

Cristian Gómez

Programación I

Actividad: Git y GitHub

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

¿Qué es GitHub?

GitHub es una plataforma en línea para alojar proyectos que utilizan el sistema de control de versiones Git. Permite a los desarrolladores colaborar, compartir código, gestionar tareas y revisar cambios. Ofrece funcionalidades como issues, pull requests, acciones automatizadas y más.

¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, primero se inicia sesión, después se clickea en el botón '+' y se selecciona 'New repository'. Luego, se completa el nombre, descripción (opcional), elegir si será público o privado, y clickear en 'Create repository'.

¿Cómo crear una rama en Git?

Utilizar el comando: `git branch nombre_de_rama`

¿Cómo cambiar a una rama en Git?

Utilizar el comando: `git checkout nombre_de_rama`

¿Cómo fusionar ramas en Git?

Primero cambiar a la rama destino con `git checkout main` (por ejemplo) y luego usar `git merge nombre_de_rama`

¿Cómo crear un commit en Git?

Primero preparar los archivos con `git add`, luego ejecutar: `git commit -m "mensaje del commit"`

¿Cómo enviar un commit a GitHub?

Usar el comando: `git push origin nombre_de_rama`

¿Qué es un repositorio remoto?

Es una versión alojada en la nube de tu repositorio local, generalmente en una plataforma como GitHub.

¿Cómo agregar un repositorio remoto a Git?

Usar el comando: `git remote add origin url_del_repositorio`

¿Cómo empujar cambios a un repositorio remoto?

Usar el comando: `git push origin nombre_de_rama`

¿Cómo tirar de cambios de un repositorio remoto?

Usar el comando: `git pull origin nombre_de_rama`

¿Qué es un fork de repositorio?

Es una copia de un repositorio que permite experimentar con cambios sin afectar el original.

¿Cómo crear un fork de un repositorio?

Hacer clic en el botón 'Fork' en la parte superior del repositorio de GitHub que se desea copiar.

¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

Después de hacer cambios en un fork, hacer clic en 'Compare & pull request' y luego en 'Create pull request'.

¿Cómo aceptar una solicitud de extracción?

El administrador del repositorio revisa los cambios y, si está de acuerdo, hace clic en 'Merge pull request'.

¿Qué es una etiqueta en Git?

Es un marcador que se utiliza para señalar puntos específicos en la historia del proyecto, como versiones.

¿Cómo crear una etiqueta en Git?

Usar el comando: `git tag nombre_etiqueta`

¿Cómo enviar una etiqueta a GitHub?

Usar el comando: `git push origin nombre_etiqueta`

¿Qué es un historial de Git?

Es el registro de todos los commits realizados en un proyecto.

¿Cómo ver el historial de Git?

Usar el comando: `git log`

¿Cómo buscar en el historial de Git?

Usar `git log --grep='texto'` para buscar por mensajes, o `git log -S'código'` para buscar por contenido.

¿Cómo borrar el historial de Git?

Se puede reiniciar el historial con: `rm -rf .git` y luego `git init` nuevamente.

¿Qué es un repositorio privado en GitHub?

Es un repositorio solo accesible por su creador y las personas invitadas. No es visible públicamente.

¿Cómo crear un repositorio privado en GitHub?

Al crear el repositorio, seleccionar la opción 'Private' antes de hacer clic en 'Create repository'.

¿Cómo invitar a alguien a un repositorio privado en GitHub?

Ir a la pestaña 'Settings' → 'Collaborators' → 'Add people' y escribir su nombre de usuario o correo electrónico.

¿Qué es un repositorio público en GitHub?

Es un repositorio accesible a cualquier persona. Puede ser clonado y visto sin restricciones.

¿Cómo crear un repositorio público en GitHub?

