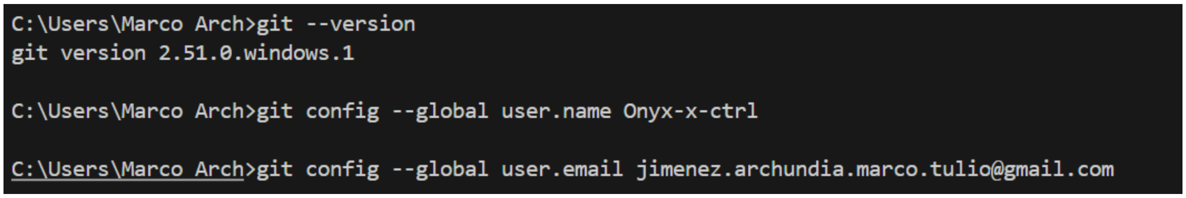
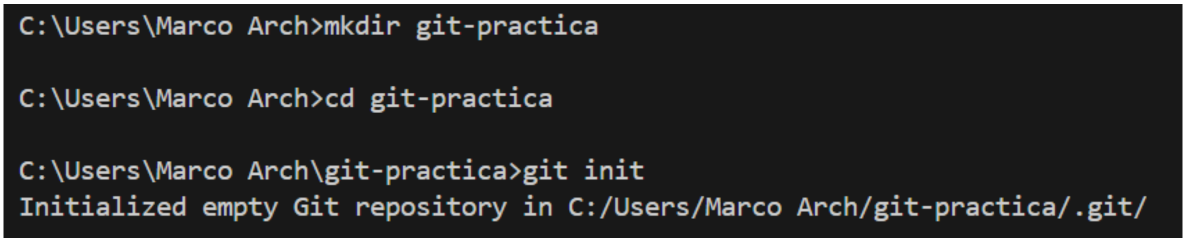
Jiménez Archundia Marco Tulio

1. Configuración Inicial



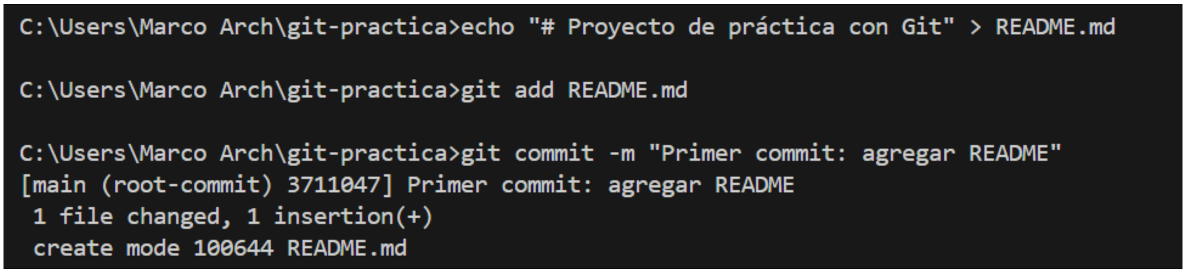
```
C:\Users\Marco Arch>git --version
git version 2.51.0.windows.1

C:\Users\Marco Arch>git config --global user.name Onyx-x-ctrl

C:\Users\Marco Arch>git config --global user.email jimenez.archundia.marco.tulio@gmail.com
```

2. Crear un repositorio nuevo



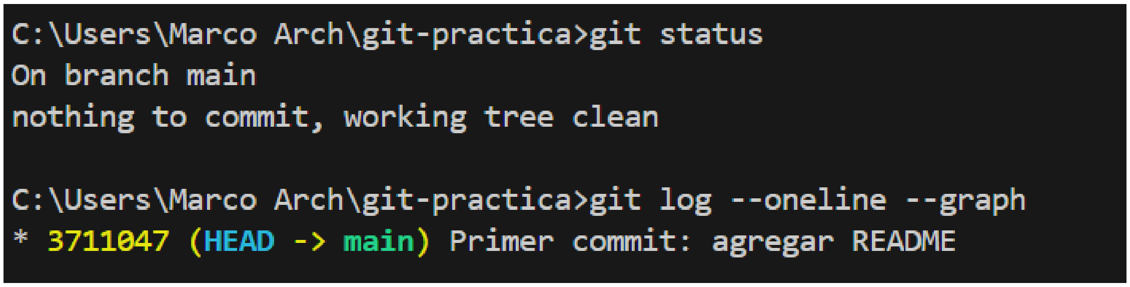
```
C:\Users\Marco Arch>mkdir git-practica

C:\Users\Marco Arch>cd git-practica

C:\Users\Marco Arch\git-practica>git init
Initialized empty Git repository in C:/Users/Marco Arch/git-practica/.git/
```

