

EJERCICIOS

Ejercicio 1: Manejo avanzado de ramas y resolución de conflictos

Previamente, en la rama main se creó el archivo main.py y se añadió al área de preparación. Luego, se realizó el commit con un mensaje.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> echo "print('Hello World')" > main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add .
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Add main.py"
[main 84c3b0b] Add main.py
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.py
```

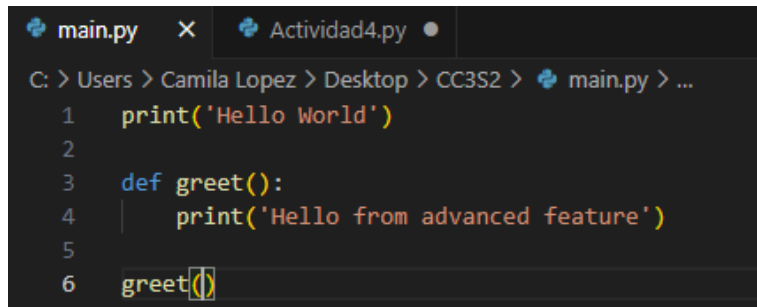
1. Crear una nueva rama para una característica:

Se creó una nueva rama desde la rama main y luego se cambió a esa rama.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git branch feature/advanced-feature
PS C:\Users\Camila Lopez\Desktop\CC3S2> git checkout feature/advanced-feature
Switched to branch 'feature/advanced-feature'
```

2. Modificar archivos en la nueva rama

Se modificó el contenido de main.py para incluir una función adicional.



```
main.py x Actividad4.py
C: > Users > Camila Lopez > Desktop > CC3S2 > main.py > ...
1 print('Hello World')
2
3 def greet():
4     print('Hello from advanced feature')
5
6 greet()
```

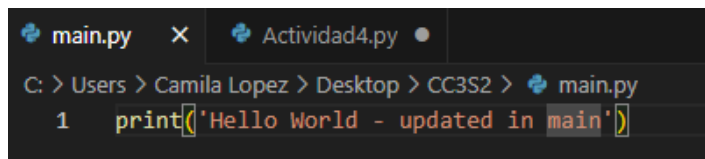
Luego, se añadió y se confirmaron los cambios

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Add greet function in advanced feature"
[feature/advanced-feature 0a68787] Add greet function in advanced feature
1 file changed, 0 insertions(+), 0 deletions(-)
```

3. Simular un desarrollo paralelo en la rama main

Se cambió a la rama main y se editó el archivo main.py

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> 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)
PS C:\Users\Camila Lopez\Desktop\CC3S2> code main.py
```



```
main.py x Actividad4.py
C: > Users > Camila Lopez > Desktop > CC3S2 > main.py
1 print('Hello World - updated in main')
```

