

Computación en Física

Harvey Rodriguez Gil

Universidad EIA

27 de Agosto de 2024

Ramas

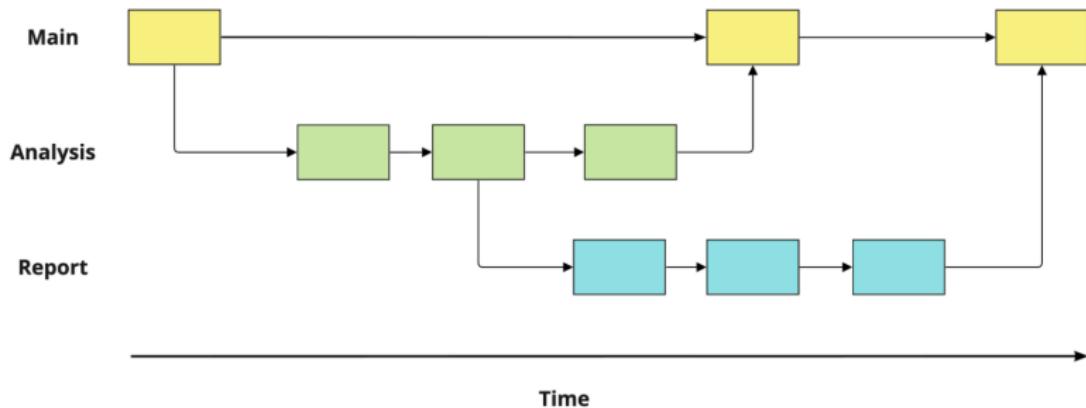
Git utiliza el concepto de ramas para registrar sistemáticamente múltiples versiones de los archivos. En cada rama se puede:

- ▶ Encontrar que los archivos son los mismos
- ▶ Algunos archivos son diferentes
- ▶ Algunos archivos ni siquiera existen



Manejo de las Ramas

Tenemos una rama principal llamada `main`, a partir de la cual se crea una rama secundaria llamada `Analysis`, y de esta última se crea una nueva rama llamada `Report`. Al final todas se mezclan con la rama principal `main`.



Manejo de las Ramas

- ▶ Cuando dos ramas se mezclan:
 - ▶ Los commits son llamados **commits padres**
 - ▶ La rama desde donde se quiere hacer la mezcla o unión se conoce como **source**
 - ▶ La rama donde quiero mezclar los cambios se conoce como **destination**
- ▶ Cuando se realiza la mezcla o unión de Analysis a Main:
 - ▶ Analysis es el **source**
 - ▶ Main es el **destination**

Manejo de las Ramas

Los beneficios de trabajar con ramas son:

- ▶ Evitar los subdirectorios sin fin.
- ▶ Multiples usuarios pueden trabajar simultaneamente
- ▶ Cada archivo o directorio es registrado
- ▶ Minimiza el riesgo de conflictos entre versiones

Manejo de las Ramas

El comando de git para conocer las ramas es:

```
git branch
```

El comando para crear y cambiarnos a dichar rama será:

```
git checkout -b new_branch
```

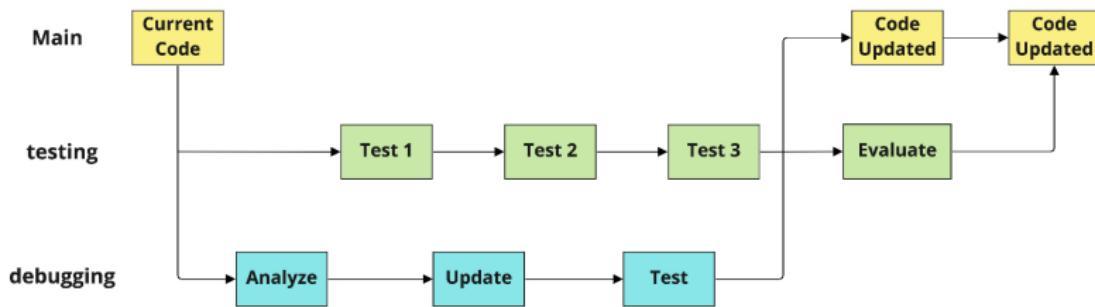
Para ver las diferencias entre ramas:

```
git diff main new_branch
```

