Informe de Actividad: Git y Python

Autor: Bryam Alejandro Mora Mendoza

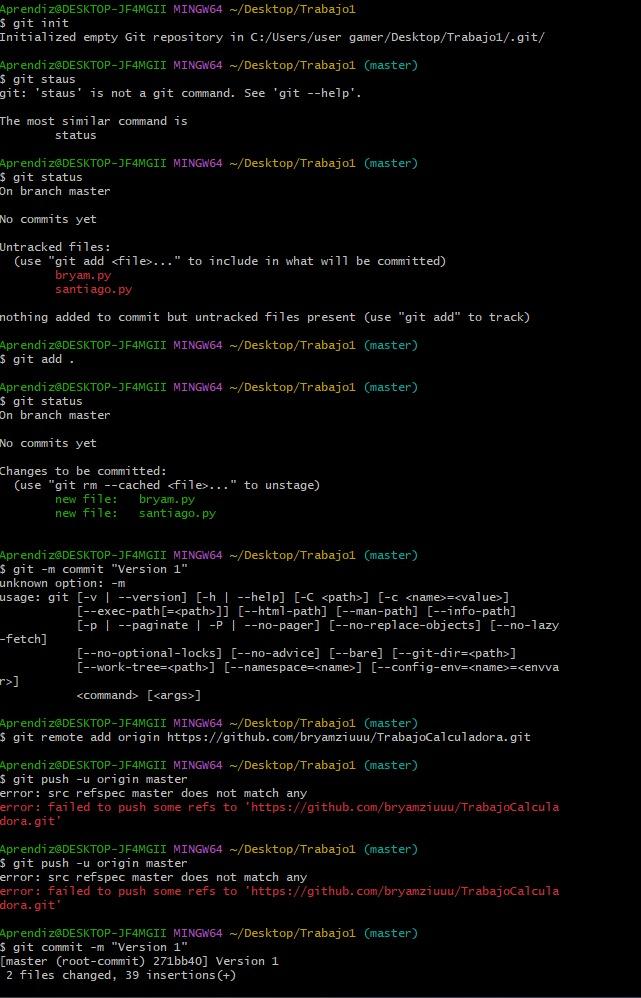
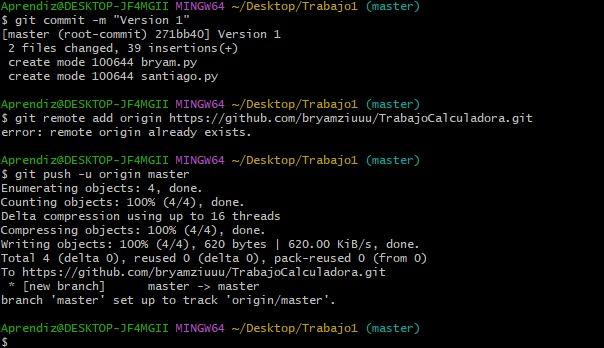
Materia: Programación en Python y Control de Versiones con Git

# Introducción

En este informe se describe el procedimiento para crear un repositorio en GitHub, clonarlo en un equipo local, trabajar con ramas y unir los cambios realizados en la rama principal (master). Además, se documenta con imágenes cada uno de los pasos realizados en el proceso.

# Desarrollo

1. Creamos una carpeta que en este caso fue nombrada “Trabajo1”
2. Dentro de esa carpeta abrimos la terminal de git
3. Ingresamos los siguientes comandos:

* Git init: Inicia un repositorio git en la carpeta
* Gif status: Revisar el estado de los archivos
* Git add: Guarda los archivos
* Git commit -m “version1”: Guarda los cambios en la zona de preparación, el mensaje debe ser una descripción breve de los cambios realizados.
* Gif remote add origin url: Crea el repositorio a GitHub
* Git pus -U origin máster: Sube los cambios realizados en un repositorio remoto
* 
* 

Con esto logramos crear nuestro repositorio en GitHub

Después creamos nuestra parte de la calculadora en Visual Studio Code:

