

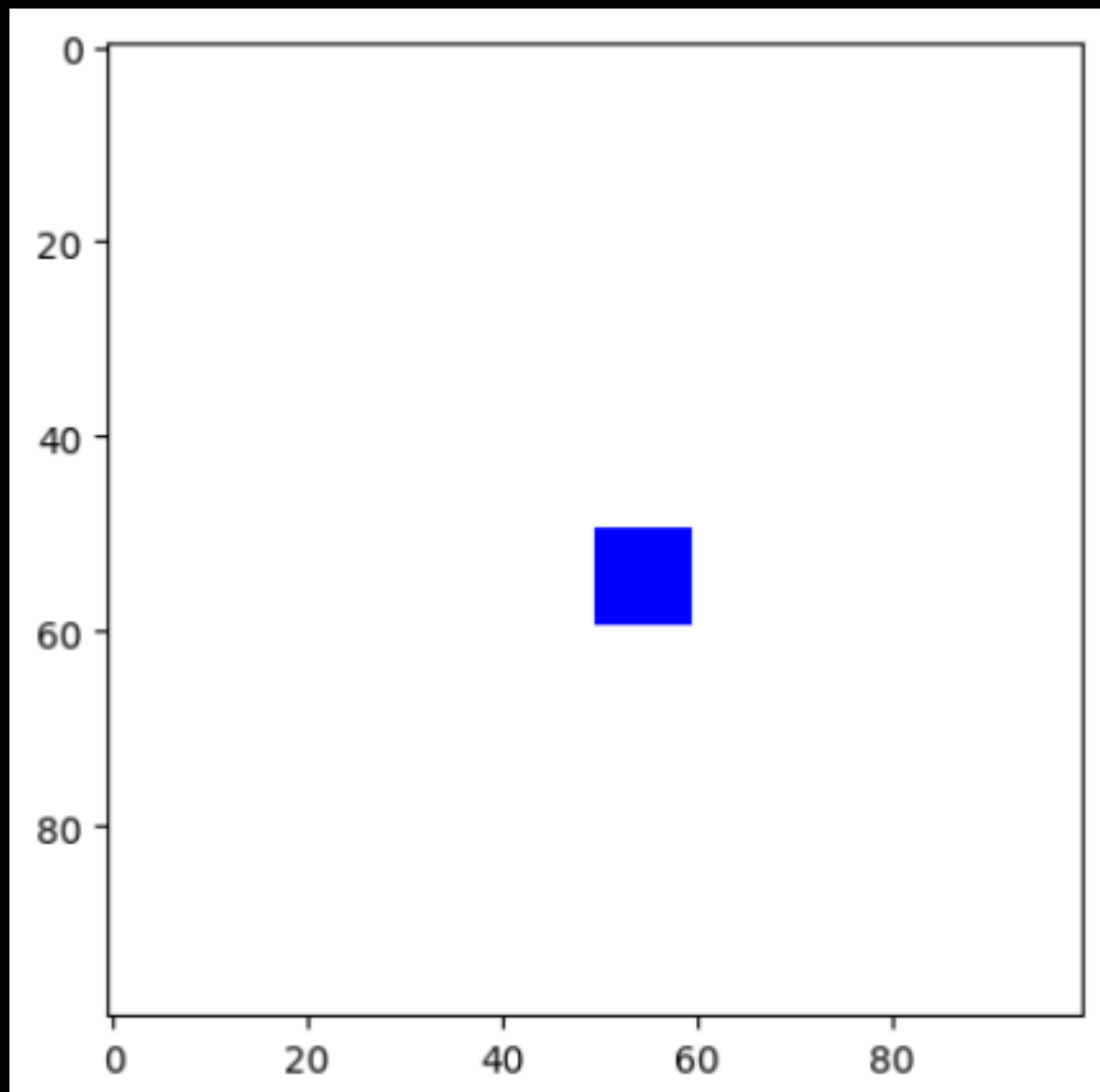
A person is seen from behind, sitting at a desk in a dimly lit room. They are looking at a large computer monitor that displays a vibrant cityscape at sunset, with the sun low on the horizon and buildings silhouetted against the orange and yellow sky. The person is wearing a dark sweater. On the desk, there is a white mug with steam rising from it, a computer mouse, and various cables. To the right of the person, there are server racks with glowing blue lights. The background is filled with a dense network of glowing, colorful lines (blue, red, yellow) that resemble data connections or fiber optics, creating a futuristic, digital atmosphere. The overall scene suggests a focus on technology, data, and digital art.