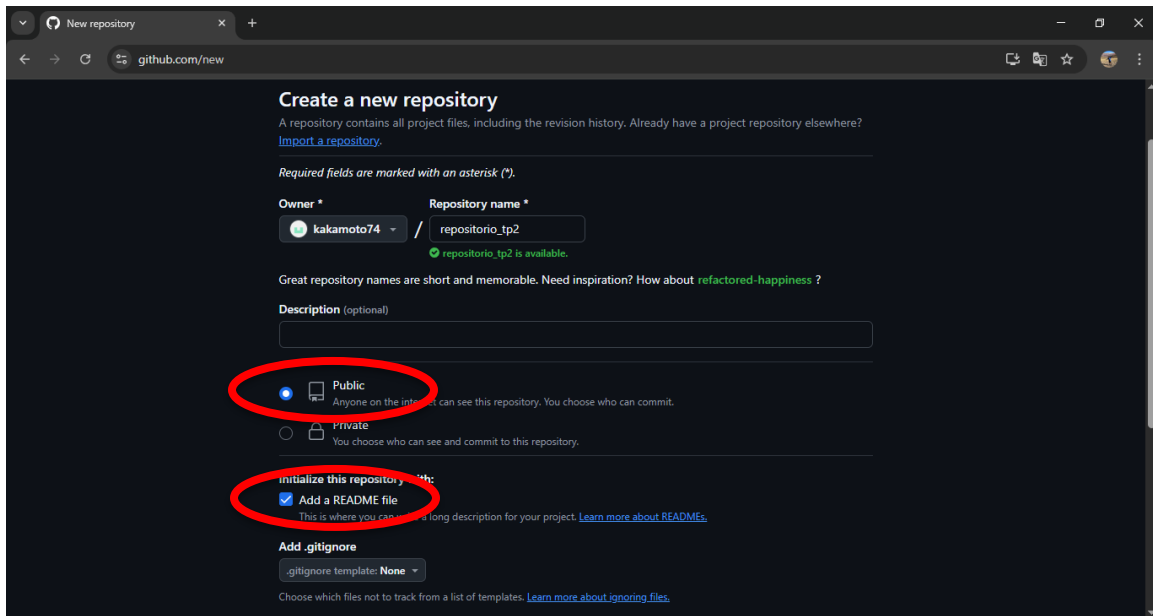
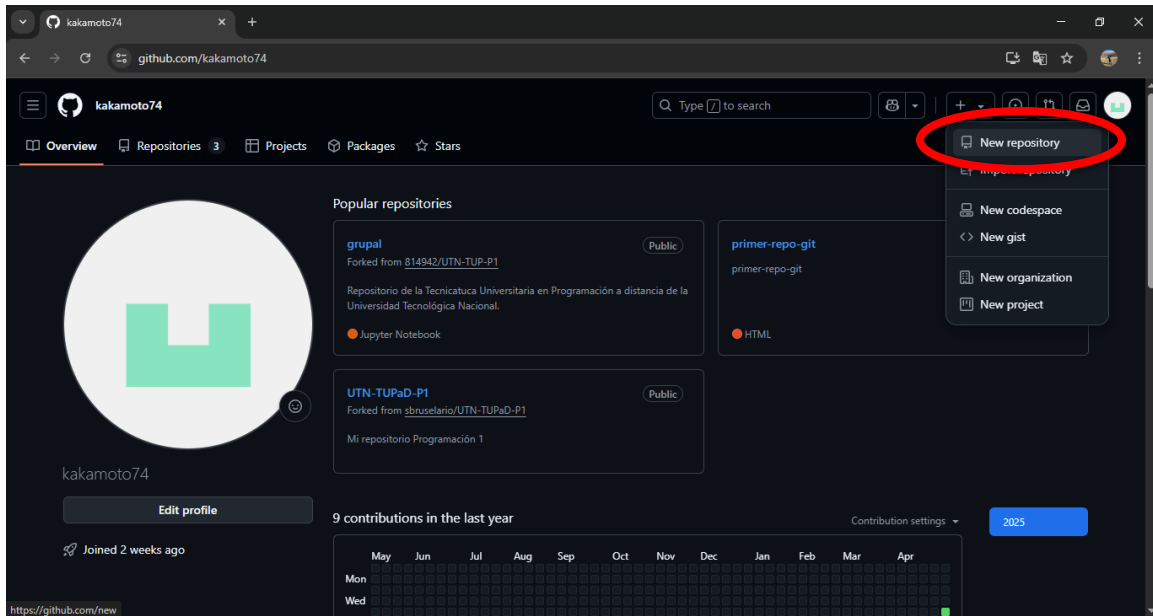
Durante la creación del repositorio, seleccionar 'Public' antes de hacer clic en 'Create repository'.

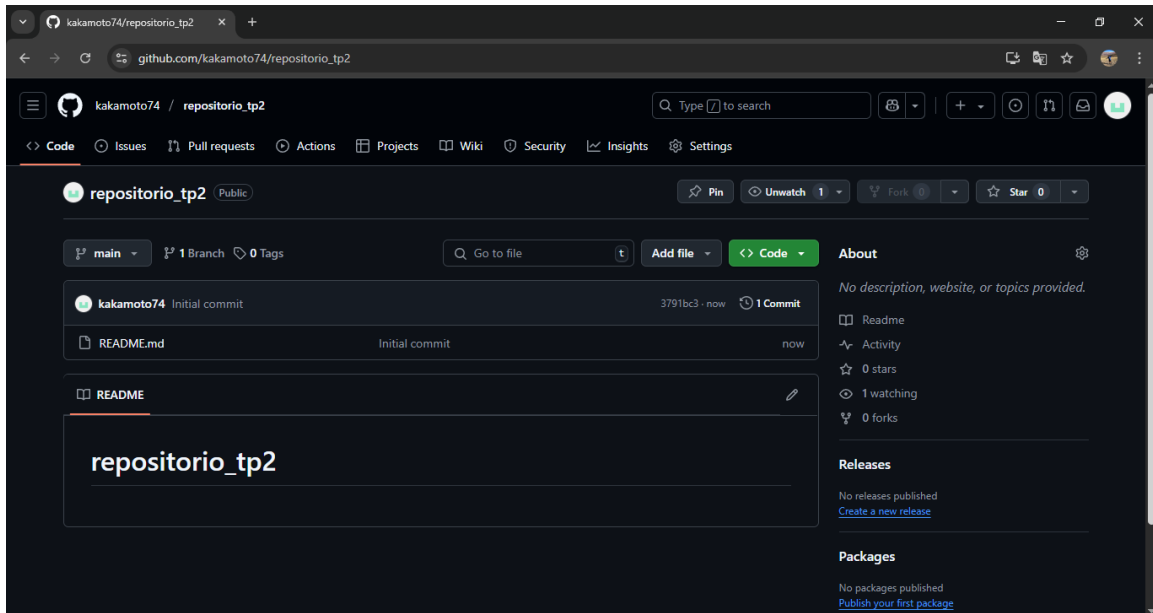
¿Cómo compartir un repositorio público en GitHub?

