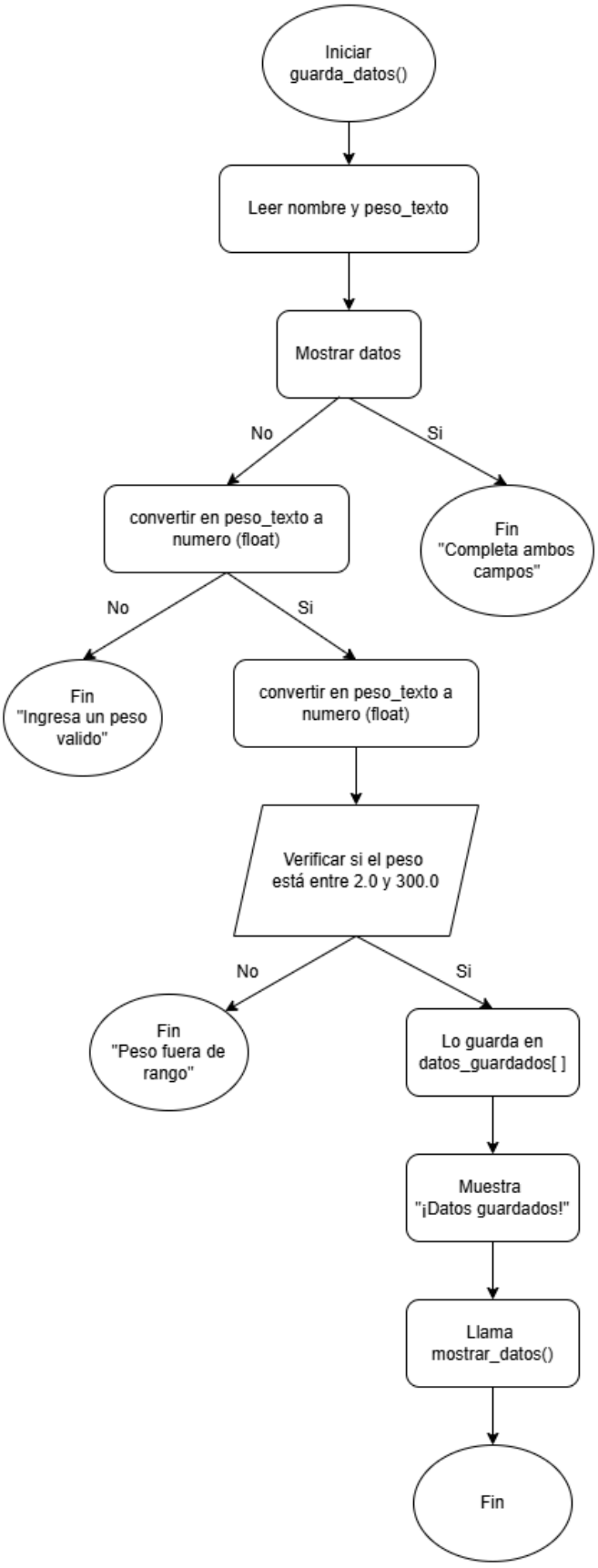
Práctica de desarrollo, cobertura y pruebas

Link GitHub
https://github.com/nicoohb0/registro-personal.git

Función de guardar_datos()

```python
datos_guardados = []

def guardar_datos():
    """Función para guardar el nombre y peso"""

    nombre = entrada_nombre.get()
    peso_texto = entrada_peso.get()

    if not nombre or not peso_texto:
        resultado.config(text="Por favor, completa ambos campos")
        return

    try:
        peso = float(peso_texto)

        peso_minimo = 2.0
        peso_maximo = 300.0

        if peso < peso_minimo or peso > peso_maximo:
            resultado.config(text=f"El peso debe estar entre {peso_minimo} y {peso_maximo} kg")
            return

        datos = {
            "nombre": nombre,
            "peso": peso
        }
        datos_guardados.append(datos)

        resultado.config(text="¡Datos guardados correctamente!")

        entrada_nombre.delete(0, 'end')
        entrada_peso.delete(0, 'end')

        mostrar_datos()

    except ValueError:
        resultado.config(text="Por favor, ingresa un peso válido (número)")

```

1. Diagrama de flujo.



2. Calcular V(G).

Es VG = 4

3. Determinar los caminos independientes de ejecución.

- Camino 1: Inicio -> Campo vacio (True) -> Mostrar mensaje de error -> Fin
- Camino 2: Inicio ->Campo vacio (False) -> Bloque try -> Error de conversión a float(except ValueError) (True) -> Mostrar mensaje de error de peso no valido -> Fin
- Camino 3: Inicio -> Campo vacio (False) -> Bloque try -> Conversión exitosa -> Peso fuera de rango (peso < 2.0 o peso > 300.0) (True) -> Mostrar mensaje de error de rango -> Fin
- Camino 3: Inicio -> Campo vacio (False) -> Bloque try -> Conversión exitosa -> Peso fuera de rango (False) -> Guardar datos -> Mostrar mensaje de éxito -> Fin