Trabajando con ramas

Por qué necesitamos intercambiar de ramas:

- ▶ Para trabajar en diferentes componentes de un mismo proyecto simultáneamente
- ▶ Las ramas nos permiten mantener el progreso concurrentemente



Trabajando con ramas

El comando de git para crear una ramas es:

```
git checkout -b new_branch
```

El comando para cambiar a una rama ya existente será:

```
git checkout other_branch
```

Trabajando con ramas

Por qué debemos mezclar o unir las ramas?

- ▶ Tomamos como fuente de la verdad a la rama `main`
- ▶ Cada rama que se crea, debe ser para ejecutar una actividad en específico
- ▶ Una vez que la actividad se ha terminado o completado, se deben mezclar o unir los cambios a la rama principal `main` para mantener todo al día.

Trabajando con ramas

Para mezclar o unir las ramas ejecutaremos el comando:

```
git merge source destination
```

Por ejemplo, para unir la rama `testing` a `main` ejecutamos:

```
git merge testing main
```

Resumen

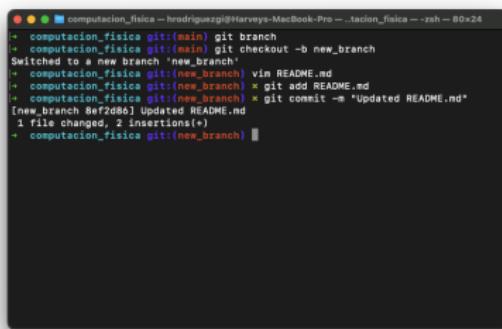
Creamos una rama y nos cambiamos a ella, hacemos una modificación en el archivo README.md, persistimos el cambio y por último mezclamos los cambios con la rama main

```
mi_proyecto ~ hrodriguezgj@Harveys-MacBook-Pro ~ A/mi_proyecto ~ zsh - 80x24
+ mi_proyecto git:(main) git branch
+ mi_proyecto git:(main) git checkout -b new_branch
Switched to a new branch 'new_branch'.
+ mi_proyecto git:(new_branch) vim README.md
+ mi_proyecto git:(new_branch) x git add README.md
+ mi_proyecto git:(new_branch) x git commit -m "Updated README.md"
[New branch ca321a4] Updated README.md
1 file changed, 3 insertions(+)
+ mi_proyecto git:(new_branch) |
```

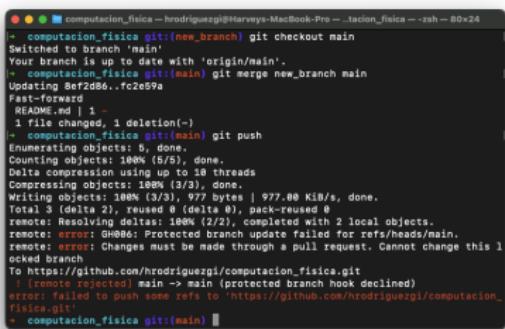
```
mi_proyecto ~ hrodriguezgj@Harveys-MacBook-Pro ~ A/mi_proyecto ~ zsh - 80x24
+ mi_proyecto git:(new_branch) git checkout main
Switched to branch 'main'
+ mi_proyecto git:(main) git merge new_branch main
Updating 80075c4..ca321a4
Fast-forward
 README.md | 3 +++
 1 file changed, 3 insertions(+)
+ mi_proyecto git:(main) |
```

