

INGENIERIA EN SISTEMAS

GRUPO LOS MAGIOS

NOMBRE DE LOS ESTUDIANTES

**Limber David Quispe Osco**

**Yeri Torrico Ribera**

**Salomon Leon Pesoa**

**Beymar Ferrufino Peredo**

**Daniel Alanes Caceres**

DOCENTE

**Jimmy Nataniel Requena Llorentty**

Santa Cruz – Bolivia 16/07/2025

Índice

[INTRODUCCION A GITHUB Y REPLIT. 3](#_Toc203586385)

[1. Introducción a GitHub. 3](#_Toc203586386)

[2. Segundo paso para crear cuenta de GitHub. 4](#_Toc203586387)

[3. Verificación de cuenta. 5](#_Toc203586388)

[4. Inicio de sesión en GitHub. 6](#_Toc203586389)

[5. iniciar sesión ya con nuestra cuenta. 7](#_Toc203586390)

[6. Creación de repositorios. 8](#_Toc203586391)

[7. Creación y colocado de nombre a repositorio. 9](#_Toc203586392)

[8. Edición de archivo “Readme.md”. 10](#_Toc203586393)

[9. Editando archivo “Readme.md”. 11](#_Toc203586394)

[10. Agregado de colaboradores a nuestro repositorio. 12](#_Toc203586395)

[11. Creación de carpetas. 13](#_Toc203586396)

[12. Creacion de carpetas para los integrantes. 14](#_Toc203586397)

[13. Introducción a Replit. 15](#_Toc203586398)

[14. Vinculación de cuenta. 16](#_Toc203586399)

[15. Creación de repositorios en Replit. 17](#_Toc203586400)

[16. Creación de folder Replit. 18](#_Toc203586401)

[17. Ejecutar códigos mediante la consola “Shell”. 19](#_Toc203586402)

[EJECUCION DE CODIGOS. 20](#_Toc203586403)

[1. Código Hola mundo. 20](#_Toc203586404)

[2. Código Tabla de multiplicar. 21](#_Toc203586405)

[3. Código Verificador de Edad para Película. 22](#_Toc203586406)

[4. Código Tabla de Multiplicar. 23](#_Toc203586407)

[5. Código Área de Rectángulo. 24](#_Toc203586408)

[6. Código Promedio de notas. 25](#_Toc203586409)

[7. Código Presentación de Grupo. 26](#_Toc203586410)

[8. Código Lista invertida 27](#_Toc203586411)

[9. Código Encuentra al Mayor. 28](#_Toc203586412)

[10. Código Factorial 29](#_Toc203586413)

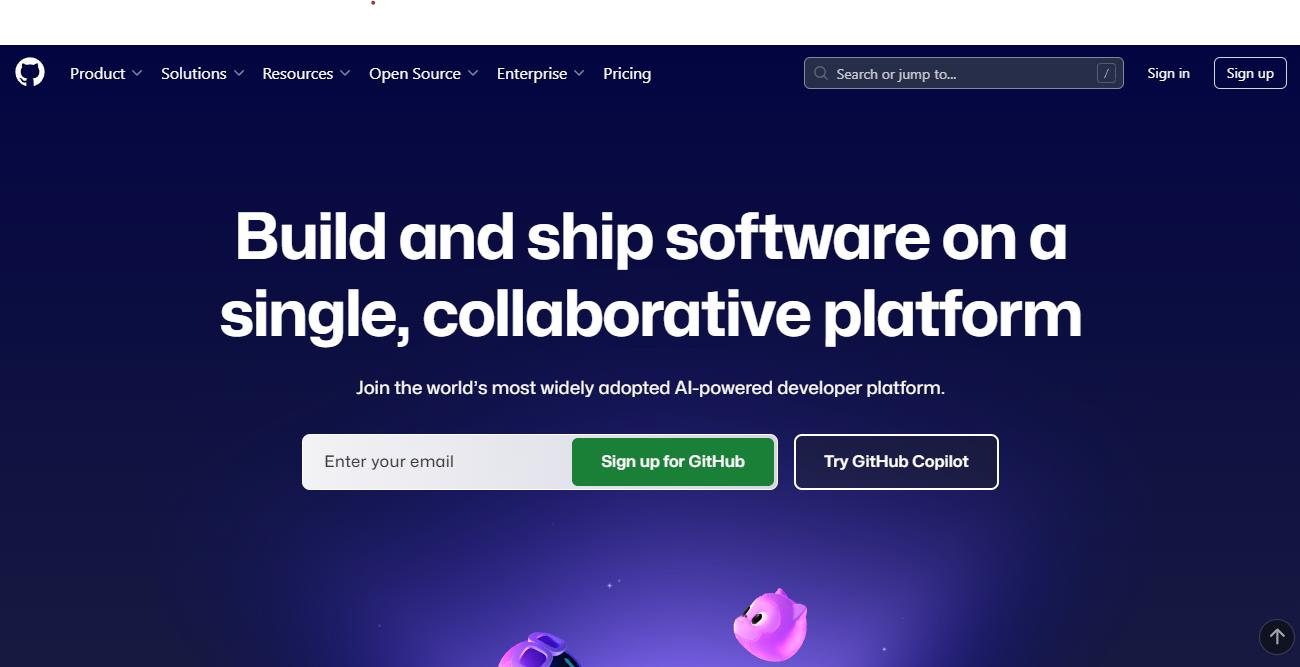
[11. Código Funciones. 30](#_Toc203586414)

# INTRODUCCION A GITHUB Y REPLIT.

## Introducción a GitHub.

El primer paso para para poder tener acceso a GitHub es crear una cuenta. Lo cual tenemos que ir a la página de GitHub.

Después haber buscado la página procedemos a empezar a crear nuestra cuenta y le damos click en en “Sing up”.



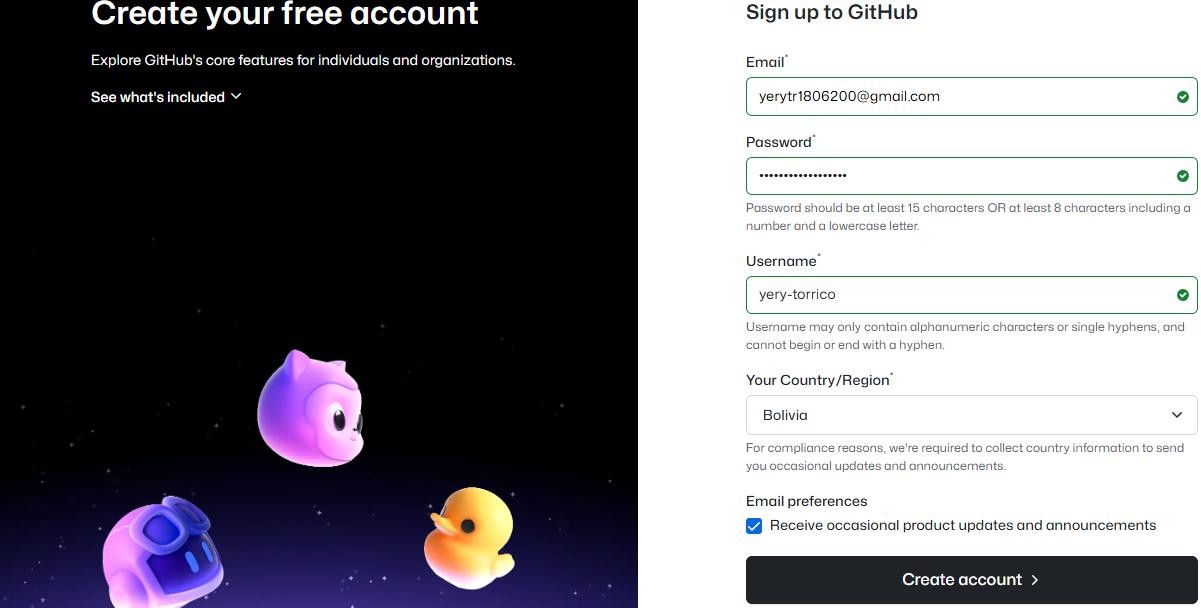
## Segundo paso para crear cuenta de GitHub.