Se añadió y se confirmó dicho cambio

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> code main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Update main.py message in main branch"
[main 2ffc203] Update main.py message in main branch
1 file changed, 1 insertion(+), 1 deletion(-)
```

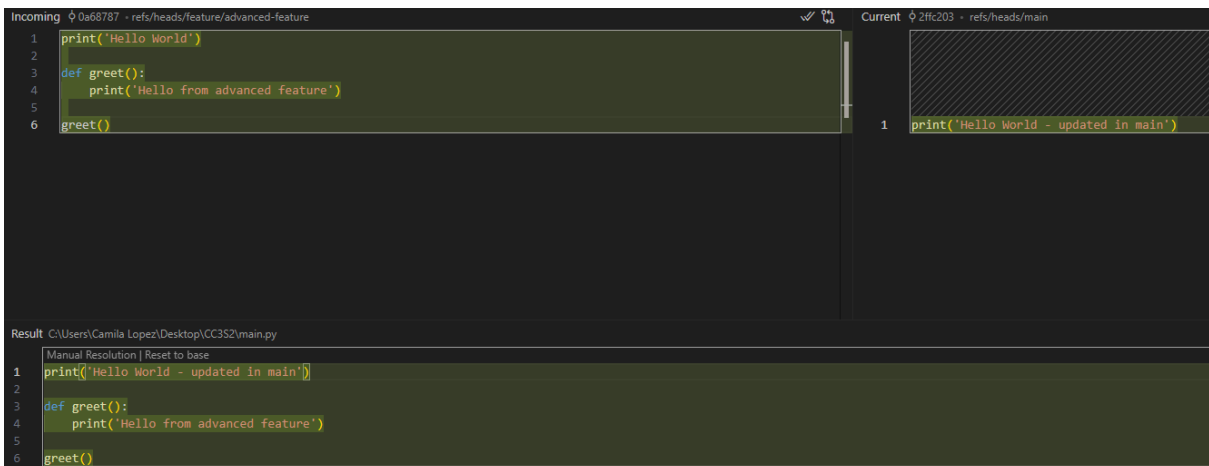
4. Intentar fusionar la rama feature/advanced-feature en main

Se intentó fusionar ambas ramas, sin embargo no se realizó la fusión automática por conflictos con el archivo main.py

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git merge feature/advanced-feature
warning: Cannot merge binary files: main.py (HEAD vs. feature/advanced-feature)
Auto-merging main.py
CONFLICT (add/add): Merge conflict in main.py
Automatic merge failed; fix conflicts and then commit the result.
```

5. Resolver el conflicto de fusión

Se resolvió el conflicto de manera manual en el IDE.



Luego, se añadió el archivo resuelto y se completó la fusión con un commit.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Resolve merge conflict between main and feature/advanced-feature"
[main 5251cf6] Resolve merge conflict between main and feature/advanced-feature
```

6. Eliminar la rama fusionada

Una vez realizada la fusión, se eliminó la rama feature/advanced-feature.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git branch -d feature/advanced-feature
Deleted branch feature/advanced-feature (was 0a68787).
```

Ejercicio 2: Exploración y manipulación del historial de commits

1. Exploración y manipulación del historial de commits

Se exploró el historial de commits a detalle

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log -p
```

Este commit finaliza la fusión entre las ramas, integrando los cambios del commit 2ffc203 de la rama main y el commit 0a68787 de la rama feature/advanced-feature, luego de haber resuelto manualmente el conflicto en el archivo main.py.

```
commit 5251cf626aa17aedaa77dc9f9893cbe33f01959f (HEAD -> main)
Merge: 2ffc203 0a68787
Author: Ariana Lopez <ariana.lopez.j@uni.pe>
Date: Sun Apr 6 22:44:58 2025 -0500

    Resolve merge conflict between main and feature/advanced-feature
```

Se actualizó una línea en el archivo main.py dentro de la rama main. Además, se muestra el contenido del archivo antes del cambio y después del cambio.

```
commit 2ffc20348121304259b757eb2ef165b71e8e3e26
Author: Ariana Lopez <ariana.lopez.j@uni.pe>
Date: Sun Apr 6 22:37:25 2025 -0500

    Update main.py message in main branch

diff --git a/main.py b/main.py
index d3e46f2..78f55b5 100644
--- a/main.py
+++ b/main.py
@@ -1,1 @@
- print('Hello World')
\ No newline at end of file
+ print('Hello World - updated in main')
\ No newline at end of file
```

2. Filtrar commits por autor

Se listaron exclusivamente los commits realizados por el autor Ariana

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --author="Ariana"
commit 5251cf626aa17aedaa77dc9f9893cbe33f01959f (HEAD -> main)
Merge: 2ffc203 0a68787
Author: Ariana Lopez <ariana.lopez.j@uni.pe>
Date: Sun Apr 6 22:44:58 2025 -0500

    Resolve merge conflict between main and feature/advanced-feature

commit 2ffc20348121304259b757eb2ef165b71e8e3e26
Author: Ariana Lopez <ariana.lopez.j@uni.pe>
Date: Sun Apr 6 22:37:25 2025 -0500

    Update main.py message in main branch

commit 0a687870f3defa334bb08c94719774293f725993
Author: Ariana Lopez <ariana.lopez.j@uni.pe>
Date: Sun Apr 6 22:35:40 2025 -0500

    Add greet function in advanced feature
```

