

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Práctico 2: Git y GitHub

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

- **¿Qué es GitHub?**

GitHub es una plataforma basada en la web que permite el almacenamiento, gestión y control de versiones de código fuente mediante el uso de Git. Facilita la colaboración entre desarrolladores y equipos al proporcionar herramientas como repositorios, ramas, control de cambios y revisión de código.

- **¿Cómo crear un repositorio en GitHub?**

1. Crear una cuenta e iniciar sesión en GitHub.
2. Hacer clic en el botón 'New Repository' en la página principal o desde el perfil.
3. Ingresar un nombre y una descripción opcional.
4. Elegir si el repositorio será público o privado.
5. Hacer clic en 'Create Repository'.

- **¿Cómo crear una rama en Git?**

Usando el código: `git branch nombreRama`

- **¿Cómo cambiar a una rama en Git?**

Usando el código: `git checkout nuevaRama`

- **¿Cómo fusionar ramas en Git?**

Primero nos ubicamos en la rama principal: (main o master)

Usando el código: `Git checkout main`

Estando en la rama principal se usa el siguiente código para fusionar las dos ramas:

`Git merge nombreRama`

- **¿Cómo crear un commit en Git?**

Primero se guardan los cambios del archivo específico o todos.

Usando el código: `git add archivo` (el archivo en específico) o `git add` (todos los archivos)

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Después se hace el commit describiendo los cambios realizados

Usando el código: `git commit -m (cambios)`

- **¿Cómo enviar un commit a GitHub?**

Después de guardar los cambios y realizar el commit se usa el siguiente código:

Usando el código: `git push origin nombreRama`

- **¿Qué es un repositorio remoto?**

Es una versión de un proyecto que esta alojada en un servidor externo (como GitHub), que permite compartir, sincronizar, leer o realizar cambios del código entre equipos.

- **¿Cómo agregar un repositorio remoto a Git?**

Usando el código: `git remote add origin url_repositorio`

- **¿Cómo empujar cambios a un repositorio remoto?**

Se recomienda antes de empujar cambios, actualizar los últimos cambios del repositorio:

Usando el código: `git pull origin nombreRama`

Una vez estamos seguros de que se obtuvieron los últimos cambios del repositorio se pushea:

Usando el código: `git push origin nombreRama`

- **¿Cómo tirar de cambios de un repositorio remoto?**

Usando el código: `git pull origin nombreRama`

- **¿Qué es un fork de repositorio?**

Es una copia de un repositorio de una cuenta diferente que se crea en la nuestra permitiendo crear cambios sin afectar el original.

- **¿Cómo crear un fork de un repositorio?**

Inicia sesión en github

Ingresar al repositorio original

Hacer click en el botón “Fork” de la esquina superior derecha

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Opcional cambiar la información de la descripción

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

Ir a la pestaña de “pull requests” en GitHub

Hacer click en “New pull requests”. Se vera una ventana los cambios realizados comparados con el repositorio original

Hacer click n “Create pull requests”

- **¿Cómo aceptar una solicitud de extracción?**

Ir a la pestaña de “Pull requests” del repositorio

Seleccinar la solicitud y hacer click en “Merge Pull Requests”

- **¿Qué es una etiqueta en Git?**

Git tiene la posibilidad de etiquetar puntos específicos del historial como importantes, es una referencia que señala un punto específico en el historial de un repositorio. Esta funcionalidad se usa típicamente para marcar versiones de lanzamiento (v1.0, por ejemplo).

- **¿Cómo crear una etiqueta en Git?**

Git tiene dos tupos de etiquetas principales: ligeras o anotadas. Se recomienda realizar etiquetas anotadas ya que aportan mas información y quedan guardadas en la base de datos

Usando el código: `git tag nombreEtiqueta`

- **¿Cómo enviar una etiqueta a GitHub?**

Antes se debe crear una etiqueta en el repositorio local y después crear la etiqueta ligera/anotada

Usando el código: `git push origin nombreEtiqueta`

En el caso de enviar todas las etiquetas a GitHub:

Usando el código: `git push origin -tags`

- **¿Qué es un historial de Git?**

Es un registro de todos los cambios hechos en un repositorio de Git guardados como un commit donde se incluye toda la información.