En este parte tenemos rellenar el cual nos pide las siguientes cosas:

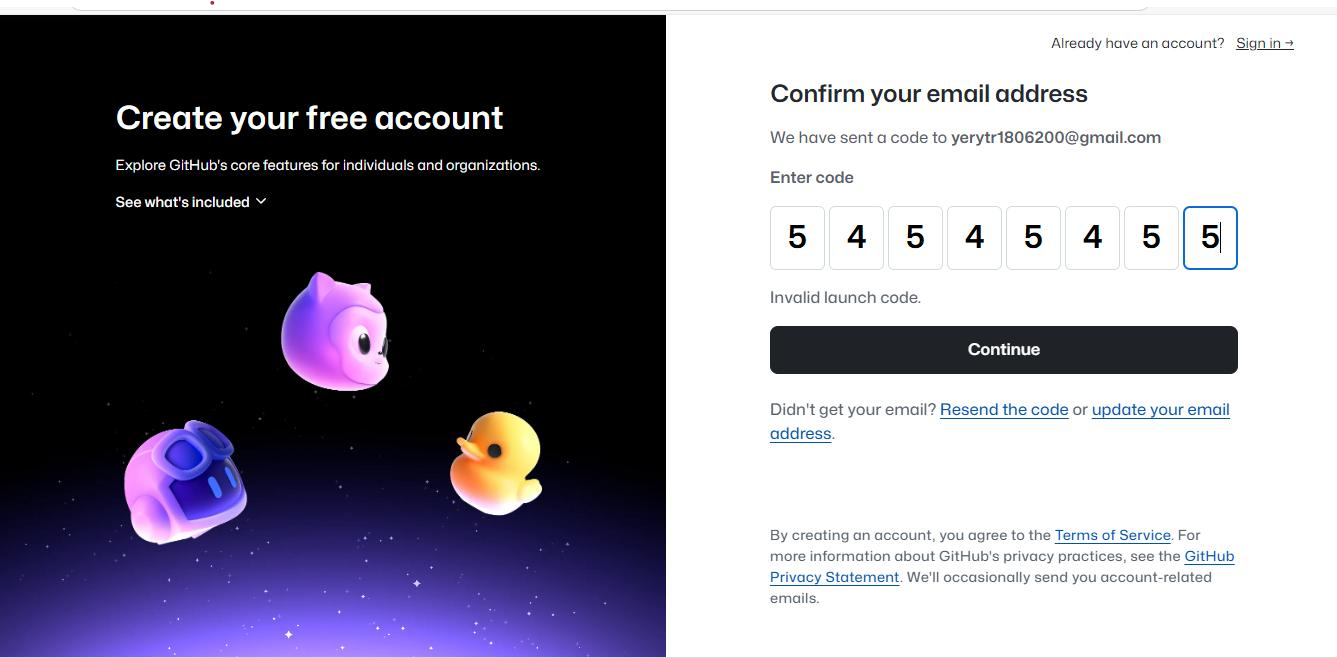
* + Email, podemos usar nuestro correo personal porque nos van a enviar mensaje de verificación.
  + Password, es la contraseña con la cual vamos a acceder después de crear la cuenta.
  + Username, es el nombre con el cual va estar tu cuenta de GitHub.
  + Your Country/Región, ahí uno tiene el país de donde se encuentra.

Después de haber rellenado todo correctamente le damos click en “Create account” ya para que

se cree la cuenta.



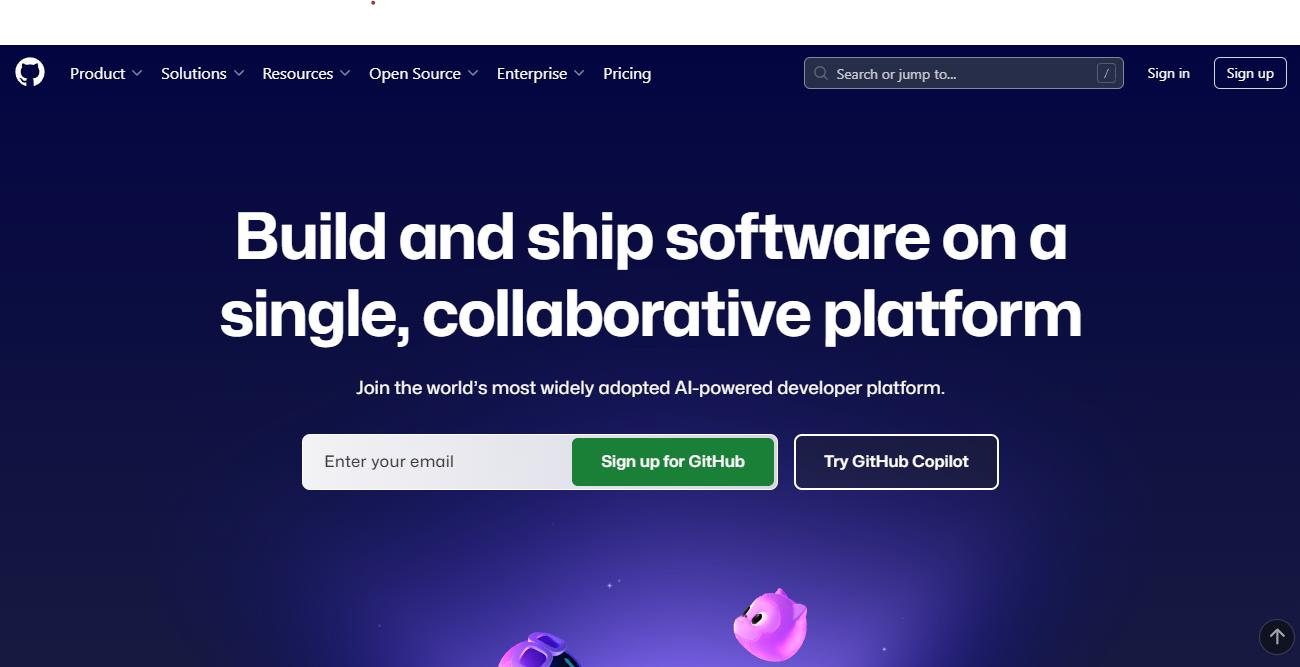
## Verificación de cuenta.

Aquí nos van enviar un código de confirmación a nuestro correo y es solo copiar el código de verificación y le damos click en “Continue”.

1. OBJETIVOS

## Inicio de sesión en GitHub.

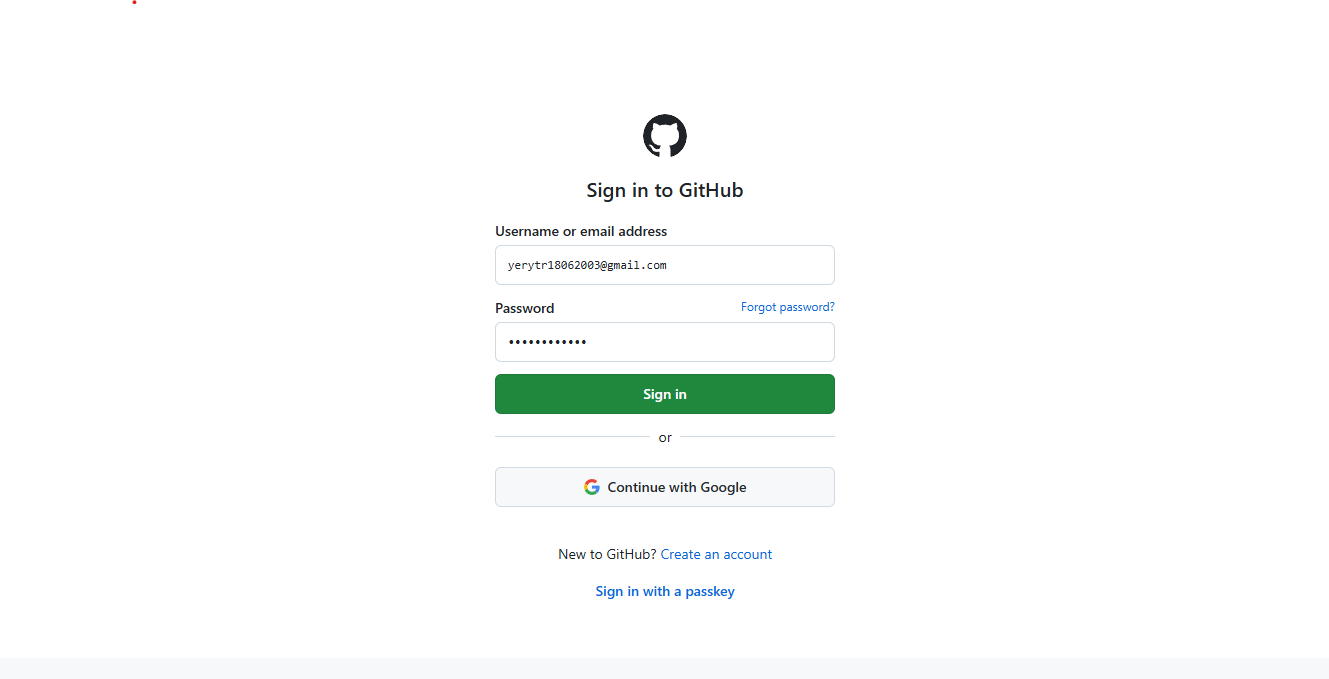
Le damos click en “Sing in” ya para poder iniciar sesión.



