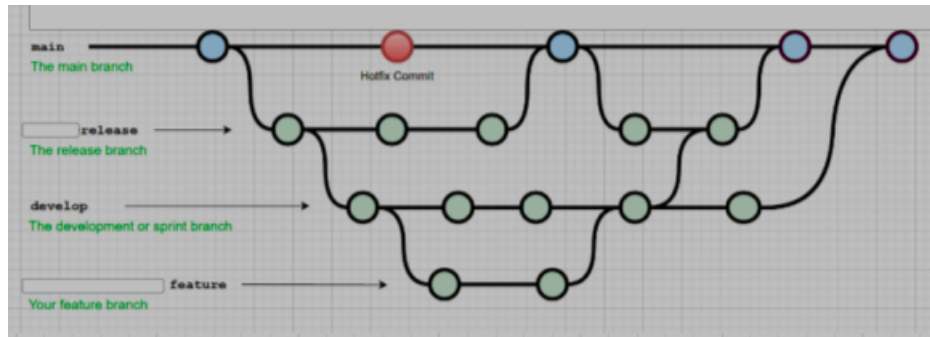


Ejercicio 4

Curso IA

En este documento tiene como intención documentar la recreación de un gitflow similar al siguiente:



1. Pasos iniciales

Inicialmente debemos crear y clonar el repositorio de git. Para ello nos dirigimos a la web github.com/new y creamos el repositorio sobre el cual vamos a trabajar.

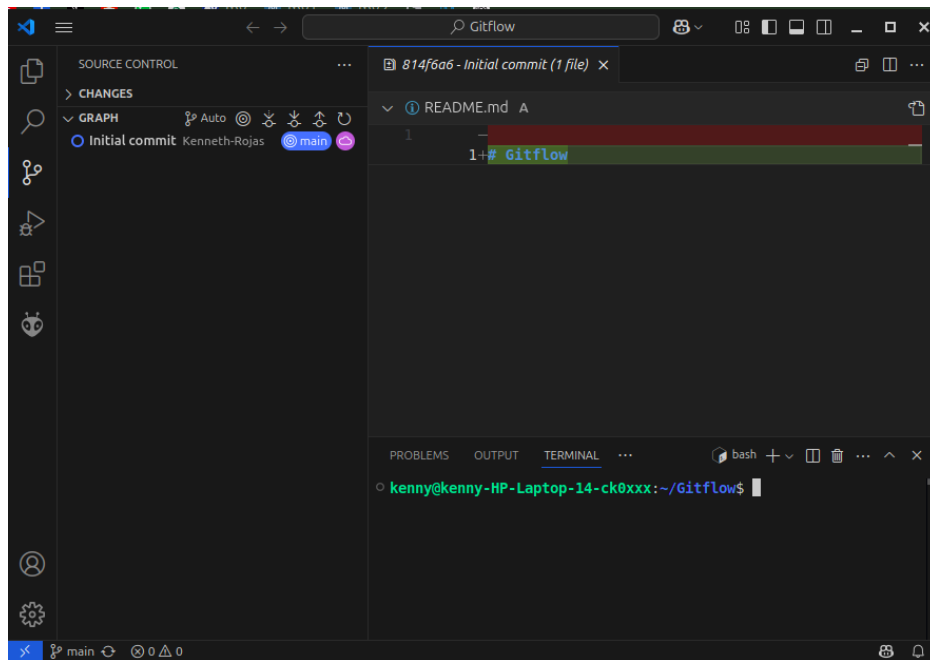
La imagen muestra la interfaz de usuario de GitHub para crear un nuevo repositorio. El título es 'Create a new repository'. Se indica que un repositorio contiene todos los archivos de proyecto, incluyendo el historial de revisiones. Se pide al usuario que marque los campos requeridos con un asterisco (*). El propietario es 'Kenneth-Rojas' y el nombre del repositorio es 'Gitflow'. Se indica que 'Gitflow' está disponible. Se sugiere que los nombres de repositorio sean cortos y memorables. Se pide una descripción opcional. Se elige la visibilidad 'Public'. Se indica que se puede inicializar el repositorio con un README, un archivo .gitignore y una licencia. Se elige 'None' para cada uno de estos. Se indica que se creará un repositorio privado en la cuenta personal. El botón 'Create repository' está en la parte inferior derecha.

