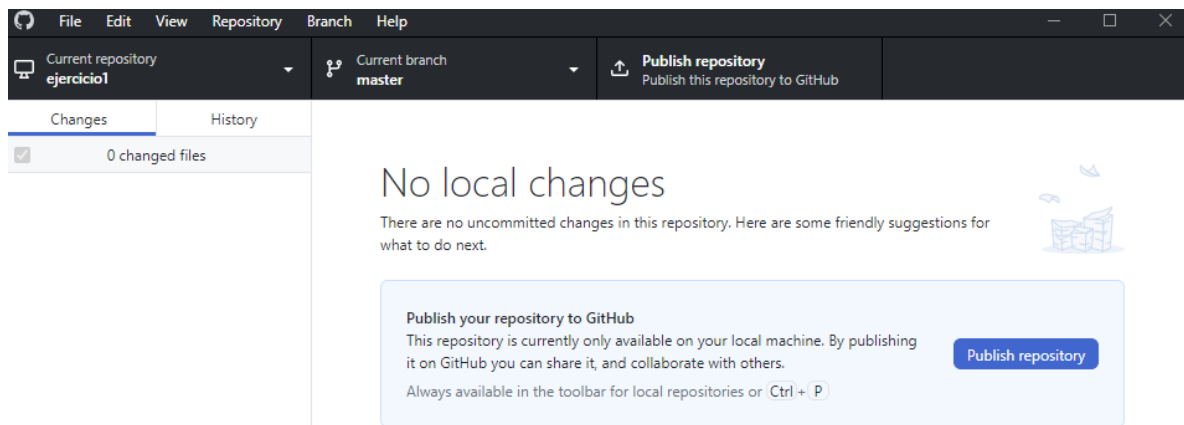
En Github Desktop iremos y daremos click en el repositorio actual.



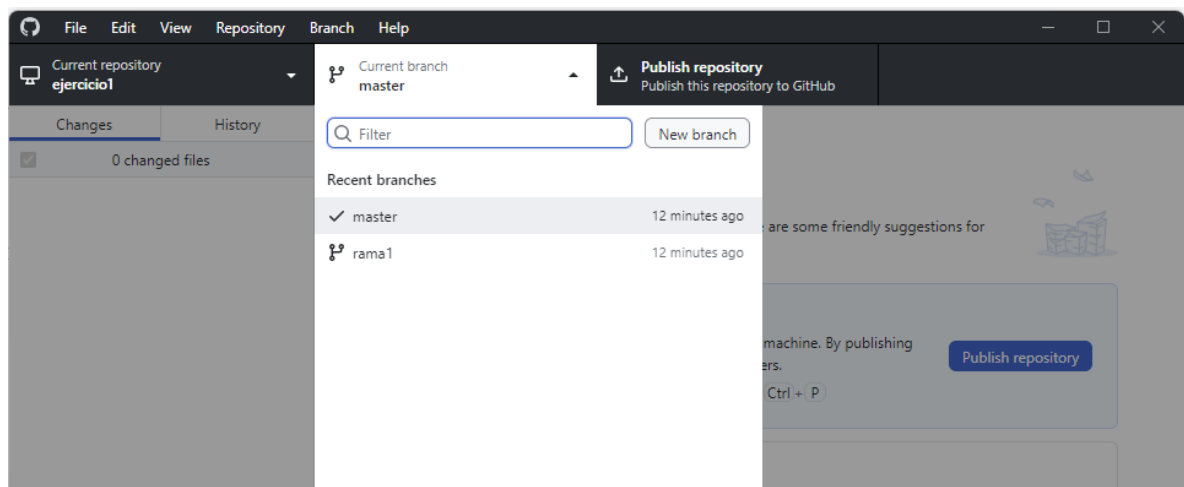
Estos nos abrirá un lista de opciones donde podremos agregar un repositorio existente.