## iniciar sesión ya con nuestra cuenta.

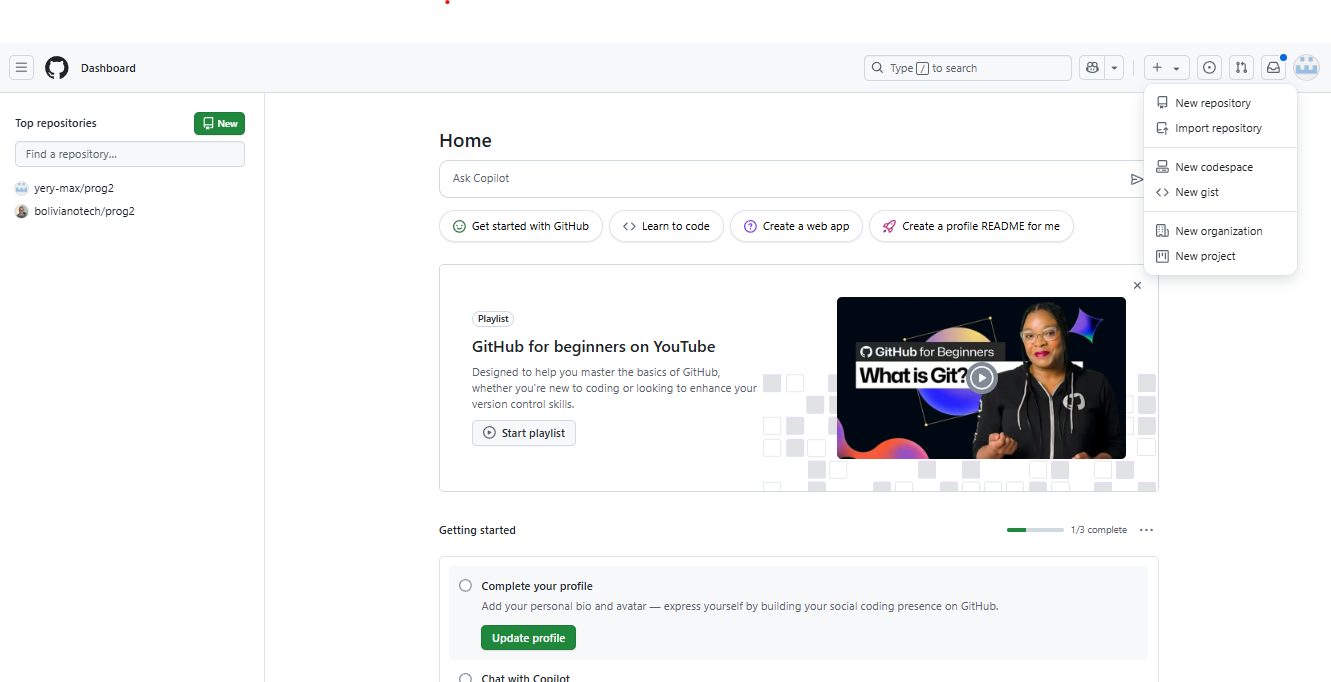
En esta parte ponemos nuestro “Usarname” o “Email” y en la parte de “Password” nuestra contraseña.

Ya ingresaríamos a nuestra cuenta de GitHub.



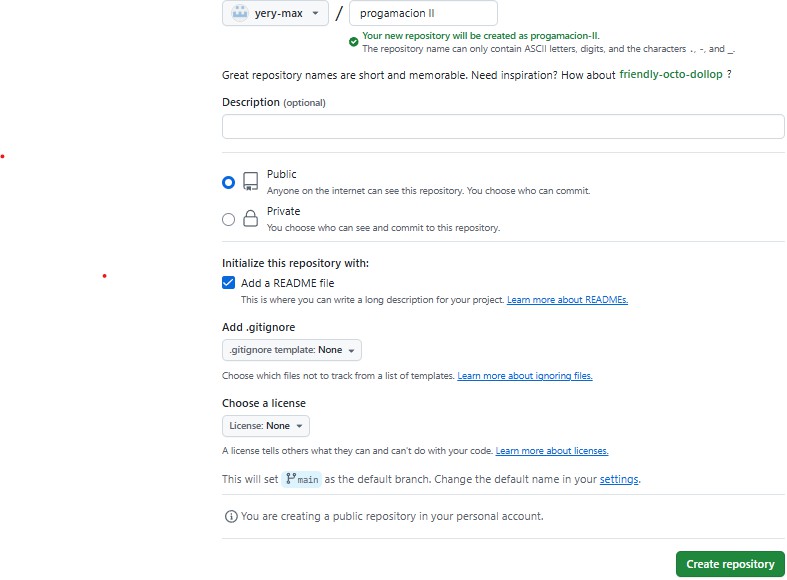
## Creación de repositorios.

Ya estando dentro de nuestra cuenta procedemos a crear nuestro primer repositorio la cual creamos dándole click en “New repository”.



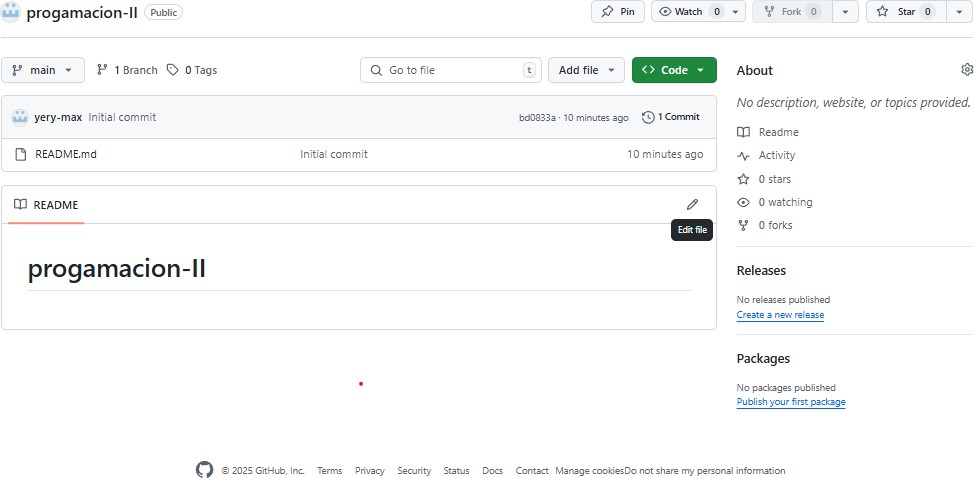
## Creación y colocado de nombre a repositorio.

En esta parte tenemos que elegir un nombre para nuestro repositorio. También la vamos a dejar publico nuestro repositorio y también le damos click en la opción “Add a README file” para que nos cree un archivo “Readme.md” que es documentación inicial de un proyecto, ya después haber hecho eso ya le damos click en “Create repository” para que cree el repositorio.



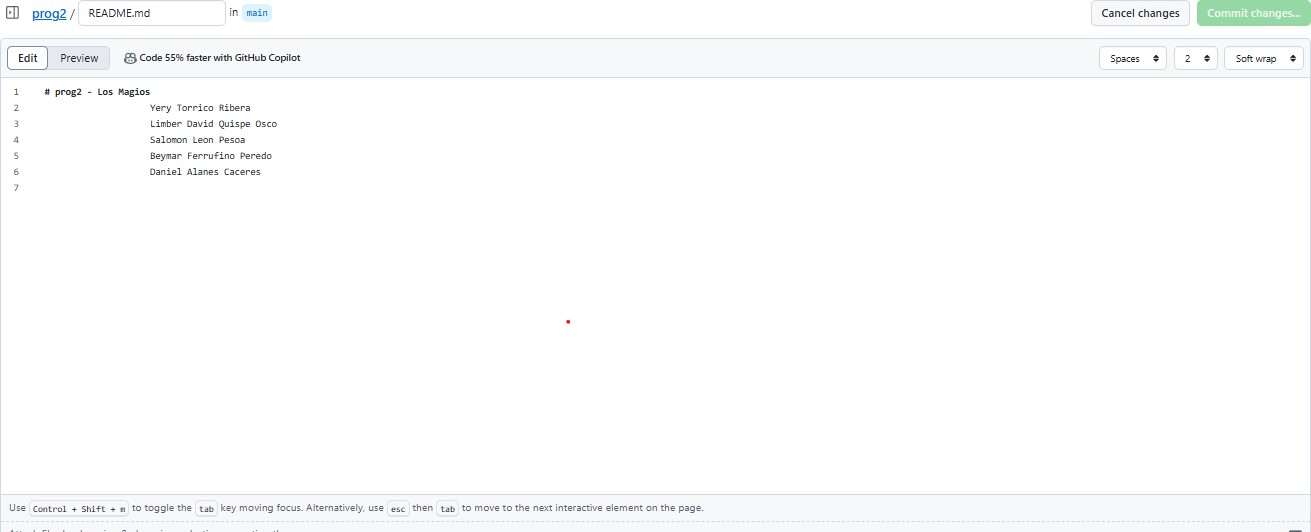
## Edición de archivo “Readme.md”.

Le damos click en la parte de donde dice “Edit file”.



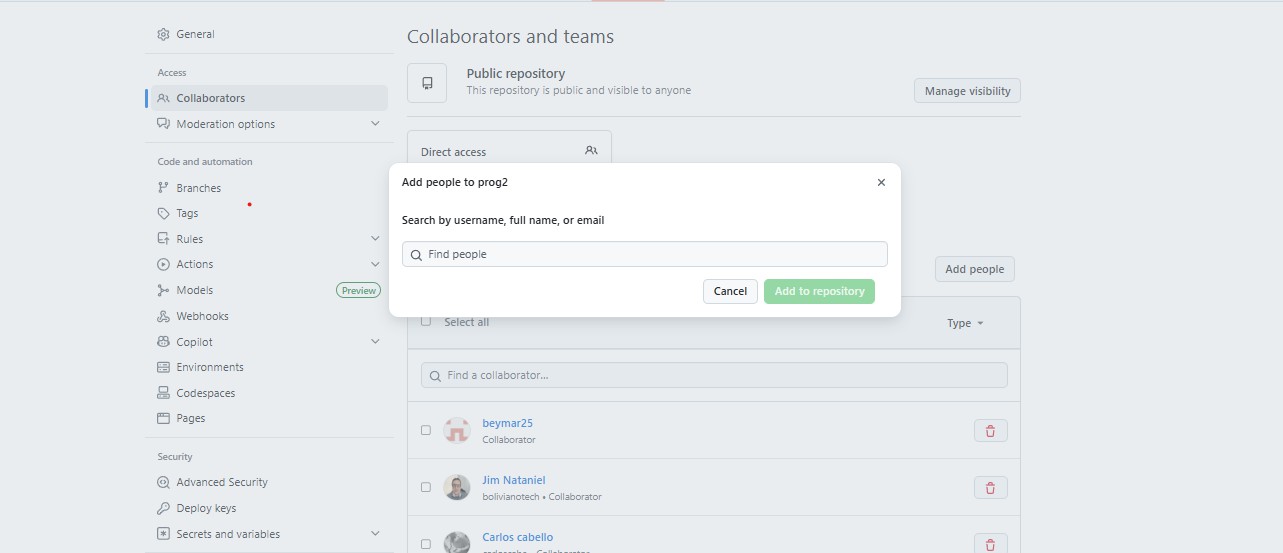
## Editando archivo “Readme.md”.