1. Dentro de la app de VSC(Visual Studio Code), abrimos la carpeta que creamos que sería trabajo1.
2. Creamos un archivo .py, que en este caso lo nombramos [bryam.py](http://bryam.py)
3. Dentro de ese archivo iniciamos poniendo un #Suma, que es un comentareado para no confundirnos.
4. Después ponemos un input("Hola, bienvenido a la calculadora de Bryam"), el input es una función incorporada que se usa para solicitar datos al usuario a través de la consola, lo cual me confundí, debería ser print.
5. Con el print("Esta es la operacion suma"), nos mostrará un mensaje para avisar que empieza la operación
6. def resta(n1, n2):

return n1 - n2

Con esta línea lo que hacemos es definir una función llamada suma que recibe dos parámetros (n1 y n2) y devuelve la suma de ambos.

1. n1 = int(input("Ingrese el primer numero: ")) Esta línea pide que el usuario ingrese un número, lo convierte en entero (int) y lo guarda en la variable n1
2. n2 = int(input("Ingrese el segundo numero: ")) Es lo mismo que el paso anterior, pero guarda el segundo número en la variable n2
3. print("La suma es: ", suma(n1, n2)) y ya para finalizar esta línea llama a la función suma con los valores que ingreso el usuario y muestra el resultado en la pantalla
4. Acá finalizaríamos con el operador de la suma, ahora seguiríamos con la resta.
5. print("Esta es la operacion resta") Muestra un mensaje para avisar que empieza la resta
6. def resta(n1, n2):

return n1 - n2

Definimos una función llamada resta que recibe dos parámetros y devuelve la resta.

1. n1 = int(input("Ingrese el primer numero: "))Esta línea pide que el usuario ingrese un número, lo convierte en entero (int) y lo guarda en la variable n1

n2 = int(input("Ingrese el segundo numero: ")) Es lo mismo que el paso anterior, pero guarda el segundo número en la variable n2

1. print("La resta es: ", resta(n1, n2)) Llama a la función resta con los valores ingresados y muestra el resultado en pantalla.



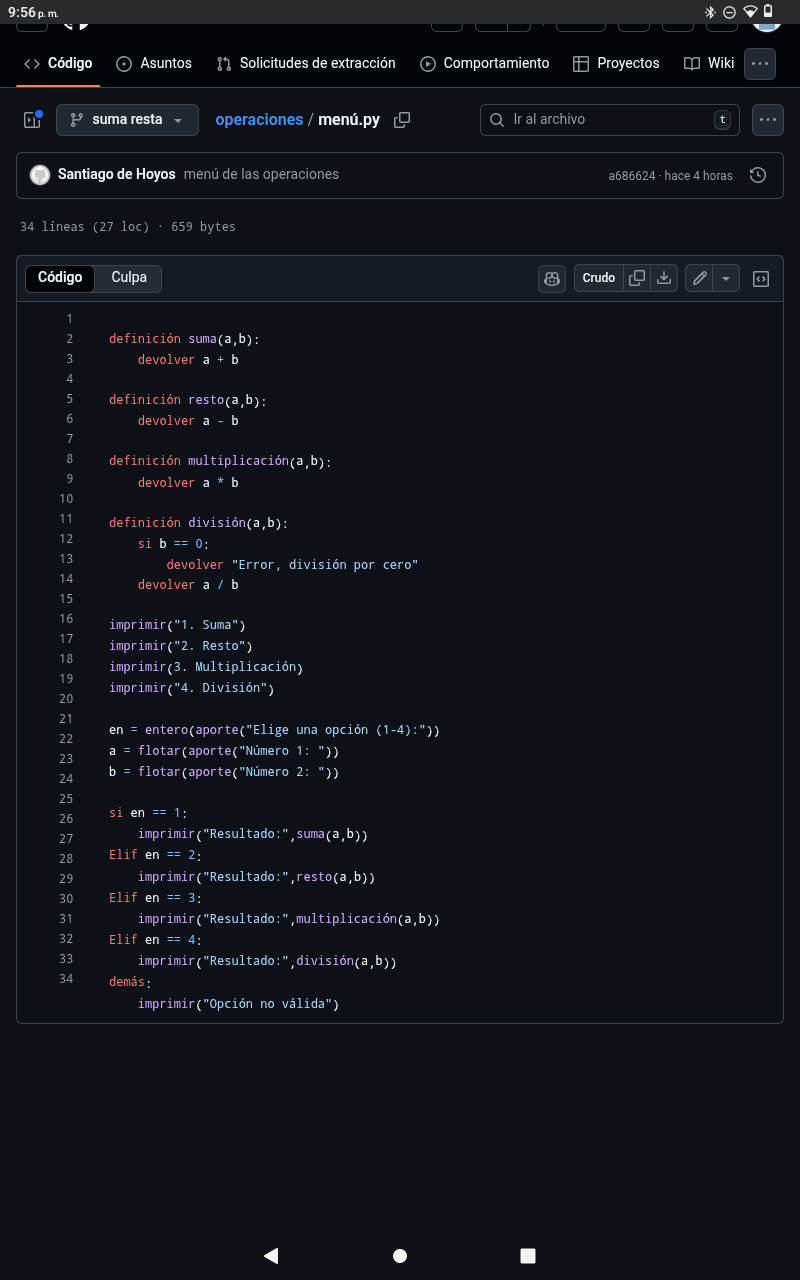
Después seguiremos haciendo la plantilla