3. Primer archivo y primer commit



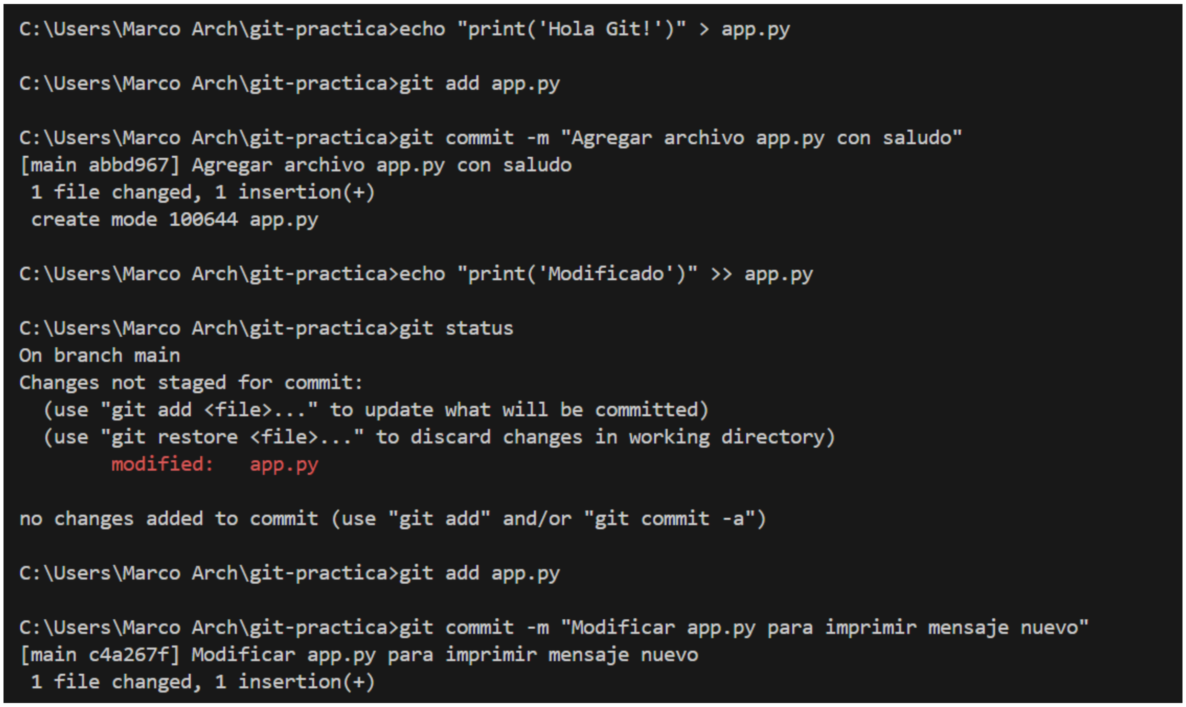
```
C:\Users\Marco Arch\git-practica>echo "# Proyecto de práctica con Git" > README.md

C:\Users\Marco Arch\git-practica>git add README.md

C:\Users\Marco Arch\git-practica>git commit -m "Primer commit: agregar README"
[main (root-commit) 3711047] Primer commit: agregar README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

4. Revisar historial y estado



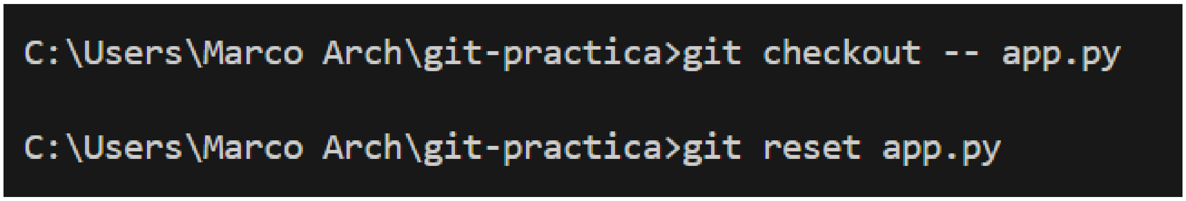
```
C:\Users\Marco Arch\git-practica>git status
On branch main
nothing to commit, working tree clean

