

## Practica 3

**Jose Antonio Vilcanqui Chambi**

**2024-119017**

### Ejercicio 1:

Pseudocodigo:

INICIO

Definir función ej1(x, y):

    retornar [  $x^2 + y^2 - 4$  ,  $\exp(x) + y - 1$  ]

Usar método numérico fsolve para resolver ej1(x, y) con punto inicial (0.5, 0.5)

Guardar la solución en sol1

Mostrar en pantalla: "Ejercicio 1: ", sol1

Definir un rango de valores para x entre -3 y 3 con 400 puntos

Definir un rango de valores para y entre -3 y 3 con 400 puntos

Crear una malla 2D (X, Y) con los rangos de x e y

Graficar la curva de la primera ecuación:  $x^2 + y^2 - 4 = 0$  (en rojo)

Graficar la curva de la segunda ecuación:  $\exp(x) + y - 1 = 0$  (en azul)

Marcar el punto de intersección (sol1) con un punto verde

Agregar título "Ejercicio 1"

Etiquetar ejes x y y

Activar la grilla

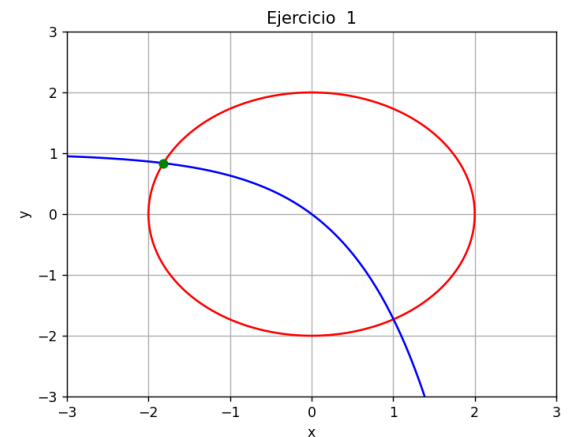
Mostrar el gráfico

FIN

```

EJERCICIO1 X
LAB4 > EJERCICIO1 > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.optimize import fsolve
4
5 def ej1(vars):
6     x, y = vars
7     return [x**2 + y**2 - 4, np.exp(x) + y - 1]
8
9 #Resolver
10 sol1 = fsolve(ej1, [0.5, 0.5])
11 print("Ejercicio 1:", sol1)
12
13 # Graficar
14 x = np.linspace(-3, 3, 400)
15 y = np.linspace(-3, 3, 400)
16 X, Y = np.meshgrid(x,y)
17
18 plt.contour(X,Y, X**2 + Y**2 - 4, [0], colors = 'r')
19 plt.contour(X, Y, np.exp(X)+Y -1,[0], colors='b')
20 plt.plot(sol1[0], sol1[1], 'go')
21 plt.title("Ejercicio 1")
22 plt.xlabel("x"); plt.ylabel("y")
23 plt.grid(True); plt.show()
24
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Store o deshabilitar este acceso directo desde Configurac
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Lc
Traceback (most recent call last):
  File "C:\Users\usuario\Desktop\MODELADO\LAB4\EJERCICIO1", line 3,
    from scipy.optimize import fsolve
ModuleNotFoundError: No module named 'scipy'
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Lc
Ejercicio 1: [-1.81626407  0.8373678 ]

```



## Ejercicio 2:

Pseudocodigo:

INICIO

Definir función ej2(x, y):

retornar [  $\sin(x) + y^2 - 1$  ,  $x^2 + y - 2$  ]

Usar fsolve para resolver ej2(x, y) con punto inicial (1, 1)

Guardar resultado en sol2

Mostrar en pantalla: "Ejercicio 2:", sol2

Definir un rango de valores de x entre -3 y 3 con 400 puntos

Definir un rango de valores de y entre -3 y 3 con 400 puntos

Construir la malla 2D (X, Y) con esos rangos

Graficar la curva  $\sin(x) + y^2 - 1 = 0$  en color rojo

Graficar la curva  $x^2 + y - 2 = 0$  en color azul

Marcar la solución encontrada sol2 con un punto verde

Agregar título "Ejercicio 2"

