```
clc
clear
close all

load audio_gravado.mat

x = Y;

Ts=1/FS;
fs = 1/Ts;

t= linspace(0,length(x)/fs,length(x));


[f,H] = obterEspectro(x,t)
```

```
f = 1×19841
10^3 ×
         0    0.0002    0.0004    0.0006    0.0008    0.0010    0.0012    0.0014 ···
H = 19841×1
    0.2591
    0.4800
    0.4536
    0.5079
    0.4665
    0.4180
    0.3317
    0.2657
    0.2494
    0.2758
      ⋮
```

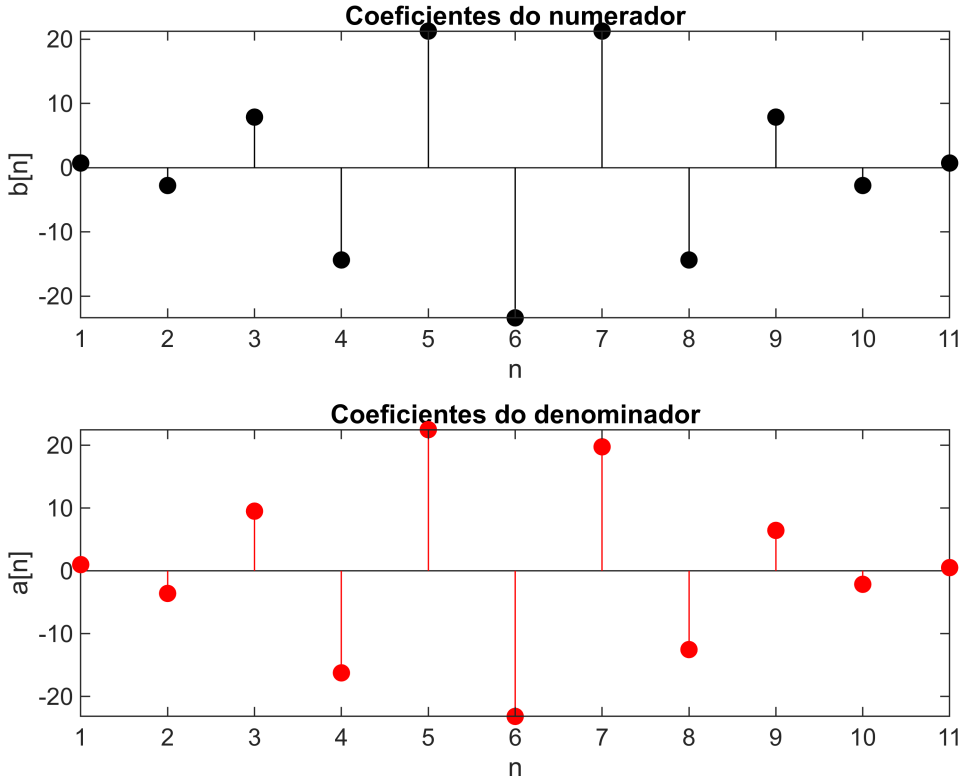
Projeto do filtro IIR

```
fN = fs/2;

%Butterworth
Rp = 1;
Rs = 40;
Wp = [1400 1600]/fN;
Ws = [1200 2000]/fN;

[n,Wn,] = buttord(Wp,Ws,Rp,Rs)
```

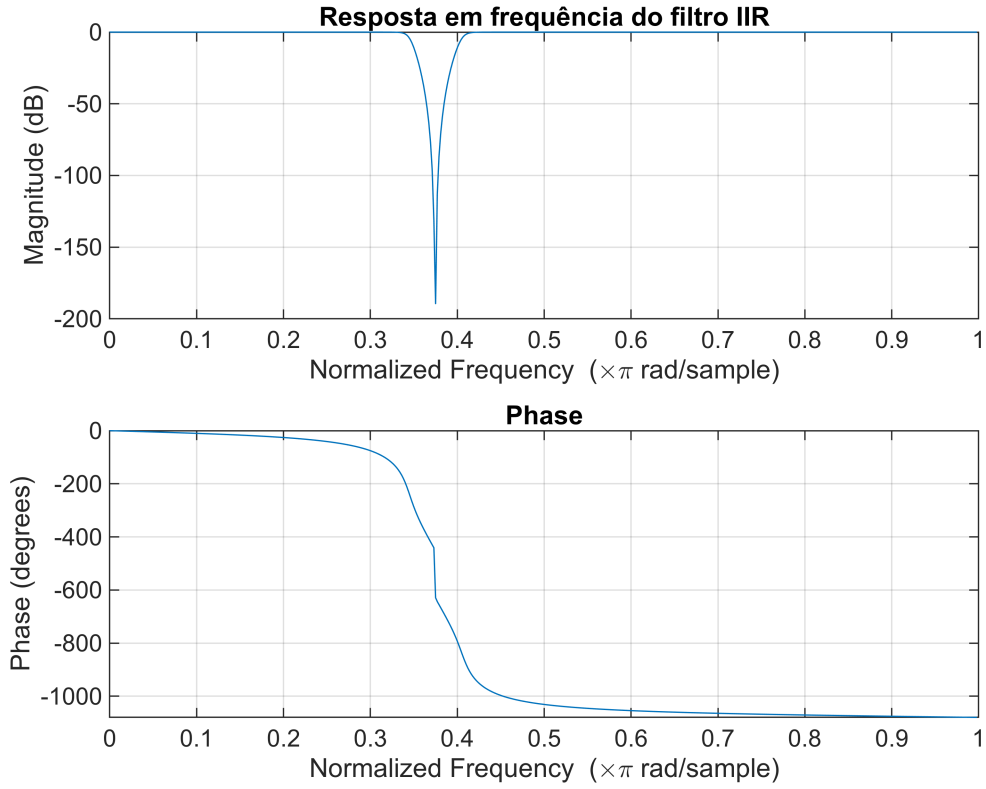
```
n = 5
Wn = 1×2
    0.3434    0.4071
```

```


[b,a] = butter(n,Wn,'stop');