C:\Users\Marco Arch\git-practica>git log --oneline --graph
* 3711047 (HEAD -> main) Primer commit: agregar README
```

5. Crear y modificar archivos



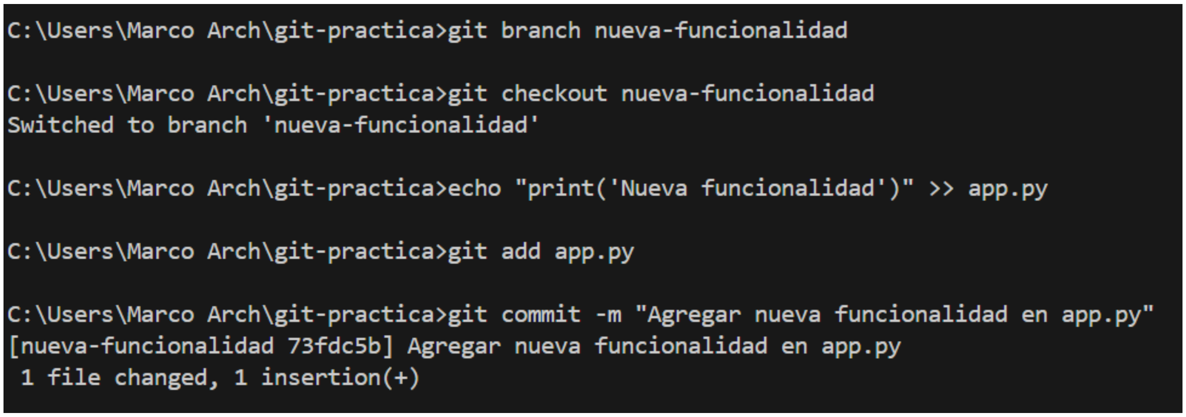
```
C:\Users\Marco Arch\git-practica>echo "print('Hola Git!')" > app.py

C:\Users\Marco Arch\git-practica>git add app.py

C:\Users\Marco Arch\git-practica>git commit -m "Agregar archivo app.py con saludo"
[main abbd967] Agregar archivo app.py con saludo
 1 file changed, 1 insertion(+)
 create mode 100644 app.py

C:\Users\Marco Arch\git-practica>echo "print('Modificado')" >> app.py

C:\Users\Marco Arch\git-practica>git status
On branch main
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:   app.py

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

C:\Users\Marco Arch\git-practica>git add app.py

C:\Users\Marco Arch\git-practica>git commit -m "Modificar app.py para imprimir mensaje nuevo"
[main c4a267f] Modificar app.py para imprimir mensaje nuevo
 1 file changed, 1 insertion(+)
```

6. Deshacer cambios



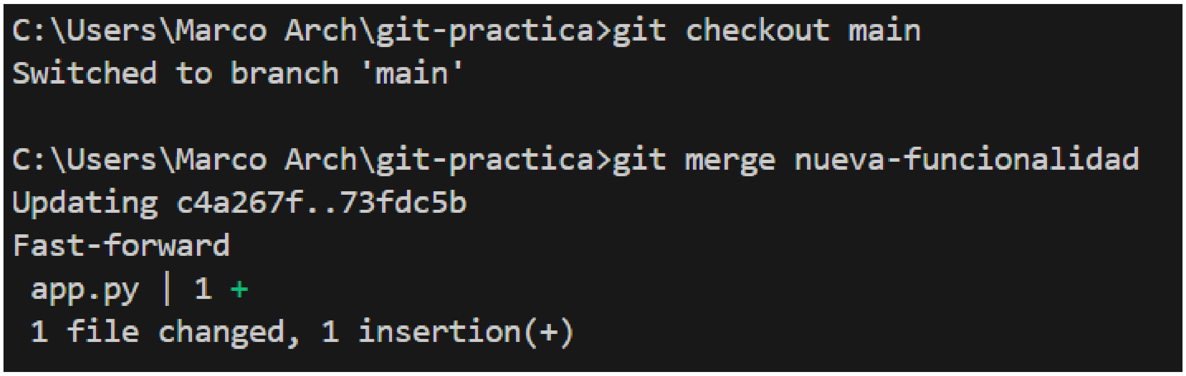
```
C:\Users\Marco Arch\git-practica>git checkout -- app.py

