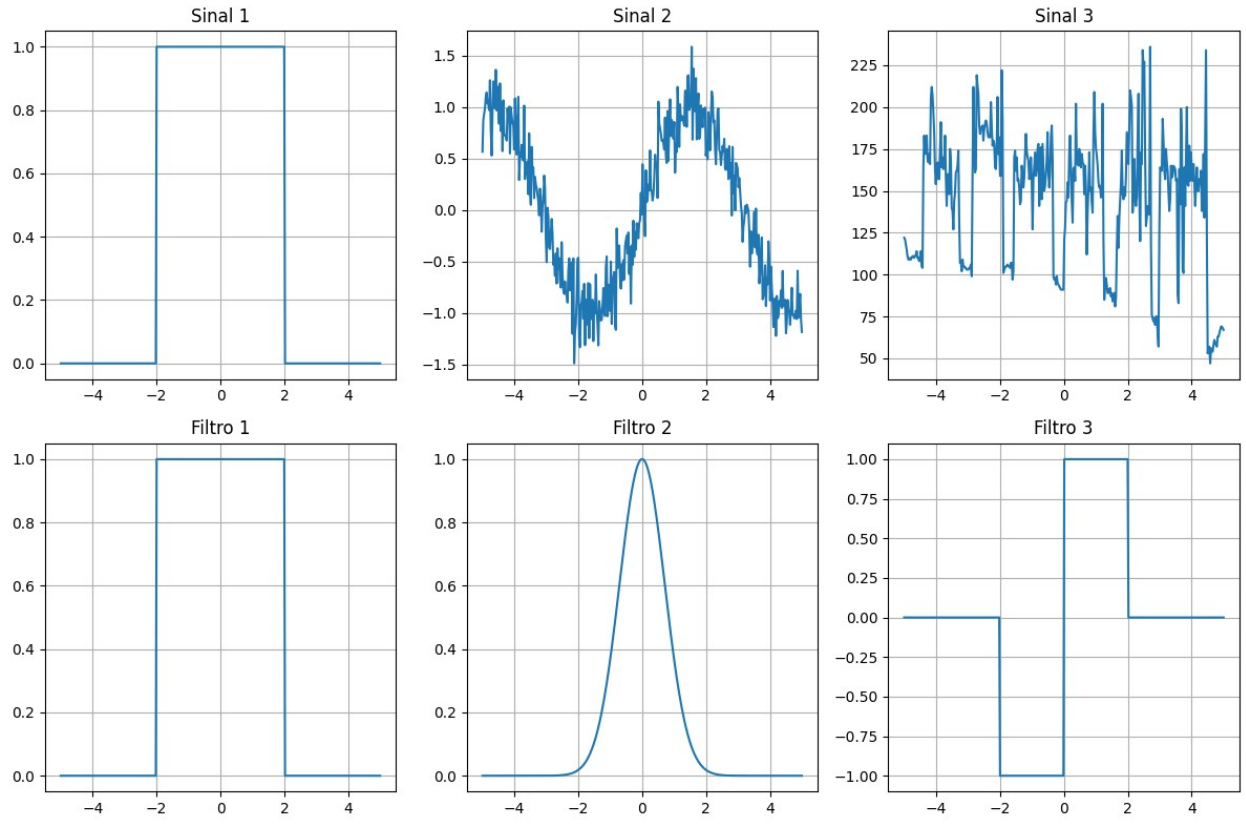
Sabrina Araújo nUSP 12566182

Exercício 4: Convolução

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import data
from numpy import random
import cv2

# Sinais de entrada

# g(t) = 1, a <= t < b; 0 caso contrário, a,b reais, a<b (janela
quadrada)
def sinal1(t):
    a = -2
    b = 2
    g = np.where((t >= a) & (t < b), 1, 0)
    return g

# g(t) = sin(t) + GaussianNoise(media, desvioPadrao) (seno com ruído
gaussiano)
def sinal2(t):
    media = 0
    desvio_padrao = 0.2
    g = np.sin(t) + random.normal(media, desvio_padrao, len(t))
    return g