A continuación procedemos a clonar el repositorio creado desde la terminal con el siguiente comando en consola:

```
git clone <Codigo_ssh>
```

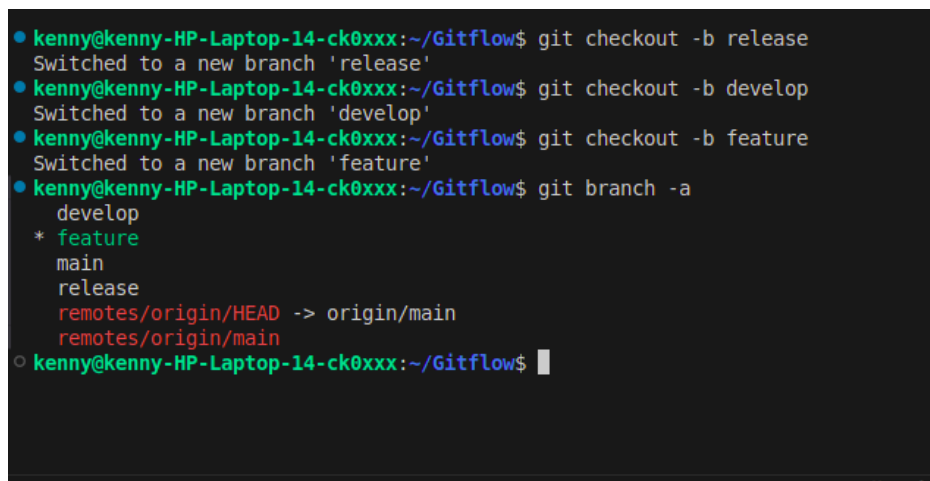
```
kenny@kenny-HP-Laptop-14-ck0xxx:~$ git clone git@github.com:Kenneth-Rojas/Gitflow.git
Cloning into 'Gitflow'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
kenny@kenny-HP-Laptop-14-ck0xxx:~$
```

Para una mejor visualización de nuestro gitflow, ingresaremos y modificaremos nuestro repositorio a través de Visual Studio Code.



A continuación vamos a crear las ramas *release*, *develop* y *feature*, adicionalmente podemos ver las ramas con los siguientes comandos respectivamente:

```
git checkout -b <nombre_de_la_rama>
git branch -a
```

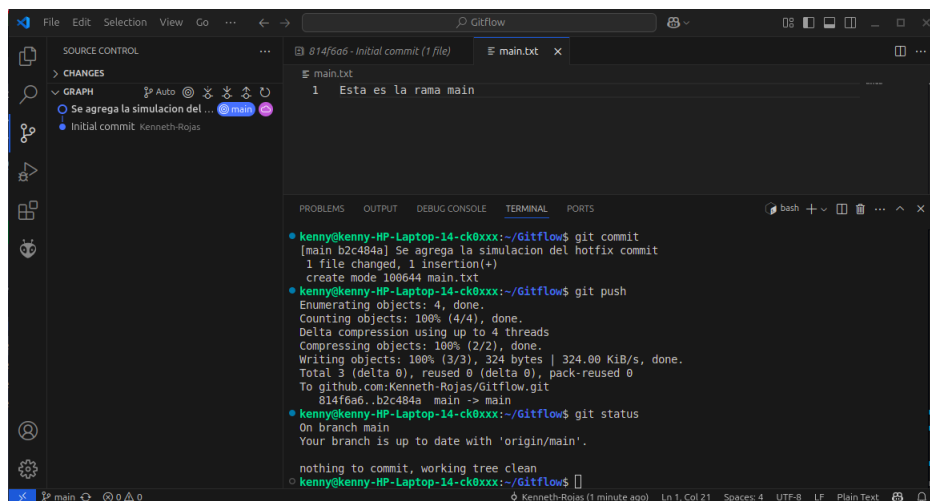


Para asegurarnos de guardar los cambios en el repositorio remoto haremos archivos de texto en cada rama para poder hacer un *push*. De este modo todas las ramas empiezan por **remotes/origin** haciendonos saber que esta correctamente sincronizado con el repositorio remoto.

```
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git checkout features
error: pathspec 'features' did not match any file(s) known to git
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git checkout feature
Switched to branch 'feature'
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git branch -a
develop
feature
* main
release
remotes/origin/HEAD -> origin/main
remotes/origin/develop
remotes/origin/feature
remotes/origin/main
remotes/origin/release
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$
```