3. Revertir un commit

Se revirtió el último commit que corresponde a una fusión utilizando el comando `git revert -m 1 HEAD`. La opción `-m 1` indica que se desea mantener el historial del primer padre, es decir, el historial de la rama principal (main) al deshacer el merge.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git revert -m 1 HEAD
[main e336b26] Revert "Resolve merge conflict between main and feature/advanced-feature"
1 file changed, 1 insertion(+), 6 deletions(-)
```

Verificamos que el commit de reversión esté en el historial

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log
commit e336b26b90ec66d5f9e0b2264720f2f93791c553 (HEAD -> main)
Author: Ariana Lopez <ariana.lopez.j@uni.pe>
Date: Sun Apr 6 22:57:09 2025 -0500

    Revert "Resolve merge conflict between main and feature/advanced-feature"

    This reverts commit 5251cf626aa17aadaa77dc9f9893cbe33f01959f, reversing
    changes made to 2ffc20348121304259b757eb2ef165b71e8e3e26.
```

4. Rebase interactivo

Previamente, se creó el archivo `prueba.py`, el cual fue añadido y confirmado con un commit. Luego, se realizaron dos modificaciones en el archivo, cada una de las cuales se añadió y guardó mediante un nuevo commit.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add prueba.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Archivo Prueba"
[main 16a2336] Archivo Prueba
1 file changed, 1 insertion(+)
create mode 100644 prueba.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> code prueba.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add prueba.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Archivo Prueba V1"
[main 0417726] Archivo Prueba V1
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Camila Lopez\Desktop\CC3S2> code prueba.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add prueba.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Archivo Prueba V2"
[main 7467182] Archivo Prueba V2
1 file changed, 1 insertion(+), 1 deletion(-)
```

Se realizó un rebase interactivo de los últimos 3 commits

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git rebase -i HEAD~3
[detached HEAD 4df5a5e] Archivo Prueba
Date: Sun Apr 6 23:43:49 2025 -0500
1 file changed, 1 insertion(+)
create mode 100644 prueba.py
Successfully rebased and updated refs/heads/main.
```

En el editor, se utilizó la opción `squash` para combinar cada commit con el anterior. El último commit mantiene la opción `pick` porque no hay otro commit previo con el que fusionarlo.

```
pick 16a2336 Archivo Prueba
squash 0417726 Archivo Prueba V1
squash 7467182 Archivo Prueba V2
```

5. Visualización gráfica del historial

(HEAD -> main) indica que actualmente estamos en la rama main.

Debido al rebase realizado, los tres últimos commits ahora aparecen combinados en un solo commit (e336b26) dentro del historial.

Los commit 0a68787 y 84c3b0b están desplazados hacia la derecha porque pertenecen a la rama feature/advanced-feature.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --graph --oneline --all
* 4df5a5e (HEAD -> main) Archivo Prueba
* e336b26 Revert "Resolve merge conflict between main and feature/advanced-feature"
* 5251cf6 Resolve merge conflict between main and feature/advanced-feature
| \
| * 0a68787 Add greet function in advanced feature
| * 84c3b0b Add main.py
* | 2ffc203 Update main.py message in main branch
* | 0cb25ff Add main.py
|/
```

Ejercicio 3: Creación y gestión de ramas desde commits específicos

1. Crear una nueva rama desde un commit específico

Visualizamos de manera compacta en una sola línea cada comando

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --oneline
4df5a5e (HEAD -> main) Archivo Prueba
e336b26 Revert "Resolve merge conflict between main and feature/advanced-feature"
5251cf6 Resolve merge conflict between main and feature/advanced-feature
2ffc203 Update main.py message in main branch
0a68787 Add greet function in advanced feature
0cb25ff Add main.py
84c3b0b Add main.py
53b7cf0 (origin/main, origin/HEAD) Eliminar archivo Actividad4.py
60cee2f Actividad 3
59e2675 Actividad2
d9ed92f Actividad 1
```