En esta parte hacemos lista con los integrantes del grupo que hicimos en clases y le ponemos el nombre que eligió de como se va a llamar el grupo en este caso Los Magios.



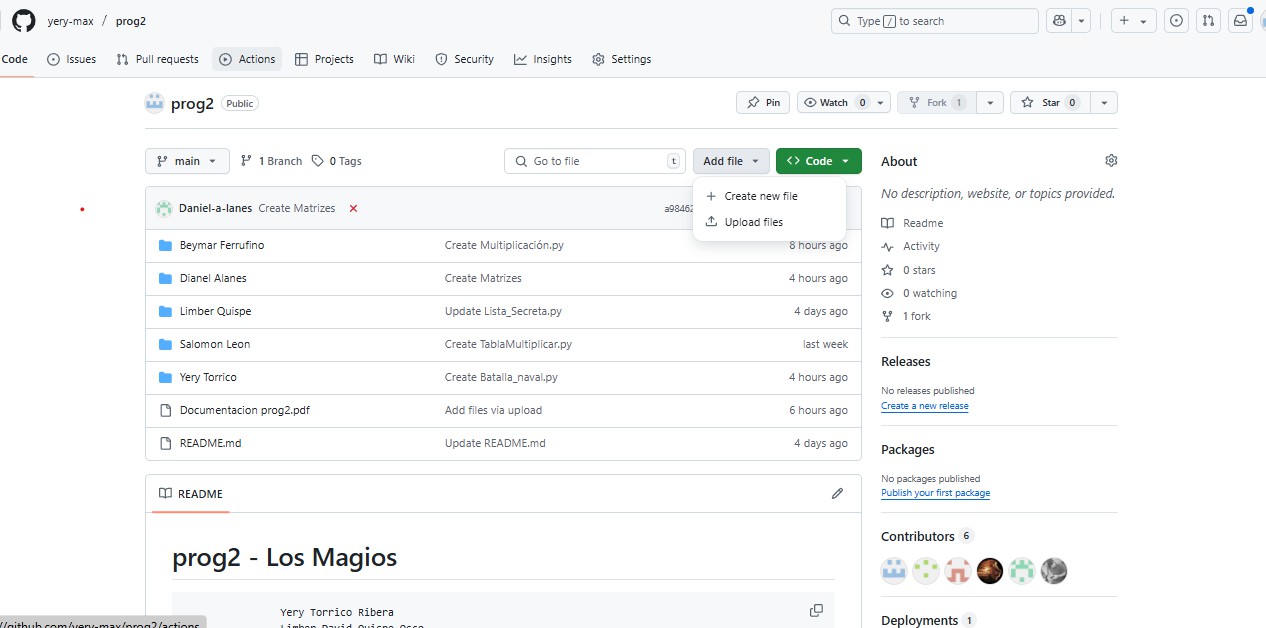
## Agregado de colaboradores a nuestro repositorio.

En esta parte agregamos a los integrantes del grupo para que ellos pueden ver y editar el repositorio.



## Creación de carpetas.

Aquí para poder crear carpetas para que cada uno de los integrantes del grupo le dimos click en “Add file” y después en “+ Create new file”.

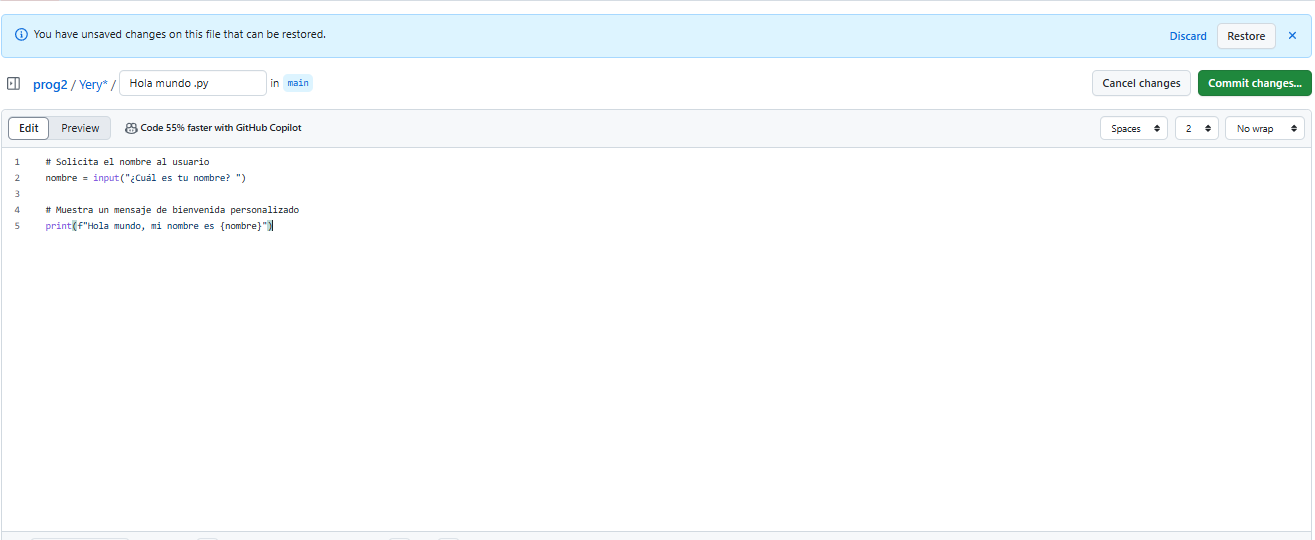


## Creacion de carpetas para los integrantes.

En esta parte primero creamos la carpeta lo cual llamaremos en este caso “Yery” que es uno de lo integrantes del grupo y dentro de esa carpeta creamos otra carpeta ya para poder copiar los códigos, en este caso copiamos el código que lleva por nombre “Hola\_mundo.py”.

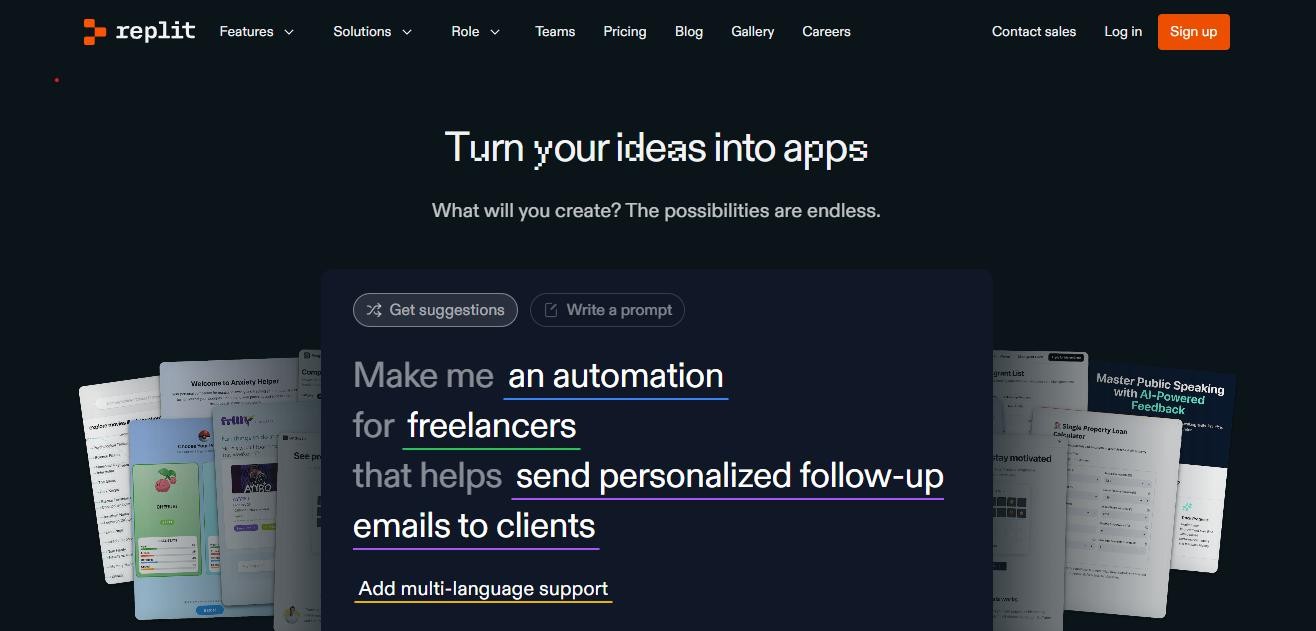
Después haber hecho eso le damos click en “Commit changes” para que se guarden los que añadimos.

Ya cada uno de los integrantes del grupo va a ir subiendo sus códigos a sus carpetas.