2. Cambios de main

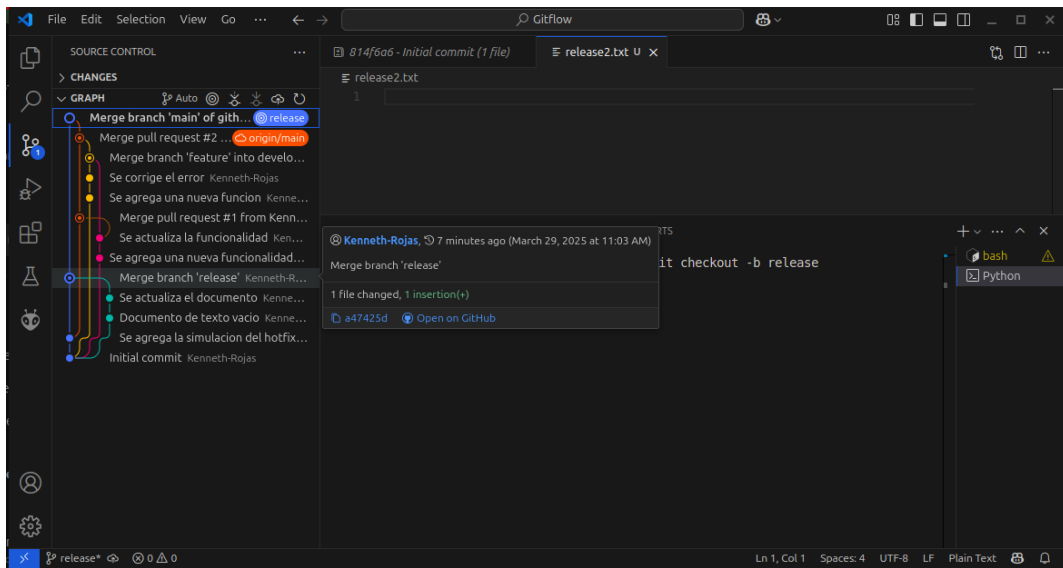
El primer cambio de la rama main es hacer un hotfix commit, sin embargo realizaremos solo un commit. Para esto creamos un archivo *main.txt* y hacemos commit y push para finalizar.



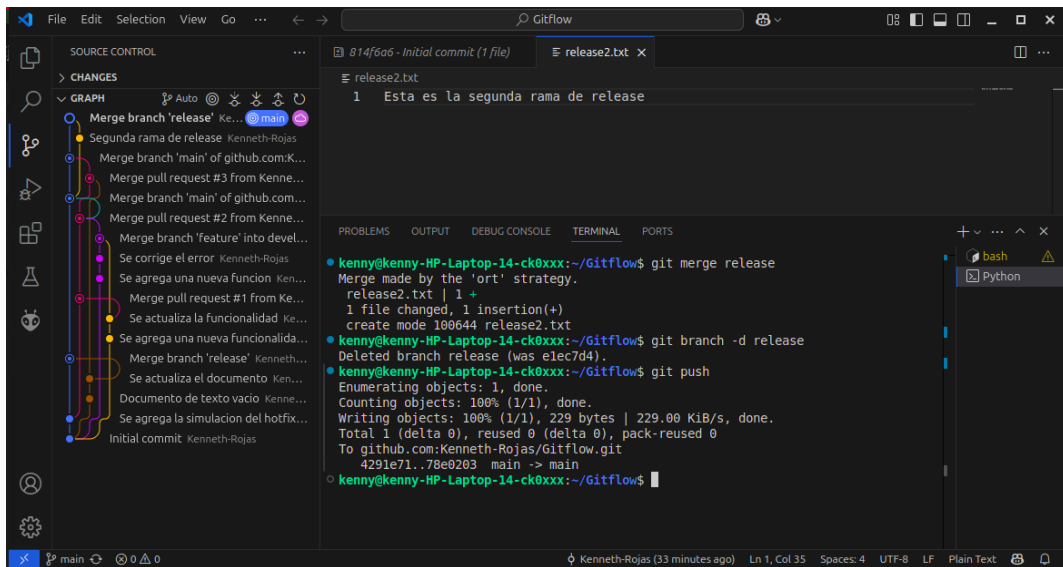
```
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git commit
[main b2c484a] Se agrega la simulacion del hotfix commit
1 file changed, 1 insertion(+)
create mode 100644 main.txt
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 324.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Kenneth-Rojas/Gitflow.git
814f6a6..b2c484a  main -> main
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$
```