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

- **¿Cómo ver el historial de Git?**

Usando el código: `git log`

- **¿Cómo buscar en el historial de Git?**

Para buscar un texto en específico de un commit

Usando el código: `git log --grep="texto a buscar"`

Para buscar por el nombre del autor del commit

Usando el código: `git log --author="nombre del autor"`

Para buscar un rango de fechas

Usando el código: `git log --since"fecha inicial" --until"fecha final"`

- **¿Cómo borrar el historial de Git?**

Para borrar el historial se usa `reset` con un puntero HEAD

Usando el código: `git reset nombre(archivo, carpeta)`

- **¿Qué es un repositorio privado en GitHub?**

Es un repositorio que solo pueden ver y modificar las personas con permisos autorizados

- **¿Cómo crear un repositorio privado en GitHub?**

Iniciar sesión en GitHub

Crear un repositorio como se mencionó anteriormente y seleccionar la pestaña de "private" en configuración de privacidad

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

Ir a "settings" del repositorio

Seleccionar "Manage Access"

Hacer click en "Invite collaborator"

Hacer click en "Add people"

Ingresar el nombre de usuario de GitHub de la persona invitada

Hacer click en "Add"

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

- **¿Qué es un repositorio público en GitHub?**

Es un repositorio visible para cualquier en GitHub

- **¿Cómo crear un repositorio público en GitHub?**

Iniciar sesión en GitHub

Crear un repositorio como se mencionó anteriormente y seleccionar la pestaña de “Public” en configuración de privacidad

- **¿Cómo compartir un repositorio público en GitHub?**

Se accede al repositorio y se hace click en “<> Code”

Se comparte el enlace del repositorio (URL)

2) Realizar la siguiente actividad:

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

URL= <https://github.com/newtoncas/TP2.git>

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

- Agregando un Archivo

o Crea un archivo simple, por ejemplo, "mi-archivo.txt".

o Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.

o Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git init
Initialized empty Git repository in C:/Users/newto/OneDrive/Documents/PROGRAMACION/GIT/.git/

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git add .

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git commit -m "Este es mi primer commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'newto@DESKTOP-D3B5060.(none)')

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git config --global user.email "sernachristian700@gmail.com"

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git config --global user.name "newtoncas"

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git commit -m "Este es mi primer commit"
[master (root-commit) 3d4f2b0] Este es mi primer commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Mi-archivo.txt
```

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git remote add origin https://github.com/newtoncas/TP2

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 228 bytes | 228.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/newtoncas/TP2/pull/new/master
remote:
To https://github.com/newtoncas/TP2
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

• Creando Branchs

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

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (nuevaRama)
$ git branch
  master
* nuevaRama

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (nuevaRama)
$ git add .

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (nuevaRama)
$ git status
On branch nuevaRama
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Mi-archivo.txt

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (nuevaRama)
$ git commit -m "Cambios en el archivo de la nueva rama"
[nuevaRama 68a7d9e] Cambios en el archivo de la nueva rama
 1 file changed, 2 insertions(+), 1 deletion(-)

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (nuevaRama)
$ git status
On branch nuevaRama
nothing to commit, working tree clean

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (nuevaRama)
$ git push origin nuevaRama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 326 bytes | 326.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nuevaRama' on GitHub by visiting:
remote:   https://github.com/newtoncas/TP2/pull/new/nuevaRama
remote:
To https://github.com/newtoncas/TP2
 * [new branch]      nuevaRama -> nuevaRama
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

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".

<https://github.com/newtoncas/conflict-exercise.git>

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:
- Entra en el directorio del repositorio:

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ git clone https://github.com/newtoncas/conflict-exercise.git
Cloning into 'conflict-exercise'...
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.

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ cd conflict-exercise

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (
main)
$ |
```