Copiar la URL del repositorio y compartirla. Otros podrán acceder y clonar el proyecto.

2) Realizar la siguiente actividad:

- Crear un repositorio.
 - Dale un nombre al repositorio.
 - Elije el repositorio sea público.
 - Inicializa el repositorio con un archivo





- Agregando un Archivo

- Crea un archivo simple, por ejemplo, "mi-archivo.txt".
- Realiza los comandos `git add .` y `git commit -m "Agregando mi-archivo.txt"` en la línea de comandos.
- Sube los cambios al repositorio en GitHub con `git push origin main` (o el nombre de la rama correspondiente).

```
MINGW64/c/Users/Criss/UTN-TUPaD-P1/repositorio_tp2
C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1 (main)
$ git clone https://github.com/kakamoto74/repositorio_tp2
Cloning into 'repositorio_tp2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1 (main)
$ cd repositorio_tp2

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ ls
README.md

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ touch mi-archivo.txt

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ ls
README.md  mi-archivo.txt

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

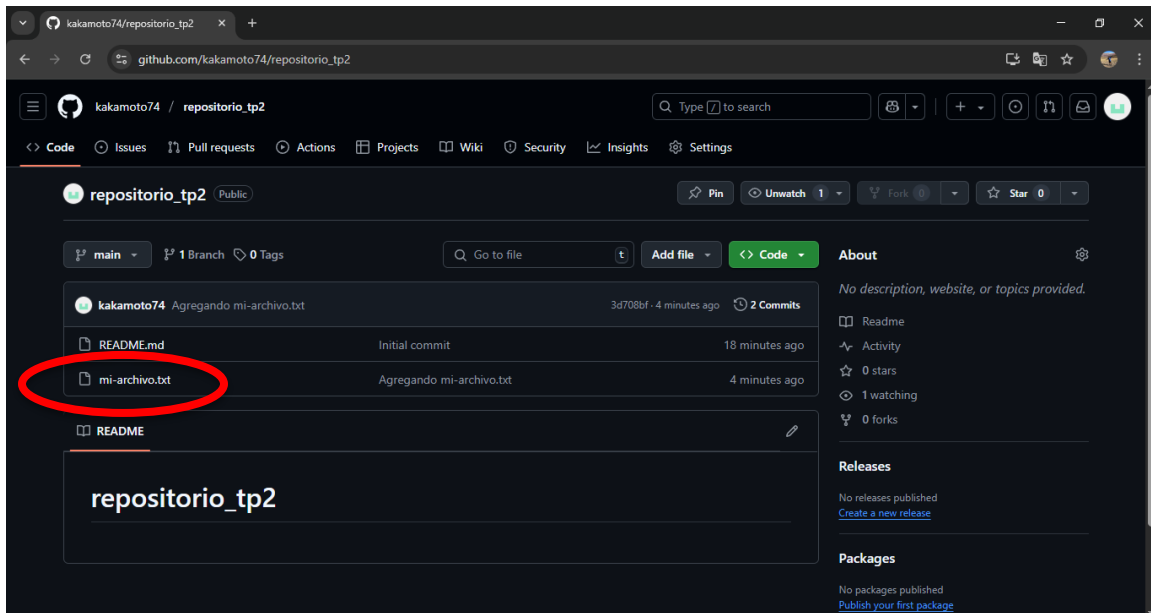
nothing added to commit but untracked files present (use "git add" to track)
C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ git add .

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ git commit -m "Agregando mi-archivo.txt"
[main 3d708bf] Agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt

C:\ISS\DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ 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
```

```
nothing to commit, working tree clean
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 97.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kakamoto74/repositorio_tp2
   3751bc3..3d708bf  main -> main
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$
```

