

UNIVERSIDAD PRIVADA DOMINGO SAVIO



Actividad 01 Capturas

DOCENTE: Jimmy N. Requena Llorentty

TURNO: Mañana

CARRERA: Ing. En Sistemas

MATERIA: Programación II

ESTUDIANTE: Franz A. Almanza Galindo

Santa Cruz- Bolivia

Fecha y hora actual: 2025-06-16 20:53:07

Captura #1 README.md



The screenshot shows a GitHub repository page for a file named 'README.md'. The repository is owned by 'arastoru'. The file has 24 lines (17 loc) and 917 bytes. It was last updated 13 minutes ago. The README content is as follows:

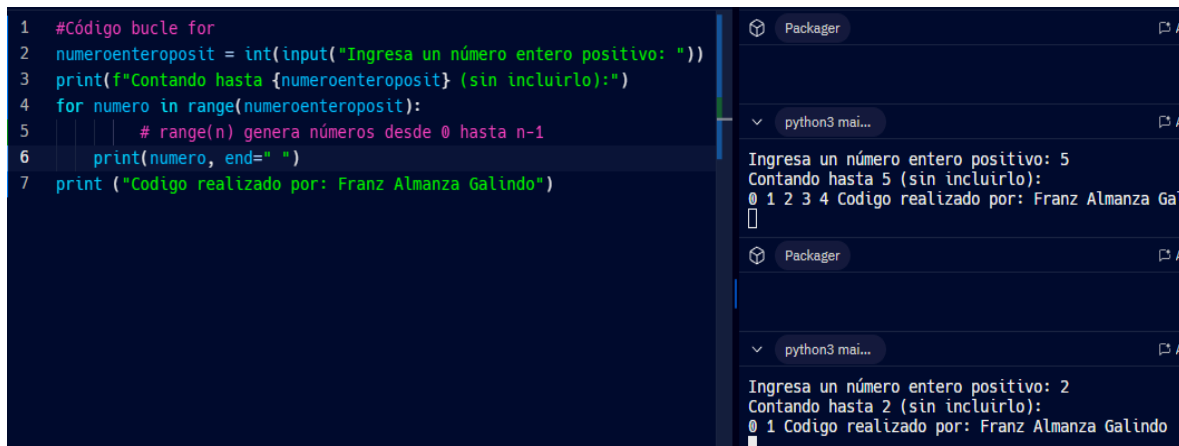
Clasificación de Películas por Edad

Este es un pequeño programa en Python que recomienda películas según la edad del usuario. El programa solicita la edad al usuario y le muestra recomendaciones de clasificación de películas apropiadas para su edad.

Cómo funciona

1. El usuario ingresa su edad.
2. Dependiendo del valor ingresado, el programa recomienda una clasificación de película:
 - o **Edad negativa:** Muestra un mensaje de error.
 - o **18 años o más:** Recomendación de películas clasificadas **R**.
 - o **Entre 13 y 17 años:** Recomendación de películas **PG-13**.

Captura #2 Bucle “for”