TECNICATURA UNIVERSITARIA

EN PROGRAMACIÓN A DISTANCIA

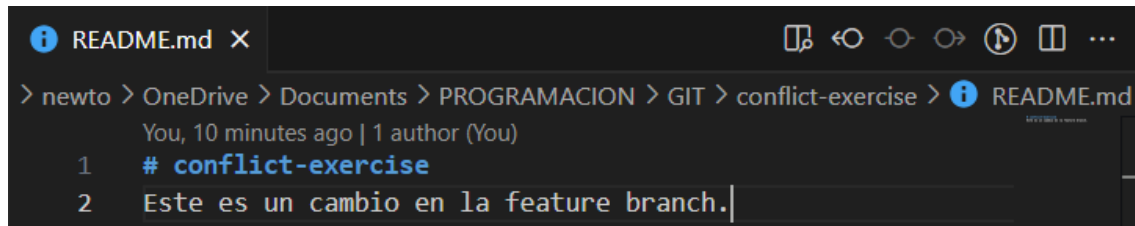
Paso 3: Crear una nueva rama y editar un archivo

- Crea una nueva rama llamada feature-branch:

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT (master)
$ cd conflict-exercise

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (
main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

- Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:



- Guarda los cambios y haz un commit:

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (
feature-branch)
$ git add README.md

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (
feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 208d75b] Added a line in feature-branch
1 file changed, 2 insertions(+), 1 deletion(-)
```

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

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

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise
feature-branch)
$ git checkout -b master
Switched to a new branch 'master'

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise
master)
$ |
```

- Edita el archivo README.md de nuevo, añadiendo una línea diferente:
- Guarda los cambios y haz un commit:

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise
master)
$ git add README.md

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise
master)
$ git commit -m "Added a line in main branch"
[master d81f603] Added a line in main branch
1 file changed, 1 insertion(+), 1 deletion(-)
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Paso 5: Hacer un merge y generar un conflicto

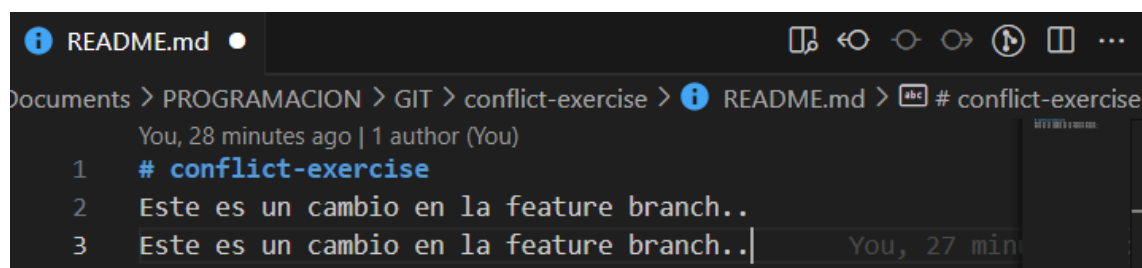
- Intenta hacer un merge de la feature-branch en la rama main:

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (master)
$ git merge feature-branch
Already up to date.
```

- 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.
- 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



```
Documents > PROGRAMACION > GIT > conflict-exercise > README.md > # conflict-exercise
You, 28 minutes ago | 1 author (You)
1 # conflict-exercise
2 Este es un cambio en la feature branch..
3 Este es un cambio en la feature branch..
```

Paso 7: Subir los cambios a GitHub

- Añade el archivo resuelto y completa el merge:

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (feature-branch)
$ git add README.md

newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (feature-branch)
$ git commit -m "Resolved merge conflict"
```