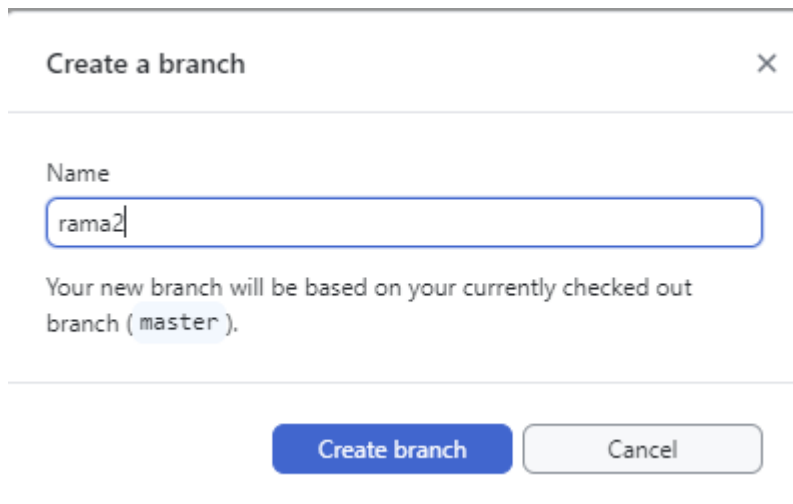
Agregaremos el repositorio que creamos anteriormente.



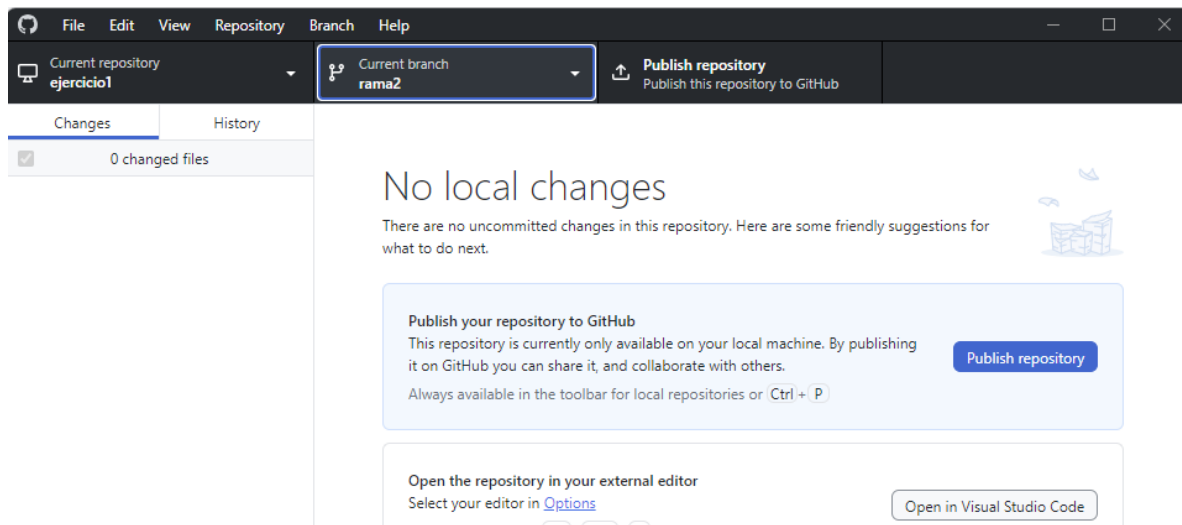
Crearemos una rama llamada **rama2** dando click en **current branch** y click en **New branch**.



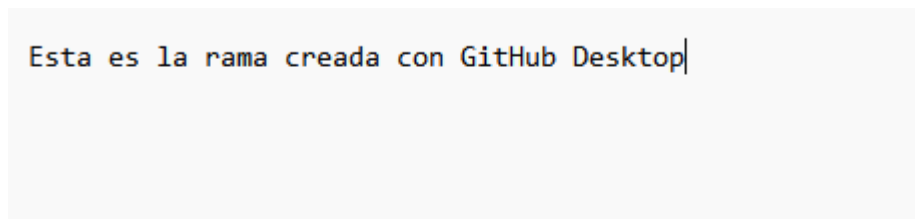
Colocaremos el nombre de la rama y daremos click en **Create branch**.