Este programa que hice es básicamente una calculadora en Python. Lo primero que hice fue crear funciones para cada operación: una para sumar, otra para restar, otra para multiplicar y otra para dividir. En la de división también puse una condición para que no se pueda dividir por cero, porque eso daría error.

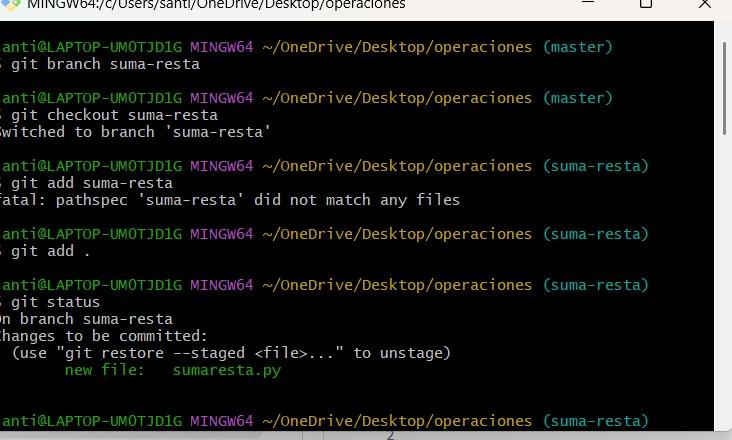
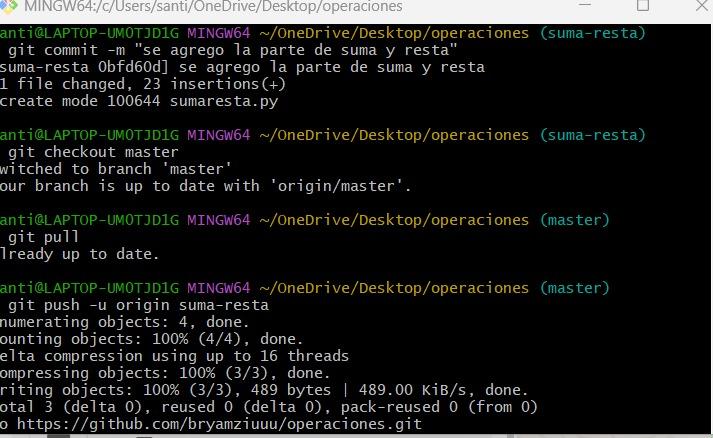
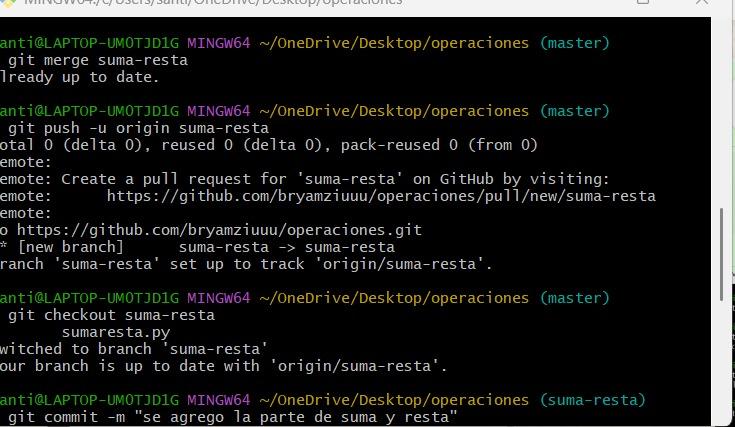
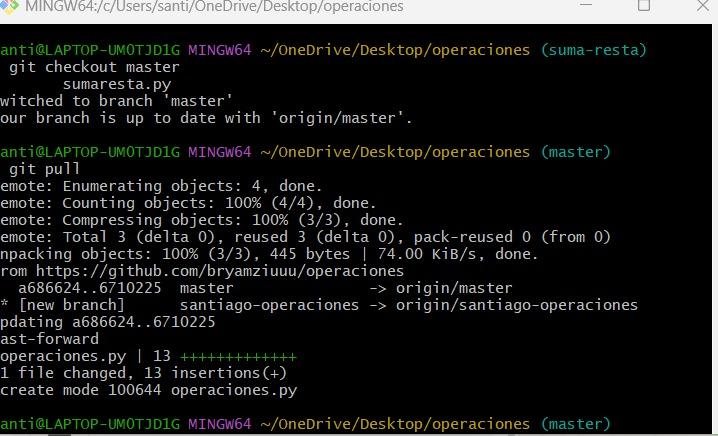
Después de eso, el programa muestra un menú con cuatro opciones: suma, resta, multiplicación y división. El usuario simplemente elige escribiendo un número del 1 al 4.