COMPUTAÇÃO GRÁFICA

Aula 3 – Formas de Imagens

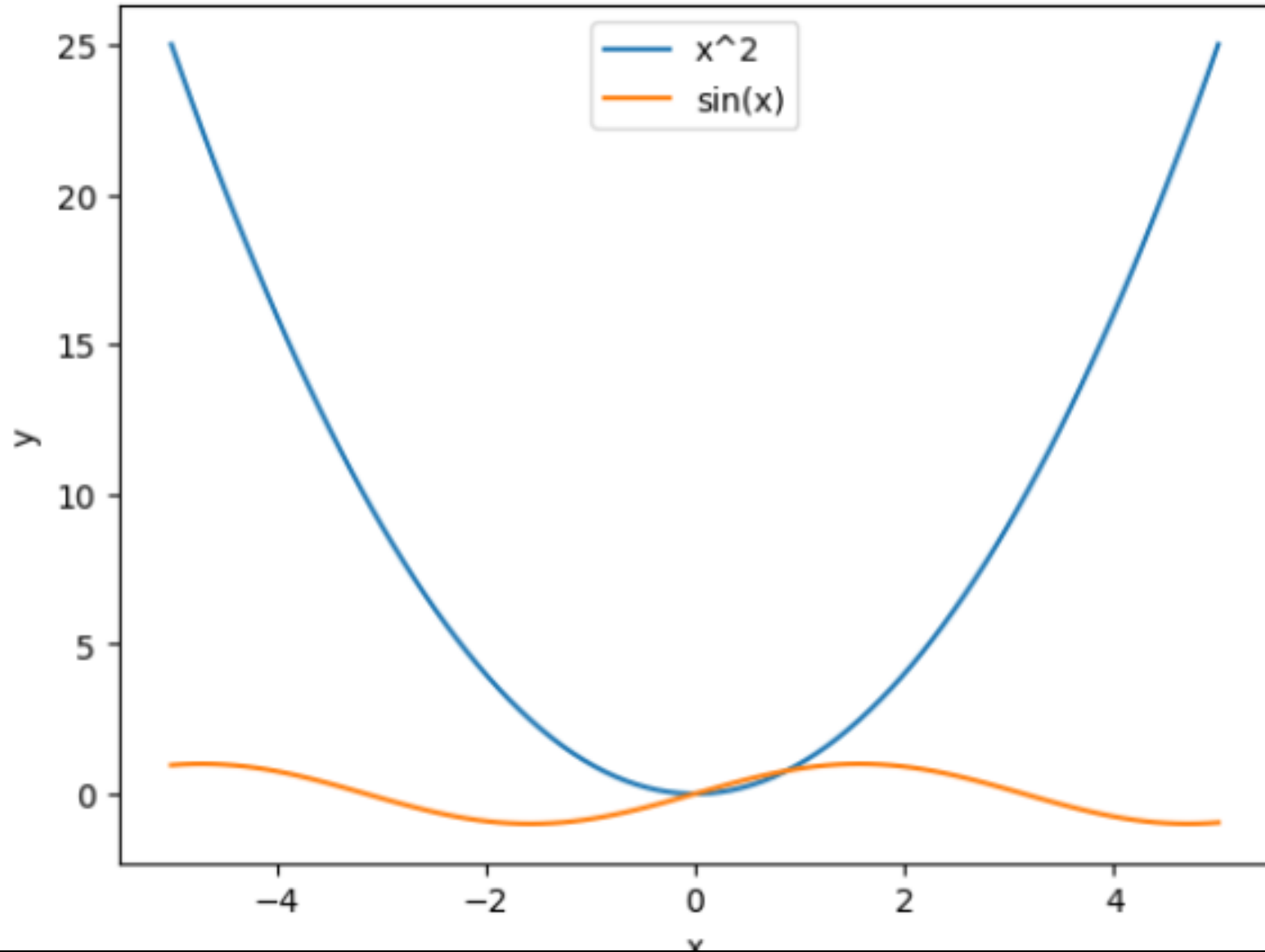
Curso de Ciência da Computação
Dr. Rodrigo Xavier de Almeida Leão
Cientista de Dados

```
1 # prompt: gerar imagem de matriz RGB toda em branco cujas cores dos pixels podem
2
3 # Criando uma matriz RGB toda branca (255, 255, 255)
4 largura = 100
5 altura = 100
6 matriz_rgb = np.full((altura, largura, 3), 255, dtype=np.uint8)
7
8 # Alterando a cor de alguns pixels (exemplo)
9 matriz_rgb[50:60, 50:60, :] = [0, 0, 255] # Define uma região azul
10
11 # Criando uma imagem a partir da matriz
12 imagem = Image.fromarray(matriz_rgb)
13
14 # Exibindo a imagem
15 plt.imshow(imagem)
16 plt.show()
17
```



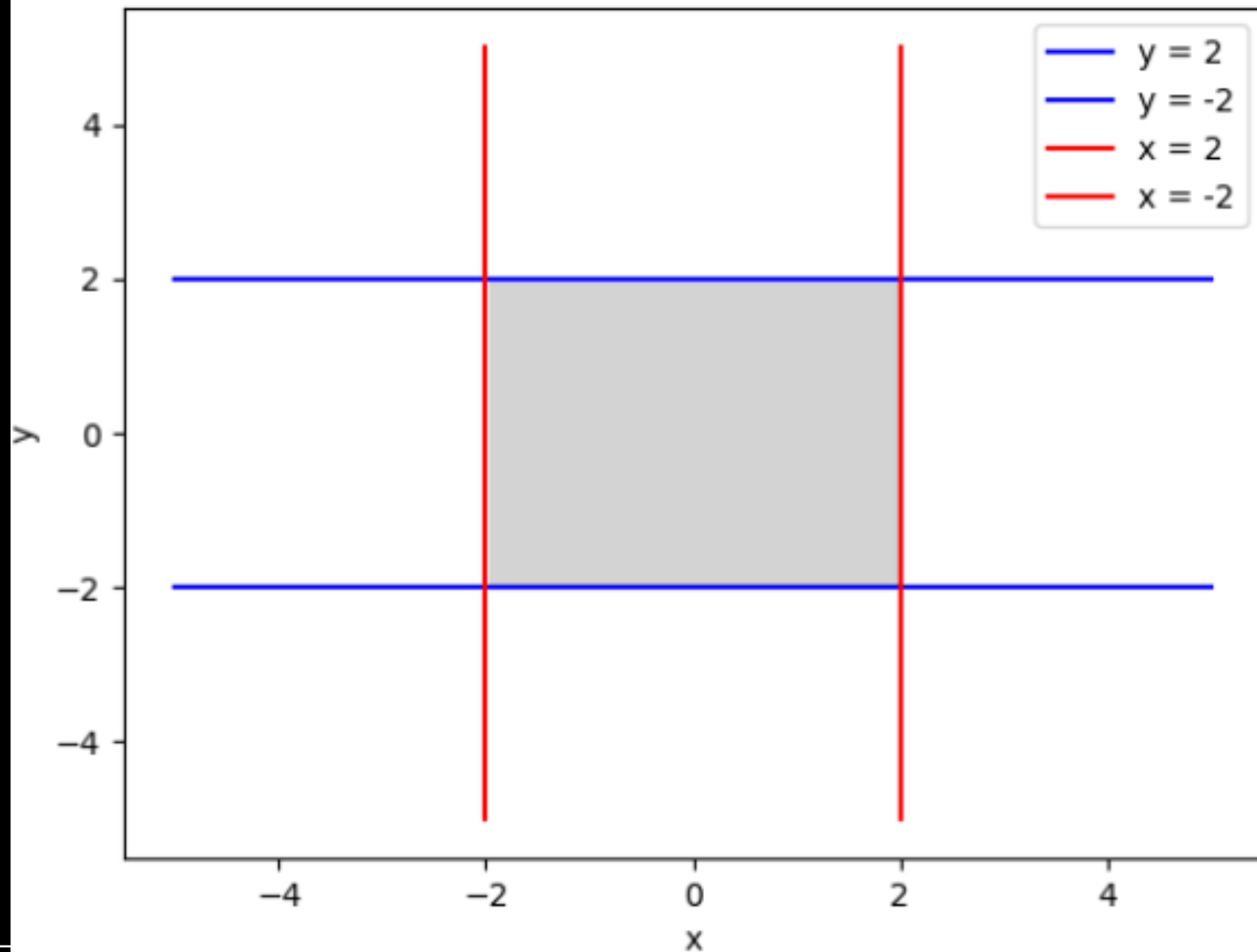
```
1 # prompt: plotar mais de uma equação no mesmo gráfico
2
3 # Dados para a primeira equação
4 x1 = np.linspace(-5, 5, 100)