figure
subplot(2,1,1)
stem(b,'filled','k')
```

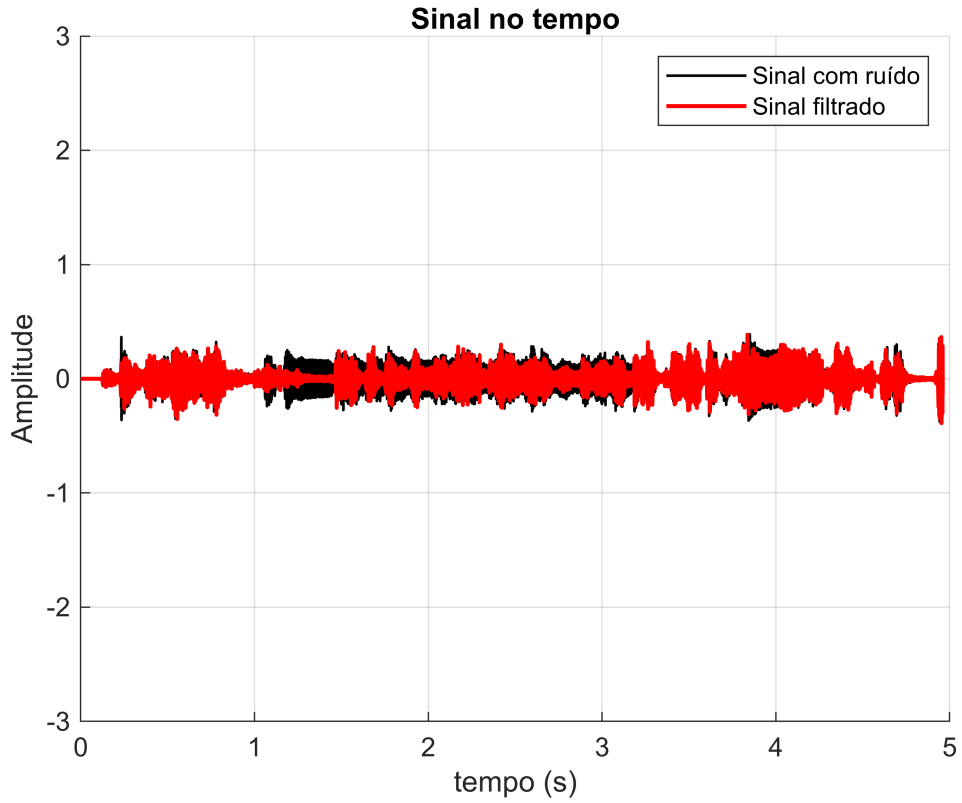
```
xlabel('n')
ylabel('b[n]')
title('Coeficientes do numerador')
subplot(2,1,2)
stem(a,'filled','r')
xlabel('n')
ylabel('a[n]')
title('Coeficientes do denominador')
```




```
figure
freqz(b, a)
title('Resposta em frequência do filtro IIR')
```




```
% fvtool(b,a)

%% Aplicar filtro

x_filtrado = filter(b,a,x);

% plotar sinais

figure
hold all
plot(t,x,'k','LineWidth',1)
plot(t,x_filtrado,'r','LineWidth',1.5)
xlabel('tempo (s)')
ylabel('Amplitude')
legend({'Sinal com ruído','Sinal filtrado'})
title('Sinal no tempo')
grid on
ylim([-3 3])
```

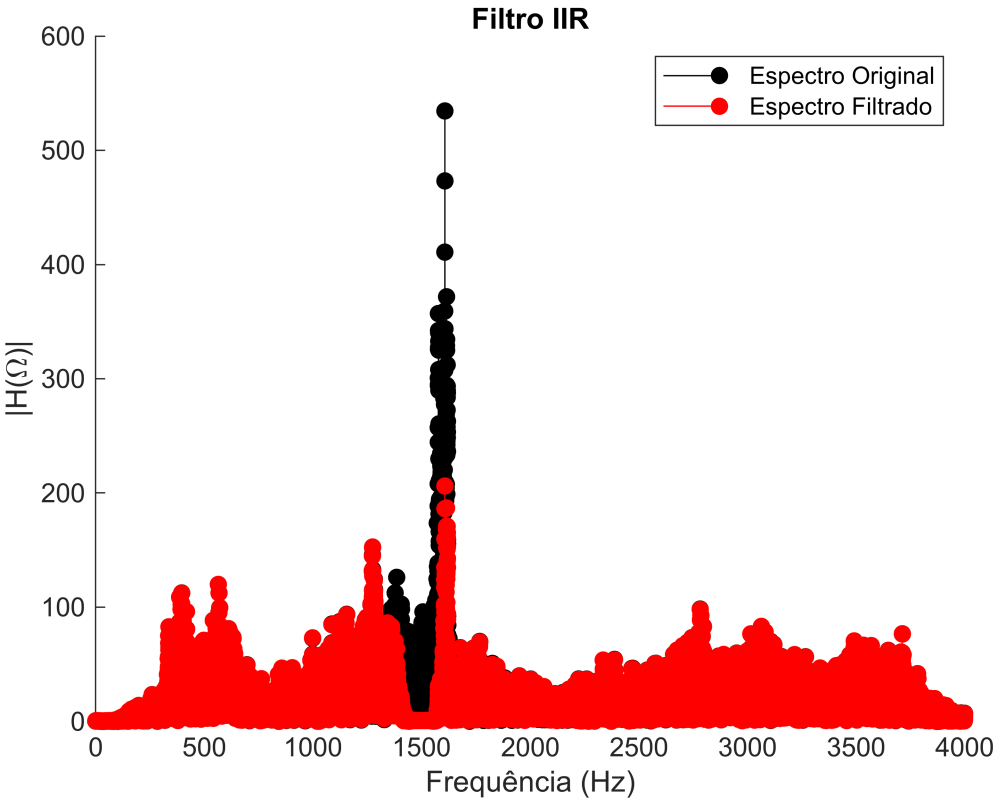



```