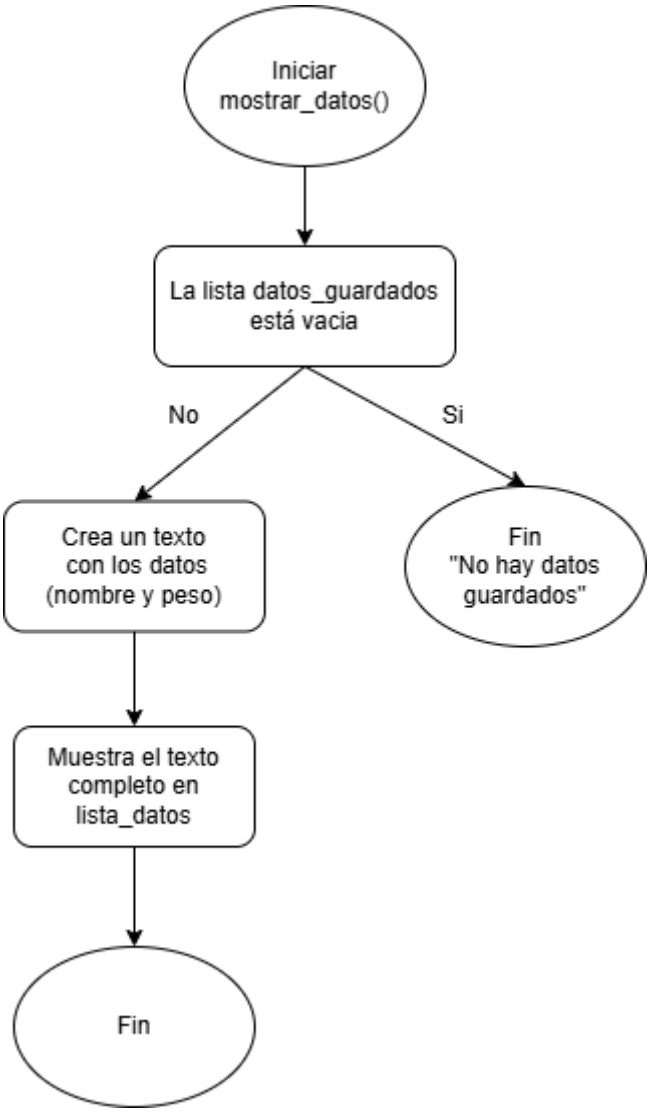
4. Determinar los valores de entrada de la función para probar cada camino independiente.

Camino	Condiciones logicas	Valor de entrada (nombre, peso_texto)	Resultado
Camino 1	nombre o peso_texto es vacio (True)	(“”, “100.0”) o (“Juan”, “”) o (“”, “”)	“Completa en ambos campos”
Camino 2	Peso_texto no es un número válido	(“Juan”, “cien”) o (“Ana”, “10.5”)	“Ingresa un peso válido (número)”
Camino 3	Peso es < 2.0 o > 300.0	(“Pedro”, “1.9”) o (“Luis”, “300.1”)	“El peso debe estar entre 2.0 y 300.0 Kg”
Camino 4	Campos NO vacíos, peso_texto es número, 2.0 < peso < 300.0	(“Maria”, “55.5”) o (“Carlos”, “2.0”)	“¡Datos guardados correctamente!”

Función de mostrar_datos()

```python
def mostrar_datos():
    """Función para mostrar todos los datos guardados"""

    if not datos_guardados:
        lista_datos.config(text="No hay datos guardados")
        return

    texto_datos = "Datos guardados:\n\n"
    for i, dato in enumerate(datos_guardados, 1):
        texto_datos += f"{i}. Nombre: {dato['nombre']} - Peso: {dato['peso']} kg\n"

    lista_datos.config(text=texto_datos)
```

1. Diagrama de flujo



2. Calcular V(G).

Es VG = 2

3. Determinar los caminos independientes de ejecución.

- Camino 1: Inicio -> Lista vacía (True) -> Mostrar “No hay datos guardados” -> Fin
- Camino 2: Inicio -> Lista vacía (False) -> Crear el texto con los datos -> Mostrar texto completo -> Fin

4. Determinar los valores de entrada de la función para probar cada camino independiente.

Camino	Condiciones logicas	Estado de datos_guardados	Resultado
Camino 1	Datos_guardados está vacía (True)	[ ] Lista vacía	“No hay datos guardados”
Camino 2	Datos_guardados tiene elementos (False)	[{‘nombre’: ‘A’, ‘peso’: 50.0}] Con uno o más elementos	Mostrar “Datos guardados:\n\n1. Nombre: A Peso: 50.0 Kg”