Etiquetar ejes x y y

Activar la grilla

Mostrar el gráfico

FIN

```

LAB4 > EJERCICIO2 > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.optimize import fsolve
4
5 #EJERCICIO 2
6 def ej2(vars):
7     x, y = vars
8     return [np.sin(x) + y**2 - 1, x**2 + y - 2]
9
10 # Resolver
11 sol2 = fsolve(ej2, [1, 1])
12 print("Ejercicio 2:", sol2)
13
14 x = np.linspace(-3, 3, 400)
15 y = np.linspace(-3, 3, 400)
16 X, Y = np.meshgrid(x, y)
17
18 # Graficar
19 plt.contour(X, Y, np.sin(X) + Y**2 - 1, [0], colors='r')
20 plt.contour(X, Y, X**2 + Y - 2, [0], colors='b')
21 plt.plot(sol2[0], sol2[1], 'go')
22 plt.title("Ejercicio 2")
23 plt.xlabel("x"); plt.ylabel("y")
24 plt.grid(True); plt.show()

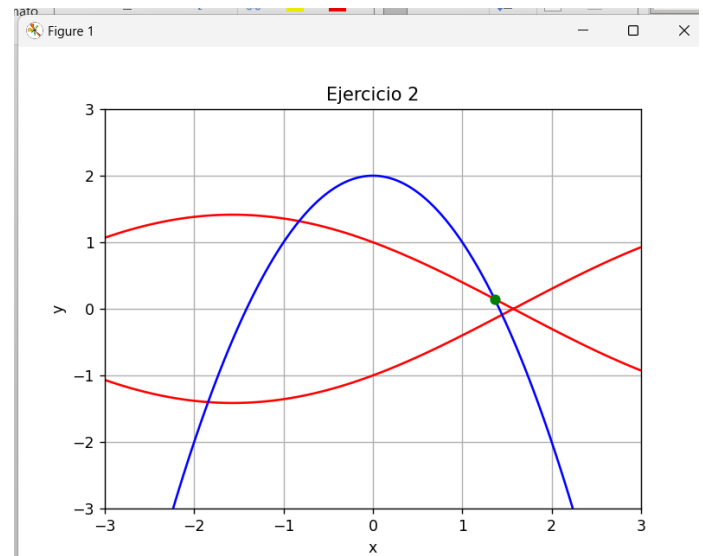
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

Ejercicio 2: [1.36080308 0.14821497]
Traceback (most recent call last):
  File "c:\Users\usuario\Desktop\MODELADO\LAB4\EJERCICIO2", line 15, in <module>
    plt.contour(X, Y, np.sin(X) + Y**2 - 1, [0], colors='r')
    ^
NameError: name 'X' is not defined
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python39-6/Scripts/python.exe c:\Users\usuario\Desktop\MODELADO\LAB4\EJERCICIO2.py
Ejercicio 2: [1.36080308 0.14821497]

```



### Ejercicio 3:

Pseudocodigo:

INICIO

Definir función ej3(x, y):

    retornar [  $x^3 - y$  ,  $x + y^2 - 4$  ]

Usar fsolve para resolver ej3(x, y) con punto inicial (1, 1)

Guardar resultado en sol3

Mostrar en pantalla: "Ejercicio 3:", sol3

Definir un rango de valores de x entre -3 y 3 con 400 puntos

Definir un rango de valores de y entre -3 y 3 con 400 puntos

Construir malla 2D (X, Y) con los rangos

Graficar la curva  $x^3 - y = 0$  en color rojo

Graficar la curva  $x + y^2 - 4 = 0$  en color azul

Marcar la solución encontrada sol3 con un punto verde

Agregar título "Ejercicio 3"