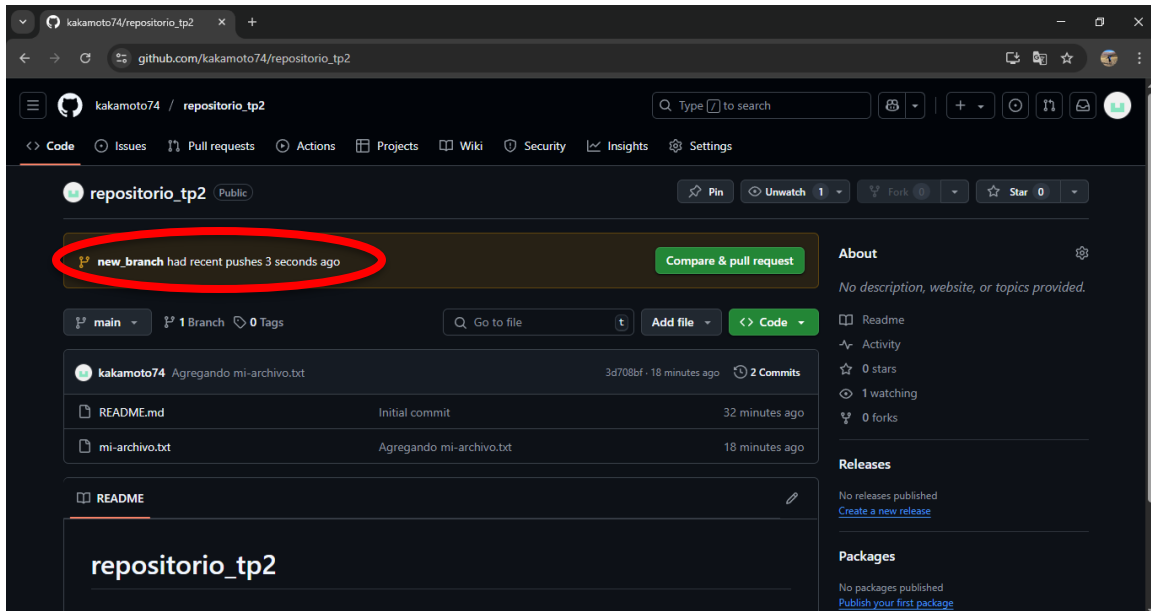


• Creando Branchs

- Crear una Branch
- Realizar cambios o agregar un archivo
- Subir la Branch

```
MINGW64/c/Users/Criss/UTN-TUPaD-P1/repositorio_tp2
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ git branch
* main
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (main)
$ git checkout -b new_branch
Switched to a new branch 'new_branch'
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
$ touch nuevo.txt
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
$ ls
README.md  mi-archivo.txt  nuevo.txt
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
$ git status
On branch new_branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nuevo.txt

nothing added to commit but untracked files present (use "git add" to track)
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
$ git add .
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
$ git commit -m "nueva rama y creación de archivo nuevo.txt"
[new_branch cdc2ae] nueva rama y creación de archivo nuevo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nuevo.txt
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
$ git push origin new_branch
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 304 bytes | 101.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'new_branch' on GitHub by visiting:
remote:   https://github.com/kakamoto74/repositorio_tp2/pull/new/new_branch
remote:
To https://github.com/kakamoto74/repositorio_tp2
   [new branch]  new_branch -> new_branch
Criss@DESKTOP-N3BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2 (new_branch)
```



3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

- Ve a GitHub e inicia sesión en tu cuenta.
- Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
- Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
- Opcionalmente, añade una descripción.
- Marca la opción "Initialize this repository with a README".
- Haz clic en "Create repository".

Paso 2: Clonar el repositorio a tu máquina local

- Copia la URL del repositorio (usualmente algo como <https://github.com/tuusuario/conflict-exercise.git>).
- Abre la terminal o línea de comandos en tu máquina.
- Clona el repositorio usando el comando:
git clone <https://github.com/tuusuario/conflict-exercise.git>
- Entra en el directorio del repositorio:
cd conflict-exercise

Paso 3: Crear una nueva rama y editar un archivo

- Crea una nueva rama llamada feature-branch:

```
git checkout -b feature-branch
```

- Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo: Este es un cambio en la feature branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in feature-branch"
```

Paso 4: Volver a la rama principal y editar el mismo archivo

- Cambia de vuelta a la rama principal (main):

```
git checkout main
```

- Edita el archivo README.md de nuevo, añadiendo una línea diferente:

Este es un cambio en la main branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in main branch"
```

Paso 5: Hacer un merge y generar un conflicto

- Intenta hacer un merge de la feature-branch en la rama main:

```
git merge feature-branch
```

- Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

Paso 6: Resolver el conflicto

- Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:

```
<<<<<<< HEAD
```

Este es un cambio en la main branch.

```
=====
```

Este es un cambio en la feature branch.