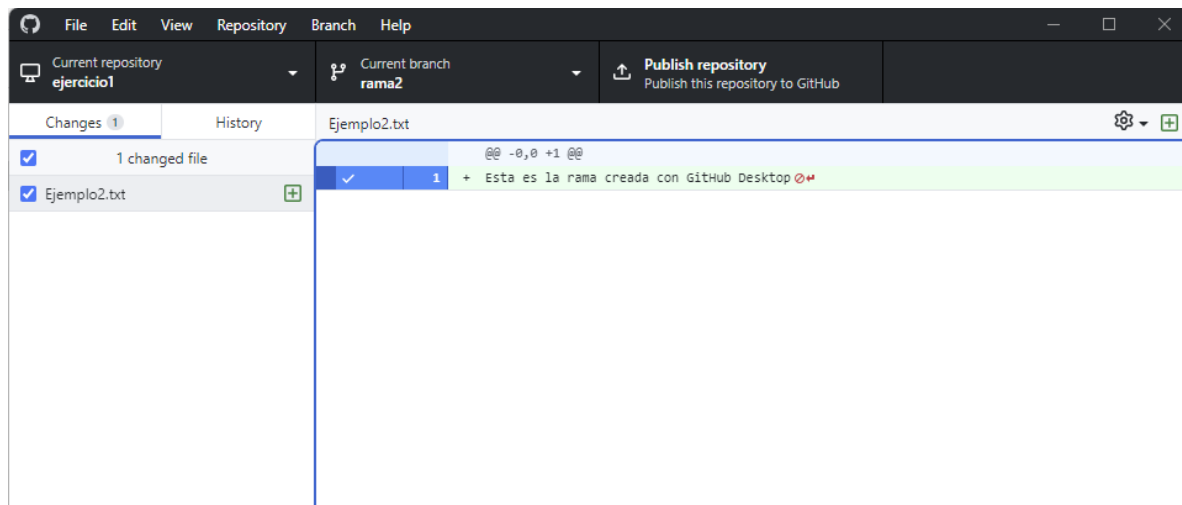
Ahora ya estamos en la **rama2**.




Agregaremos un nuevo archivo llamado **Ejemplo2**.



Y ahora podremos ver los cambios realizados en la interfaz de Github Desktop.



Para guardar los cambios dejaremos un mensaje y daremos click en **commit to rama2**.