Etiquetar ejes x y y

Activar la grilla

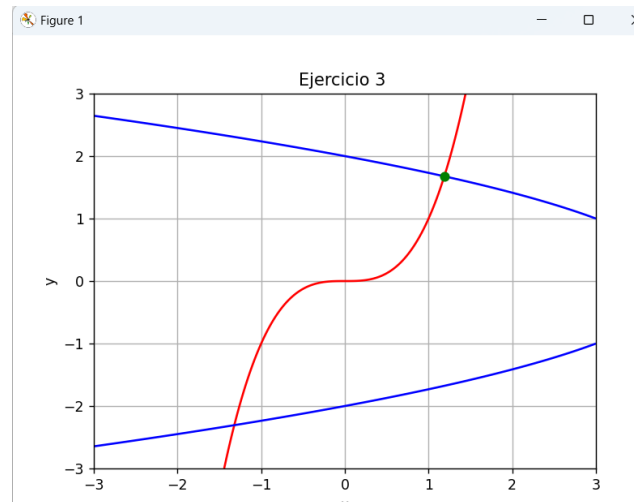
Mostrar el gráfico

FIN

```

EJERCICIO1  EJERCICIO2  EJERCICIO3 X
LAB4 > EJERCICIO3 > ...
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.optimize import fsolve
4
5  def ej3(vars):
6      x,y = vars
7      return [x**3 - y, x+y**2-4]
8
9  #Resolver
10 sol3 = fsolve(ej3, [1,1]) #fsolve es de scipy.optimize busca un
11 print("Ejercicio 3:", sol3)
12
13 #Crear malla para graficar
14 x = np.linspace(-3,3,400)
15 y = np.linspace(-3,3,400)
16 X,Y = np.meshgrid(x,y)
17
18 #Graficar
19 plt.contour(X, Y, X**3 - Y, [0], colors='r')
20 plt.contour(X,Y,X + Y**2 - 4, [0], colors='b')
21 plt.plot(sol3[0], sol3[1], 'go')#soluciones cerca a 1,1
22 plt.title("Ejercicio 3 ")
23 plt.xlabel("x"); plt.ylabel("y")
24 plt.grid(True); plt.show() #activa la grilla, líneas de referen
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
plt.contour(X, Y, X**3 - Y, [0], colors='r')
^
NameError: name 'X' is not defined
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Pro
Ejercicio 3: [1.18804969 1.67688709]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Pro
Ejercicio 3: [1.18804969 1.67688709]

```



#### Ejercicio 4:

Pseudocódigo:

INICIO

Definir función ej4(x, y):

retornar [  $\exp(x) + y - 3$  ,  $x^2 + y^2 - 5$  ]

Usar fsolve para resolver ej4(x, y) con punto inicial (1, 1)

Guardar resultado en sol4

Mostrar en pantalla: "Ejercicio 4:", sol4

Definir un rango de valores de x entre -3 y 3 con 400 puntos

Definir un rango de valores de y entre -3 y 3 con 400 puntos

Construir malla 2D (X, Y) con los rangos

Graficar la curva  $\exp(x) + y - 3 = 0$  en color rojo

Graficar la curva  $x^2 + y^2 - 5 = 0$  en color azul

Marcar la solución encontrada sol4 con un punto verde

Agregar título "Ejercicio 4"

