

```

p = 0; %vector de fase

%%Definir la frecuencia de muestreo
%Definir el tiempo
fs = 1024; % frecuencia de muestreo en Hz
t = 5; % tiempo de la señal en segundos

%Señal en el dominio del tiempo
[y_t,t_s] = f_creasenal(a,f,p,t,fs);

figure, plot(t_s,y_t),title('Señal original');

%%Grafica del espectro
%Determinar el numero de muestras
n = length(y_t);

%%%%Transformada de fourier, fft devuelve la transformada de fourier
%%%%de la señal de la forma  $F(W) = A(w) + iB(w)$ 
y_f1 = fftshift(y_f);

vf1 = linspace(-fs/2,fs/2,n); %vector de frecuencias de acuerdo ala salida del modulo d

my_f1 = abs(y_f1); %Espectro de magnitud

figure,plot(vf1,my_f1),title('Espectro de magnitud de la fft'),grid on;

my_f2 = 2 * (my_f(1:(n)/2)/n);

vf2 = linspace(0,fs/2,n/2);

%Se obtiene la grafica
figure,plot(vf2,my_f2),title('Mitad de Espectro de Frecuencias'),grid on;

% %%Diseño de filtro pasa bajas-----
%
```

```

%Aplicacion del filtro
my_f2_f = my_f2.*bpf;%Señal filtrada

%Grafica de a señal filtrada y el filtro
figure,stem(vf2,my_f2_f);
hold on
plot(vf2,bpf,'-r','LineWidth',2),grid on,title('Filtro vs senal filtrada');
xlabel('Frecuencia(Hz)');
ylabel('Amplitud');
legend('Senal filtrada , filtro');

%%Muestra las frecuencias despues del filtro
figure,stem(vf2,my_f2_f,'LineWidth',2,'MarkerFaceColor','red'),grid on,title('Frecuencia d
xlabel('Frecuencia(Hz)');
ylabel('Amplitud');
%xlim([min(vf2)-10 (max(vf2)+10)]);

```