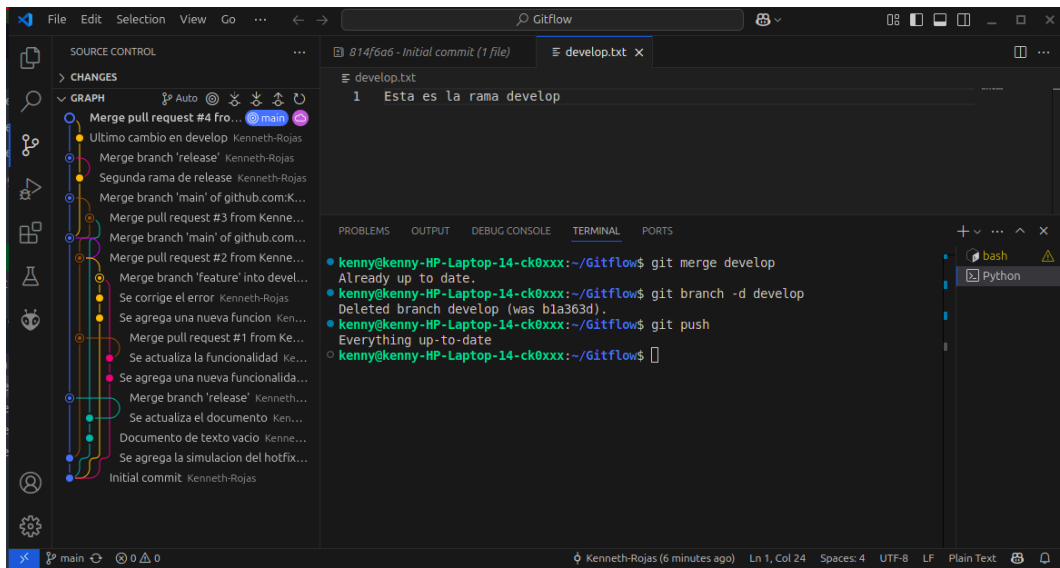
Seguidamente se realiza el primer merge a la rama *main* de la rama *release*. Se elimina esta última rama como se muestra en el gitflow.



A continuación se hace el merge de la rama *release* a la rama *main* por segunda vez.

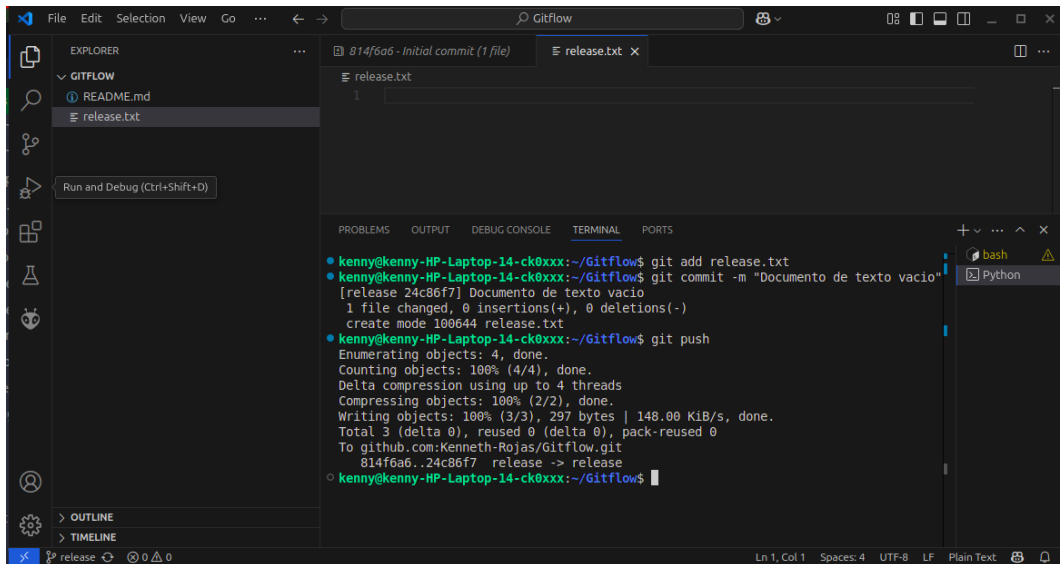


Para finalizar el ejercicio hacemos un merge de la rama *develop* a la rama *main*, eliminamos *develop* y hacemos push.

