```
import numpy as np
import sounddevice as sd
import soundfile as sf
import matplotlib.pyplot as plt

from scipy.signal import lfilter

#Noise function
def noiseAudio(voice, a1, a2):
    a1 = 0.01
    a2 = 0.01
    n = np.arange(len(voice)) #Time axis
    noise = a1 * np.cos(0.76 * np.pi * n) + a2 * np.cos(0.8 * np.pi * n)
    noisyVoice = noise + voice
    return noisyVoice #Return audio with noise

# Set-up
recording_time = 5 # seconds
sample_rate = 44100 # sample rate(Hz)
std_audio = "Audio padrao"
noisy_audio = "Audio com Ruído"
filtered_audio = "Audio filtrado"

# Audio recording
print("Gravando..")
voice= sd.rec(int(recording_time * sample_rate), samplerate=sample_rate, channels=1)
sd.wait()

# Noisy audio
noisyVoice = noiseAudio(voice.flatten(), 0.7, 0.7)

# Applying filter
def filter(sign, coefs):
    filtered_sign = lfilter(coefs, 1, sign)
    return filtered_sign

# Filter settings
N = 37 #Filter lenght
m2 = 36/2
wc = 0.57 * np.pi #Cut Off Frequency (rad)
#print(wc)

# Retangular window
nf = np.arange(N)
#print(nf)
wn = 1

# Window applied to filter transfer
hd = wn * (np.sin(wc * (nf - m2)) / (np.pi * (nf - m2)))
hd[18] = 0.57

# Noise removal filter
filtered_sign = filter(noisyVoice, hd)

# Save WAV file
file_name_wav1 = noisy_audio + ".wav"
sf.write(file_name_wav1, noisyVoice, sample_rate)

file_name_wav2 = std_audio + ".wav"
sf.write(file_name_wav2, voice.flatten(), sample_rate)

```

```python
file_name_wav3 = filtered_audio + ".wav"
sf.write(file_name_wav3, filtered_sign, sample_rate)

print("""Gravado com sucesso.
    Arquivo salvo com sucesso.""")

#  Charts plot
t = np.linspace(0, recording_time, num=len(voice))
plt.figure(figsize=(12, 10))

# Original audio file
plt.subplot(3, 1, 1)
plt.plot(t, voice.flatten(), color = "purple")
plt.title("Sinal Original")
plt.xlabel("Tempo(s)")
plt.ylabel("Amplitude")
plt.ylim(-0.10, 0.10)

# Noisy audio
plt.subplot(3, 1, 2)
plt.plot(t, noisyVoice, color = "lightgreen")
plt.title("Sinal com Ruído")
plt.xlabel("Tempo (s)")
plt.ylabel("Amplitude")

# Filtered audio
plt.subplot(3, 1, 3)
plt.plot(t, filtered_sign, color = "lightblue")
plt.title("Sinal Filtrado")
plt.xlabel("Tempo (s)")
plt.ylabel("Amplitude")
plt.ylim(-0.10, 0.10)

plt.tight_layout()
plt.savefig("plots.png")
plt.show()
```