Escogemos el commit 84c3b0b, que corresponde a la primera vez que se agregó el archivo main.py, y a partir de ese commit creamos una nueva rama llamada bugfix/rollback-feature.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git branch bugfix/rollback-feature 84c3b0b
PS C:\Users\Camila Lopez\Desktop\CC3S2> git checkout bugfix/rollback-feature
Switched to branch 'bugfix/rollback-feature'
```

2. Modificar y confirmar cambios en la nueva rama

Se realizó una modificación en main.py que simula una corrección de errores, luego se añadió y confirmó los cambios.

```
main.py M X  Merging: main.py M
C: > Users > Camila Lopez > Desktop > CC3S2 > m
1  print('Hello World')
2
3  def greet():
4  |  print('Fixed bug in feature')
```

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Fix bug in rollback feature"
[bugfix/rollback-feature 91a67e8] Fix bug in rollback feature
1 file changed, 0 insertions(+), 0 deletions(-)
```

3. Fusionar los cambios en la rama principal

Se cambió a la rama main y se intentó fusionar la rama bugfix/rollback-feature.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 7 commits.
(use "git push" to publish your local commits)
PS C:\Users\Camila Lopez\Desktop\CC3S2> git merge bugfix/rollback-feature
warning: Cannot merge binary files: main.py (HEAD vs. bugfix/rollback-feature)
Auto-merging main.py
CONFLICT (content): Merge conflict in main.py
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Camila Lopez\Desktop\CC3S2> code main.py
```

Se solucionó el conflicto de manera manual, luego se añadió el archivo corregido al área de preparación y se confirmó el cambio, finalizando el proceso de fusión.

```
Result C:\Users\Camila Lopez\Desktop\CC3S2\main.py
Current + Incoming | Remove Current | Remove Incoming
1 print('Hello World - updated in main')
2 print('Hello World')
3
4 def greet():
5     print('Fixed bug in feature')
```

4. Explorar el historial después de la fusión

Se visualiza la fusión del commit 91a67e8 de la rama bugfix/rollback-feature en la rama main, quedando registrado como el commit bca56e3.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --graph --oneline
* bca56e3 (HEAD -> main) resolve merge
| \
| * 91a67e8 (bugfix/rollback-feature) Fix bug in rollback feature
| * 4df5a5e Archivo Prueba
| * e336b26 Revert "Resolve merge conflict between main and feature/advanced-feature"
| * 5251cf6 Resolve merge conflict between main and feature/advanced-feature
| \ \
| * 0a68787 Add greet function in advanced feature
| | /
| * 84c3b0b Add main.py
| * 2ffc203 Update main.py message in main branch
| * 0cb25ff Add main.py
```

5. Eliminar la rama bugfix/rollback-feature

Después de la fusión, se eliminó la rama bugfix/rollback-feature

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git branch -d bugfix/rollback-feature
Deleted branch bugfix/rollback-feature (was 91a67e8).
```


Ejercicio 4: Manipulación y restauración de commits con git reset y git restore

1. Hacer cambios en el archivo main.py

Se realizó un cambio en el archivo main.py, se añadió y confirmó el cambio.