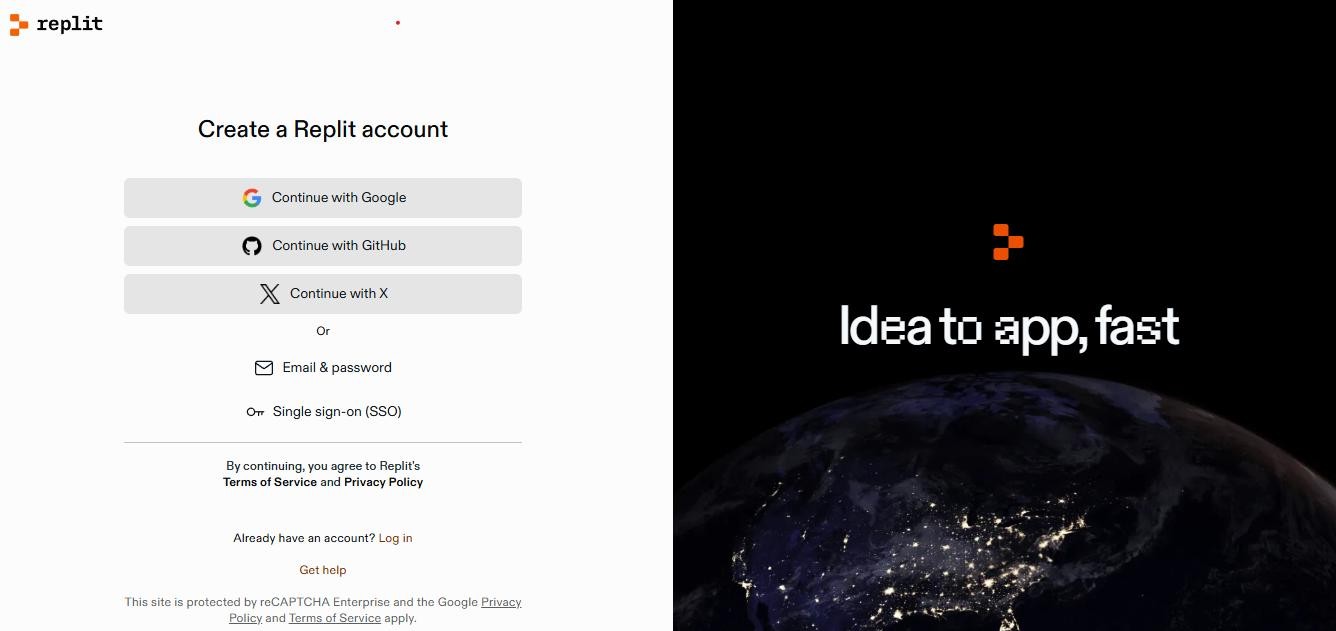
## Introducción a Replit.

En el navegador buscamos la página de replit y le damos click en “Sing Up”.



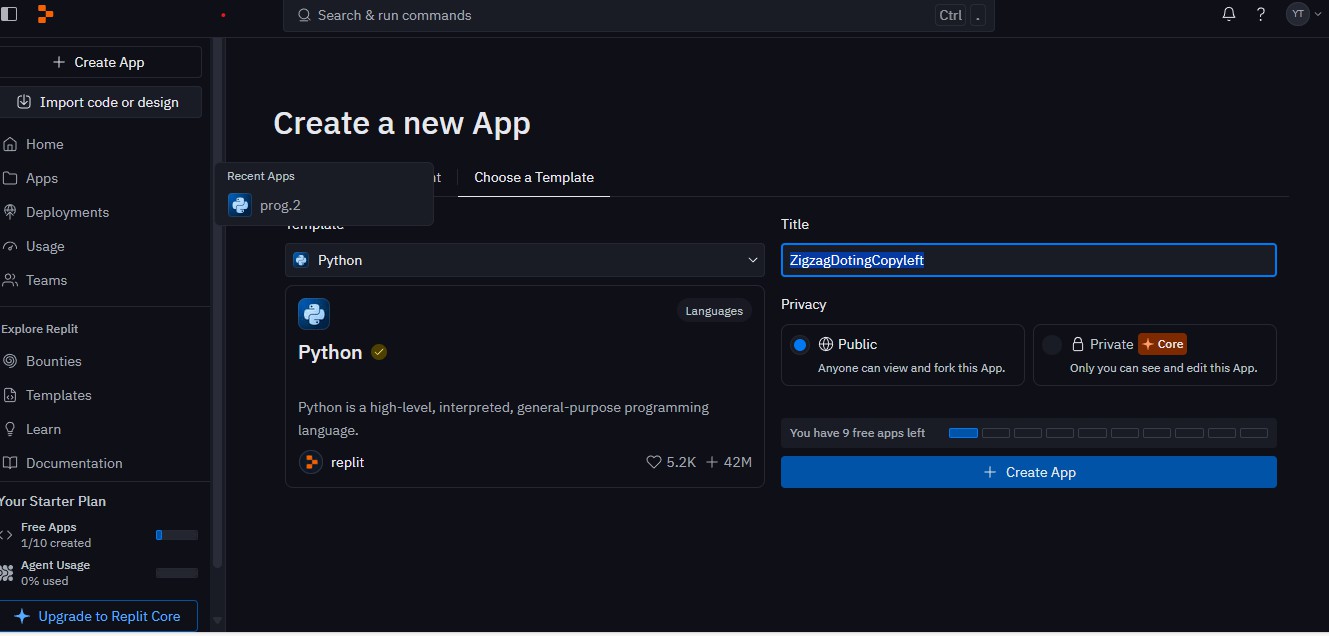
## Vinculación de cuenta.

En esta parte vamos a vincular con nuestra de cuenta de GitHub para poder iniciar sesión en Replit.



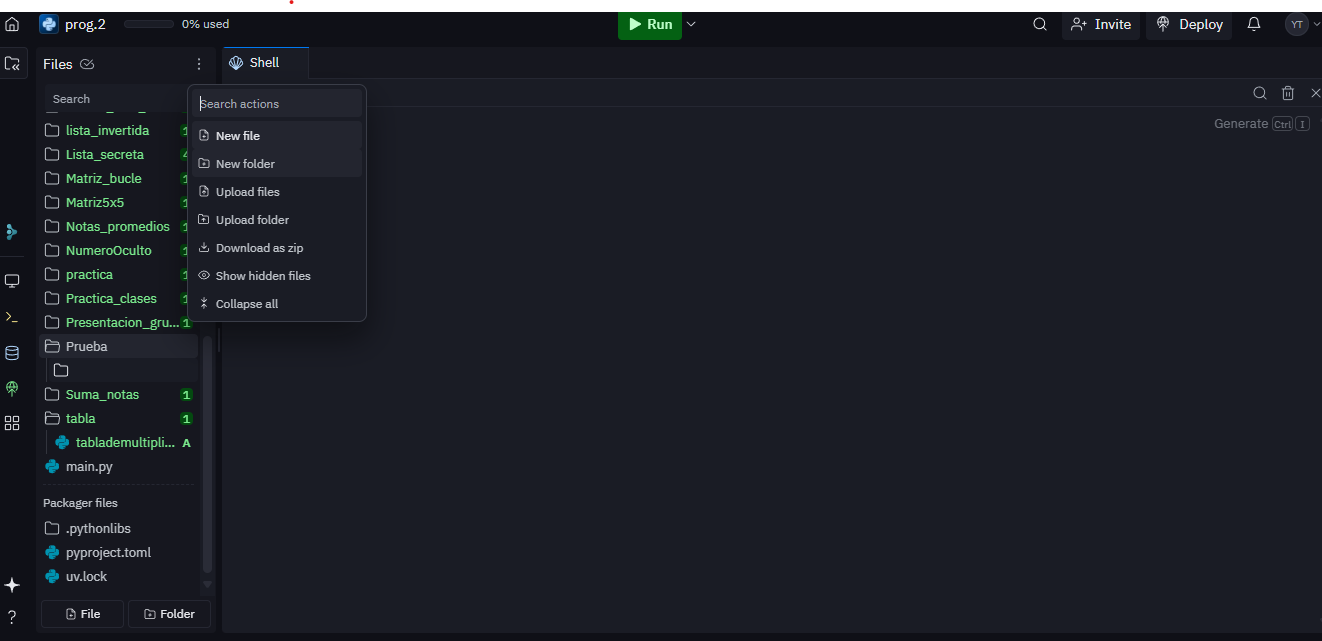
## Creación de repositorios en Replit.

En esta parte le damos click en “+ Create a new App” y después le damos click en “Chosse a Template” y buscamos el lenguaje de programación Python y le asignamos un nombre al repositorio y ya procedemos a darle click en “+ Create App”.