Pull Request - Merge Request

En algunos repositorios remotos, no es permitido realizar la mezcla o unión de una rama contra la principal `main`, por lo que es necesario realizar el proceso de pull request o merge request.



```
+ computacion_fisica git:(main) git branch
computacion_fisica git:(main) git checkout -b new_branch
Switched to branch 'new_branch'
+ computacion_fisica git:(new_branch) vim README.md
+ computacion_fisica git:(new_branch) x git add README.md
+ computacion_fisica git:(new_branch) x git commit -m "Updated README.md"
[New branch 8ef2d8b] Updated README.md
1 file changed, 2 insertions(+)
+ computacion_fisica git:(new_branch) |
```



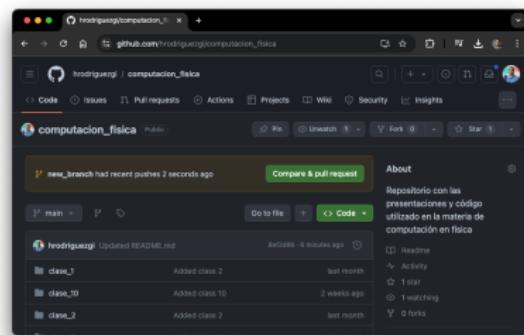
```
+ computacion_fisica git:(new_branch) git checkout main
Switched to branch 'main'
Your branch is up-to-date with 'origin/main'.
+ computacion_fisica git:(main) git merge new_branch main
Updating 8ef2d8b..fc2e59a
Fast-forward
 README.md | 1 -
 1 file changed, 1 deletion(-)
+ computacion_fisica git:(main) git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 53 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote: error: GH006: Protected branch update failed for refs/heads/main.
remote: error: Changes must be made through a pull request. Cannot change this locked branch
To https://github.com/hrodriguezgi/computacion_fisica.git
 ! [remote rejected] main -> main (protected branch hook declined)
error: failed to push some refs to 'https://github.com/hrodriguezgi/computacion_fisica.git'
+ computacion_fisica git:(main) |
```

Pull Request - Merge Request

Por lo tanto, primero deberemos publicar la rama que hemos creado, para que en la interfaz web, se pueda activar la opción de pull request:

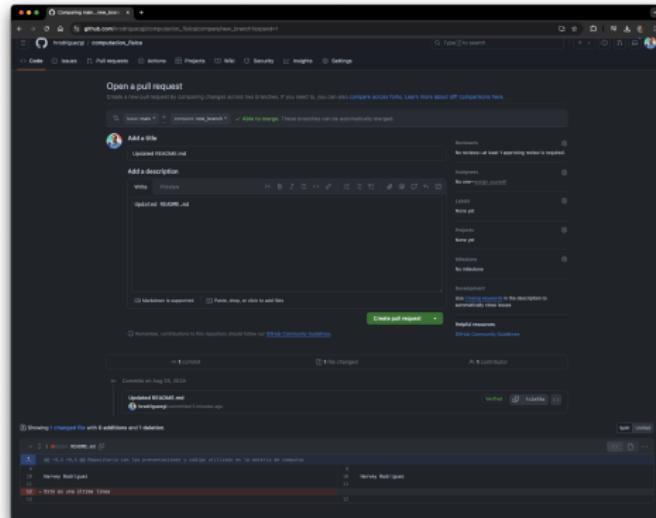
```
computacion_fisica$ git checkout new_branch
Switched to branch 'new_branch'
computacion_fisica$ git push --set-upstream origin new_branch

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 10 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 977 bytes | 977.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'new_branch' on GitHub by visiting:
remote:   https://github.com/hrodiguezgi/computacion_fisica/pull/new/new_bra
nch
remote:
To https://github.com/hrodiguezgi/computacion_fisica.git
 * [new branch]      new_branch -> new_branch
branch 'new_branch' set up to track 'origin/new_branch'.
computacion_fisica$
```