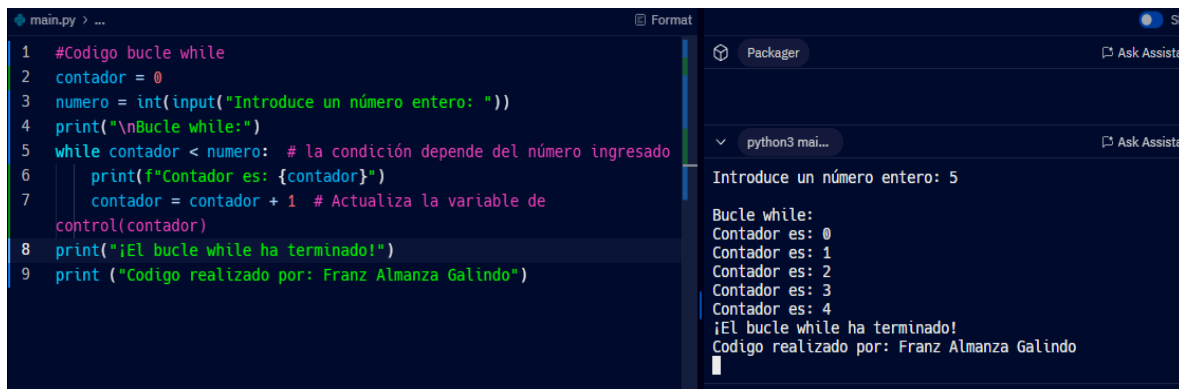
The screenshot shows a code editor with a Python script using a 'for' loop. The code is as follows:

```
1 #Código bucle for
2 numeroenteroposit = int(input("Ingresa un número entero positivo: "))
3 print(f"Contando hasta {numeroenteroposit} (sin incluirlo):")
4 for numero in range(numeroenteroposit):
5     # range(n) genera números desde 0 hasta n-1
6     print(numero, end=" ")
7 print ("Codigo realizado por: Franz Almanza Galindo")
```

The output of the program is shown in the terminal:

```
Ingresa un número entero positivo: 5
Contando hasta 5 (sin incluirlo):
0 1 2 3 4 Codigo realizado por: Franz Almanza Galindo
```

Captura #3 Bucle “while”



The screenshot shows a code editor with a Python script using a 'while' loop. The code is as follows:

```
1 #Codigo bucle while
2 contador = 0
3 numero = int(input("Introduce un número entero: "))
4 print("\nBucle while:")
5 while contador < numero: # la condición depende del número ingresado
6     print(f"Contador es: {contador}")
7     contador = contador + 1 # Actualiza la variable de control(contador)
8 print("¡El bucle while ha terminado!")
9 print ("Codigo realizado por: Franz Almanza Galindo")
```

The output of the program is shown in the terminal:

```
Introduce un número entero: 5

Bucle while:
Contador es: 0
Contador es: 1
Contador es: 2
Contador es: 3
Contador es: 4
¡El bucle while ha terminado!
Codigo realizado por: Franz Almanza Galindo
```

Captura #4 Clasificación películas en cine

```
1 # Código de que películas puedes ver en el cine
2 edad_str = input("Bienvenido al cine, ¿cuál es tu edad?: ")
3 edad = int(edad_str)
4
5 if edad < 0:
6     print("Edad no válida. Por favor, ingresa un número positivo.")
7 elif edad >= 18:
8     print("¡Puedes ver películas clasificadas R!")
9 elif edad >= 13: # Python llega aquí solo si edad NO es >= 18 y NO
10    es < 0
11    print("Puedes ver películas clasificadas PG-13.")
12 else: # Si no es < 0, ni >= 18, ni >= 13, entonces debe ser < 13 y
13    >= 0
14    print("Te recomendamos películas clasificadas G o P.")
15 print("Codigo realizado por Franz Almanza Galindo")
```

Package Manager

python3 mai... Ask Assistant

Bienvenido al cine, ¿cuál es tu edad?: 16
Puedes ver películas clasificadas PG-13.
Codigo realizaod por Franz Almanza Galindo

Package Manager Ask Assistant

python3 mai... Ask Assistant

Bienvenido al cine, ¿cuál es tu edad?: 18
¡Puedes ver películas clasificadas R!
Codigo realizaod por Franz Almanza Galindo

Package Manager Ask Assistant

python3 mai... Ask Assistant

Bienvenido al cine, ¿cuál es tu edad?: 14
Puedes ver películas clasificadas PG-13.
Codigo realizado por Franz Almanza Galindo

