



MANUAL DE USUARIO	
CODIFICACION:	DTI-MU-2024-001
FECHA:	04-04-2024

Manual Usuario

Manejo de Git

Dirección de Tecnologías e Información

Elaborado por:

Mgs. Norma Arcos

Versión 1.0
Fecha de emisión: 04 de Abril de 2024
Quito – Ecuador

Contenido

1. CREAR UN DIRECTORIO EN GIT WEB	3
2. CREAR UN DIRECTORIO EN GIT POR CONSOLA.....	3
3. COMANDOS ÚTILES EN GIT	3
4. ELIMINAR UN ARCHIVO	3
5. CREAR RAMAS EN GIT	4
6. GIT EN VISUAL STUDIO COMMUNITY	5
7. NOTAS IMPORTANTES	7
8. BIBLIOGRAFIA.....	7
9. ANEXOS	7
10. FIRMAS	7

1. CREAR UN DIRECTORIO EN GIT WEB

1. Hacer clic en el botón Créate new file
2. En la caja de texto que aparece escribir el nombre del directorio que se va a crear.
3. Al oprimir la tecla /, automáticamente, el nombre escrito en el punto anterior se convierte en directorio y aparece una nueva caja de texto.
4. Seguir escribiendo la ruta. Los directorios **deben** terminar con /.
5. Escribir el nombre de un archivo (sin terminación en /) para crear toda la ruta.
6. Hacer *commit* en el archivo.

2. CREAR UN DIRECTORIO EN GIT POR CONSOLA

Para crear un directorio en git por consola realizar lo siguiente

```
mkdir prueba
```

```
cd prueba
```

```
git clone https://web_del_proyecto.git .
```

-> se baja en el pc al directorio que se creó y se ingresa al mismo

`mkdir "mi_nuevo_directorio_con_o_sin_contenido"` -> debe de estar dentro del directorio anterior para que el próximo comando no dé error

```
git add "mi_nuevo_directorio_con_o_sin_contenido"
```

```
git commit -m "Nuevo directorio"
```

```
git push origin
```

3. COMANDOS ÚTILES EN GIT

`git log --oneline` Para revisar el último cambio

`git status` Permite ver los archivos que están por subir

4. ELIMINAR UN ARCHIVO

1. Dirigirse al archivo que deseas eliminar de tu repositorio.
2. En la esquina superior derecha, selecciona el menú desplegable; después, haz clic en **Eliminar archivo**.



3. En el campo de "Mensaje de confirmación", escriba un mensaje de confirmación corto y significativo que describa la modificación que hizo en el archivo.
4. Haz clic en **Confirmar cambios** o **Proponer cambios**.

5. CREAR RAMAS EN GIT

Hay dos métodos para crear ramas en Git.

```
// crear una rama y moverse a esa rama
$ git checkout -b <nombre-rama>
```

```
// crear sólo una rama
$ git branch <nombre-rama>
```

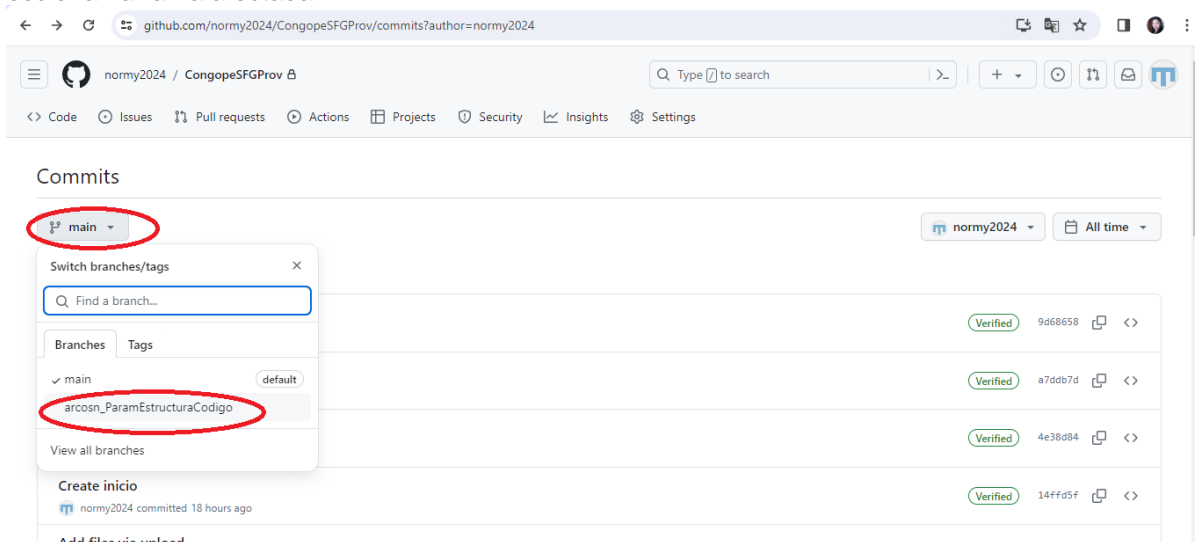
Para ver la lista de todas las ramas disponibles, se puede usar este comando:

```
$ git branch
```