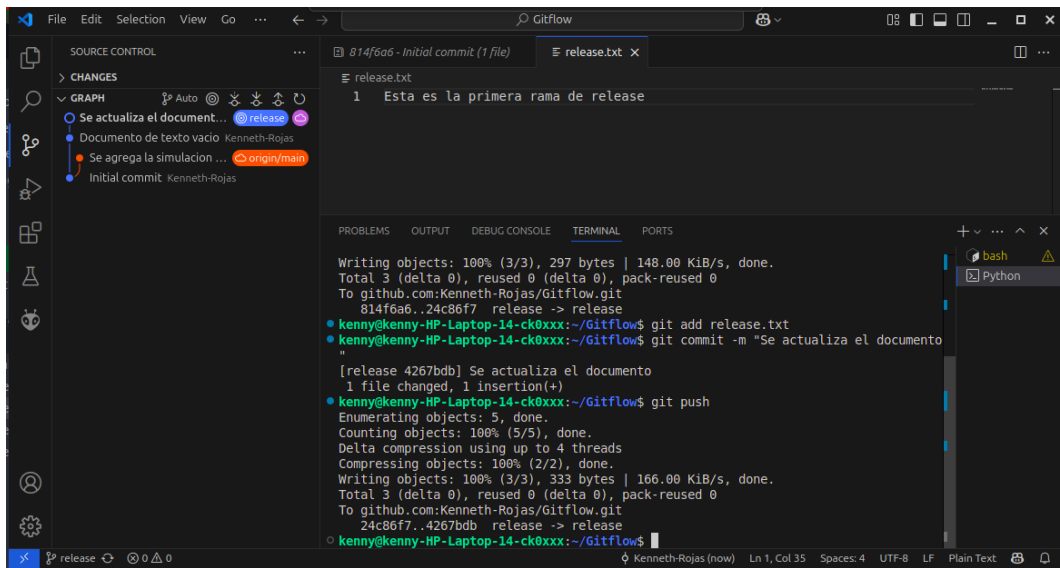


3. Cambios de release

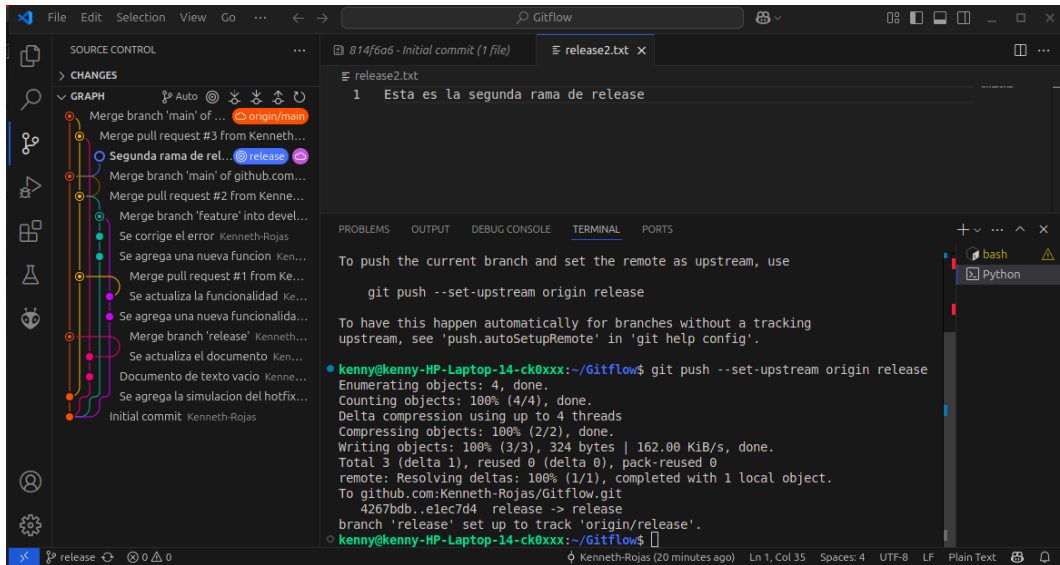
Inicialmente creamos y agregamos un documento de texto vacío que será modificado más adelante. Hacemos commit y push para actualizar los cambios.