Captura #5 Tabla multiplicar

```
1 # Código de la tabla de multiplicar
2 num_tabla = int(input("Ingresa un número para ver su tabla de
3 multiplicar: "))
4 print(f"--- Tabla del {num_tabla} ---")
5
6 for i in range(1, 11): # i tomará valores de 1 a 10
7     resultado = num_tabla * i
8     print(f"{num_tabla} x {i} = {resultado}")
9 print("Codigo realizado por: Franz Almanza Galindo")
```

Package Manager Ask Assistant

python3 mai... Ask Assistant

Ingresa un número para ver su tabla de multiplicar: 5
--- Tabla del 5 ---
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
Codigo realizado por: Franz Almanza Galindo

Captura #6 Refactorizar cálculo área rectángulo

```
#Codigo de refactorizacion (incluimos listas) rectangulo
def calcular_area_rectangulo(base, altura):
    return base * altura

rectangulos = [] #es una variable para almacenar los datos de los rectangulos
contador = 1 #es la variable para contar los rectangulos
while True:
    base = int(input("Ingrese la base del rectángulo: "))
    altura = int(input("Ingrese la altura del rectángulo: "))
    area = calcular_area_rectangulo(base, altura)
    rectangulos.append((contador, base, altura, area)) #el append te permite agregar
    elementos a la lista
    print(f"El área del rectángulo {contador} ({base}x{altura}) es: {area}")
    contador += 1
    continuar = input("¿Desea ingresar otro rectángulo? Presione Enter para continuar o 'n'
para salir: ")
    if continuar.lower() == 'n':
        break
ver_datos = input("¿Ver los datos de los rectángulos guardados? Presione 1 para ver: ")
if ver_datos == '1':
    for rectangulo in rectangulos:
        numero, base, altura, area = rectangulo
        print(f"Rectángulo {numero}: Base = {base}, Altura = {altura}, Área = {area}")
print("Codigo realizado por: Franz Almanza Galindo")
```

python3 mai... Ask Assistant ✓

Ingrese la base del rectángulo: 2
Ingrese la altura del rectángulo: 4
El área del rectángulo 1 (2x4) es: 8
¿Desea ingresar otro rectángulo? Presione Enter para continuar o 'n' para salir: 45
Ingrese la base del rectángulo: 4
Ingrese la altura del rectángulo: 7
El área del rectángulo 2 (4x7) es: 28
¿Desea ingresar otro rectángulo? Presione Enter para continuar o 'n' para salir: n
¿Desea ver los datos de los rectángulos guardados? Presione 1 para ver: 1
Rectángulo 1: Base = 2, Altura = 4, Área = 8
Rectángulo 2: Base = 4, Altura = 7, Área = 28
Codigo realizado por: Franz Almanza Galindo

Captura #7 Adivina el número secreto

```
# Código del juego adivina el número secreto
# El número secreto ahora es: 6 pero puede ser cambiado por cualquier otro número entero.
numero_secreto = 6
print("¡Bienvenido al juego Adivina el Número Secreto!")
print("Estoy pensando en el...")
# Iniciamos la variable adivinanza con un valor diferente al número secreto para que el
bucle WHILE se ejecute al menos una vez.
adivinanza = 0
# Bucle que continua mientras la adivinanza no sea correcta
while adivinanza != numero_secreto:
    try:
        # Pide al usuario que adivine el número.
        adivinanza = int(input("Ingresa el número que crees que es el secreto: "))
        # Aqui le damos una pista si el número no es correcto
        if adivinanza > numero_secreto:
            print("Demasiado alto. Intenta de nuevo.")
        elif adivinanza < numero_secreto:
            print("Demasiado bajo. Intenta de nuevo.")
    except ValueError:
        # Maneja el error si el usuario no ingresa un número entero (int)
        print("Ingresa un número entero.")
# Mensaje (print) que aparece cuando el número fue adivinado
print(f"¡Correcto! El número era {numero_secreto}. ¡Felicidades, adivinaste!")
```

python3 mai... Ask Assistant ✓

¡Bienvenido al juego Adivina el Número Secreto!
Estoy pensando en el...
Ingresa el número que crees que es el secreto: 4
Demasiado bajo. Intenta de nuevo.
Ingresa el número que crees que es el secreto: 8
Demasiado alto. Intenta de nuevo.
Ingresa el número que crees que es el secreto: 7
Demasiado alto. Intenta de nuevo.
Ingresa el número que crees que es el secreto: 6
¡Correcto! El número era 6. ¡Felicidades, adivinaste!
Codigo realizado por: Franz Almanza Galindo

