1. \*\*Clonar el repositorio y crear una rama nueva\*\*:

```bash

# Clonar el repositorio

git clone <url\_del\_repositorio>

cd <nombre\_del\_repositorio>

# Crear una nueva rama llamada "nueva-funcionalidad"

git checkout -b nueva-funcionalidad

```

2. \*\*Agregar el archivo "hola\_mundo.py" a la rama "nueva-funcionalidad"\*\*:

```python

# Contenido de "hola\_mundo.py"

print("Hola mundo")

```

3. \*\*Crear y agregar un módulo que suma enteros aleatorios\*\*:

```python

# Contenido del módulo "suma\_aleatoria.py" en la rama "nueva-funcionalidad"

import random

def suma\_aleatoria():

num1 = random.randint(1, 100)

num2 = random.randint(1, 100)

resultado = num1 + num2

return resultado

```

4. \*\*Modificar "hola\_mundo.py" para usar el módulo\*\*:

```python

# Contenido actualizado de "hola\_mundo.py" en la rama "nueva-funcionalidad"

from suma\_aleatoria import suma\_aleatoria

resultado\_suma = suma\_aleatoria()

print(f"Hola mundo: {resultado\_suma}")

```

5. \*\*Confirmar los cambios y comentarlos\*\*:

```bash

# Agregar los archivos al área de preparación

git add hola\_mundo.py suma\_aleatoria.py

# Confirmar los cambios con un mensaje descriptivo

git commit -m "Agregado módulo para sumar enteros aleatorios y mostrar resultado en hola\_mundo.py"

```

6. \*\*Volver a la rama main y fusionar los cambios\*\*:

```bash

# Cambiar de vuelta a la rama main

git checkout main

# Fusionar la rama "nueva-funcionalidad" con la rama main

git merge nueva-funcionalidad

```

7. \*\*Eliminar la rama "nueva-funcionalidad" (opcional)\*\*:

```bash

# Eliminar la rama "nueva-funcionalidad" después de fusionar los cambios

git branch -d nueva-funcionalidad

```

8. \*\*Subir los cambios al repositorio remoto\*\*:

```bash

# Subir los cambios fusionados a la rama main al repositorio remoto

git push origin main

```

Con estos pasos, has agregado una nueva funcionalidad a tu proyecto, comentado los cambios, fusionado la rama con la rama principal (main) y subido los cambios al repositorio remoto.

Si has completado exitosamente la fusión de la rama adicional (en este caso, "nueva-funcionalidad") con la rama principal (main) y ya no necesitas mantener esa rama separada, puedes eliminarla de tu repositorio local y remoto. Aquí está cómo hacerlo:

1. \*\*Eliminar la rama adicional localmente\*\*:

```bash

git branch -d nueva-funcionalidad

```

Este comando eliminará la rama "nueva-funcionalidad" de tu repositorio local. Si hay cambios que aún no se han fusionado en la rama principal, Git te advertirá y no eliminará la rama hasta que esos cambios se fusionen o se descarten.

2. \*\*Eliminar la rama adicional en el repositorio remoto\*\*:

```bash

git push origin --delete nueva-funcionalidad

```

Este comando eliminará la rama "nueva-funcionalidad" del repositorio remoto. Asegúrate de haber subido todos los cambios relevantes antes de eliminar la rama remota.

Una vez que hayas eliminado la rama adicional local y remota, ya no estará presente en tu historial de Git y solo conservarás los cambios fusionados en la rama principal (main). Esto ayuda a mantener tu historial de Git limpio y organizado.

Para eliminar los archivos del sistema de archivos que estaban asociados con la rama que estás eliminando, tendrías que realizar pasos adicionales. Aquí te muestro cómo hacerlo:

1. \*\*Eliminar la rama adicional localmente y forzar la eliminación de commits\*\* (tal como se mencionó anteriormente):

```bash

git checkout main # Asegurarse de estar en la rama main

git branch -d nueva-funcionalidad # Eliminar la rama localmente

git branch -D nueva-funcionalidad # Forzar la eliminación de los commits

```

2. \*\*Eliminar los archivos asociados con la rama del sistema de archivos\*\*:

```bash

git checkout main # Asegurarse de estar en la rama main

git branch -d nueva-funcionalidad # Eliminar la rama localmente

git branch -D nueva-funcionalidad # Forzar la eliminación de los commits

git checkout -- . # Restaurar el estado de los archivos al de la rama main

```

El comando `git checkout -- .` restaurará los archivos al estado de la rama `main`, es decir, eliminará los cambios no confirmados y restaurará los archivos a su estado en la rama `main`. Esto eliminará cualquier archivo que se haya agregado en la rama `nueva-funcionalidad`, pero no existía en `main`. Asegúrate de tener una copia de seguridad de los archivos si necesitas preservarlos antes de ejecutar este comando.