C:\Users\Marco Arch\git-practica>git reset app.py
```

7. Crear y cambiar de ramas



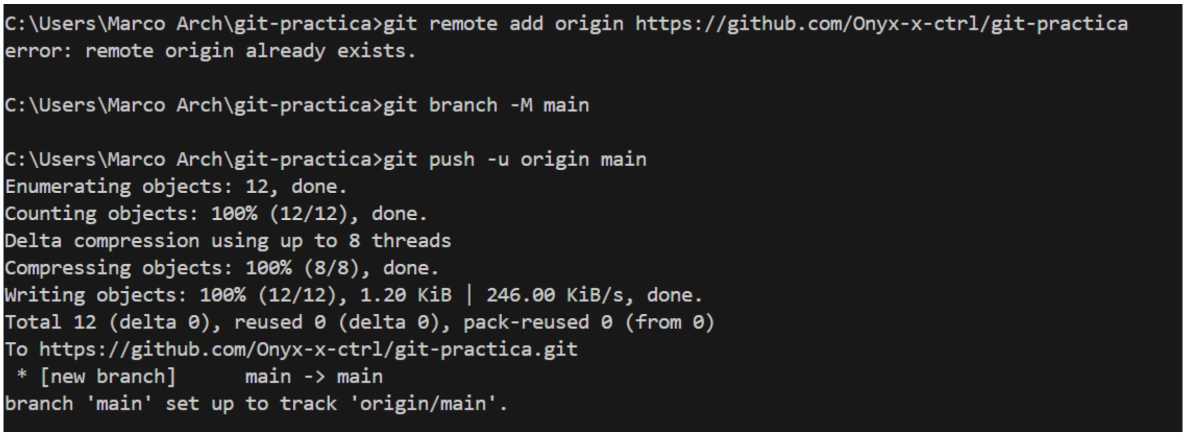
```
C:\Users\Marco Arch\git-practica>git branch nueva-funcionalidad

C:\Users\Marco Arch\git-practica>git checkout nueva-funcionalidad
Switched to branch 'nueva-funcionalidad'

C:\Users\Marco Arch\git-practica>echo "print('Nueva funcionalidad')" >> app.py

C:\Users\Marco Arch\git-practica>git add app.py

C:\Users\Marco Arch\git-practica>git commit -m "Agregar nueva funcionalidad en app.py"
[nueva-funcionalidad 73fdc5b] Agregar nueva funcionalidad en app.py
 1 file changed, 1 insertion(+)
```

8. Fusionar ramas (merge)



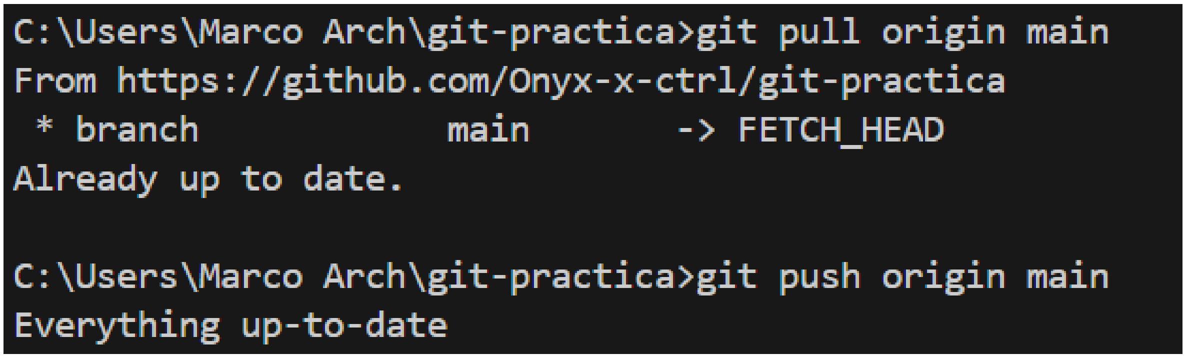
```
C:\Users\Marco Arch\git-practica>git checkout main
Switched to branch 'main'

C:\Users\Marco Arch\git-practica>git merge nueva-funcionalidad
Updating c4a267f..73fdc5b
Fast-forward
 app.py | 1 +
 1 file changed, 1 insertion(+)
```

9. Conectar con un repositorio remoto



```
C:\Users\Marco Arch\git-practica>git remote add origin https://github.com/Onyx-x-ctrl/git-practica
error: remote origin already exists.