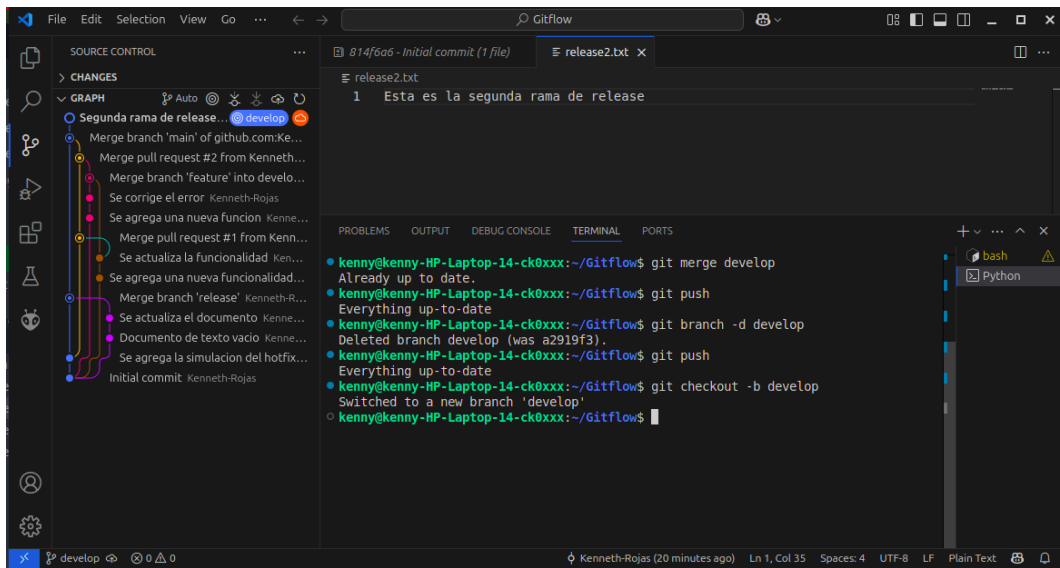
A continuacion editamos el documento de texto, hacemos commit y push y la rama está lista para realizar el primer merge mostrado en el gitflow.



Después del merge anterior se vuelve a crear la rama *release* con un nuevo documento de texto.

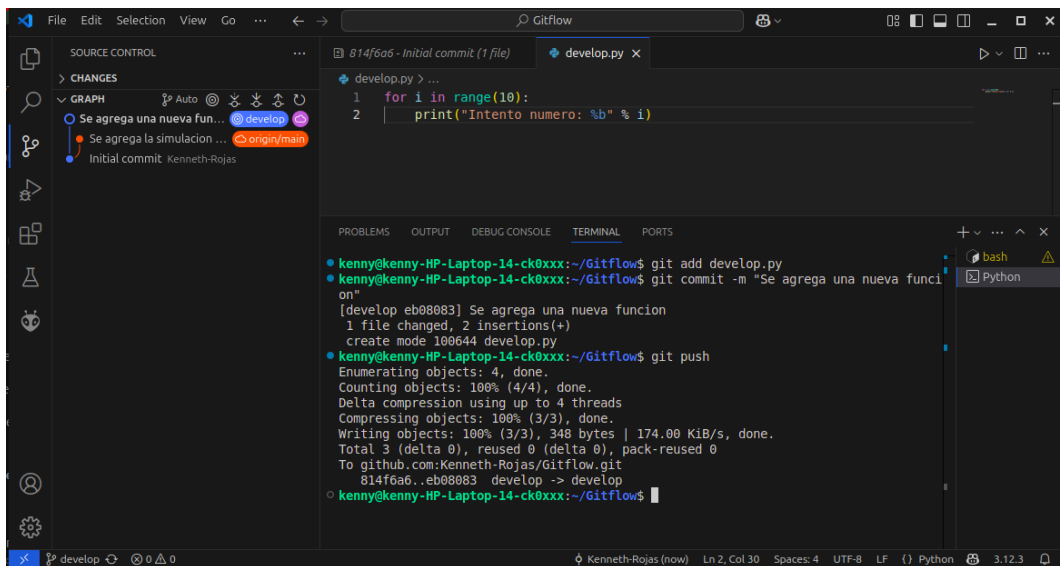


Para finalizar el flujo de esta nueva rama, se hará un merge de la rama *develop* sobre la nueva rama *release*. Erroneamente se eliminó la rama *develop* así que se volverá a crear para seguir don el orden del gitflow.