%plotar espectro

[f,H_filtrado] = obterEspectro(x_filtrado,t)
```

```
f = 1×19841
10^3 ×
         0    0.0002    0.0004    0.0006    0.0008    0.0010    0.0012    0.0014 ···
H_filtrado = 19841×1
    0.3334
    0.6283
    0.5942
    0.6511
    0.6127
    0.5658
    0.4786
    0.4120
    0.3974
    0.4239
      ⋮
```

```
figure
hold all
stem(f,H,'Filled','k')
stem(f,H_filtrado,'Filled','r')
xlabel('Frequência (Hz)')
ylabel('|H(\Omega)|')
title('Filtro IIR')
```

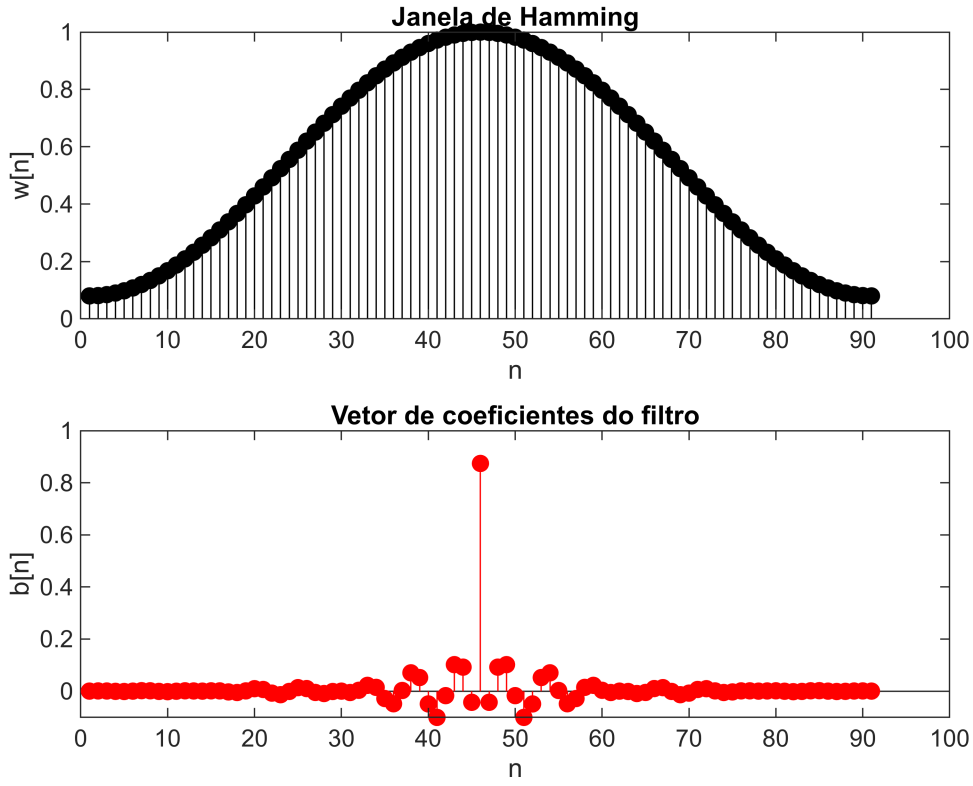
```
legend({'Espectro Original','Espectro Filtrado'})
```



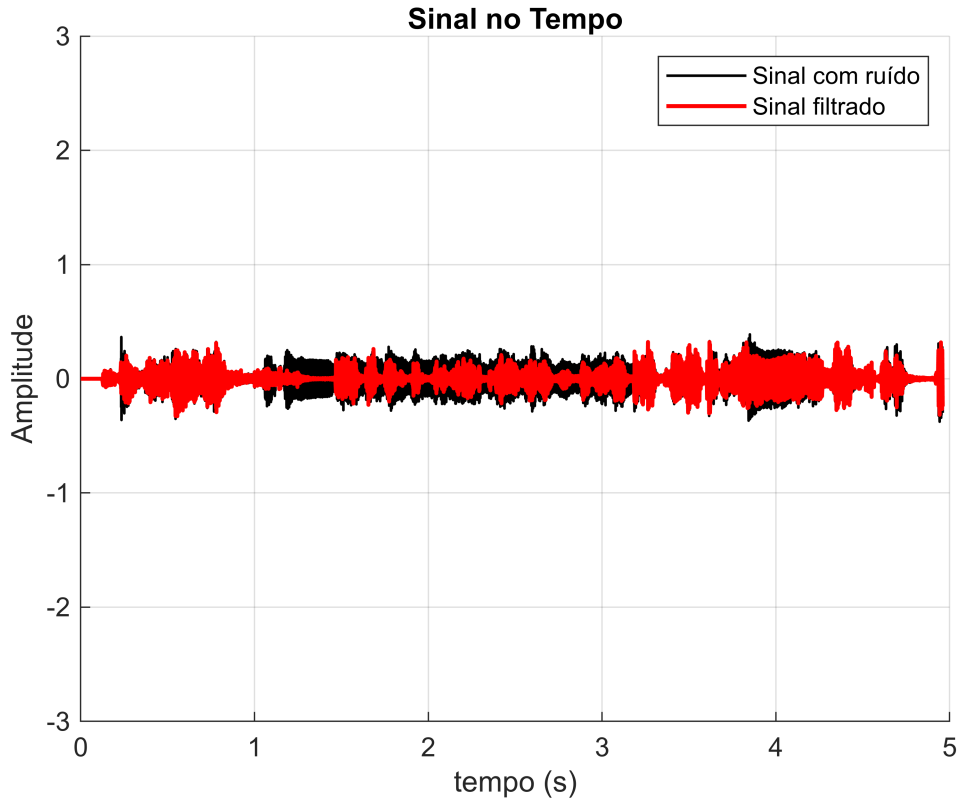
Projeto filtro FIR

```
%% Encontrar ordem do filtro FIR

f = [1200 1400 1600 2000 ];
a = [1 0 1]; %aumentar