# g(t) = linha de uma imagem do Scikit Image, como feito no 1o
exercício do curso.
def sinal3(t):
    image = data.coins()
    linha = 50
    g = image[linha, :]
    return g

# Filtros

# h(t) = 1 caso a <= t < b; 0 caso contrário, a,b reais, a<b (janela
quadrada)
def filtro1(t):
    a = -2
    b = 2
    h = np.where((a <= t) & (t < b), 1, 0)
    return h

# h(t) = exp(-t**2) (gaussiana)
def filtro2(t):
    h = np.exp(-t**2)
    return h
```

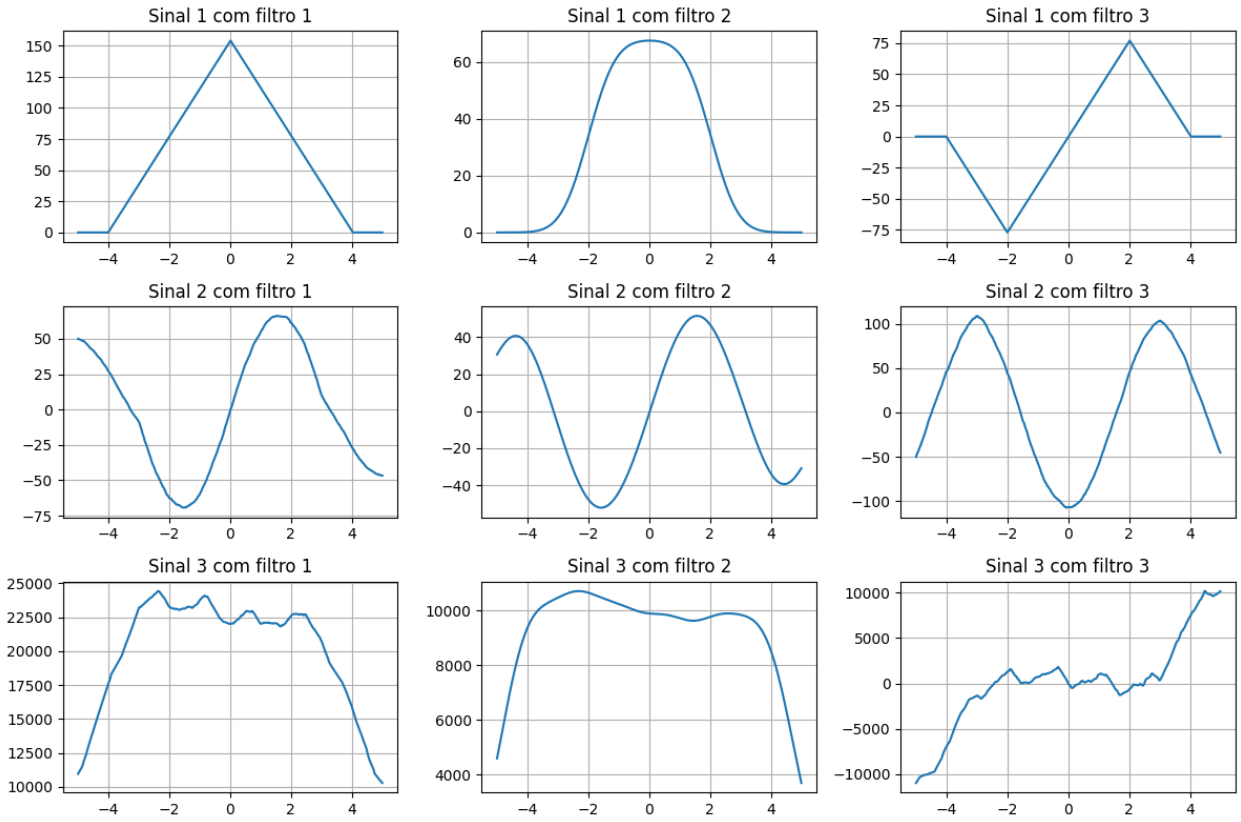
```
# h(t) = -1 caso -a <= t < 0; -1, 1 caso 0 <= t < a; 0 caso contrário,
a real (filtro de diferenciação)
def filtro3(t):
    a = 2
    h = np.where((-a <= t) & (t < 0), -1, np.where((0 <= t) & (t < a),
1, 0))
    return h

t = np.linspace(-5, 5, 384)

sinal_1 = sinal1(t)
sinal_2 = sinal2(t)
sinal_3 = sinal3(t)

filtro_1 = filtro1(t)
filtro_2 = filtro2(t)
filtro_3 = filtro3(t)