5 y1 = x1**2
6
7 # Dados para a segunda equação
8 x2 = np.linspace(-5, 5, 100)
9 y2 = np.sin(x2)
10
11 # Plotando as duas equações no mesmo gráfico
12 plt.plot(x1, y1, label='x^2')
13 plt.plot(x2, y2, label='sin(x)')
14
15 # Adicionando legenda, título e rótulos dos eixos
16 plt.legend()
17 plt.title('Gráfico com Duas Equações')
18 plt.xlabel('x')
19 plt.ylabel('y')
20
21 # Exibindo o gráfico
22 plt.show()
```

Gráfico com Duas Equações



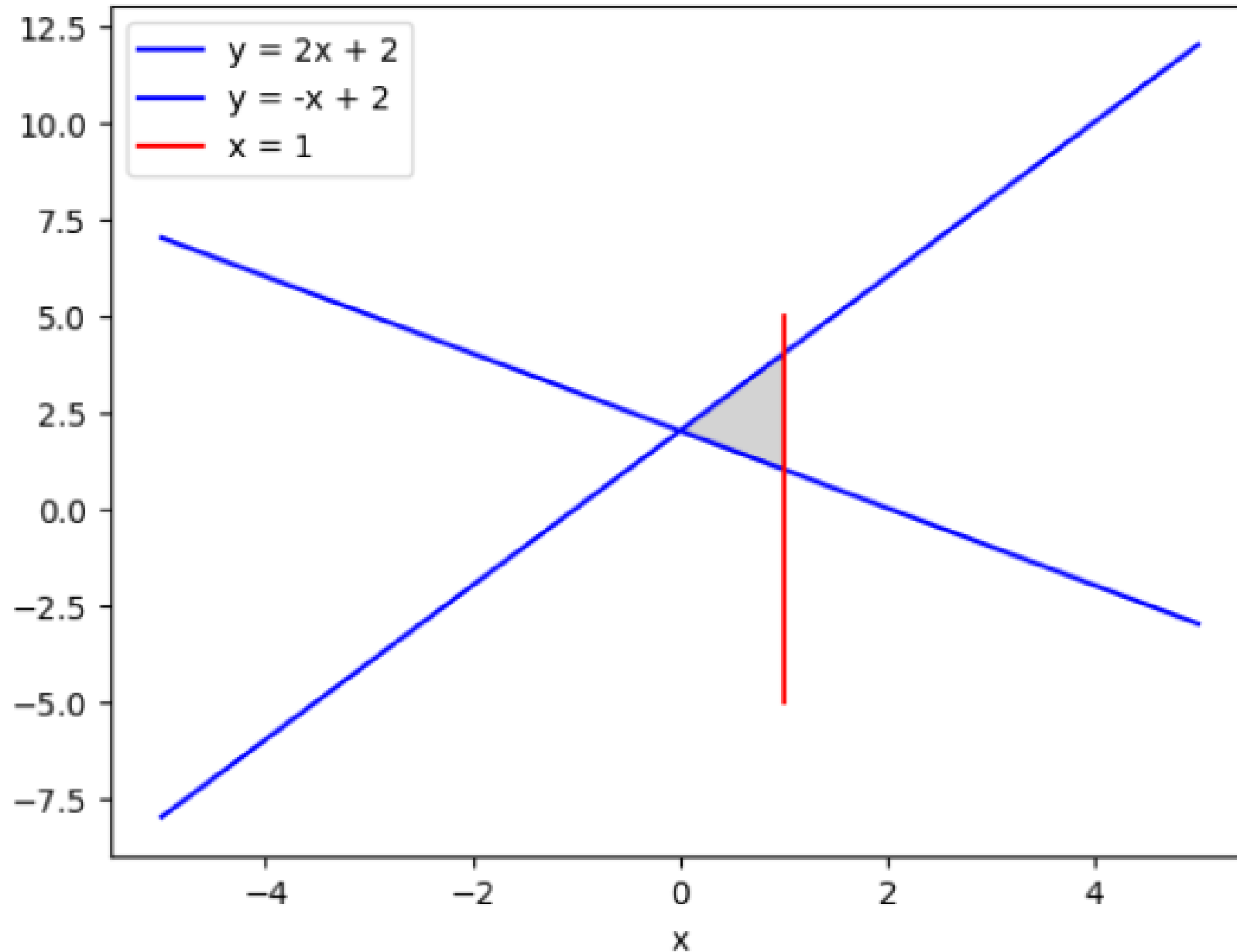
```
1 # prompt: utilizar o código anterior para gerar um quadrado a partir das equações de 4 retas e pintar
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Definindo as equações das retas
7 x = np.linspace(-5, 5, 100)
8 y1 = 2*np.ones_like(x) # Reta horizontal superior - np.ones_like(x) -> array de valor 1 do tamanho de x
9 y2 = -2*np.ones_like(x) # Reta horizontal inferior