dev = [0.01 0.1 0.1]; %aumentar


[n,Wn,beta,ftype] = kaiserord(f,a,dev,fs);
w = window(@hamming,n+1);

b = fir1(n,Wn,ftype,w);

figure
subplot(2,1,1)
stem(w,'filled','k')
xlabel('n')
ylabel('w[n]')
title('Janela de Hamming')
subplot(2,1,2)
stem(b,'filled','r')
xlabel('n')
ylabel('b[n]')
```

```
title('Vetor de coeficientes do filtro')
```




```
% fvtool(b,1)

%% Aplicar o filtro

x_filtrado = conv(x,b,'same');

%plotar sinal

figure
hold all
plot(t,x,'k','LineWidth',1)
plot(t,x_filtrado,'r','LineWidth',1.5)
xlabel('tempo (s)')
ylabel('Amplitude')
legend({'Sinal com ruído','Sinal filtrado'})
title('Sinal no Tempo')
grid on
ylim([-3 3])
```

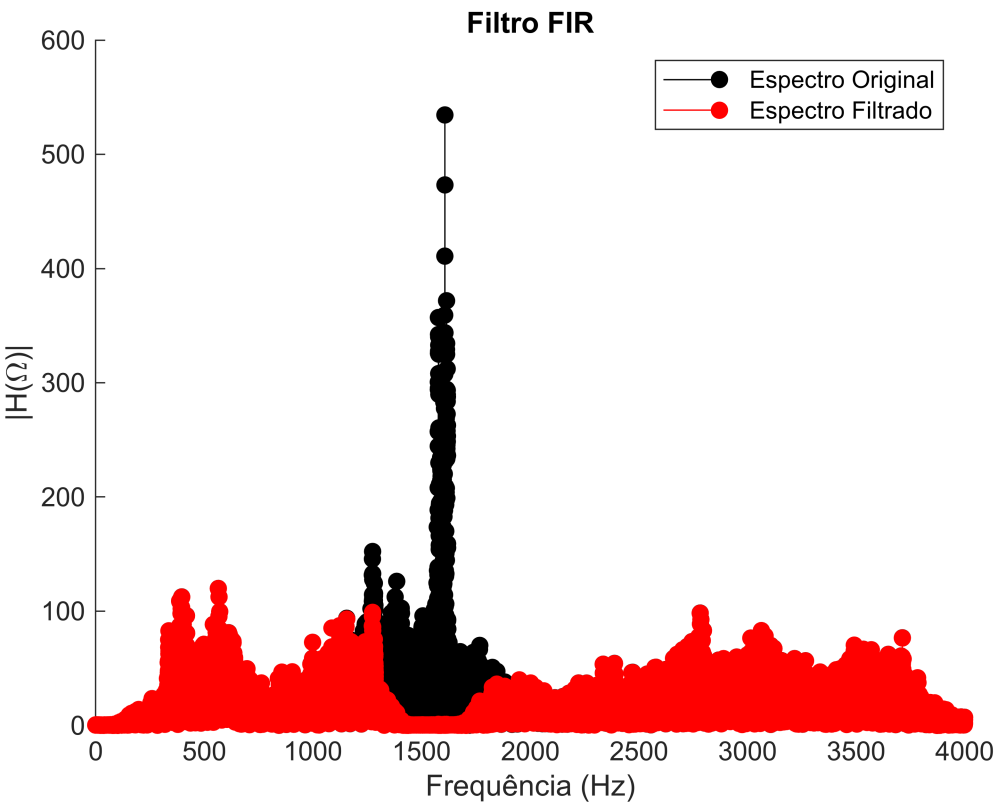