Etiquetar ejes x y y

Activar la grilla

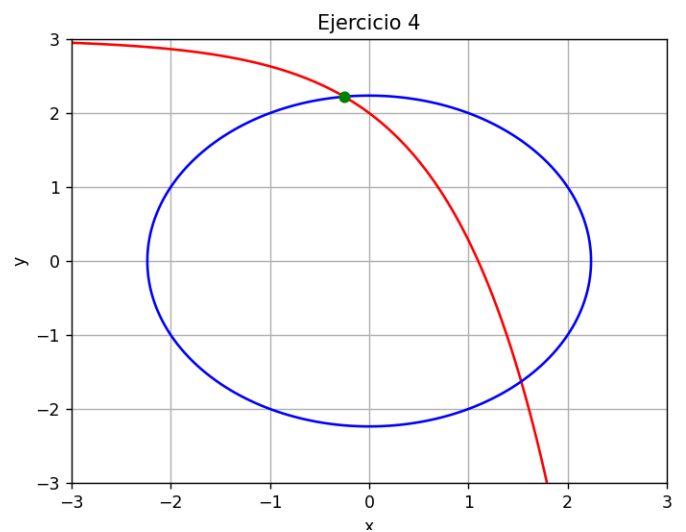
Mostrar el gráfico

FIN

```

EJERCICIO1  EJERCICIO2  EJERCICIO3  EJERCICIO4 > ...
LAB4 > EJERCICIO4 > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.optimize import fsolve
4
5 def ej4(vars):
6     x, y = vars
7     return [np.exp(x) + y - 3, x**2 + y**2 - 5]
8
9 #resolver
10 sol4 = fsolve(ej4, [1,1])
11 print("Ejercicio 4:", sol4)
12
13 #graficar
14 x = np.linspace(-3, 3, 400)
15 y = np.linspace(-3, 3, 400)
16 X,Y = np.meshgrid(x,y) #guarda las coordenadas
17
18 plt.contour(X,Y, np.exp(X) + Y - 3, [0], colors='r')
19 plt.contour(X,Y, X**2 + Y**2 - 5, [0], colors='b')
20 plt.plot(sol4[0], sol4[1], 'go')
21 plt.title("Ejercicio 4")
22 plt.xlabel("x"); plt.ylabel("y")
23 plt.grid(True); plt.show()
24
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Traceback (most recent call last):
  File "c:\Users\usuario\Desktop\MODELADO\LAB4\EJERCICIO4", line
  plt.contour(X,Y, np.exp(X) + Y - 3, [0], colors='r')
AttributeError: module 'matplotlib.pyplot' has no attribute 'count
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/
Ejercicio 4: [-0.25095313  2.22194116]

```



### Ejemplo 5:

Pseudocódigo:

INICIO

Definir función ej5(x, y):

retornar [  $\log(x + 2) + y - 1$ ,  $x^2 - y - 2$  ]

Usar fsolve para resolver ej5(x, y) con punto inicial (1, 1)

Guardar resultado en sol5

Mostrar en pantalla: "Ejercicio 5:", sol5

Definir un rango de valores de x entre -1.9 y 3 con 400 puntos

Definir un rango de valores de y entre -3 y 3 con 400 puntos

Construir malla 2D (X, Y) con los rangos

Graficar la curva  $\log(x + 2) + y - 1 = 0$  en color rojo

Graficar la curva  $x^2 - y - 2 = 0$  en color azul

Marcar la solución encontrada sol5 con un punto verde

Agregar título "Ejercicio 5"

Etiquetar ejes x y y

Activar la grilla

Mostrar el gráfico

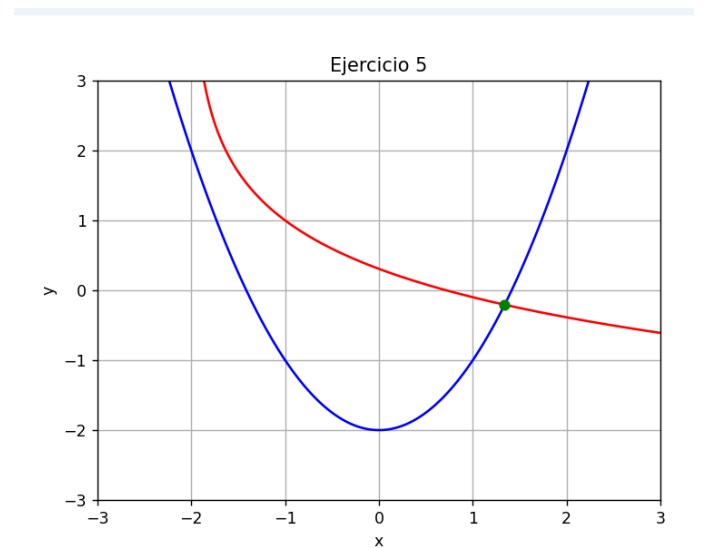
FIN

```

LAB4 > EJERCICIOS > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.optimize import fsolve
4
5 def ej5(vars):
6     x, y = vars
7     return [np.log(x+2) + y - 1, x**2 - y - 2]
8
9 #resolver
10 sol5 = fsolve(ej5, [1,1])
11 print("Ejercicio 5: ", sol5)
12
13 #graficar
14 x = np.linspace(-1.9, 3, 400)
15 y = np.linspace(-3, 3, 400)
16 X,Y = np.meshgrid(x,y)
17
18 plt.contour(X,Y,np.log(X+2) + Y - 1,[0], colors='r')
19 plt.contour(X,Y, X**2 - Y - 2, [0], colors = 'b' )
20 plt.plot(sol5[0], sol5[1], 'go')
21 plt.title("Ejercicio 5")
22 plt.xlabel("x"); plt.ylabel("y")
23 plt.grid(True); plt.show()
24
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

n <module>
sol5 = fsolve(ej5, [1,1])
      ^^^
NameError: name 'ej5' is not defined. Did you mean: 'ej2'?
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData
Ejercicio 5: [ 1.33947269 -0.20581292]
c:\Users\usuario\Desktop\MODELADO\LAB4\EJERCICIOS:18: RuntimeWarn
plt.contour(X,Y,np.log(X+2) + Y - 1,[0], colors='r')
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData
Ejercicio 5: [ 1.33947269 -0.20581292]
PS C:\Users\usuario\Desktop\MODELADO>