10 x1 = 2*np.ones_like(x) # Reta vertical direita
11 x2 = -2*np.ones_like(x) # Reta vertical esquerda
12
13 # Plotando as retas
14 plt.plot(x, y1, 'b-', label='y = 2')
15 plt.plot(x, y2, 'b-', label='y = -2')
16 plt.plot(x1, x, 'r-', label='x = 2')
17 plt.plot(x2, x, 'r-', label='x = -2')
18
19 # Preenchendo o quadrado
20 plt.fill_between(x, y2, y1, where=(x >= -2) & (x <= 2), color='lightgray')
21
22 # Adicionando legenda, título e rótulos dos eixos
23 plt.legend()
24 plt.title('Quadrado Formado por 4 Retas')
25 plt.xlabel('x')
26 plt.ylabel('y')
27
28 # Exibindo o gráfico
29 plt.show()
```

Quadrado Formado por 4 Retas



```
1 # prompt: utilizar o código anterior para gerar um triângulo a partir da das equações de 3 retas
2
3 # Definindo as equações das retas
4 x = np.linspace(-5, 5, 100)
5 y1 = 2*x + 2 # Reta 1
6 y2 = -x + 2 # Reta 2
7 x1 = 1*np.ones_like(x) # Reta vertical
8
9 # Plotando as retas
10 plt.plot(x, y1, 'b-', label='y = 2x + 2')
11 plt.plot(x, y2, 'b-', label='y = -x + 2')
12 plt.plot(x1, x, 'r-', label='x = 1')
13
14 # Preenchendo o triângulo
15 plt.fill_between(x, y1, y2, where=(x >= 0) & (x <= 1), color='lightgray')
16
17 # Adicionando legenda, título e rótulos dos eixos
18 plt.legend()
19 plt.title('Triângulo Formado por 3 Retas')
20 plt.xlabel('x')
21 plt.ylabel('y')
22
23 # Exibindo o gráfico
24 plt.show()
```

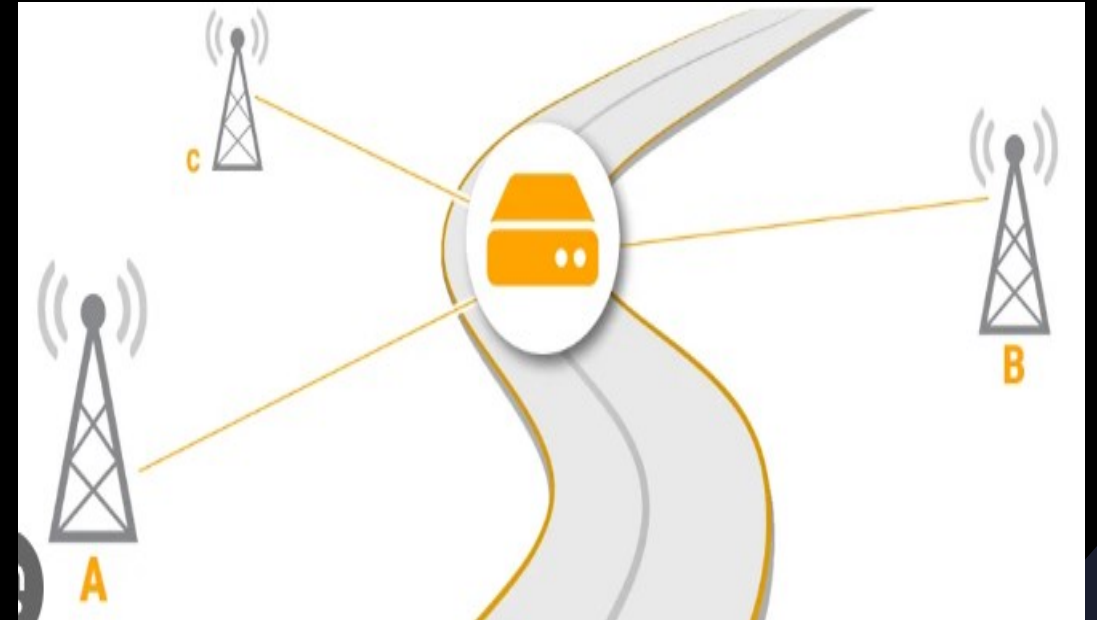
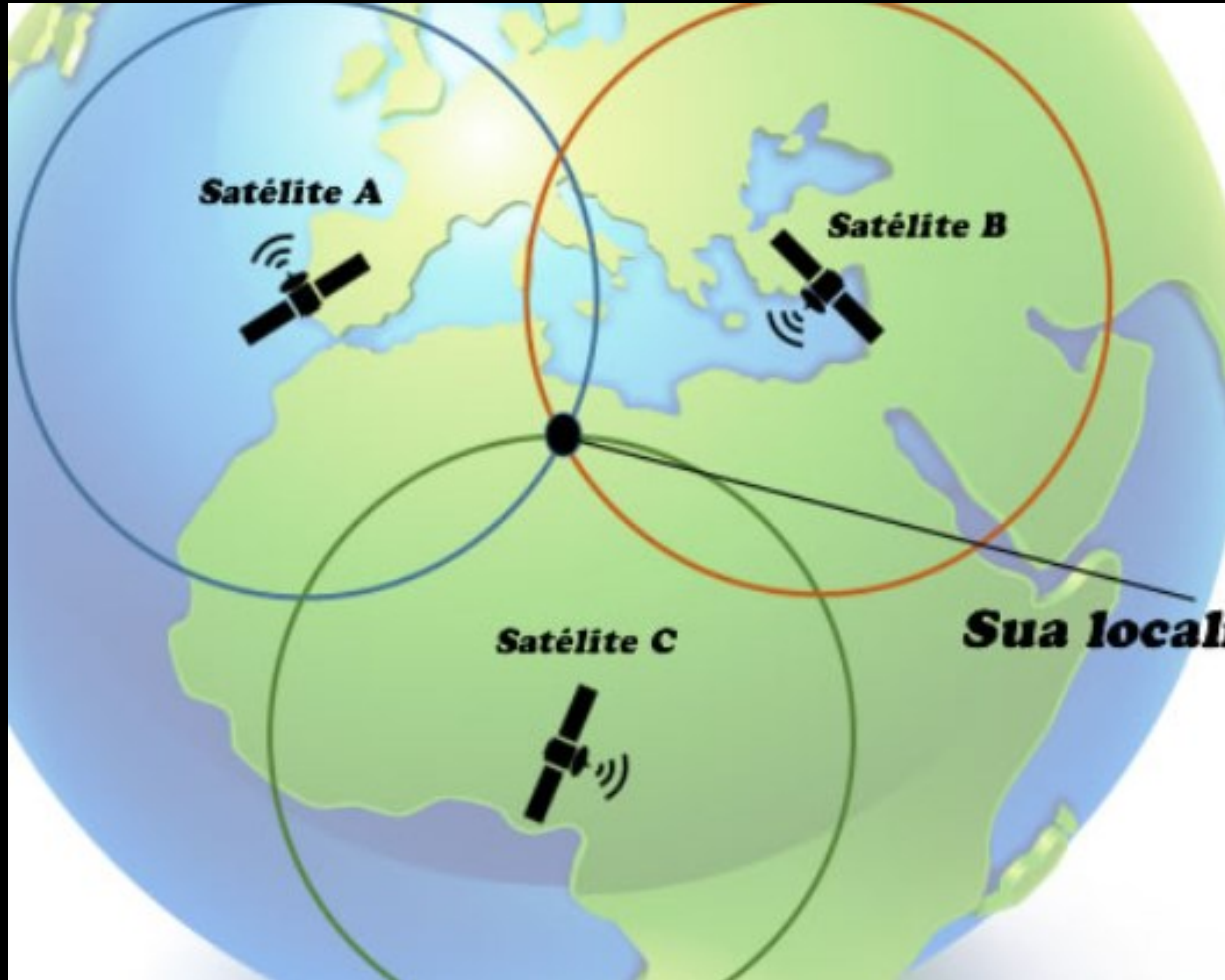

Triângulo Formado por 3 Retas



```
1 # prompt: carregar uma figura do drive e plotar um círculo sobre ela
2
3 from google.colab import drive
4 drive.mount('/content/drive')
5
6 # Import the Image module from PIL
7 from PIL import Image
8
9 # Carregando a imagem do Drive (substitua pelo caminho correto)