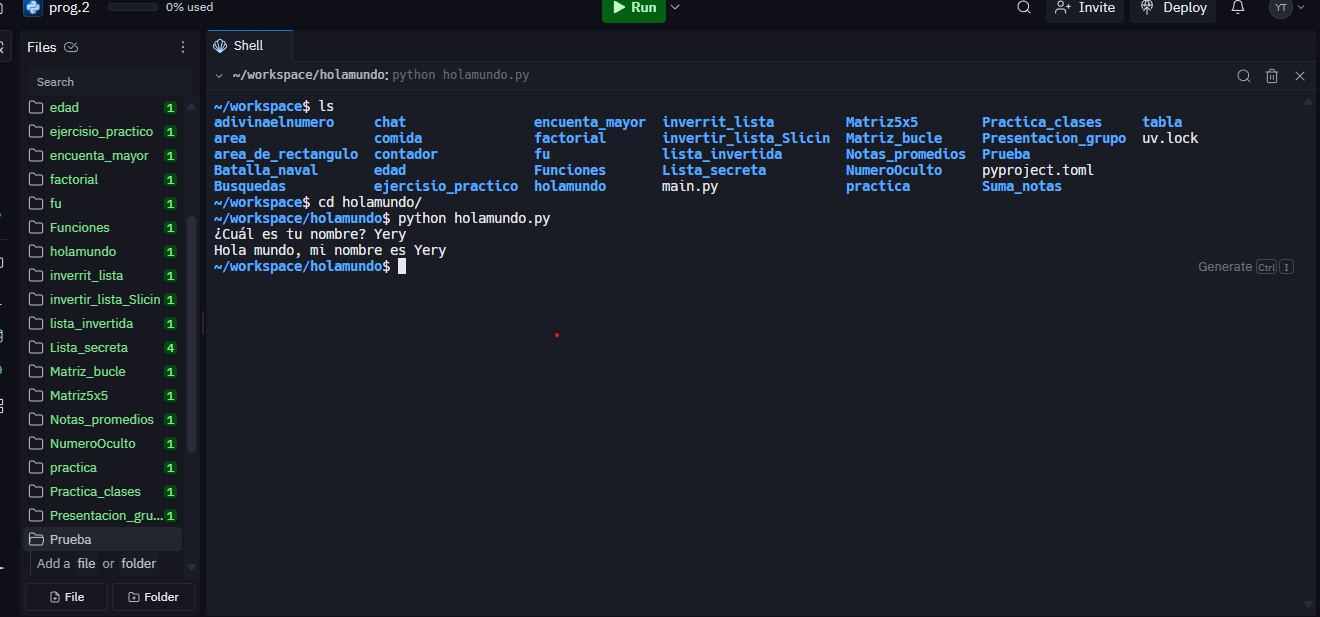
## Creación de folder Replit.

En esta parte creamos los folder donde vamos a meter vuestros códigos para poder tenerlo de mejor manera organizado para que cada código tenga su folder.



## Ejecutar códigos mediante la consola “Shell”.

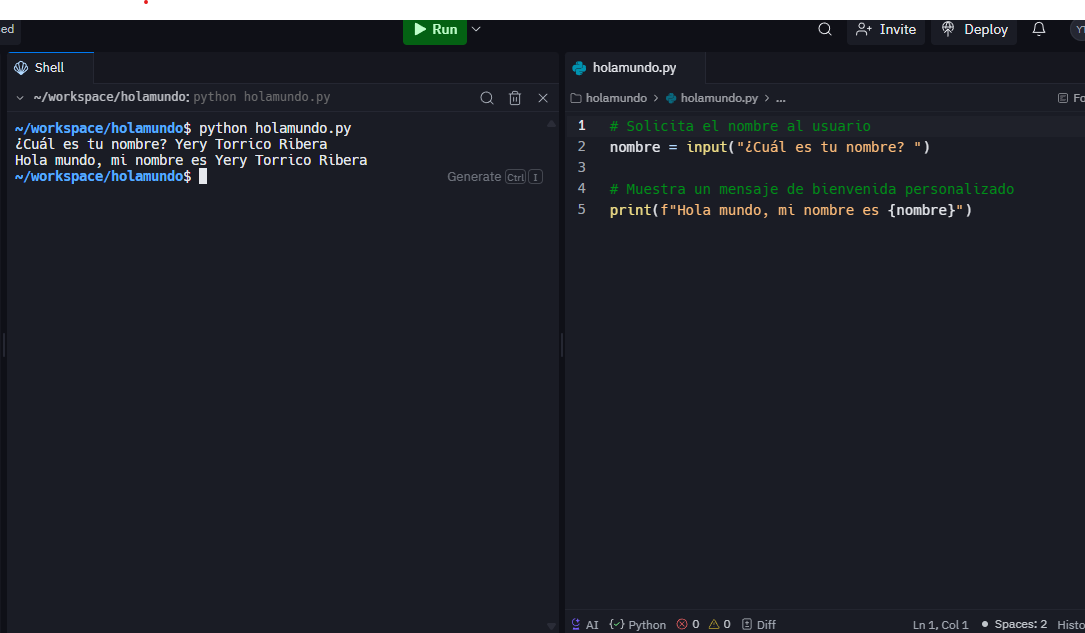
Para poder ejecutar abrimos la consola y escribimos el comando “ls” el cual te va a mostrar todos los folders que uno haya creado ya después de eso para poder entra a un folder se hace con el comando “cd (nombre de folder)” y ya una vez dentro del folder metemos el comando “Python (nombre del código)” y ya de esa manera ya va correr el código y el programa que hallas programado.



## EJECUCION DE CODIGOS.

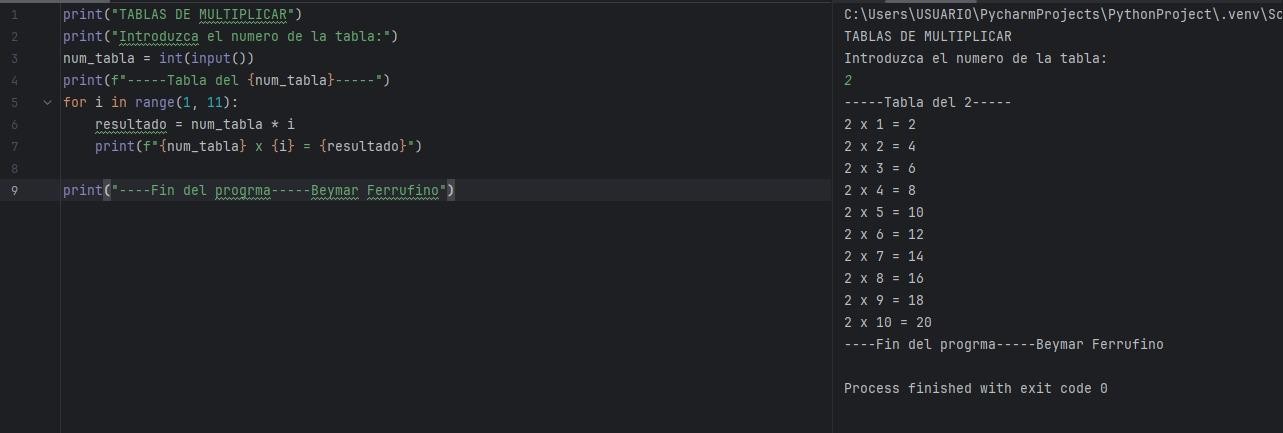
## Código Hola mundo.

En este código se utilizó las funciones “input” y la función “print”. Para poder meter información mediante el teclado usamos la función “input” y para poder ver usamos la función “print” información que nos da el código.



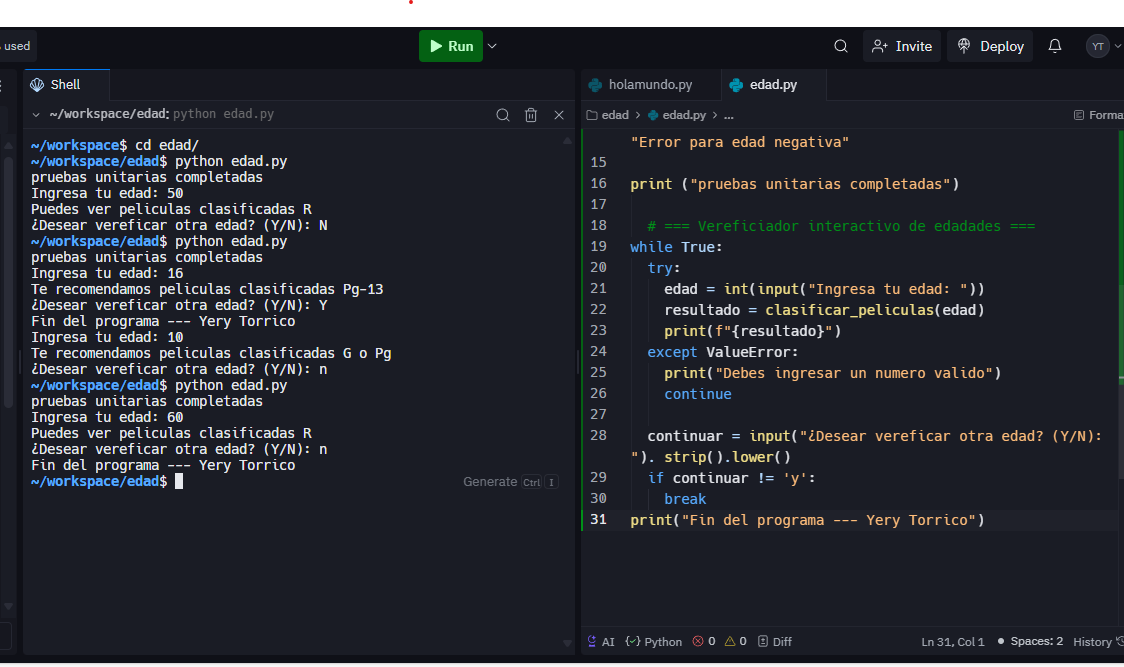
## Código Tabla de multiplicar.

Este código Python solicita al usuario un número, luego usa un bucle “for” para calcular y mostrar la tabla de multiplicar de ese número del 1 al 10. Utiliza "input()” para obtener el número, “int()” para convertirlo a entero y “print() con f-strings” para formatear la salida.