C:\Users\Marco Arch\git-practica>git branch -M main

C:\Users\Marco Arch\git-practica>git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.20 KiB | 246.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Onyx-x-ctrl/git-practica.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

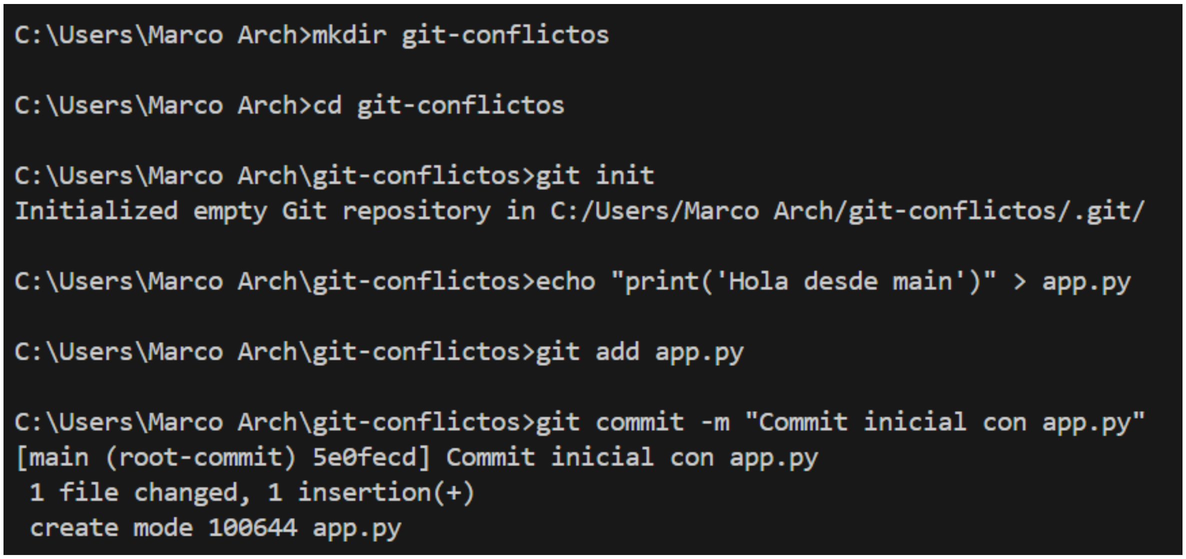
10. Actualizar y colaborar



```
C:\Users\Marco Arch\git-practica>git pull origin main
From https://github.com/Onyx-x-ctrl/git-practica
 * branch            main       -> FETCH_HEAD
Already up to date.

C:\Users\Marco Arch\git-practica>git push origin main
Everything up-to-date
```

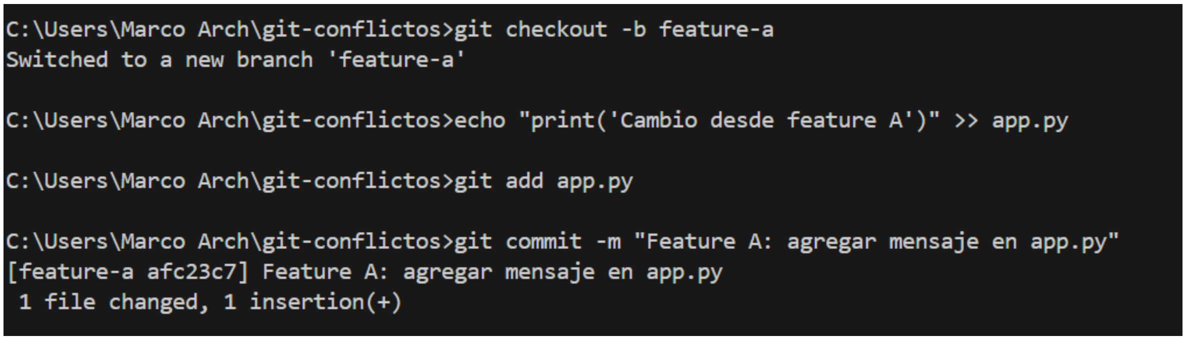
Ejercicio Guiado de Git.