10 img_path = '"/content/drive/My Drive/Colab Notebooks/DATASETS/mapa_Guarapari.png"'
11 img = Image.open(img_path)
12
13 # Plotando a imagem
14 plt.imshow(img)
15
16 # Desenhando um círculo
17 circle = plt.Circle((100, 100), 50, color='red', fill=False) # (x, y), raio
18 plt.gca().add_patch(circle)
19
20 plt.show()
```



TRIÂNGULAÇÃO DE SINAL

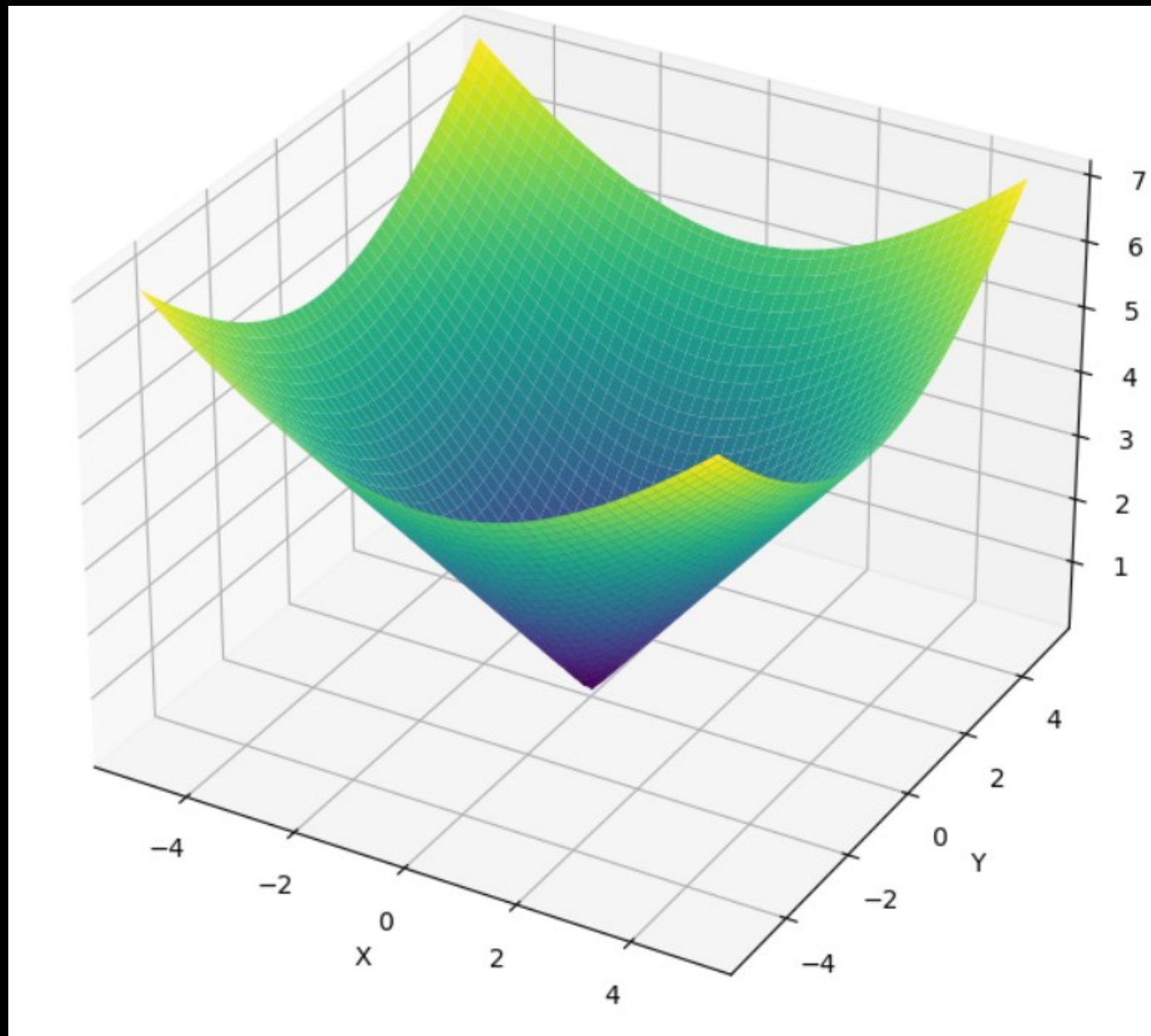


ATIVIDADE – TRIÂNGULAÇÃO

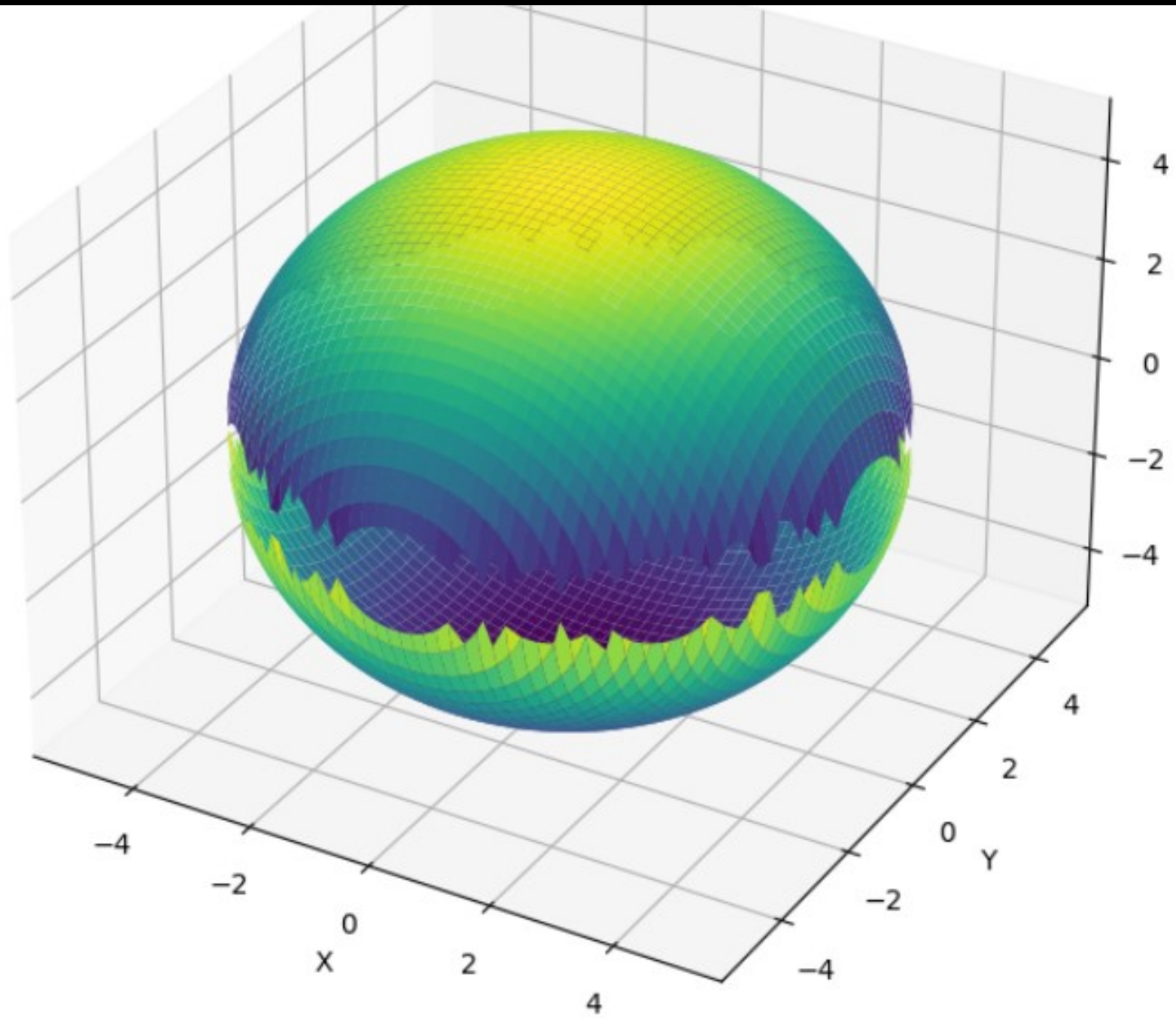
Sabendo que um sinal de celular foi captado por 3 antenas nas posições $(150, 100)$ $(200, 600)$ e $(550, 550)$ ao mesmo tempo, encontre no mapa a região de onde partiu o sinal.



```
3 # Criando figura e eixos 3D
4 fig = plt.figure(figsize=(10, 8))
5 ax = fig.add_subplot(111, projection='3d')
6
7 # Dados para o cone
8 raio = 5 # Raio da base do cone
9 x = np.linspace(-raio, raio, 100)