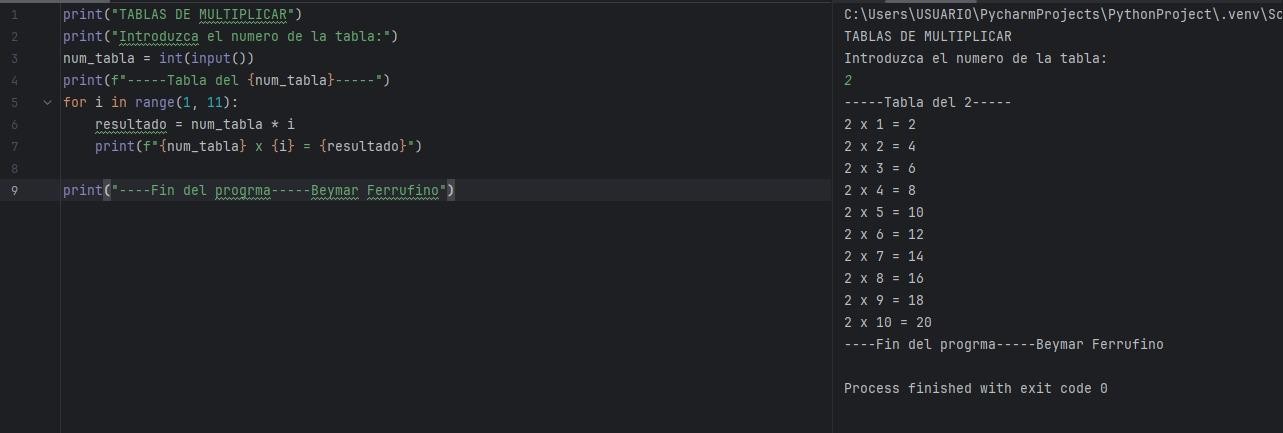
## Código Verificador de Edad para Película.

Este script interactivo solicita la edad del usuario para una clasificación de películas. Utiliza un bucle “while True” para permitir múltiples consultas, con un bloque “try-except” para manejar entradas no numéricas. Después de cada clasificación (realizada por una función externa clasificar\_peliculas), pregunta al usuario si desea continuar o finalizar el programa.



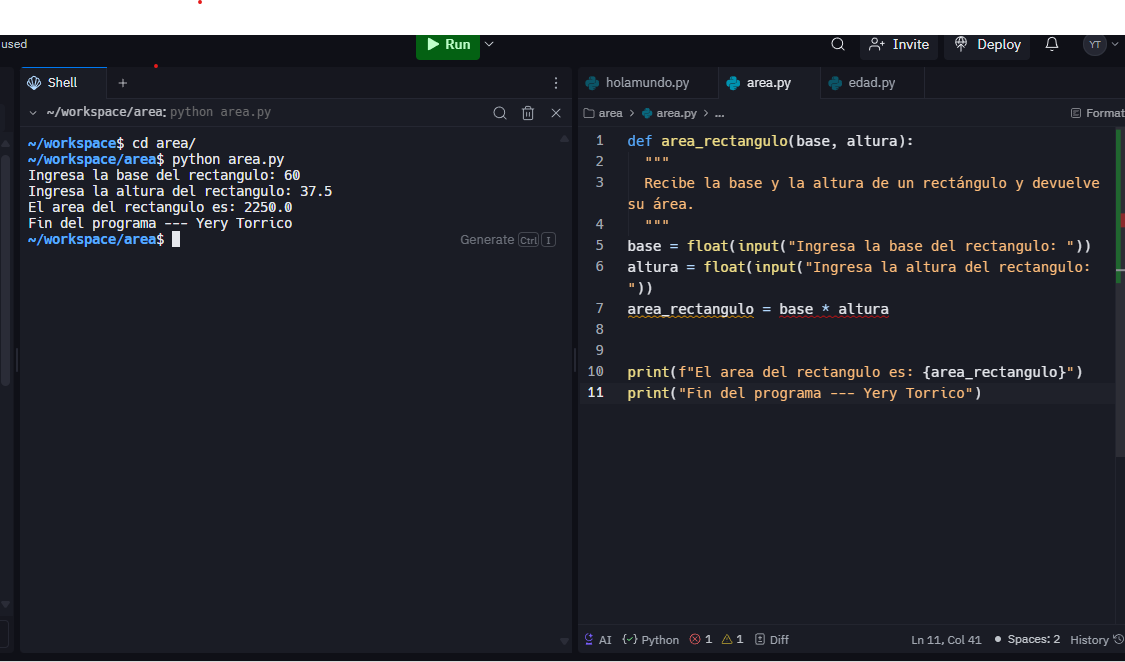
## Código Tabla de Multiplicar.

Este programa pide al usuario mediante en “print” un número y luego calcula y muestra su tabla de multiplicar del 1 al 10. Utiliza la función “input” para obtener el número, lo convierte a un valor numérico y luego un ciclo “for” repite las multiplicaciones e imprime cada resultado.



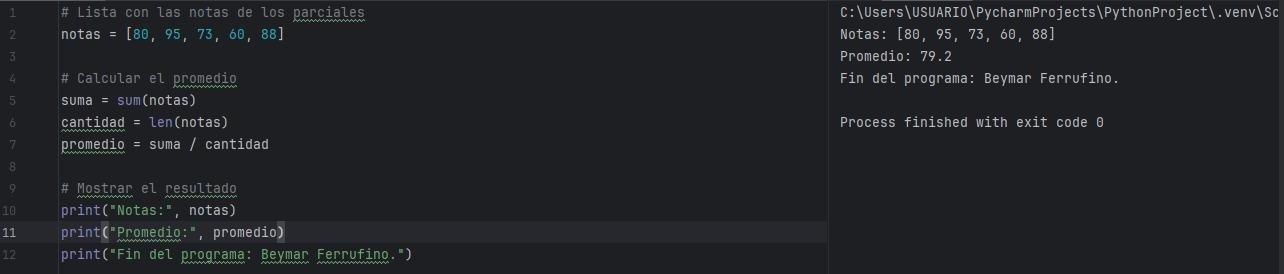
## Código Área de Rectángulo.

En el código se utilizó la función “def” la cual implementamos para definir una función (area\_rectangulo), igual utilizamos “input” para poder ingresar los valores de la base y altura, utilizamos igual La función “float” la cual toma esa cadena de texto y la convierte en un número de punto flotante (un número con decimales) y ya de ultimo utilizamos la función “print” para poder mostrar el resultado.



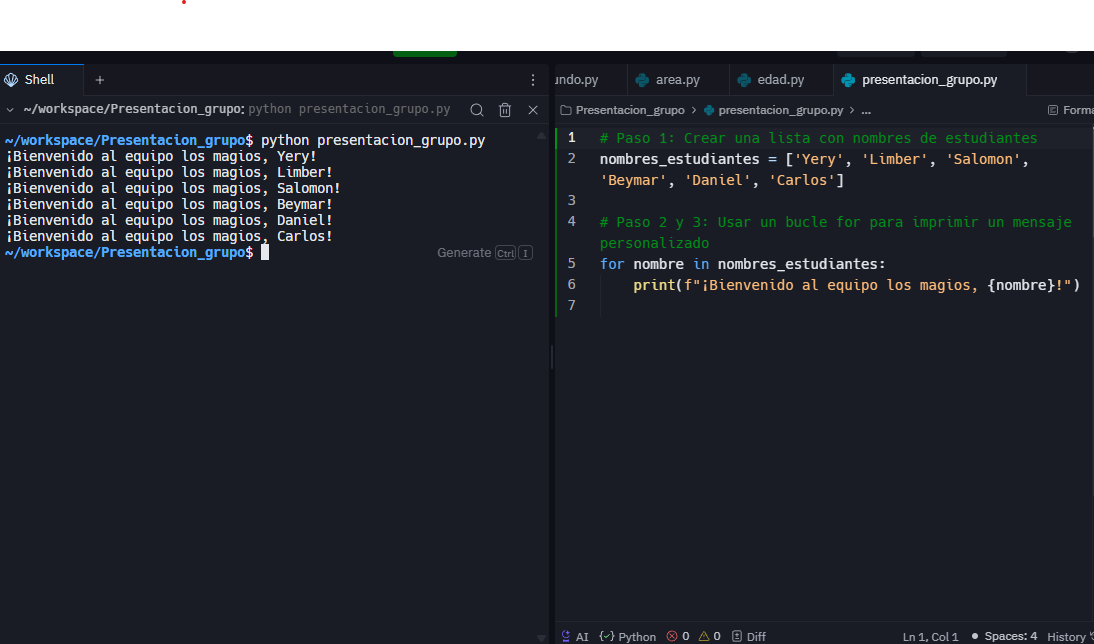
## Código Promedio de notas.

En el código utilizamos “#” para poder implementar la línea de comentario, utilizamos una lista llamada “notas = [80,95,73,60,88]”. Después usamos la función de “sum” que es la función suma y también la función “len” para poder leer las cantidades de valores que tiene la lista y así poder sacar el promedio de notas ya después de haber definido ya podemos poner la operación para poder sacar promedios que es “promedio = suma / Cantidad” y ya utilizamos “print” para poder mostraren pantalla el resultado.