Captura #8 Comida boliviana favorita

```
#Codigo lista de comidas bolivianas favoritas
# Crea una lista llamada comidas_favoritas con los nombres (strings) de tus 3 comidas
bolivianas favoritas.
comidas_favoritas = ["Picante de Pollo", "Pique Macho", "Sopa de mani"]

# Modificamos el primer elemento (índice 0) a "Majadito"
comidas_favoritas[0] = "Majadito"

# Imprimimos la lista completa para verificar que se creó bien.
print("Mi lista de comidas favoritas es:")
print(comidas_favoritas)

# Imprime un mensaje que diga: "Mi segunda comida favorita es: [el nombre de la segunda
comida]".
print(f"\nMi segunda comida favorita es: {comidas_favoritas[1]}")

# Usamos len() al imprimir un mensaje
# len() nos da la cantidad de elementos en la lista.
print(f"Mi lista de comidas favoritas tiene {len(comidas_favoritas)} elementos.")
print("Codigo realizado por: Franz Almanza")
```

Packager Ask Assistant

python3 mai... Ask Assistant

Mi lista de comidas favoritas es:
['Picante de Pollo', 'Pique Macho', 'Sopa de mani']

Mi segunda comida favorita es: Pique Macho
Mi lista de comidas favoritas tiene 3 elementos.
Codigo realizado por: Franz Almanza

Packager Ask Assistant

python3 mai... Ask Assistant

Mi lista de comidas favoritas es:
['Majadito', 'Pique Macho', 'Sopa de mani']

Mi segunda comida favorita es: Pique Macho
Mi lista de comidas favoritas tiene 3 elementos.
Codigo realizado por: Franz Almanza