# Plot dos sinais e filtros
plt.figure(figsize=(12, 8))

plt.subplot(2, 3, 1)
plt.plot(t, sinal_1)
plt.title('Sinal 1')
plt.grid(True)

plt.subplot(2, 3, 2)
plt.plot(t, sinal_2)
plt.title('Sinal 2')
plt.grid(True)

plt.subplot(2, 3, 3)
plt.plot(t, sinal_3)
plt.title('Sinal 3')
plt.grid(True)

plt.subplot(2, 3, 4)
plt.plot(t, filtro_1)
plt.title('Filtro 1')
plt.grid(True)

plt.subplot(2, 3, 5)
plt.plot(t, filtro_2)
plt.title('Filtro 2')
plt.grid(True)

plt.subplot(2, 3, 6)
plt.plot(t, filtro_3)
plt.title('Filtro 3')
```

```
plt.grid(True)

plt.tight_layout()
plt.show()
```




```
# Cálculo da convolução entre os sinais e filtros
convolucao1_filtro1 = np.convolve(sinal_1, filtro_1, mode='same')
convolucao1_filtro2 = np.convolve(sinal_1, filtro_2, mode='same')
convolucao1_filtro3 = np.convolve(sinal_1, filtro_3, mode='same')

convolucao2_filtro1 = np.convolve(sinal_2, filtro_1, mode='same')
convolucao2_filtro2 = np.convolve(sinal_2, filtro_2, mode='same')
convolucao2_filtro3 = np.convolve(sinal_2, filtro_3, mode='same')

convolucao3_filtro1 = np.convolve(sinal_3, filtro_1, mode='same')
convolucao3_filtro2 = np.convolve(sinal_3, filtro_2, mode='same')
convolucao3_filtro3 = np.convolve(sinal_3, filtro_3, mode='same')

# Plot das convoluções
plt.figure(figsize=(12, 8))

plt.subplot(3, 3, 1)
plt.plot(t, convolucao1_filtro1)
plt.title('Sinal 1 com filtro 1')
```

```
plt.grid(True)

plt.subplot(3, 3, 2)
plt.plot(t, convolucao1_filtro2)
plt.title('Sinal 1 com filtro 2')
plt.grid(True)

plt.subplot(3, 3, 3)
plt.plot(t, convolucao1_filtro3)
plt.title('Sinal 1 com filtro 3')
plt.grid(True)

plt.subplot(3, 3, 4)
plt.plot(t, convolucao2_filtro1)
plt.title('Sinal 2 com filtro 1')
plt.grid(True)

plt.subplot(3, 3, 5)
plt.plot(t, convolucao2_filtro2)
plt.title('Sinal 2 com filtro 2')
plt.grid(True)

plt.subplot(3, 3, 6)
plt.plot(t, convolucao2_filtro3)
plt.title('Sinal 2 com filtro 3')
plt.grid(True)

plt.subplot(3, 3, 7)
plt.plot(t, convolucao3_filtro1)
plt.title('Sinal 3 com filtro 1')
plt.grid(True)

plt.subplot(3, 3, 8)
plt.plot(t, convolucao3_filtro2)
plt.title('Sinal 3 com filtro 2')
plt.grid(True)

plt.subplot(3, 3, 9)
plt.plot(t, convolucao3_filtro3)
plt.title('Sinal 3 com filtro 3')
plt.grid(True)

plt.tight_layout()
plt.show()
```


Sinal 1 com filtro 1
Sinal 1 com filtro 2
Sinal 1 com filtro 3
Sinal 2 com filtro 1
Sinal 2 com filtro 2
Sinal 2 com filtro 3
Sinal 3 com filtro 1
Sinal 3 com filtro 2
Sinal 3 com filtro 3