## Código Presentación de Grupo.

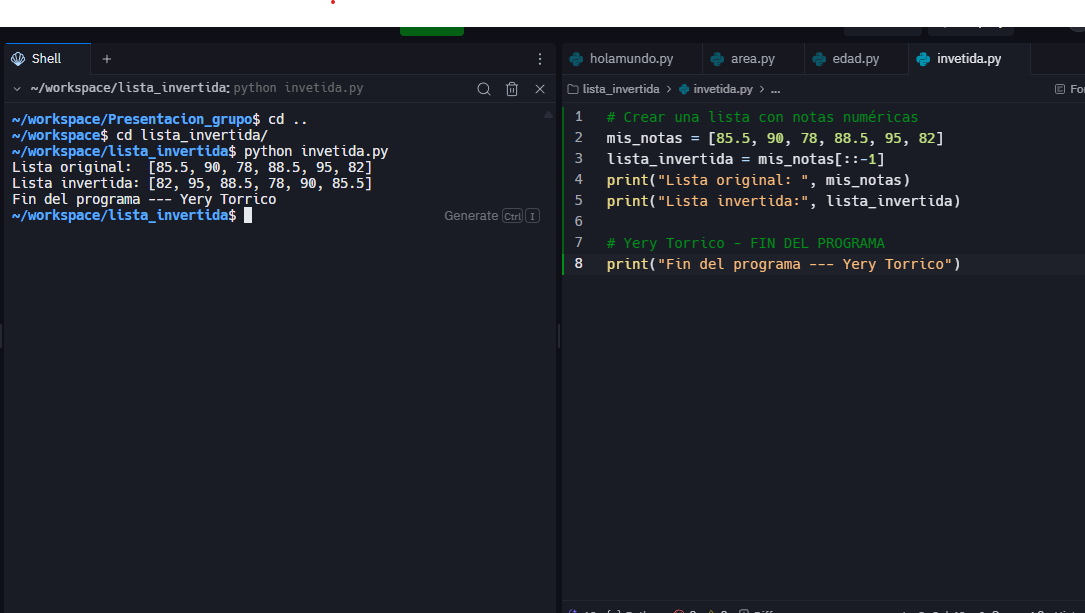
En él se código utiliza “Listas” para almacenar una colección de nombres de estudiantes de manera ordenada. Bucle “for” para iterar sobre cada nombre en la lista, permitiendo ejecutar un bloque de código para cada uno, “print” para mostrar un mensaje de bienvenida personalizado en la consola para cada estudiante.



**}**

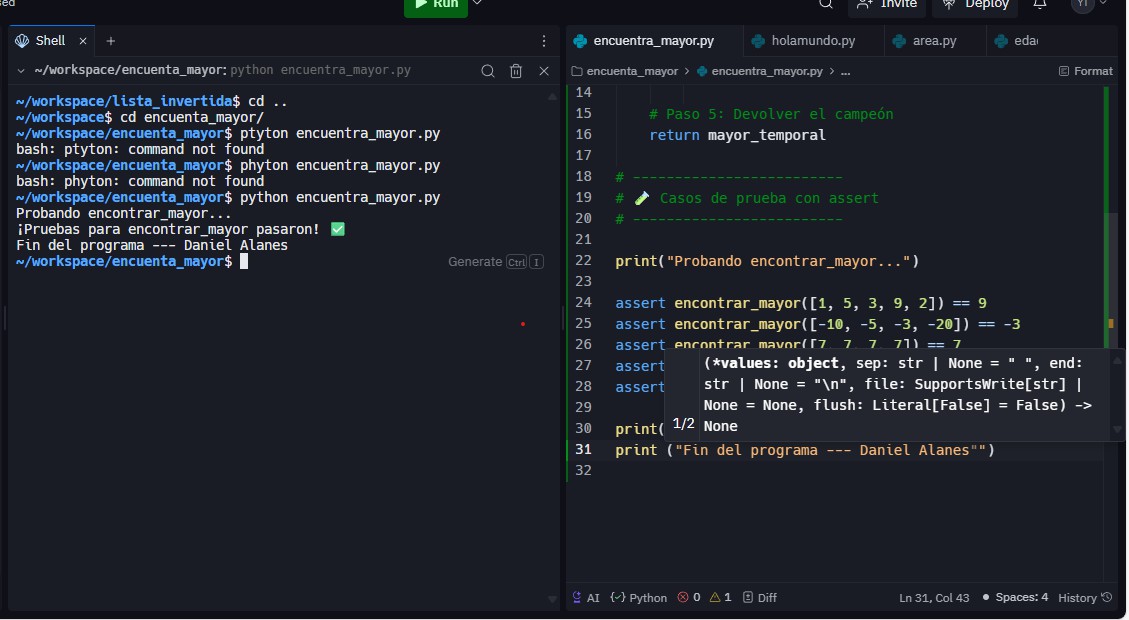
## Código Lista invertida

En este código primeramente tenemos crear un lista en este caso numérica lo cual llamamos “mis\_notas”, y después usamos Invertir la lista usando “slicing” y después usamos la función “print” para poder mostrar ambas listas.



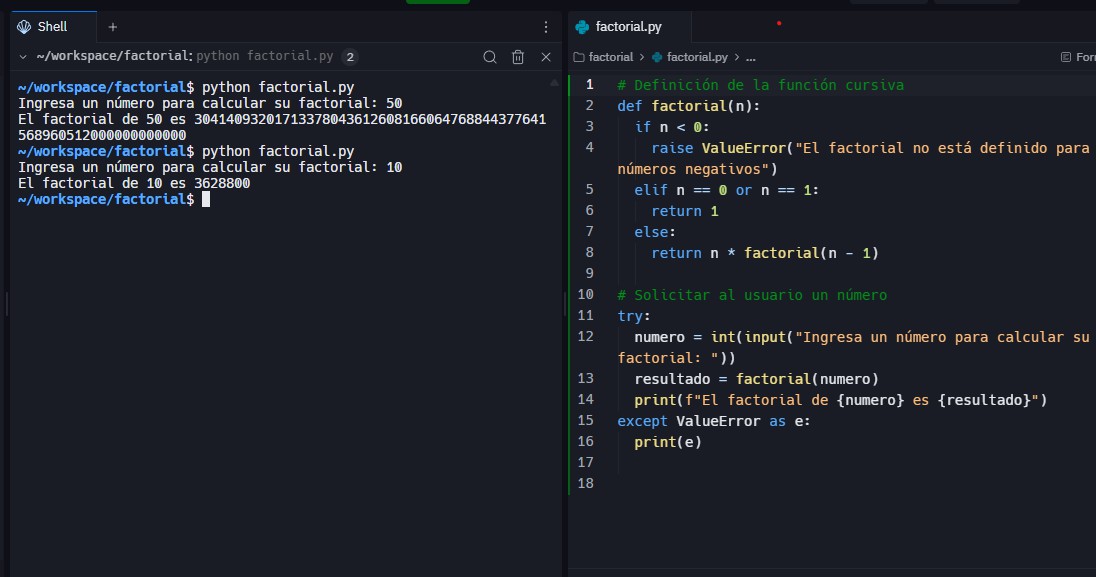
## Código Encuentra al Mayor.

El código utiliza “def” define una función (encontrar mayor) para encapsular la lógica de búsqueda del número más grande, haciéndola reutilizable, bucle “for” itera a través de cada elemento en la lista para comparar y encontrar el valor mayor, “return” devuelve el resultado final (el número mayor encontrado) de la función, “print” muestra mensajes al usuario, incluyendo el estado de las pruebas y el final del programa y “assert” el cual se e usa para realizar pruebas unitarias, verificando que la función encontrar mayor produce los resultados esperados para diferentes entradas.

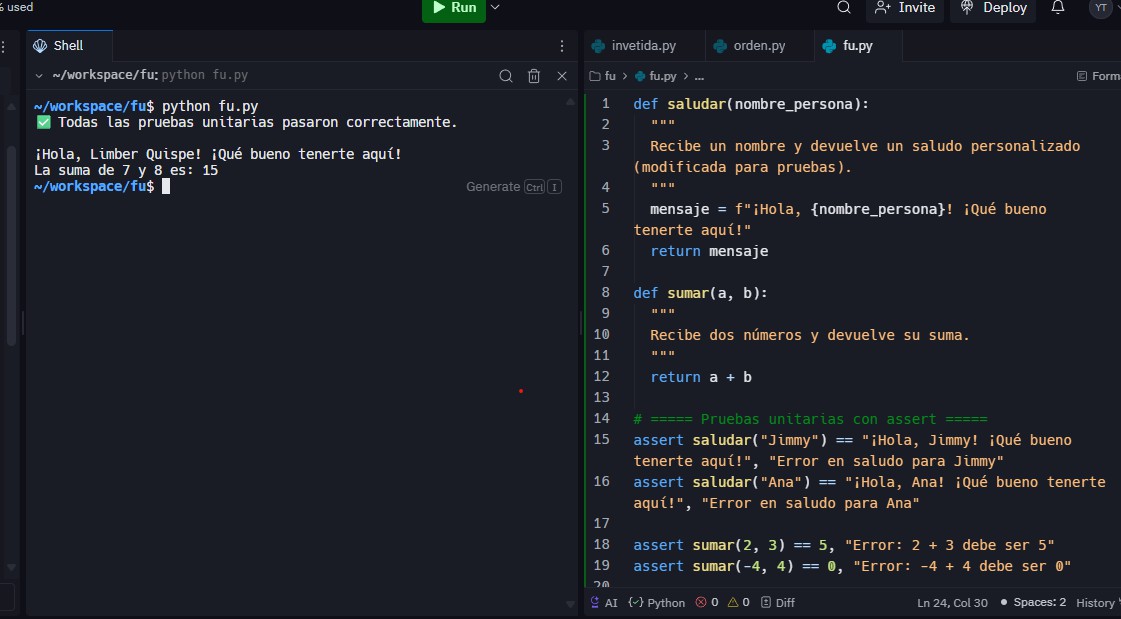


## Código Factorial

En este código usamos una función “recursiva”. Verifica si el número es negativo y lanza un error si lo es. Si es 0 o 1, retorna 1. Para otros casos, multiplica el número por el factorial del anterior. Usa try-except para manejar errores de entrada.



## Código Funciones.

En este código implementamos los “assert” para poder para verificar si una condición es verdadera