Cuando ya eligió la operación, el programa pide dos números, que pueden ser enteros o decimales. Con esos números, según la opción que el usuario escogió, se llama a la función correspondiente y se muestra el resultado.

Si la persona escribe una opción que no existe, por ejemplo un 5, el programa responde diciendo que esa opción no es válida.



Ahora vamos con los comandos de git.

* git branch suma-resta
* git checkout suma-resta
* git branch suma-resta → crea una nueva rama llamada suma-resta.
* git checkout suma-resta → se cambia a esa rama para trabajar de forma separada al master.
* 2. Agregar archivos al área de preparación
* git add :Agrega los archivos creados o modificados (en este caso sumaresta.py) al área de staging para poder confirmar los cambios después.
* 3. Confirmar los cambios (commit)
* git commit -m "se agrego la parte de suma y resta": Guarda los cambios en la rama actual con un mensaje descriptivo.
* 4. Actualizar la rama principal (master)
* git checkout master
* git pull
* git checkout master → cambia a la rama principal (master).
* git pull → trae las últimas actualizaciones del repositorio remoto de GitHub a tu repositorio local.
* 5. Subir la rama al repositorio remoto
* git push -u origin suma-resta: Sube la rama suma-resta a GitHub y la enlaza para que los próximos push y pull funcionen automáticamente.
* 6. Mezclar cambios de la rama al master
* git merge suma-resta :Integra los cambios de la rama suma-resta a la rama principal master.
* 

# Nota

Agradezco y expongo la ayuda de mi compañera Santiago De Hoyos, ya que pedí un portátil prestado en horas de la tarde, y a la hora de elaboración de este documento (10:08pm) y no me hicieron el favor, así que el me ayudó en lo que más pudo, no dispongo de un computador o portátil, lo estoy realizando mediante una tablet, así que de igual manera pido disculpas si hay algo con error.