Si se quiere agregar esta rama de forma remota se debe enviar al proveedor de Git como GitHub usando el siguiente comando:

```
$ git push -u origin <nombre-rama>
```

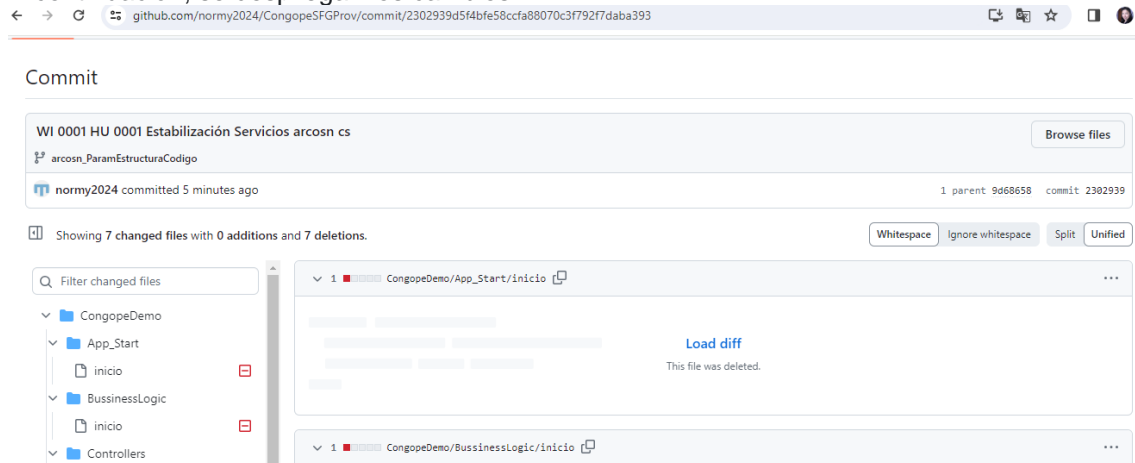
Seleccionar la rama afectada



Se verifica los cambios subidos por el numero changeset

Verified 9d68658

A continuación, se despliegan los cambios



6. GIT EN VISUAL STUDIO COMMUNITY

Para visualizar los cambios en el proyecto acceder a la sección “Cambios en Git”
Aquí se puede ver las secciones siguientes:



1.- Nombre de la rama

Permite verificar la rama sobre la cual se está trabajando y ubicarse en otra rama remota de ser el caso.

2.- Botones de control

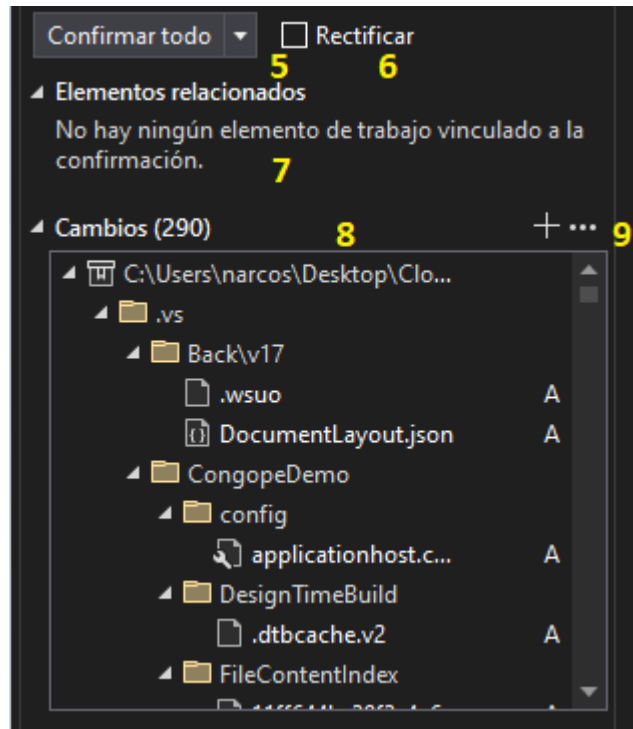
2.1 Fetch: descarga confirmaciones de rama remota y objetos de archivo a los que se hace referencia que no existen en el repositorio local y actualiza las ramas de seguimiento remoto en la memoria caché del repositorio local.