```



### Ejercicio 6:

Pseudocodigo:

INICIO

Definir funciones f1, f2, f3

$$f1(x,y,z) = x^2 + y + z - 4$$

$$f2(x,y,z) = x + y^2 + z - 5$$

$$f3(x,y,z) = x + y + z^2 - 6$$

Usar método numérico para resolver el sistema (ej: fsolve)

Mostrar resultados (x, y, z)

FIN

```
1 # EJERCICIO 6 (SIN USAR FUNCION DEF)
2 import numpy as np
3 from scipy.optimize import fsolve
4 sistema = lambda v: [
5     v[0]**2 + v[1] + v[2] - 4,
6     v[1]**2 + v[2] + v[0] - 5,
7     v[2]**2 + v[0] + v[1] - 6
8 ]
9 sol = fsolve(sistema, [1, 1, 1])
10 print("Ejercicio 6 - solución aproximada (x,y,z) =", sol)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\usuario\Desktop\MODELADO\Ejercicio6.py  
Ejercicio 6 - solución aproximada (x,y,z) = [0.73442414 1.52710987 1.93351131]  
PS C:\Users\usuario\Desktop\MODELADO>

### Ejercicio 7:

Pseudocodigo:

INICIO

Definir funciones f1, f2, f3

$$f1(x,y,z) = \exp(x) + y^2 + z - 7$$

$$f2(x,y,z) = x + \exp(y) + z^2 - 8$$

$$f3(x,y,z) = x^2 + y + \exp(z) - 9$$

Usar método numérico para resolver el sistema (ej: fsolve)

Mostrar resultados (x, y, z)

FIN

```
LAB4 > EJERCICIO7 > ...
1 # EJERCICIO 7 (SIN USAR FUNCION DEF)
2 import numpy as np
3 from scipy.optimize import fsolve
4
5 sistema = lambda v: [
6     np.exp(v[0]) + v[1]**2 + v[2] - 7,
7     v[0] + np.exp(v[1]) + v[2]**2 - 8,
8     v[0]**2 + v[1] + np.exp(v[2]) - 9
9 ]
10 sol = fsolve(sistema, [1, 1, 1])
11
12 print("Ejercicio 7 - solución aproximada (x,y,z) =", sol)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\usuario\Desktop\MODELADO\Ejercicio7.py  
Ejercicio 7 - solución aproximada (x,y,z) = [1.29891043 1.23709145 1.80430383]  
PS C:\Users\usuario\Desktop\MODELADO>

### Ejercicio 8:

Pseudocódigo:

INICIO

Definir funciones f1, f2, f3

$$f1(x,y,z) = \sin(x) + y + z^2 - 3$$

$$f2(x,y,z) = x + \cos(y) + z - 2$$

$$f3(x,y,z) = x^2 + y^2 + z - 4$$

Usar método numérico para resolver el sistema (ej: fsolve)

Mostrar resultados (x, y, z)

FIN

```
LAB4 > EJERCICIO 8 > ...
1  # EJERCICIO 8 (SIN USAR FUNCIÓN DEF)
2  import numpy as np
3  from scipy.optimize import fsolve
4  sistema = lambda v: [
5  np.sin(v[0]) + v[1] + v[2]**2 - 3,
6  v[0] + np.cos(v[1]) + v[2] - 2,
7  v[0]**2 + v[1]**2 + v[2] - 4
8  ]
9  sol = fsolve(sistema, [1, 1, 1])
10 print("Ejercicio 8 - solución aproximada (x,y,z) =", sol)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Py
Ejercicio 6 - solución aproximada (x,y,z) = [0.73442414 1.52710987 1.93351131]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Py
Ejercicio 7 - solución aproximada (x,y,z) = [1.29891043 1.23709145 1.80430383]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Py
Ejercicio 8 - solución aproximada (x,y,z) = [1.04053051 1.4437803 0.83279472]
PS C:\Users\usuario\Desktop\MODELADO>
```

### Ejercicio 9:

Pseudocódigo:

INICIO

Definir funciones f1, f2, f3

$$f1(x,y,z) = e^x + y + z - 4$$

$$f2(x,y,z) = x + e^y + z - 5$$

$$f3(x,y,z) = x + y + e^z - 6$$

Usar método numérico para resolver el sistema (ej: fsolve)

Mostrar resultados (x, y, z)

FIN



```
LAB4 > EJERCICIO9 > ...
1 import numpy as np
2 from scipy.optimize import fsolve
3
4 def sistema9(vars):
5     x, y, z = vars
6     return [
7         np.exp(x) + y + z - 4,
8         x + np.exp(y) + z - 5,
9         x + y + np.exp(z) - 6
10    ]
11
12 sol9 = fsolve(sistema9, [1, 1, 1])
13 print("Ejercicio 9 - solución aproximada (x,y,z) =", sol9)
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/usuario/Desktop/MODELADO/Ejercicio9.py
Ejercicio 6 - solución aproximada (x,y,z) = [0.73442414 1.52710987 1.93351131]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/usuario/Desktop/MODELADO/Ejercicio7.py
Ejercicio 7 - solución aproximada (x,y,z) = [1.29891043 1.23709145 1.80430383]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/usuario/Desktop/MODELADO/Ejercicio8.py
Ejercicio 8 - solución aproximada (x,y,z) = [1.04053051 1.4437803 0.83279472]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/usuario/Desktop/MODELADO/Ejercicio9.py
Ejercicio 9 - solución aproximada (x,y,z) = [0.27577005 1.16538609 1.51706905]
PS C:\Users\usuario\Desktop\MODELADO>
```

### **EJERCICIO 10:**

Pseudocódigo:

INICIO

Definir funciones f1, f2, f3

$$f1(x,y,z) = \ln(x+2) + y + z - 3$$

$$f2(x,y,z) = x + \ln(y+2) + z - 4$$

$$f3(x,y,z) = x + y + \ln(z+2) - 5$$

Usar método numérico para resolver el sistema (ej: fsolve)

Mostrar resultados (x, y, z)

FIN

```
LAB4 > EJERCICIO10 > ...
1 import numpy as np
2 from scipy.optimize import fsolve
3
4 def sistema10(vars):
5     x, y, z = vars
6     return [
7         np.log(x + 2) + y + z - 3,
8         x + np.log(y + 2) + z - 4,
9         x + y + np.log(z + 2) - 5
10    ]
11
12 sol10 = fsolve(sistema10, [1, 1, 1])
13 print("Ejercicio 10 - solución aproximada (x,y,z) =", sol10)
14 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python
Ejercicio 6 - solución aproximada (x,y,z) = [0.73442414 1.52710987 1.93351131]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python
Ejercicio 7 - solución aproximada (x,y,z) = [1.29891043 1.23709145 1.80430383]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python
Ejercicio 8 - solución aproximada (x,y,z) = [1.04053051 1.4437803 0.83279472]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python
Ejercicio 9 - solución aproximada (x,y,z) = [0.27577005 1.16538609 1.51706905]
PS C:\Users\usuario\Desktop\MODELADO> & C:/Users/usuario/AppData/Local/Programs/Python
Ejercicio 10 - solución aproximada (x,y,z) = [ 2.83682432 1.51864069 -0.09489906]
PS C:\Users\usuario\Desktop\MODELADO>
```

## Conclusiones:

En todos los ejercicios se aplicó el método numérico `fsolve` de SciPy para resolver sistemas de ecuaciones no lineales, apoyándonos en `numpy` para los cálculos y en `matplotlib` para la visualización de curvas e intersecciones. Se observó que el punto inicial influye directamente en la solución obtenida, lo que resalta la importancia de probar varias condiciones iniciales cuando existen múltiples soluciones. Además, fue necesario tener cuidado con los dominios de las funciones (por ejemplo, en logaritmos) y con errores comunes de codificación como variables no definidas o etiquetas incorrectas. Las gráficas de contorno y los puntos marcados permitieron verificar de manera visual las soluciones encontradas, complementando la verificación numérica al sustituir los valores obtenidos en las ecuaciones originales. En conclusión, este conjunto de ejercicios mostró cómo combinar herramientas numéricas y gráficas para analizar y resolver eficazmente sistemas no lineales.