10 y = np.linspace(-raio, raio, 100)
11 X, Y = np.meshgrid(x, y)
12 Z = np.sqrt(X**2 + Y**2) # Equação do cone
13
14 # Plotando a superfície do cone
15 ax.plot_surface(X, Y, Z, cmap='viridis')
16
17 # Adicionando rótulos dos eixos
18 ax.set_xlabel('X')
19 ax.set_ylabel('Y')
20 ax.set_zlabel('Z')
21
22 # Exibindo o gráfico
23 plt.show()
```




```
3 # Criando figura e eixos 3D
4 fig = plt.figure(figsize=(10, 8))
5 ax = fig.add_subplot(111, projection='3d')
6
7 # Dados para a esfera
8 raio = 5 # Raio da esfera
9 x = np.linspace(-raio, raio, 100)
10 y = np.linspace(-raio, raio, 100)
11 X, Y = np.meshgrid(x, y)
12 Z = np.sqrt(raio**2 - X**2 - Y**2) # Equação da esfera
13
14 # Plotando a superfície superior da esfera
15 ax.plot_surface(X, Y, Z, cmap='viridis')
16
17 # Plotando a superfície inferior da esfera
18 Z = -np.sqrt(raio**2 - X**2 - Y**2)
19 ax.plot_surface(X, Y, Z, cmap='viridis')
20
21 # Adicionando rótulos dos eixos
22 ax.set_xlabel('X')
23 ax.set_ylabel('Y')
24 ax.set_zlabel('Z')
25
26 # Exibindo o gráfico
27 plt.show()
```



```
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from mpl_toolkits.mplot3d import Axes3D
6
7 # Criando figura e eixos 3D
8 fig = plt.figure(figsize=(10, 8))
9 ax = fig.add_subplot(111, projection='3d')
10
11 # Dados para a esfera
12 u = np.linspace(0, 2 * np.pi, 100)
13 v = np.linspace(0, np.pi, 100)
14 x = 10 * np.outer(np.cos(u), np.sin(v))
15 y = 10 * np.outer(np.sin(u), np.sin(v))
16 z = 10 * np.outer(np.ones(np.size(u)), np.cos(v))
17
18 # Plotando a esfera
19 ax.plot_surface(x, y, z, cmap='viridis')
20
21 # Adicionando rótulos dos eixos
22 ax.set_xlabel('X')
23 ax.set_ylabel('Y')
24 ax.set_zlabel('Z')
25
26 # Exibindo o gráfico
27 plt.show()
```