>>>>>> feature-branch

- Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.
- Edita el archivo para resolver el conflicto y guarda los cambios(Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).
- Añade el archivo resuelto y completa el merge:

```
git add README.md
```

```
git commit -m "Resolved merge conflict"
```

Paso 7: Subir los cambios a GitHub

- Sube los cambios de la rama main al repositorio remoto en GitHub:

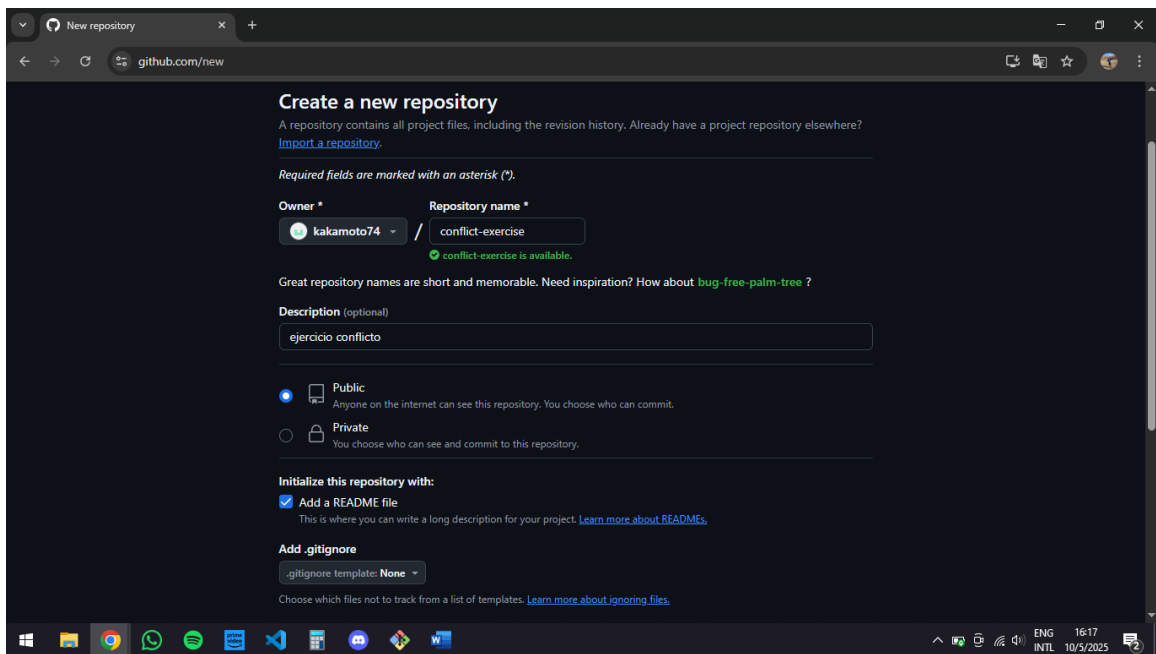
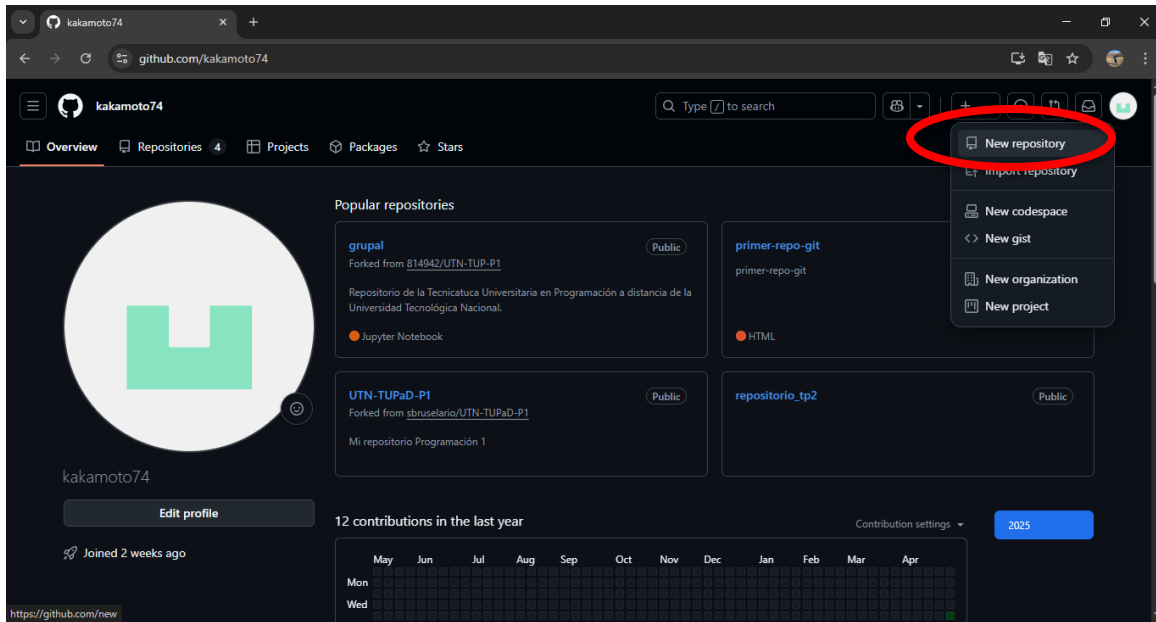
```
git push origin main
```