Captura #9 Lista de notas y promedio

```
#Codigo para calcular el promedio de notas
# Creamos una lista llamada lista_mis_notas con las notas.
lista_mis_notas = [60.5, 74, 80.5, 90.5]
# Creamos una variable suma_total inicializada en 0.
suma_total = 0

# Usamos un bucle for para recorrer lista_mis_notas.
# En cada iteración, añade la nota actual a suma_total.
for nota in lista_mis_notas:
    suma_total = suma_total + nota

# Calcula el promedio: promedio = suma_total / len(lista_mis_notas).
# len(lista_mis_notas) nos da la cantidad de elementos en la lista.
promedio = suma_total / len(lista_mis_notas)

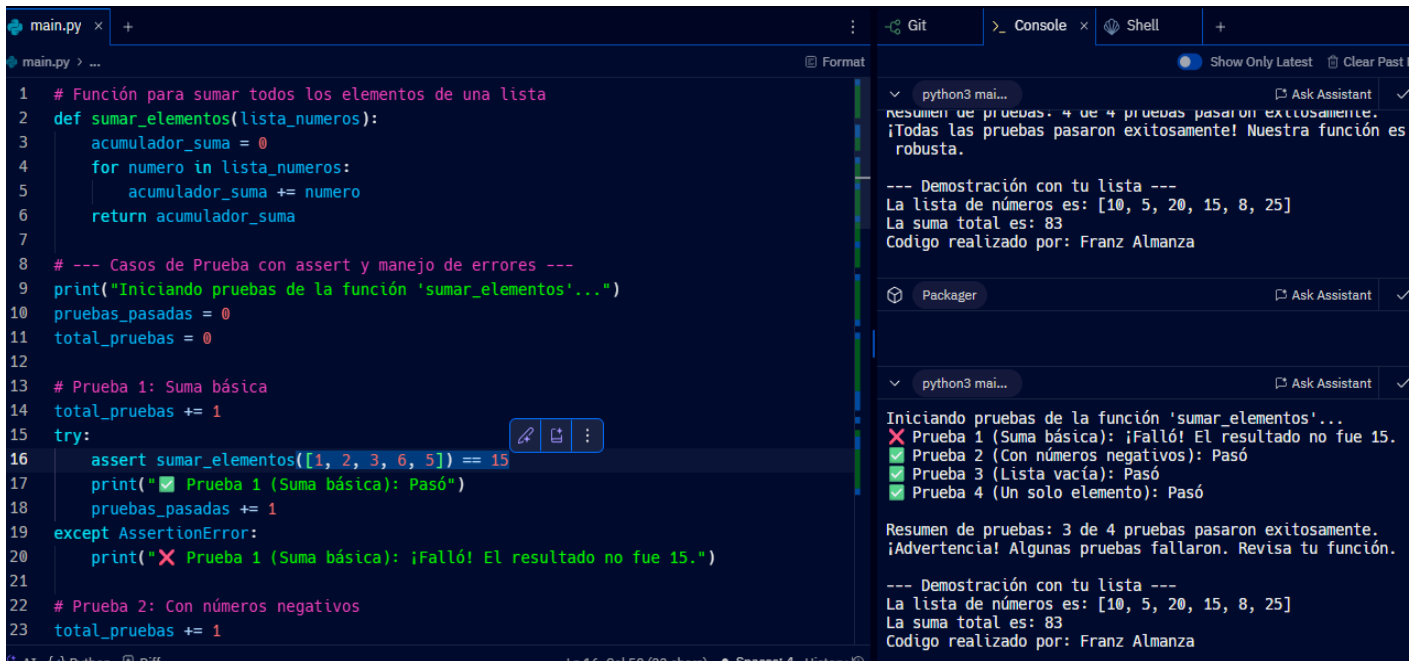
# Imprime(print) la suma total y el promedio.
print(f"Mis notas son: {lista_mis_notas}")
print(f"La suma total de las notas es: {suma_total:.2f}")
print(f"El promedio de las notas es: {promedio:.2f}") # :.2f para mostrar dos decimales
print("Codigo realizado por: Franz Almanza")
```

Packager Ask Assistant

python3 mai... Ask Assistant

Mis notas son: [60.5, 74, 80.5, 90.5]
La suma total de las notas es: 305.50
El promedio de las notas es: 76.38
Codigo realizado por: Franz Almanza

Captura #10 Sumar elementos de una lista



The screenshot shows a code editor with a Python file named `main.py`. The code defines a function `sumar_elementos` that takes a list of numbers and returns their sum. It includes test cases using `assert` and `print` statements. The right sidebar shows the output of the tests, indicating that the function passed all tests.

```
1 # Función para sumar todos los elementos de una lista
2 def sumar_elementos(lista_numeros):
3     acumulador_suma = 0
4     for numero in lista_numeros:
5         acumulador_suma += numero
6     return acumulador_suma
7
8 # --- Casos de Prueba con assert y manejo de errores ---
9 print("Iniciando pruebas de la función 'sumar_elementos'...")
10 pruebas_pasadas = 0
11 total_pruebas = 0
12
13 # Prueba 1: Suma básica
14 total_pruebas += 1
15 try:
16     assert sumar_elementos([1, 2, 3, 6, 5]) == 15
17     print("✅ Prueba 1 (Suma básica): Pasó")
18     pruebas_pasadas += 1
19 except AssertionError:
20     print("❌ Prueba 1 (Suma básica): ¡Falló! El resultado no fue 15.")
21
22 # Prueba 2: Con números negativos
23 total_pruebas += 1
```