1. Preparar entorno



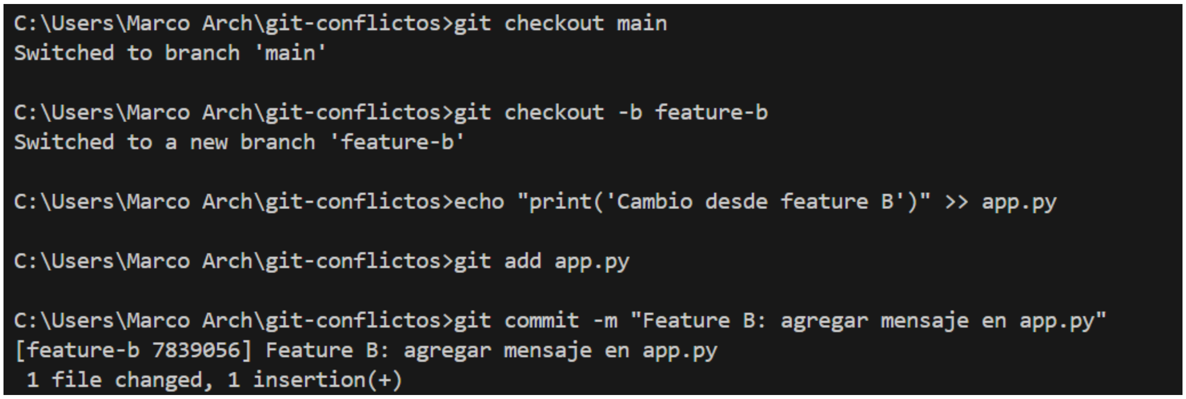
```
C:\Users\Marco Arch>mkdir git-conflictos

C:\Users\Marco Arch>cd git-conflictos

C:\Users\Marco Arch\git-conflictos>git init
Initialized empty Git repository in C:/Users/Marco Arch/git-conflictos/.git/

C:\Users\Marco Arch\git-conflictos>echo "print('Hola desde main')" > app.py

C:\Users\Marco Arch\git-conflictos>git add app.py

C:\Users\Marco Arch\git-conflictos>git commit -m "Commit inicial con app.py"
[main (root-commit) 5e0fecd] Commit inicial con app.py
 1 file changed, 1 insertion(+)
 create mode 100644 app.py
```

2. Crear y trabajar en feature-a



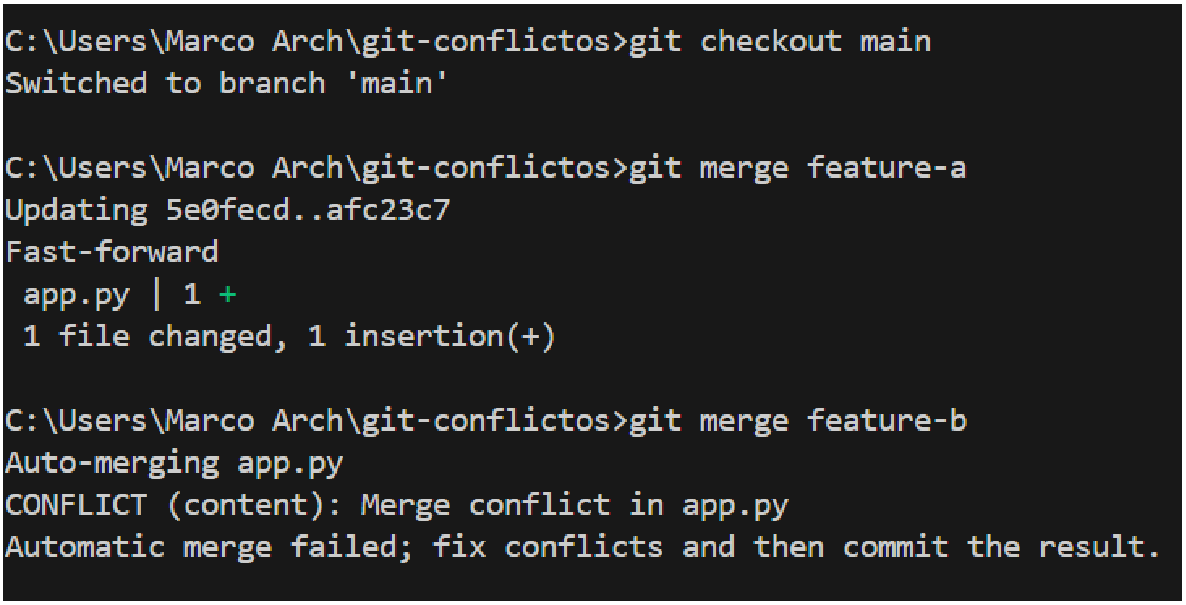
```
C:\Users\Marco Arch\git-conflictos>git checkout -b feature-a
Switched to a new branch 'feature-a'

C:\Users\Marco Arch\git-conflictos>echo "print('Cambio desde feature A')" >> app.py

C:\Users\Marco Arch\git-conflictos>git add app.py

C:\Users\Marco Arch\git-conflictos>git commit -m "Feature A: agregar mensaje en app.py"
[feature-a afc23c7] Feature A: agregar mensaje en app.py
 1 file changed, 1 insertion(+)
```