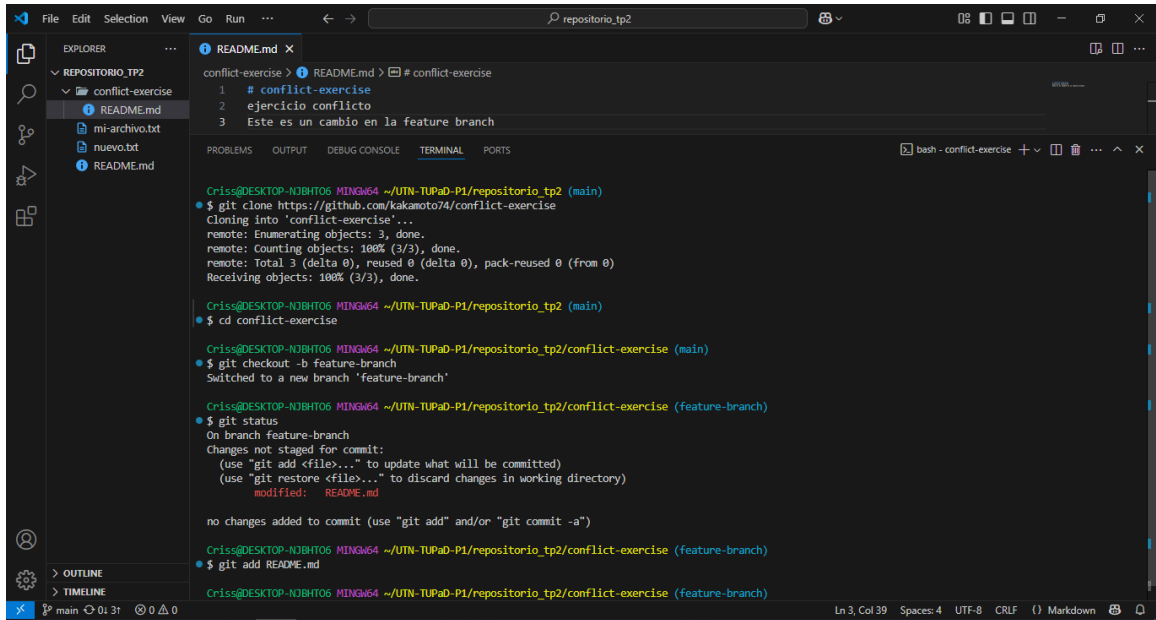
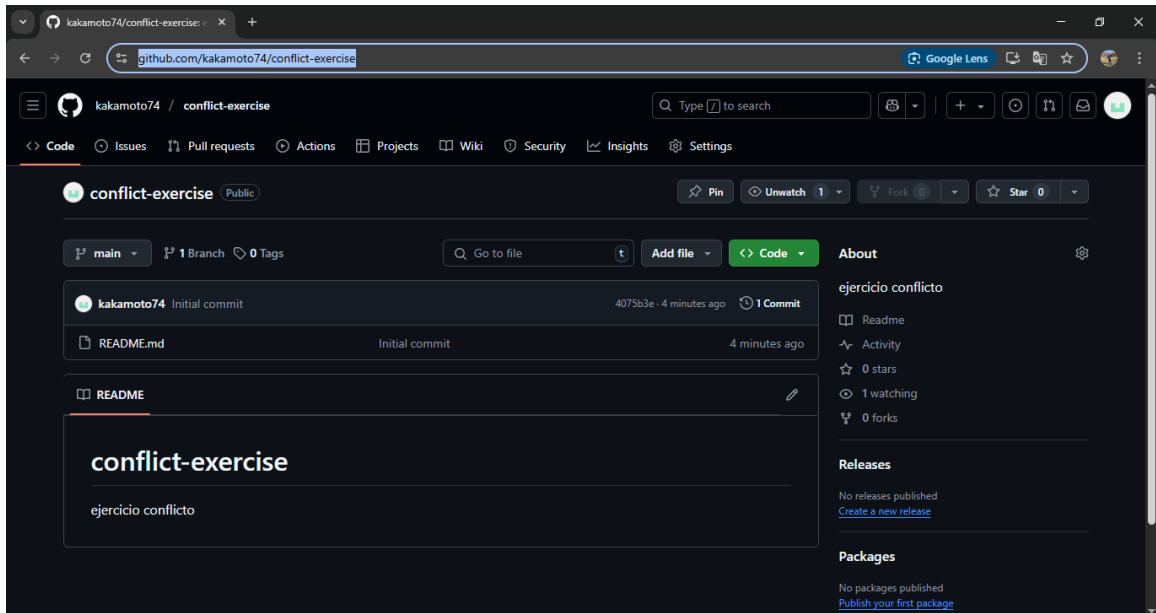
- También sube la feature-branch si deseas:

