Asignatura	Datos del alumno	Fecha
Métodos Numéricos	Apellidos: Jimenez Acosta	
	Nombre: Ronaldo	

Laboratorio: Solución

El código fuente se puede encontrar en la plataforma de GitHub, en el siguiente enlace.

Ver código fuente

Nota: los archivos sol_cuadratica_pseint.py y minimo_pseint.py son los generados por la aplicación, los otros archivos son mi propia implementación.

Ejecución de los algoritmos en Visual Studio Code

```
JIMCOSTDEV@DESKTOP-GI2F5PJ MINGW64 ~/Desktop/Cursos/Unir/III/Metodos_Numericos
$ python sol_cuadratica_pseint.py
Ingrese los coeficientes A, B y C:
1
2
-3
X1:  -3.0
X2:  1.0

JIMCOSTDEV@DESKTOP-GI2F5PJ MINGW64 ~/Desktop/Cursos/Unir/III/Metodos_Numericos
$
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

JIMCOSTDEV@DESKTOP-GI2F5PJ MINGW64 ~/Desktop/Cursos/Unir/III/Metodos_Numericos
$ python minimo_pseint.py
Ingrese cantidad de datos:
5
Ingrese el dato  1 :
2
Ingrese el dato  2 :
-3
Ingrese el dato  3 :
7
Ingrese el dato  4 :
8
Ingrese el dato  5 :
9
El minimo de los  5  numeros es:  -3
```

En este caso me genero el error TypeError: 'float' object cannot be interpreted as an integer, lo solucione cambiado la transformacón de tipo a entero (int).

Asignatura	Datos del alumno	Fecha
Métodos Numéricos	Apellidos: Jimenez Acosta	
	Nombre: Ronaldo	

Conclusiones:

Ventajas de utilizar diagramas de flujo:

Desde mi punto de vista los diagramas de flujo nos ayudan a tener una representación visual clara de la lógica del programa, lo que facilita la comprensión de lo que se quiere desarrollar. Por otro lado son muy útiles al momento de documentar un proyecto o explicar ideas y procesos a otros miembros involucrados en el mismo.

Desventajas de utilizar diagramas de flujo:

Para mi va bien para programas sencillos, por que si se desea representar un algoritmo demasiado grande y complejo es poco practico.

Importancia de las pruebas de escritorio::

Nos ayuda a verificar que la lógica del programa se ajusta a los requisitos y se implementa correctamente, también a identificar donde estamos fallando (errores).

Para concluir quiero comentar que al analizar la calidad del código con pylint me arrojaba un resultado de 0.0/10, es decir el código era funcional pero con problemas como errores de estilo, convenciones de codificación no seguidas y posibles errores lógicos.

```
JIMCOSTDEV@DESKTOP-GI2F5PJ MINGW64 ~/Desktop/Cursos/Unir/III/Metodos_Numericos
$ cd lab_1/

JIMCOSTDEV@DESKTOP-GI2F5PJ MINGW64 ~/Desktop/Cursos/Unir/III/Metodos_Numericos/lab_1
$ pylint sol_cuadratica_pseint.py
************* Module sol_cuadratica_pseint
sol_cuadratica_pseint.py:8:0: W0311: Bad indentation. Found 1 spaces, expected 4 (bad-indentation)
sol_cuadratica_pseint.py:9:0: W0311: Bad indentation. Found 1 spaces, expected 4 (bad-indentation)
sol_cuadratica_pseint.py:10:0: W0311: Bad indentation. Found 1 spaces, expected 4 (bad-indentation)
sol_cuadratica_pseint.py:11:0: W0311: Bad indentation. Found 1 spaces, expected 4 (bad-indentation)
sol_cuadratica_pseint.py:12:0: W0311: Bad indentation. Found 1 spaces, expected 4 (bad-indentation)
sol_cuadratica_pseint.py:13:0: W0311: Bad indentation. Found 2 spaces, expected 8 (bad-indentation)
sol_cuadratica_pseint.py:14:0: W0311: Bad indentation. Found 1 spaces, expected 4 (bad-indentation)
sol_cuadratica_pseint.py:15:0: W0311: Bad indentation. Found 2 spaces, expected 8 (bad-indentation)
sol_cuadratica_pseint.py:16:0: W0311: Bad indentation. Found 2 spaces, expected 8 (bad-indentation)
sol_cuadratica_pseint.py:17:0: W0311: Bad indentation. Found 2 spaces, expected 8 (bad-indentation)
sol_cuadratica_pseint.py:18:0: W0311: Bad indentation. Found 2 spaces, expected 8 (bad-indentation)
sol_cuadratica_pseint.py:19:0: W0311: Bad indentation. Found 2 spaces, expected 8 (bad-indentation)
sol_cuadratica_pseint.py:20:0: C0305: Trailing newlines (trailing-newlines)
sol_cuadratica_pseint.py:1:0: C0114: Missing module docstring (missing-module-docstring)

-----------------------------------
Your code has been rated at 0.00/10
```