Output:

```
python3 mai...
Resumen de pruebas: 4 de 4 pruebas pasaron exitosamente.
¡Todas las pruebas pasaron exitosamente! Nuestra función es robusta.

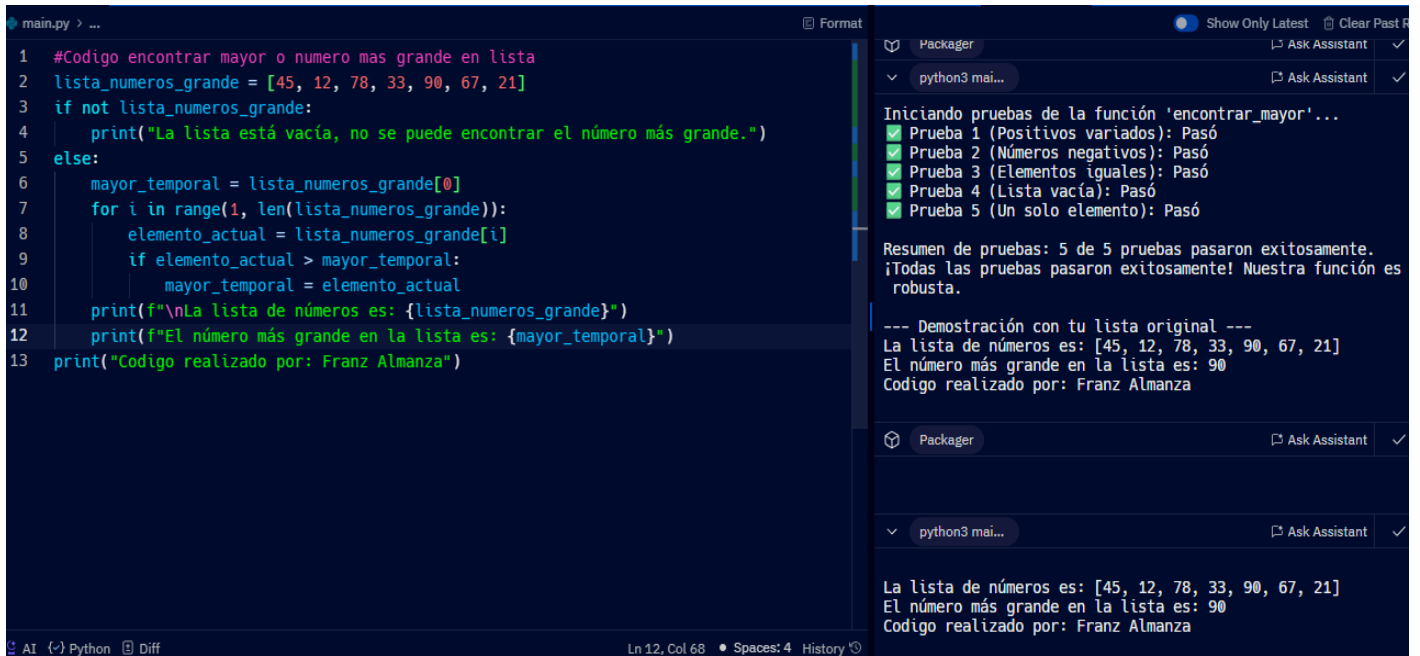
--- Demostración con tu lista ---
La lista de números es: [10, 5, 20, 15, 8, 25]
La suma total es: 83
Codigo realizado por: Franz Almanza

Packager
python3 mai...
Iniciando pruebas de la función 'sumar_elementos'...
❌ Prueba 1 (Suma básica): ¡Falló! El resultado no fue 15.
✅ Prueba 2 (Con números negativos): Pasó
✅ Prueba 3 (Lista vacía): Pasó
✅ Prueba 4 (Un solo elemento): Pasó

Resumen de pruebas: 3 de 4 pruebas pasaron exitosamente.
¡Advertencia! Algunas pruebas fallaron. Revisa tu función.

--- Demostración con tu lista ---
La lista de números es: [10, 5, 20, 15, 8, 25]
La suma total es: 83
Codigo realizado por: Franz Almanza
```