```
C: > Users > Camila Lopez > Desktop > CC3S2 > main.py > ...
```

```
1 print('Hello World - updated in main')
2 print('Hello World')
3
4 print('This change will be reset')
5 def greet():
6     print('Fixed bug in feature')
```

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> code main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add main.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Introduce a change to be reset"
[main 4ca03b2] Introduce a change to be reset
1 file changed, 1 insertion(+)
```

2. Usar git reset para deshacer el commit

Historial de commits antes de git reset

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --oneline
4ca03b2 (HEAD -> main) Introduce a change to be reset
bca56e3 resolve merge
```

Deshacemos el último commit, luego verificamos que dicho commit ya no se encuentra en el historial.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git reset --hard HEAD~1
HEAD is now at bca56e3 resolve merge
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --oneline
bca56e3 (HEAD -> main) resolve merge
91a67e8 Fix bug in rollback feature
```

Luego, verificamos que el cambio en el archivo también ha sido revertido

```
main.py X Merging: main.py Merging: main.py
C: > Users > Camila Lopez > Desktop > CC3S2 > main.py > ...
1 print('Hello World - updated in main')
2 print('Hello World')
3
4 def greet():
5     print('Fixed bug in feature')
```

3. Usar git restore para deshacer cambios no confirmados

Se realizó un cambio en README.md pero no se confirmó, verificamos su estado con git status

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> echo "Another line in README" >> README.md
PS C:\Users\Camila Lopez\Desktop\CC3S2> git status
On branch main
Your branch is ahead of 'origin/main' by 10 commits.
  (use "git push" to publish your local commits)

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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Luego, usamos git restore para deshacer este cambio no confirmado. Nuevamente verificamos con git status ya no hay cambios por confirmar.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git restore README.md
PS C:\Users\Camila Lopez\Desktop\CC3S2> git status
On branch main
Your branch is ahead of 'origin/main' by 10 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Ejercicio 5: Trabajo colaborativo y manejo de Pull Requests

1. Crear un nuevo repositorio remoto

Se clonó el repositorio del curso CC3S2

```
PS C:\Users\Camila Lopez> git clone https://github.com/aclj20/CC3S2.git
Cloning into 'CC3S2'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 2), reused 11 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 1.22 MiB | 1.30 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

2. Crear una nueva rama para desarrollo de una característica

Se creó y se cambió a una nueva rama feature/team-feature

```
PS C:\Users\Camila Lopez\CC3S2> git branch feature/team-feature
PS C:\Users\Camila Lopez\CC3S2> git checkout feature/team-feature
Switched to branch 'feature/team-feature'
```

3. Realizar cambios y enviar la rama al repositorio remoto

Se creó un archivo collaboration.py, se añadió y confirmó el cambio.


```
PS C:\Users\Camila Lopez\CC3S2> echo "print('Collaboration is key!')" > collaboration.py
PS C:\Users\Camila Lopez\CC3S2> git add .
PS C:\Users\Camila Lopez\CC3S2> git commit -m "Add collaboration script"
[feature/team-feature 656a14f] Add collaboration script
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 collaboration.py
```


Se enviaron los cambios al repositorio remoto.