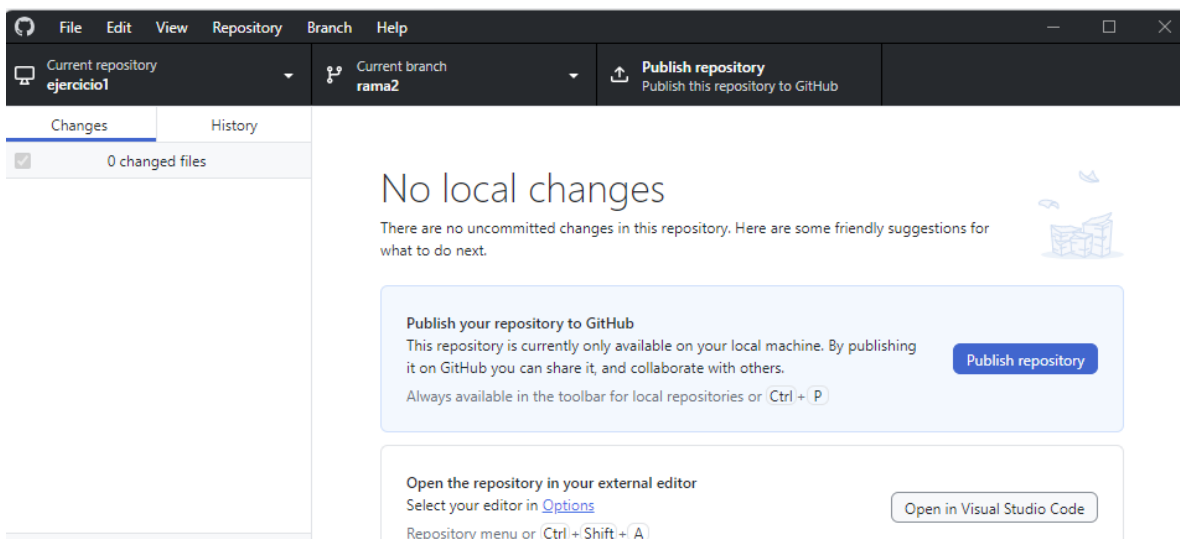


Archivo Ejemplo2 Creado

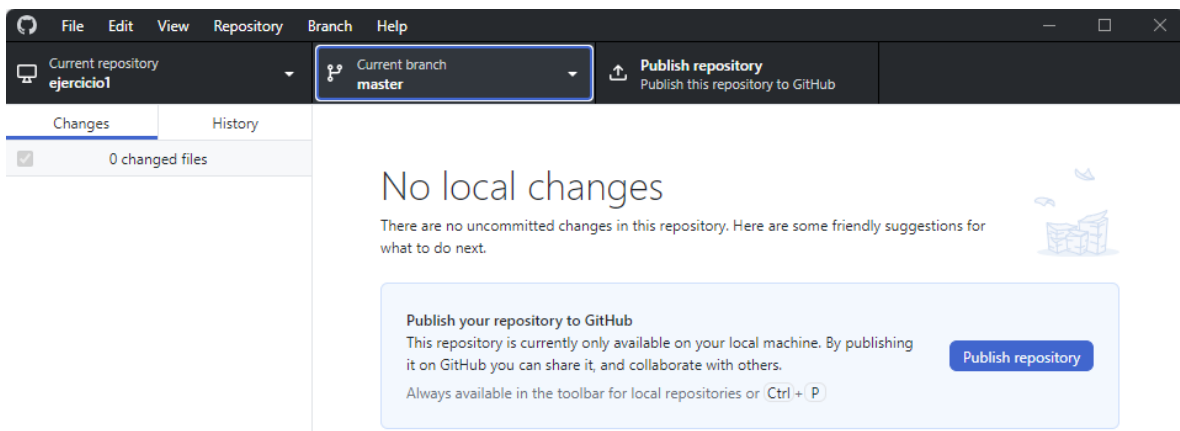
Description

Commit to rama2

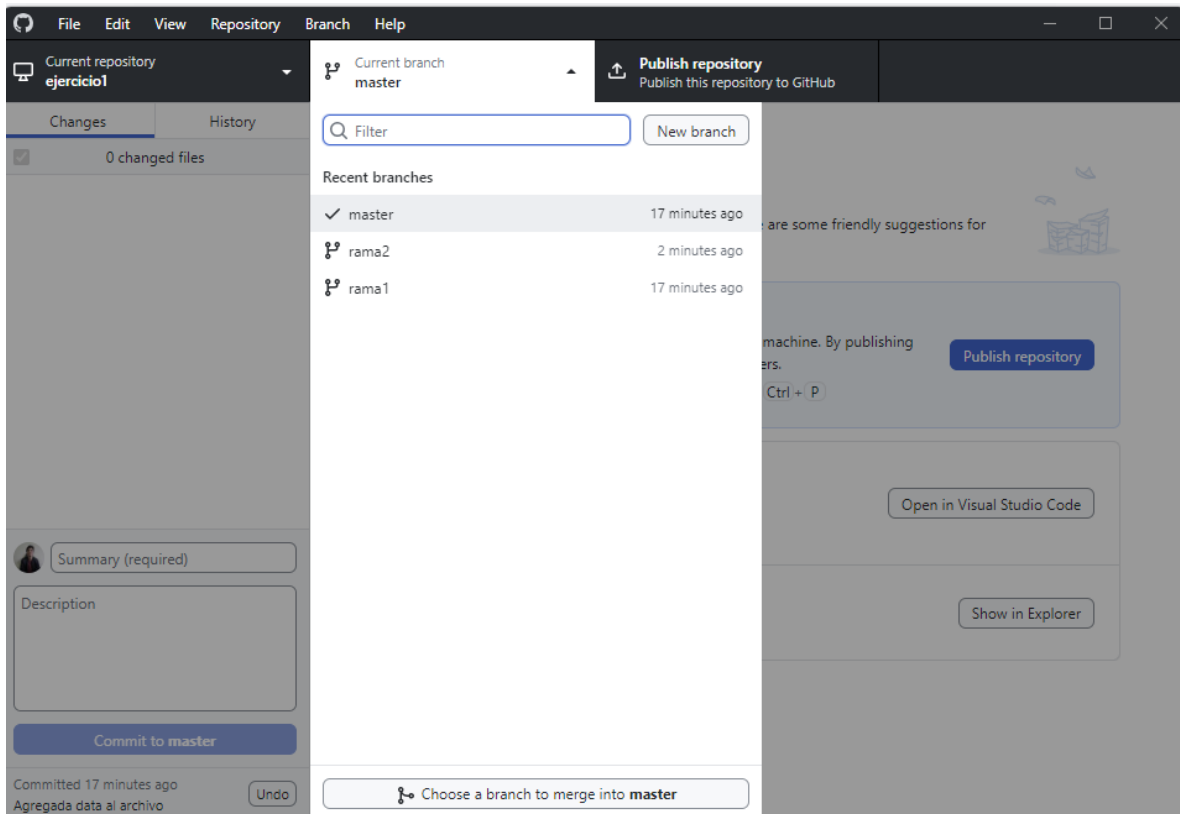
Los cambios ahora ya están salvados.



Para hacer **merge** de los cambios deberemos movernos a la rama **master**.



Daremos click en **Current branch** y hasta abajo encontraremos un botón con el mensaje **Choose a branch to merge into master**.



Elegiremos la **rama2** y daremos click en **Create merge commit**.

Merge into **master**

Q


Filter

Recent branches

✓


master

21 minutes ago



rama2

5 minutes ago



rama1

21 minutes ago

✓

This will merge **1 commit** from **rama2** into **master**

Create a merge commit

▼

Y ahora ya tendremos en nuestra rama master los cambios agregados en rama2.

FileEditViewRepositoryBranchHelp

Current repository
ejercicio1

Current branch
master

Publish repository

Publish this repository to GitHub

Changes

History

Archivo Ejemplo2 Creado

amelendezmio3 1688eee +1 -0

1 changed file

Ejemplo2.txt

@@ -0,0 +1 @@

1 + Esta es la rama creada con GitHub Desktop

Archivo Ejemplo2 Creado

amelendezmio3 • 7 minutes ago

Agregada data al archivo

amelendezmio3 • 22 minutes ago

Agregando archivo Ejemplo1

amelendezmio3 • 29 minutes ago

V. EJERCICIOS COMPLEMENTARIOS.

- Crea una tercera rama dentro de la carpeta del ejercicio creado en la cual van a eliminar el archivo 2 que se creó y agregan un tercer archivo el cual empezarán a modificar con datos de su elección.
- Investigue la forma de clonar un repositorio con Github Desktop.

VI. REFERENCIA BIBLIOGRAFICA.

- Git SCM. (s.f.). *Git Basics: Branches*. Git SCM Documentation. Recuperado de <https://git-scm.com/images/about/branches@2x.png>.
- Git SCM. (s.f.). *Git Basics: Workflow*. Git SCM Documentation. Recuperado de <https://git-scm.com/images/about/workflow-a@2x.png>.