Captura #11 Encontrar número mayor en lista



The screenshot shows a code editor with a Python file named `main.py`. The code defines a function `encontrar_mayor` that takes a list of numbers and returns the maximum value. It includes test cases using `assert` and `print` statements. The right sidebar shows the output of the tests, indicating that the function passed all tests.

```
1 #Codigo encontrar mayor o numero mas grande en lista
2 lista_numeros_grande = [45, 12, 78, 33, 90, 67, 21]
3 if not lista_numeros_grande:
4     print("La lista está vacía, no se puede encontrar el número más grande.")
5 else:
6     mayor_temporal = lista_numeros_grande[0]
7     for i in range(1, len(lista_numeros_grande)):
8         elemento_actual = lista_numeros_grande[i]
9         if elemento_actual > mayor_temporal:
10             mayor_temporal = elemento_actual
11     print(f"\nLa lista de números es: {lista_numeros_grande}")
12     print(f"El número más grande en la lista es: {mayor_temporal}")
13 print("Codigo realizado por: Franz Almanza")
```

Output:

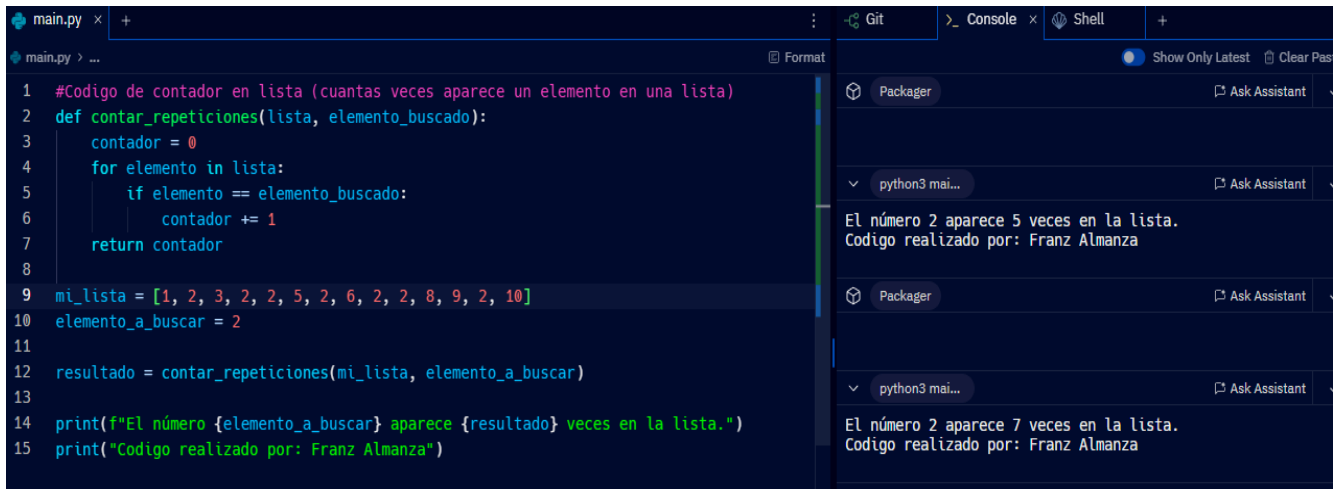
```
Packager
python3 mai...
Iniciando pruebas de la función 'encontrar_mayor'...
✅ Prueba 1 (Positivos variados): Pasó
✅ Prueba 2 (Números negativos): Pasó
✅ Prueba 3 (Elementos iguales): Pasó
✅ Prueba 4 (Lista vacía): Pasó
✅ Prueba 5 (Un solo elemento): Pasó

Resumen de pruebas: 5 de 5 pruebas pasaron exitosamente.
¡Todas las pruebas pasaron exitosamente! Nuestra función es robusta.

--- Demostración con tu lista original ---
La lista de números es: [45, 12, 78, 33, 90, 67, 21]
El número más grande en la lista es: 90
Codigo realizado por: Franz Almanza

Packager
python3 mai...
La lista de números es: [45, 12, 78, 33, 90, 67, 21]
El número más grande en la lista es: 90
Codigo realizado por: Franz Almanza
```