```
PS C:\Users\Camila Lopez\CC3S2> git push origin feature/team-feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 345 bytes | 345.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/team-feature' on GitHub by visiti
ng:
remote:      https://github.com/aclj20/CC3S2/pull/new/feature/team-feature
remote:
To https://github.com/aclj20/CC3S2.git
 * [new branch]      feature/team-feature -> feature/team-feature
```

4. Abrir un Pull Request

Abrimos un PR para fusionar las ramas y añadimos una descripción de los cambios realizados.

 **feature/team-feature** had recent pushes 3 seconds ago [Compare & pull request](#)

Add a title

Add collaboration script

Add a description

Write

Preview

H

Se agrego el script collaboration.py

5. Revisar y fusionar el Pull Request

Se revisó el código y comentó en el PR, luego se fusionó el PR en la rama main

Commit message

Merge pull request #1 from aclj20/feature/team-feature

Extended description

Add collaboration script

This commit will be authored by ariana.lopez.j@uni.pe.

[Confirm merge](#) [Cancel](#)

Pull request successfully merged and closed

You're all set — the `feature/team-feature` branch can be safely deleted.

[Delete branch](#)

6. Eliminar la rama remota y local

Se eliminó la rama en el repositorio remoto y local

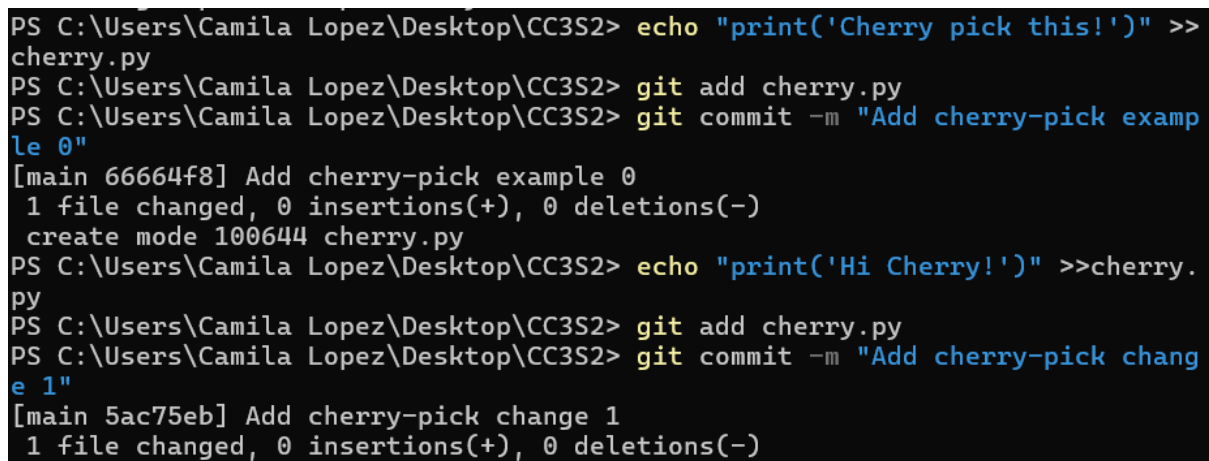


```
PS C:\Users\Camila Lopez\CC3S2> git branch -D feature/team-feature
Deleted branch feature/team-feature (was 656a14f).
```

Ejercicio 6: Cherry-Picking y Git Stash

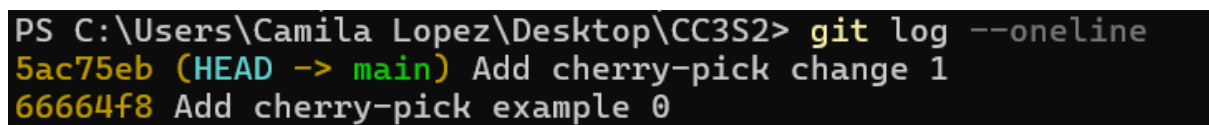
1. Hacer cambios en cherry.py y confirmarlos

Se realizaron 2 cambios en el archivo cherry.py, se añadieron y se confirmaron.



```
PS C:\Users\Camila Lopez\Desktop\CC3S2> echo "print('Cherry pick this!')" >> cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Add cherry-pick example 0"
[main 66664f8] Add cherry-pick example 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> echo "print('Hi Cherry!')" >>cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Add cherry-pick change 1"
[main 5ac75eb] Add cherry-pick change 1
1 file changed, 0 insertions(+), 0 deletions(-)
```

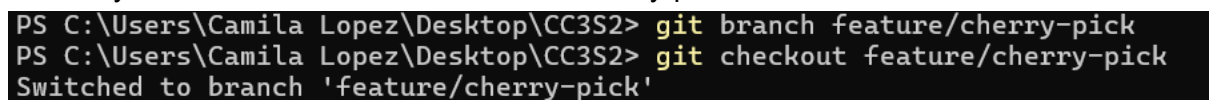
Se visualizan los commits en el historial