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

```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise (feature-branch)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 543 bytes | 271.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/newtoncas/conflict-exercise/pull/new/master
remote:
To https://github.com/newtoncas/conflict-exercise.git
* [new branch]      master -> master
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

- También sube la feature-branch si deseas:

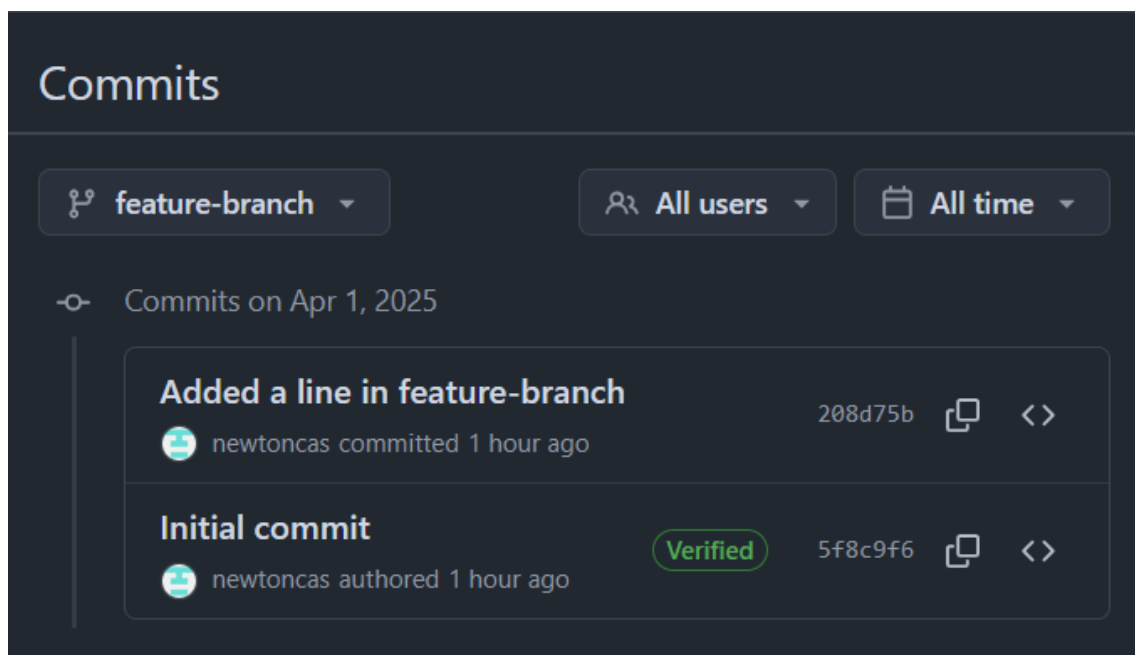
```
newto@DESKTOP-D3B5060 MINGW64 ~/OneDrive/Documents/PROGRAMACION/GIT/conflict-exercise
feature-branch)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:   https://github.com/newtoncas/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/newtoncas/conflict-exercise.git
 * [new branch]      feature-branch -> 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.

<https://github.com/newtoncas/conflict-exercise.git>

- Puedes revisar el historial de commits para ver el conflicto y su resolución.



The screenshot shows the GitHub 'Commits' page for the 'feature-branch'. At the top, there are filters for 'feature-branch', 'All users', and 'All time'. Below these, a section titled 'Commits on Apr 1, 2025' lists two commits. The first commit is titled 'Added a line in feature-branch' by 'newtoncas' committed '1 hour ago', with a hash of '208d75b'. The second commit is titled 'Initial commit' by 'newtoncas' authored '1 hour ago', with a hash of '5f8c9f6' and a 'Verified' badge. Each commit entry includes a copy icon and a compare icon.

Commit Message	Author	Time	Hash	Verified
Added a line in feature-branch	newtoncas	committed 1 hour ago	208d75b	
Initial commit	newtoncas	authored 1 hour ago	5f8c9f6	Verified