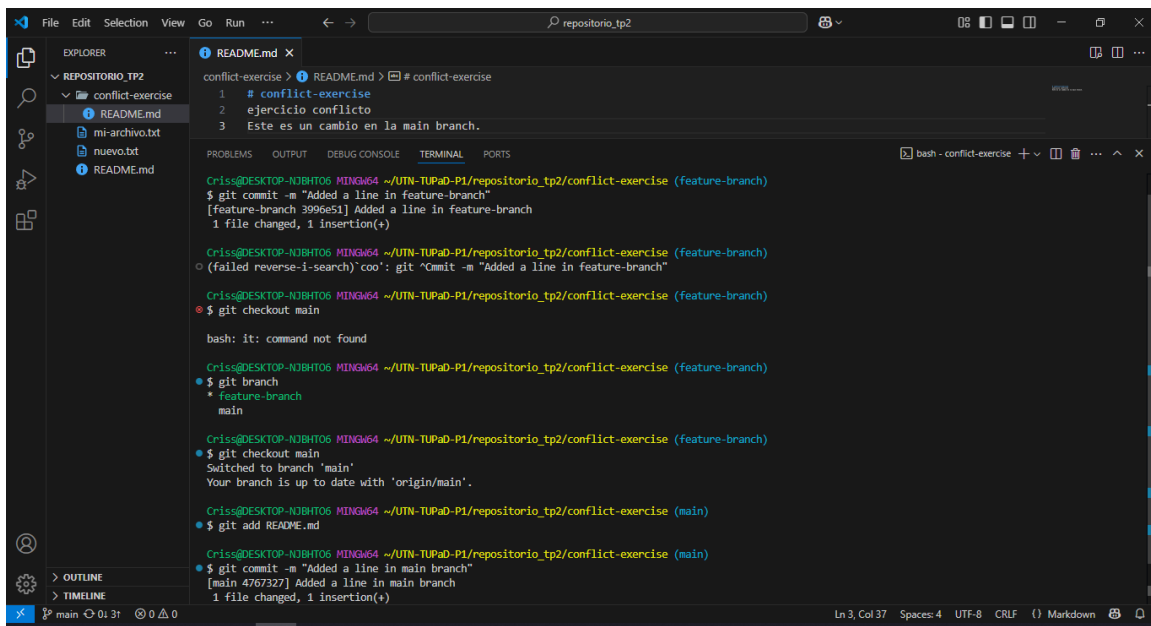
```
git push origin feature-branch
```

Paso 8: Verificar en GitHub

- Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.
- Puedes revisar el historial de commits para ver el conflicto y su resolución.







```
File Edit Selection View Go Run ... repository_tp2
EXPLORER
  REPOSITORIO_TP2
    conflict-exercise
      README.md
      mi-archivo.txt
      nuevo.txt
      README.md
  OUTLINE
  TIMELINE
  main 0:31 0 0 0

conflict-exercise > README.md # conflict-exercise
1 # conflict-exercise
2 ejercicio conflicto
3 Este es un cambio en la main branch.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - conflict-exercise + v [ ] [ ] ... ^ x

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 3996e51] Added a line in feature-branch
1 file changed, 1 insertion(+)

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (feature-branch)
$ git checkout main
(failed reverse-i-search) 'coo': git 'Commit -m "Added a line in feature-branch"'

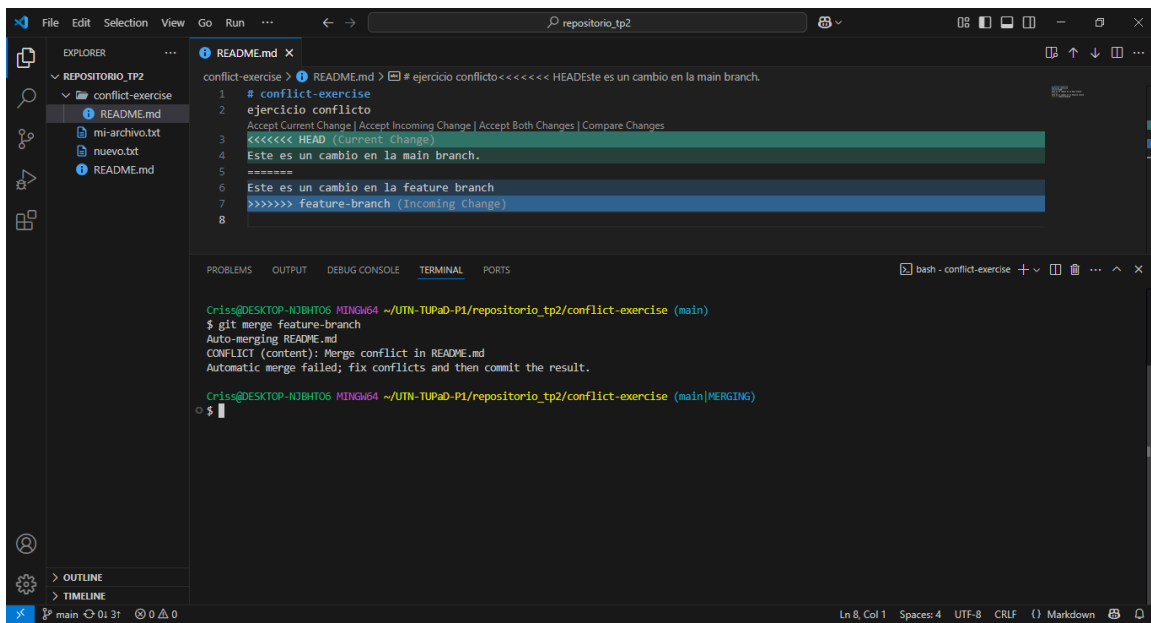
Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (feature-branch)
$ git checkout main
bash: it: command not found

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (feature-branch)
$ git branch
* feature-branch
main

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main)
$ git add README.md

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main 4767327] Added a line in main branch
1 file changed, 1 insertion(+)
```



```
File Edit Selection View Go Run ... repository_tp2
EXPLORER
  REPOSITORIO_TP2
    conflict-exercise
      README.md
      mi-archivo.txt
      nuevo.txt
      README.md
  OUTLINE
  TIMELINE
  main 0:31 0 0 0

conflict-exercise > README.md # ejercicio conflicto<<<<<< HEADEste es un cambio en la main branch.
1 # conflict-exercise
2 ejercicio conflicto
3 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 <<<<<< HEAD (Current Change)
5 Este es un cambio en la main branch.
6 =====
7 Este es un cambio en la feature branch
8 >>>>>> feature-branch (Incoming Change)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - conflict-exercise + v [ ] [ ] ... ^ x

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main)
$ git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Criss@DESKTOP-H0BHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main|MERGING)
$
```