```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --oneline
5ac75eb (HEAD -> main) Add cherry-pick change 1
66664f8 Add cherry-pick example 0
```

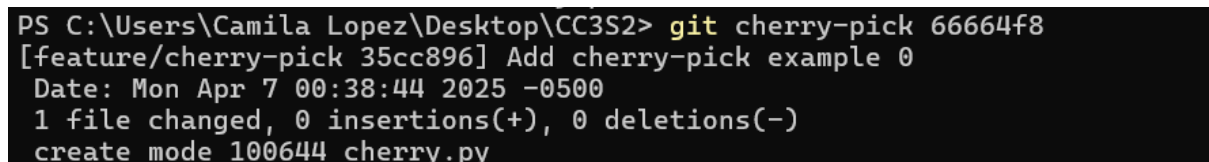
2. Crear una nueva rama y aplicar el commit específico

Se creó y cambió a una nueva rama feature/cherry-pick



```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git branch feature/cherry-pick
PS C:\Users\Camila Lopez\Desktop\CC3S2> git checkout feature/cherry-pick
Switched to branch 'feature/cherry-pick'
```

Se selecciona un comando en específico y lo aplica nuevamente a la rama actual feature/cherry-pick



```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git cherry-pick 66664f8
[feature/cherry-pick 35cc896] Add cherry-pick example 0
Date: Mon Apr 7 00:38:44 2025 -0500
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 cherry.py
```

3. Guardar temporalmente cambios no confirmados

Se realizó un cambio en cherry.py pero no se confirmó. Luego, se guardó temporalmente este cambio con git stash.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> echo "This change is stashed" >> cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git status
On branch feature/cherry-pick
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:   cherry.py

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Camila Lopez\Desktop\CC3S2> git stash
Saved working directory and index state WIP on feature/cherry-pick: 4d71dc5
Change the file cherry.py
```

4. Aplicar los cambios guardados

Se realizó otro cambio en cherry.py, esta vez si se añadió y confirmó dicho cambio.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> echo "Hi this is change" >> cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git add cherry.py
PS C:\Users\Camila Lopez\Desktop\CC3S2> git commit -m "Change the file cherry.py"
[feature/cherry-pick 4d71dc5] Change the file cherry.py
1 file changed, 0 insertions(+), 0 deletions(-)
```

Recuperamos el cambio temporal anteriormente guardado con git stash pop. Se restauró el cambio guardado y luego se eliminó ese stash.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git stash pop
On branch feature/cherry-pick
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:   cherry.py

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (ddb2e8f4aad731884a4e3adf716cc08abd57daf7)
```

5. Revisar el historial y confirmar la correcta aplicación de los cambios

Se verificó que los commits hayan sido correctamente aplicados.

```
PS C:\Users\Camila Lopez\Desktop\CC3S2> git log --oneline
4d71dc5 (HEAD -> feature/cherry-pick) Change the file cherry.py
35cc896 Add cherry-pick example 0
98828d0 Add cherry-pick change
f42d1f4 Add cherry-pick example
5-8888-111-85485-617
```

PREGUNTAS

- **¿Cómo te ha ayudado Git a mantener un historial claro y organizado de tus cambios?**

Git me ha permitido escoger qué archivos modificados puedo elegir confirmar cambios y cuáles no, lo que me resulta útil para subir pequeñas actualizaciones pequeñas sin tener que haber terminado todo el trabajo. Además, me ha ayudado a mantener un seguimiento detallado de los cambios aplicados, lo que me ayuda a visualizar un historial más organizado.

- **¿Qué beneficios ves en el uso de ramas para desarrollar nuevas características o corregir errores?**

El uso de ramas me permite implementar funcionalidades específicas o corregir errores sin afectar directamente la rama principal, es decir, evita modificar el código principal mientras trabajo en una nueva rama. Una vez completada la implementación, puedo fusionar ambas ramas de manera segura asegurando la integridad del proyecto.