3. Crear y trabajar en feature-b



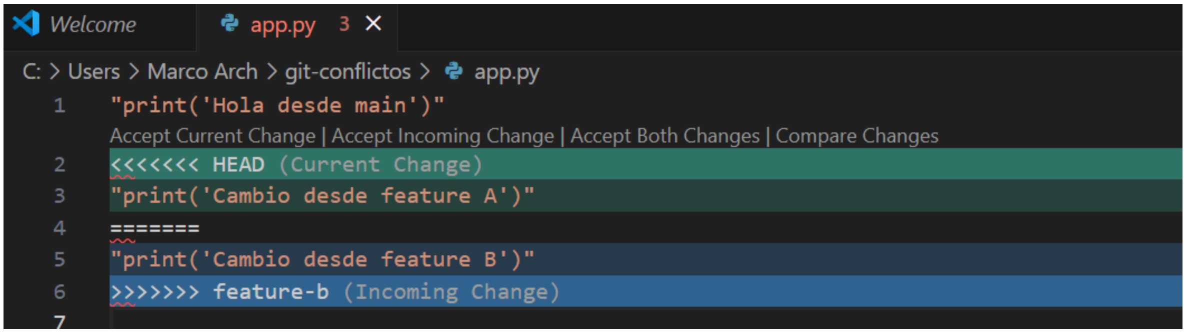
```
C:\Users\Marco Arch\git-conflictos>git checkout main
Switched to branch 'main'

C:\Users\Marco Arch\git-conflictos>git checkout -b feature-b
Switched to a new branch 'feature-b'

C:\Users\Marco Arch\git-conflictos>echo "print('Cambio desde feature B')" >> app.py

C:\Users\Marco Arch\git-conflictos>git add app.py

C:\Users\Marco Arch\git-conflictos>git commit -m "Feature B: agregar mensaje en app.py"
[feature-b 7839056] Feature B: agregar mensaje en app.py
 1 file changed, 1 insertion(+)
```

4. Simular conflicto

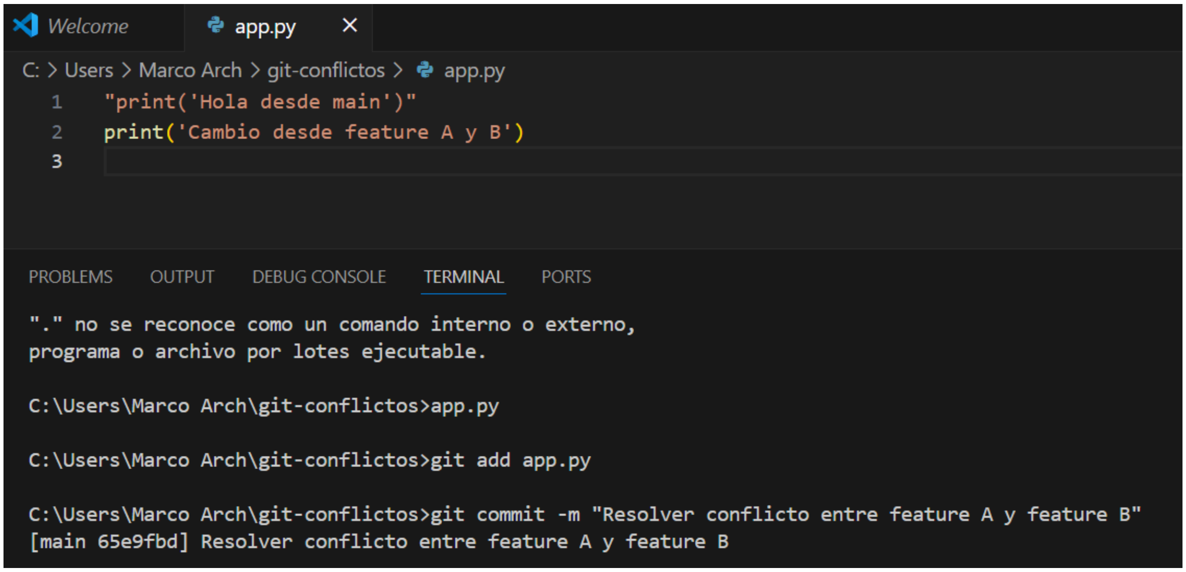


```
C:\Users\Marco Arch\git-conflictos>git checkout main
Switched to branch 'main'

C:\Users\Marco Arch\git-conflictos>git merge feature-a
Updating 5e0fecd..afc23c7
Fast-forward
 app.py | 1 +
 1 file changed, 1 insertion(+)

C:\Users\Marco Arch\git-conflictos>git merge feature-b
Auto-merging app.py
CONFLICT (content): Merge conflict in app.py
Automatic merge failed; fix conflicts and then commit the result.
```