4. Cambios de develop

Para iniciar con el flujo de esta rama agregamos un archivo *develop.py* el cual contiene un error de sintaxis que deberá ser corregido más adelante. Ejecutamos commit y push para guardar los cambios en el repositorio remoto.



A continuación actualizamos el error del archivo, ejecutamos commit y push.

The screenshot shows the VS Code Gitflow interface. On the left, the 'SOURCE CONTROL' panel displays a commit graph with three commits: 'Initial commit', 'Se agrega la simulacion...', and 'Se corrige el error'. The 'develop' branch is selected. The main editor shows a Python file named 'develop.py' with the following code:

```
1 for i in range(10):
2     print("Intento numero: %d" % i)
```

The bottom panel shows the 'TERMINAL' with the following output:

```
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 348 bytes | 174.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Kenneth-Rojas/Gitflow.git
814f6a6..eb88883 develop -> develop
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git add develop.py
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git commit -m "Se corrige el error"
[develop 8fb37e2] Se corrige el error
1 file changed, 1 insertion(+), 1 deletion(-)
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Kenneth-Rojas/Gitflow.git
eb88883..8fb37e2 develop -> develop
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$
```

Después hacemos merge de la rama *feature* sobre la rama *develop* tal como se muestra en el gitflow. Eliminamos esta rama y hacemos push

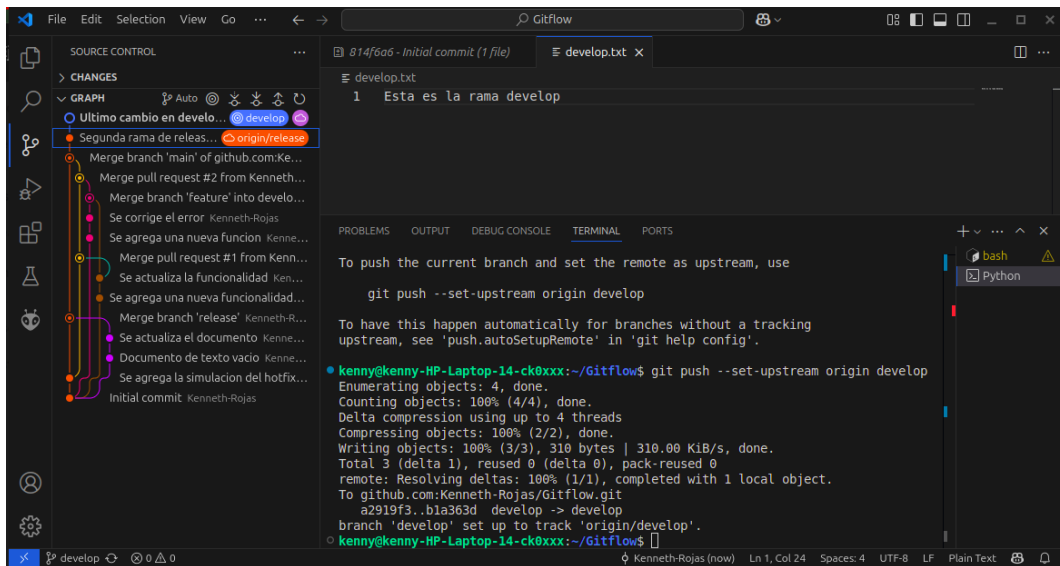
The screenshot shows the VS Code Gitflow interface. On the left, the 'SOURCE CONTROL' panel displays a commit graph with five commits: 'Initial commit', 'Se agrega la simulacion...', 'Se corrige el error', 'Se agrega una nueva funcion', and 'Merge branch 'feature''. The 'develop' branch is selected. The main editor shows a Python file named 'develop.py' with the following code:

```
1 for i in range(10):
2     print("Intento numero: %d" % i)
```

The bottom panel shows the 'TERMINAL' with the following output:

```
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git merge feature
Merge made by the 'ort' strategy.
features.py | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 features.py
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 354 bytes | 354.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Kenneth-Rojas/Gitflow.git
8fb37e2..a2919f3 develop -> develop
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git branch -d feature
Deleted branch feature (was 7b8dce9).
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git push
Everything up-to-date
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$
```

Para finalizar el flujo de esta rama se crea un ultimo documento de texto, se realiza commit, push y estaría lista para hacer merge sobre la rama *main*.