The screenshot shows a VS Code editor with a terminal window open. The terminal displays the following commands and output:

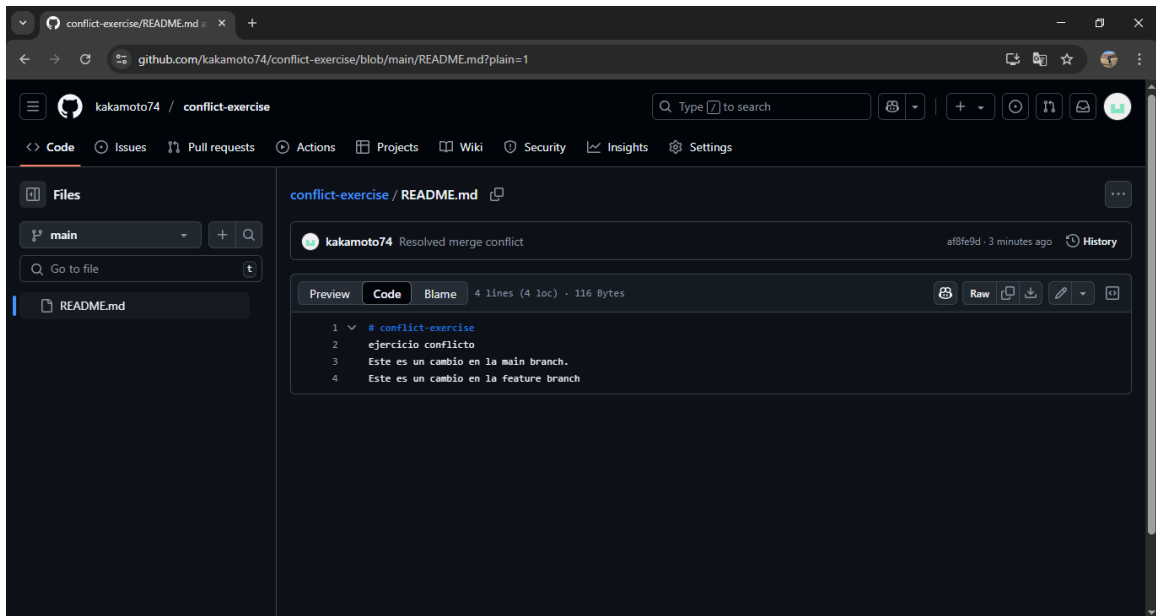
```
Criss@DESKTOP-NDJHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main)
$ git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Criss@DESKTOP-NDJHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main|MERGING)
$ git add README.md

Criss@DESKTOP-NDJHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main af8fe9d] Resolved merge conflict

Criss@DESKTOP-NDJHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 752 bytes | 94.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/kakamoto74/conflict-exercise
4075b3e..af8fe9d main -> main

Criss@DESKTOP-NDJHT06 MINGW64 ~/UTN-TUPaD-P1/repositorio_tp2/conflict-exercise (main)
$
```



Resolved merge conflict · kakam

github.com/kakamoto74/conflict-exercise/commit/af8fe9d2f6107cb4ec9ef5d18f553ca062054661

kakamoto74 / conflict-exercise

Type to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

Commit af8fe9d

Browse files

kakamoto74 committed 4 minutes ago

Resolved merge conflict

main2 parents 4767327 + 3996e51 commit af8fe9d

Filter files...README.md

1 file changed +2 -1 lines changed

Search within code

▼ README.md

@@ -1,3 +1,4 @@

1 # conflict-exercise

2 ejercicio conflicto

3 - Este es un cambio en la main branch.

1 # conflict-exercise

2 ejercicio conflicto

3 + Este es un cambio en la main branch.

4 + Este es un cambio en la feature branch

Comments 0

Lock conversation