Captura #12 Contador de elementos en lista



```
1 #Codigo de contador en lista (cuantas veces aparece un elemento en una lista)
2 def contar_repeticiones(lista, elemento_buscado):
3     contador = 0
4     for elemento in lista:
5         if elemento == elemento_buscado:
6             contador += 1
7     return contador
8
9 mi_lista = [1, 2, 3, 2, 2, 5, 2, 6, 2, 2, 8, 9, 2, 10]
10 elemento_a_buscar = 2
11
12 resultado = contar_repeticiones(mi_lista, elemento_a_buscar)
13
14 print(f"El número {elemento_a_buscar} aparece {resultado} veces en la lista.")
15 print("Codigo realizado por: Franz Almanza")
```

El número 2 aparece 7 veces en la lista.
Codigo realizado por: Franz Almanza

Captura #13 Invertir lista



```
1 #Codigo de Invertir lista con range, append
2 def invertir_lista(lista_original):
3     lista_invertida = []
4     for i in range(len(lista_original) - 1, -1, -1):
5         lista_invertida.append(lista_original[i])
6     return lista_invertida
7
8 lista_original = [1, 2, 3, 4, 5]
9 print("Lista original:", lista_original)
10 print("Lista invertida:", invertir_lista(lista_original))
11 print("Codigo realizado por: Franz Almanza")
```

Lista original: [1, 2, 3, 4, 5]
Lista invertida: [5, 4, 3, 2, 1]
Codigo realizado por: Franz Almanza

Captura #14 Búsqueda lineal



```
1 #Codigo de busqueda lineal
2 def busqueda_lineal(lista, elemento_buscado):
3     for i in range(len(lista)):
4         if lista[i] == elemento_buscado:
5             return i
6     return -1
7
8 mi_lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
9 elemento_buscado = 8
10 #sustitui clave por elemento_buscado
11 indice_encontrado = busqueda_lineal(mi_lista, elemento_buscado)
12
13 if indice_encontrado != -1:
14     print(f"El elemento {elemento_buscado} se encontró en el índice: {indice_encontrado}")
15 else:
16     print(f"El elemento {elemento_buscado} no se encontró en la lista.")
17 print("Codigo realizado por: Franz Almanza")
```

El elemento 8 se encontró en el índice: 7
Codigo realizado por: Franz Almanza

Captura #15 Búsqueda Binaria



```
1 #Codigo de busqueda binaria
2 def busqueda_binaria(lista_ordenada, clave):
3     izquierda = 0
4     derecha = len(lista_ordenada) - 1
5
6     while izquierda <= derecha:
7         medio = (izquierda + derecha) // 2
8
9         if lista_ordenada[medio] == clave:
10             return medio
11         elif clave > lista_ordenada[medio]:
12             izquierda = medio + 1
13         else:
14             derecha = medio - 1
15     return -1
16
17 # lista de prueba xd
18 lista_ordenada = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]
19 # Cuando la clave es encontrada o no
20 clave_buscada_1 = 26
21 indice_1 = busqueda_binaria(lista_ordenada, clave_buscada_1)
22 if indice_1 != -1:
23     print(f"La clave {clave_buscada_1} se encontró en el índice: {indice_1}")
```

La clave 23 se encontró en el índice: 5
Codigo realizado por: Franz Almanza

La clave 26 no se encontró en la lista.
Codigo realizado por: Franz Almanza