2.2 Extraer: Traer los cambios de la rama.

2.3 Push: Para insertar los cambios en la rama sobre la cual se esté situada.
2.4 Sincronizar: Extra e Inserta cambios.

3.- Link Confirmaciones: Para visualizar commits.

4.- Área de comentarios: Para comentarios de los commits realizados.



5.- Confirmar todo: Para opciones de confirmación como extraer e insertar o realizar estas acciones parcialmente.

6.- Rectificar: Para revertir un cambio.

7.- Elementos relacionados: Cambios relacionados a un conjunto de cambios actual.

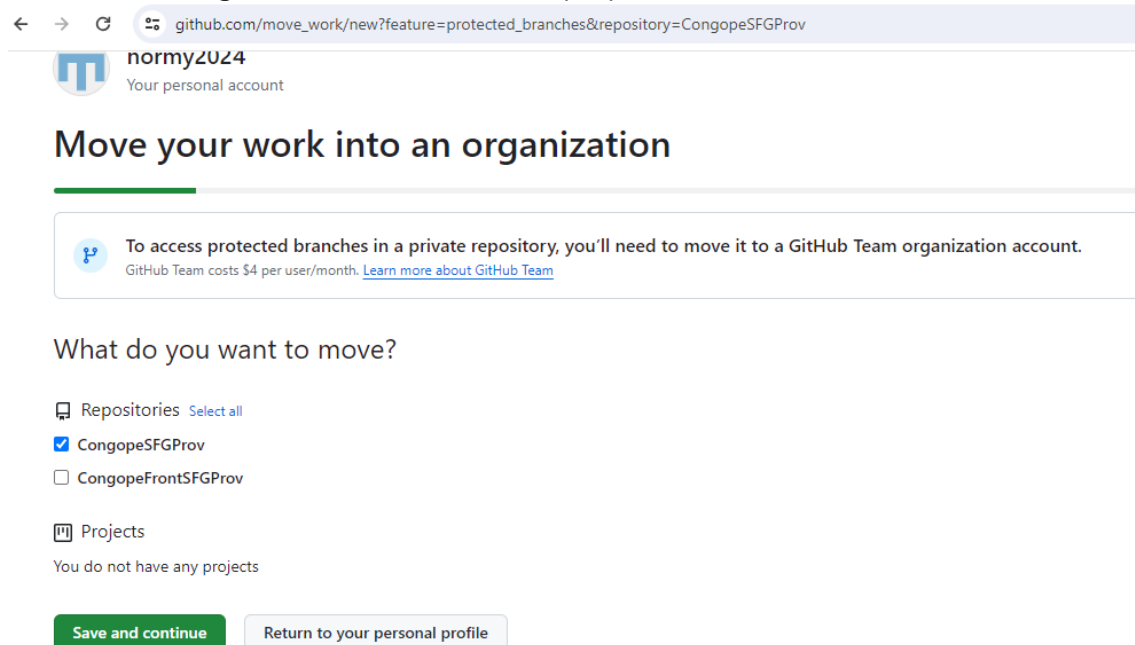
8.-Cambios: Cambios realizados en el código.

9.-Botones relacionados: para crear un stage o cambiar opciones de vista.

10.- Stashes: Cambios guardados provisionalmente

7. NOTAS IMPORTANTES

Cabe mencionar que para seguridad del repositorio creado es necesario mover el trabajo a una cuenta de organización o GitHub Team que presenta costo adicional.



The screenshot shows the GitHub interface for moving a repository. At the top, the URL is `github.com/move_work/new?feature=protected_branches&repository=CongopeSFGProv`. The user is logged in as **normy2024**. The main heading is "Move your work into an organization". A blue callout box states: "To access protected branches in a private repository, you'll need to move it to a GitHub Team organization account. GitHub Team costs \$4 per user/month. [Learn more about GitHub Team](#)". Below this, the question "What do you want to move?" is followed by two sections: "Repositories" and "Projects". Under "Repositories", there is a "Select all" link and two checkboxes: ☒ CongopeSFGProv and ☐ CongopeFrontSFGProv. Under "Projects", it says "You do not have any projects". At the bottom, there are two buttons: "Save and continue" (green) and "Return to your personal profile" (grey).

8. BIBLIOGRAFIA

Páginas Web

- <https://docs.github.com/es>
- <https://www.freecodecamp.org/espanol/news/como-crear-una-rama-local-en-git/>

9. ANEXOS

No aplica

10. FIRMAS

	CARGO	FIRMA	FECHA
ELABORADO POR	Mgs. Norma Arcos Profesional Contratado		10-04-2024