```
[f,H_filtrado] = obterEspectro(x_filtrado,t)
```

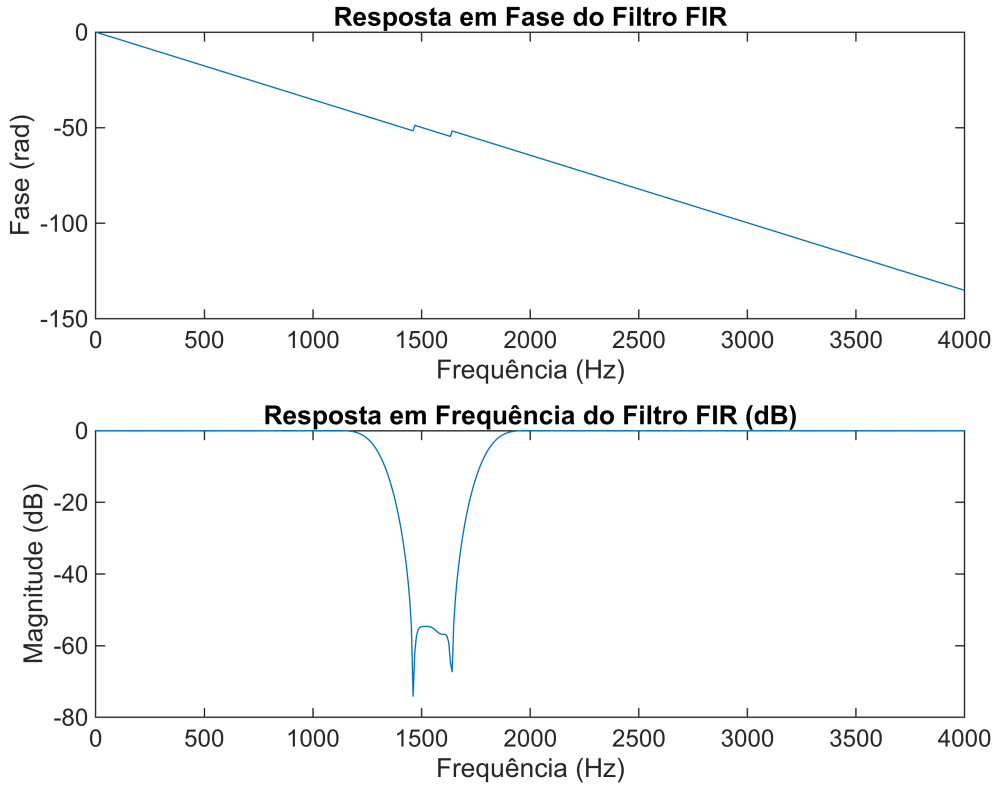
```
f = 1×19841
10^3 ×
         0    0.0002    0.0004    0.0006    0.0008    0.0010    0.0012    0.0014 ···
H_filtrado = 19841×1
    0.2647
    0.4913
    0.4641
    0.5187
    0.4776
    0.4292
    0.3428
    0.2767
    0.2606
    0.2870
      ⋮
```

```
%plotar espectro

figure
hold all
stem(f,H,'Filled','k')
stem(f,H_filtrado,'Filled','r')
xlabel('Frequência (Hz)')
ylabel('|H(\Omega)|')
```

```
title('Filtro FIR')
legend({'Espectro Original','Espectro Filtrado'})
```




```
%%

N = 1024;  % Número de pontos da FFT
H = fft(b, N);  % Cálculo da resposta em frequência do filtro FIR
f = linspace(0, fs/2, N/2+1);  % Eixo de frequência correspondente

% Plotagem da resposta em fase
figure
subplot(2,1,1) % Gráfico superior
plot(f, unwrap(angle(H(1:N/2+1))))
xlabel('Frequência (Hz)')
ylabel('Fase (rad)')
title('Resposta em Fase do Filtro FIR')


% Plotagem da resposta em frequência em dB
subplot(2,1,2) % Gráfico inferior
plot(f, 20*log10(abs(H(1:N/2+1))))
xlabel('Frequência (Hz)')
ylabel('Magnitude (dB)')
title('Resposta em Frequência do Filtro FIR (dB)')
```




```
function [f,H] = obterEspectro(x,t)

F = fft(x);
N = length(x);

magH = abs(F);
magH = magH(1:end/2+1);
magH(2:end) = magH(2:end)*2;

H=magH;

Ts = t(2)-t(1);
fs = 1/Ts;

f = (0:length(magH)-1)*(fs/2)/length(magH);

end
```