5. Resolver conflicto


Welcome
app.py 3 ×
C: > Users > Marco Arch > git-conflictos > app.py
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

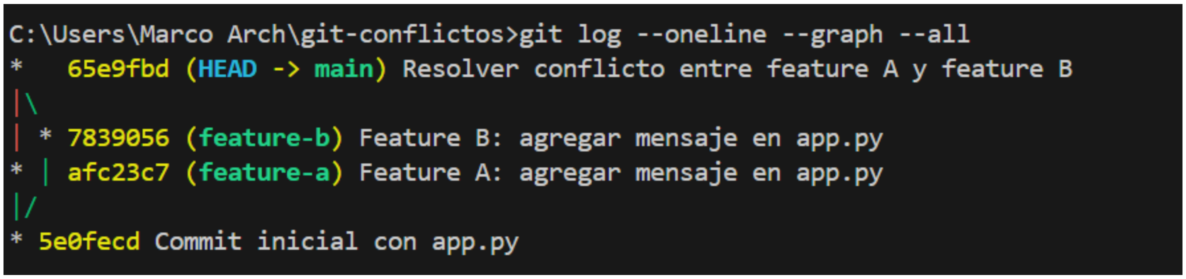
```
"print('Hola desde main')"
<<<<<<< HEAD (Current Change)
"print('Cambio desde feature A')"
=======
"print('Cambio desde feature B')"
>>>>>>> feature-b (Incoming Change)
```



```
"print('Hola desde main')"
print('Cambio desde feature A y B')
```

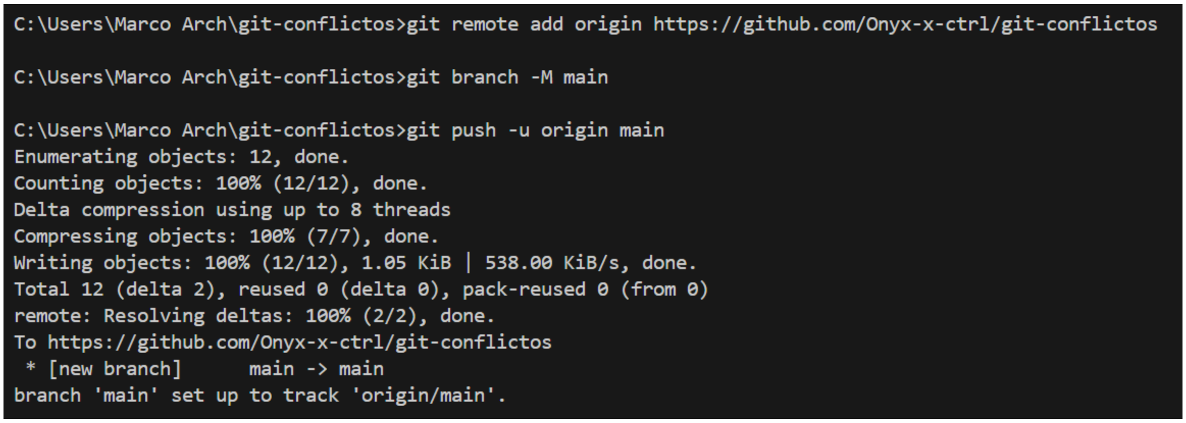
```
"." no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\Marco Arch\git-conflictos>app.py

C:\Users\Marco Arch\git-conflictos>git add app.py

C:\Users\Marco Arch\git-conflictos>git commit -m "Resolver conflicto entre feature A y feature B"
[main 65e9fbd] Resolver conflicto entre feature A y feature B
```

7. Ver historial gráfico



```
C:\Users\Marco Arch\git-conflictos>git log --oneline --graph --all
*   65e9fbd (HEAD -> main) Resolver conflicto entre feature A y feature B
|\
| * 7839056 (feature-b) Feature B: agregar mensaje en app.py
* | afc23c7 (feature-a) Feature A: agregar mensaje en app.py
|/
* 5e0fecd Commit inicial con app.py
```

8. Subir a remoto



```
C:\Users\Marco Arch\git-conflictos>git remote add origin https://github.com/Onyx-x-ctrl/git-conflictos

C:\Users\Marco Arch\git-conflictos>git branch -M main

C:\Users\Marco Arch\git-conflictos>git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 1.05 KiB | 538.00 KiB/s, done.
Total 12 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Onyx-x-ctrl/git-conflictos
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```