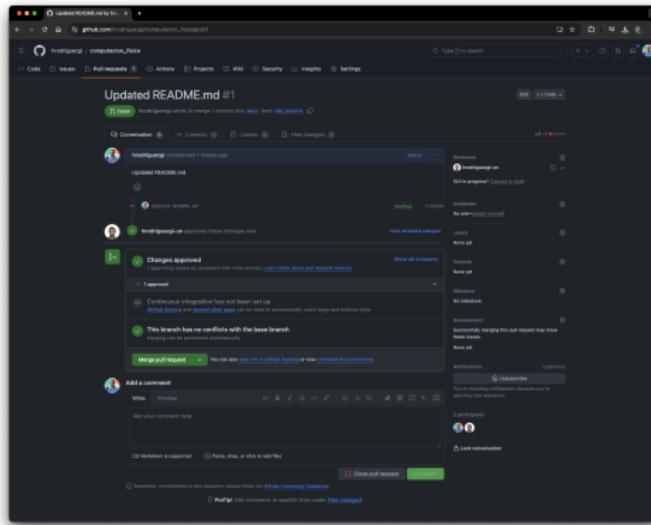
Pull Request - Merge Request

En esta pantalla se visualiza: un título del cambio junto con un breve resumen de él. Los commits que se realizaron y los cambios que estos implican:



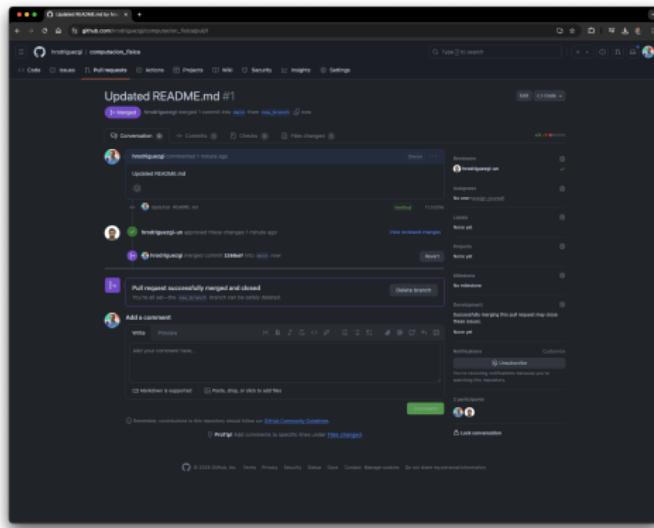
Pull Request - Merge Request

Los cambios deben ser aprobados por el dueño o administrador del repositorio para que se pueda proceder con la mezcla o unión con la rama main



Pull Request - Merge Request

Una vez realizado la mezcla o unión podemos eliminar la rama temporal que fue creada para la actividad. Tener en cuenta que esto solo la elimina del repositorio remoto



Pull Request - Merge Request

Finalmente en nuestro equipo, deberemos incorporar o actualizar los cambios que fueron llevados a la rama `main` del repositorio remoto

```
● ● ● computacion_fisica ~ hrodriguezgi@Harveys-MacBook-Pro ~..tacion_fisica ~ zsh ~ 80x24
+ computacion_fisica git:(new_branch) git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
+ computacion_fisica git:(main) git fetch
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 986 bytes | 453.00 KiB/s, done.
From https://github.com/hrodriguezgi/computacion_fisica
  8ef2d86..3280bdf main      -> origin/main
+ computacion_fisica git:(main) git pull
Updating fc2e59a..3280bdf
Fast-forward
+ computacion_fisica git:(main) git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
+ computacion_fisica git:(main)
```

Vamos a practicar

- ▶ Ingresar a <https://github.com/>
- ▶ Clonar el repositorio de `computacion_fisica`
- ▶ Crear una rama con el nombre de usuario del ssh de cada uno
- ▶ En la carpeta `clase_14/actividad` crear un archivo vacío con el nombre de usuario del ssh de cada uno
- ▶ Crear un pull request para solicitar incluir el cambio en el repositorio