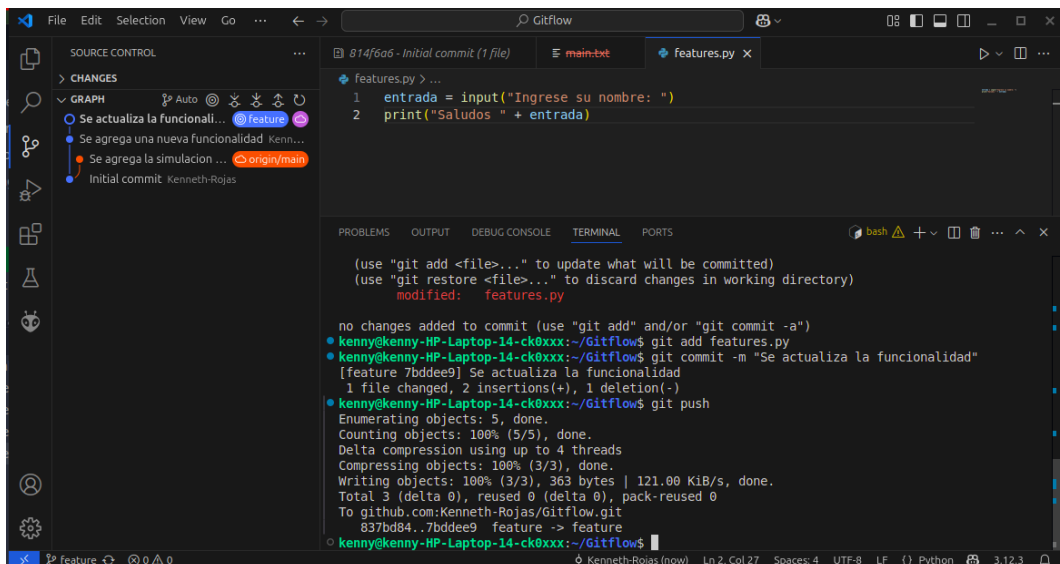


5. Cambios de feature

Inicialmente se agrega un nuevo archivo *features.py* con un "hola mundo", luego se hace commit, y para asegurarnos que todo esté en orden ejecutamos un pull, por ultimo para que los cambios se vean reflejados aplicamos un push.

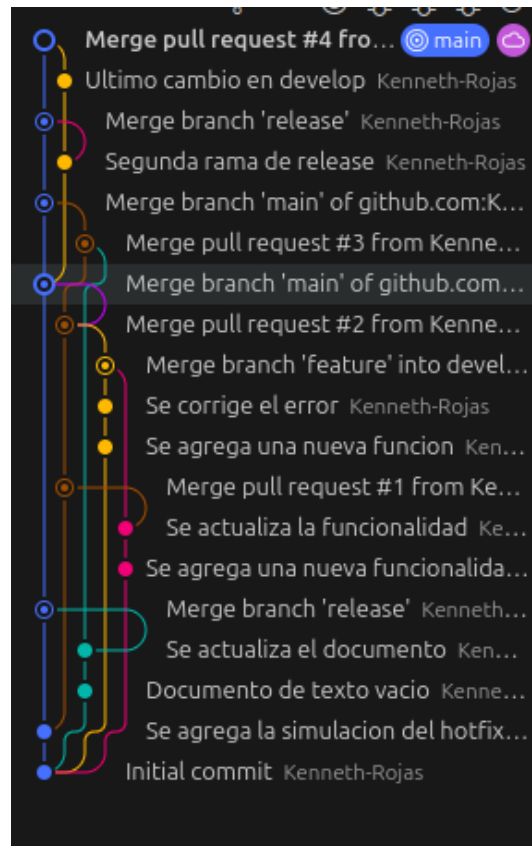
```
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git checkout feature
Switched to branch 'feature'
Your branch is up to date with 'origin/feature'.
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git add features.py
kenny@kenny-HP-Laptop-14-ck0xxx:~/Gitflow$ git commit
```

Para finalizar con esta rama, editamos el archivo de python con una actualización de la funcionalidad. Ejecutamos un commit seguido de un push y la rama estaría lista para hacer merge en el futuro.



6. Resultados finales

El gráfico del gitflow obtenido es el siguiente:



Este gitflow puede ser ligeramente diferente al objetivo inicial debido a interacciones accidentales con el repositorio remoto en github. También por un error eliminando una rama antes de tiempo.

El enlace para acceder al repositorio es el siguiente: <https://github.com/Kenneth-Rojas/Gitflow>