UPV, Facultad de Informática
Dpto. de Arquitectura y Tecnología de Computadores

PROCESADO DIGITAL DE SEÑAL

PROYECTO ESPECÍFICO - LABORATORIO 1 (Mat)

MATLAB

INTRODUCCIÓN A MATLAB PARA EL PROCESADO DE SEÑALES GENERACIÓN DE SEÑALES Y SUS CARACTERÍSTICAS

Componentes del grupo:
- Alex Beltrán
- Daniel Cañadillas
- Ainhoa Serrano

Nota: Enviar este documento "Lab1_Mat_resultados.doc" completado con las tareas solicitadas, el código generado y los comentarios y aclaraciones que consideréis oportunos, junto con los correspondientes ficheros .m, en un fichero .zip vía eGela.

SEÑALES BASICAS (P1_1_impulsos_sinusoidales.m)

a) **Impulsos**

Imágenes de los resultados obtenidos al generar las funciones (utiliza mejor el comando `stem`). Etiqueta de manera adecuada las gráficas que se adjuntan y añade los comentarios u observaciones que consideres oportunos.

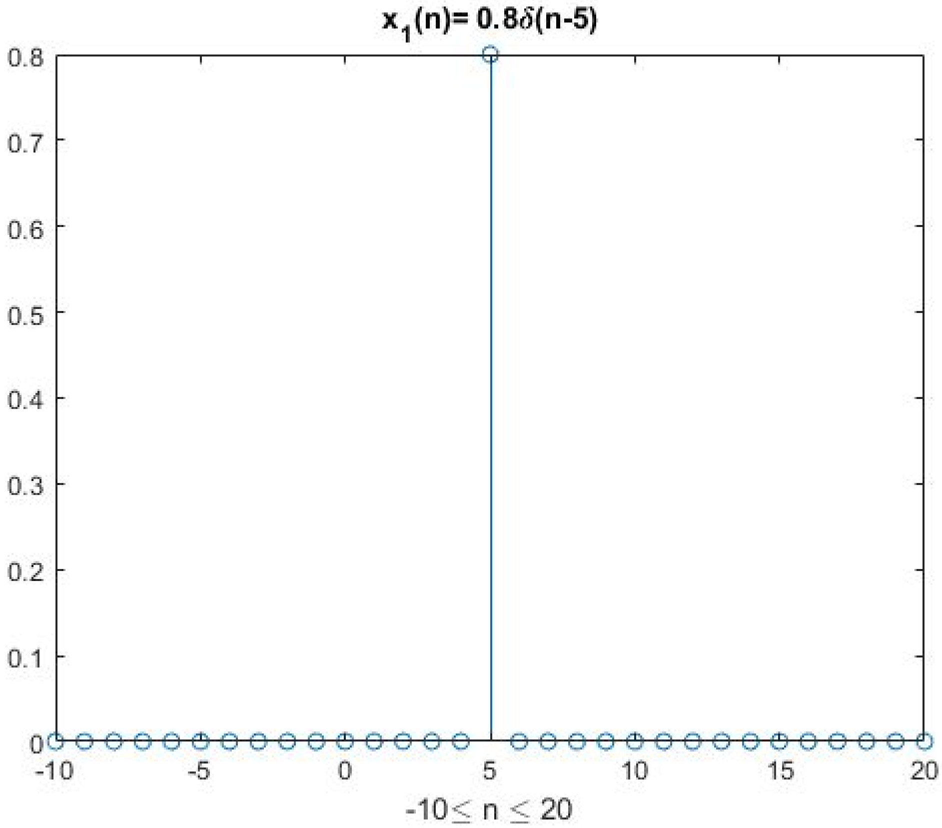
$$x_1(n) = 0.8\delta(n-5) \qquad -10 \le n \le 20$$

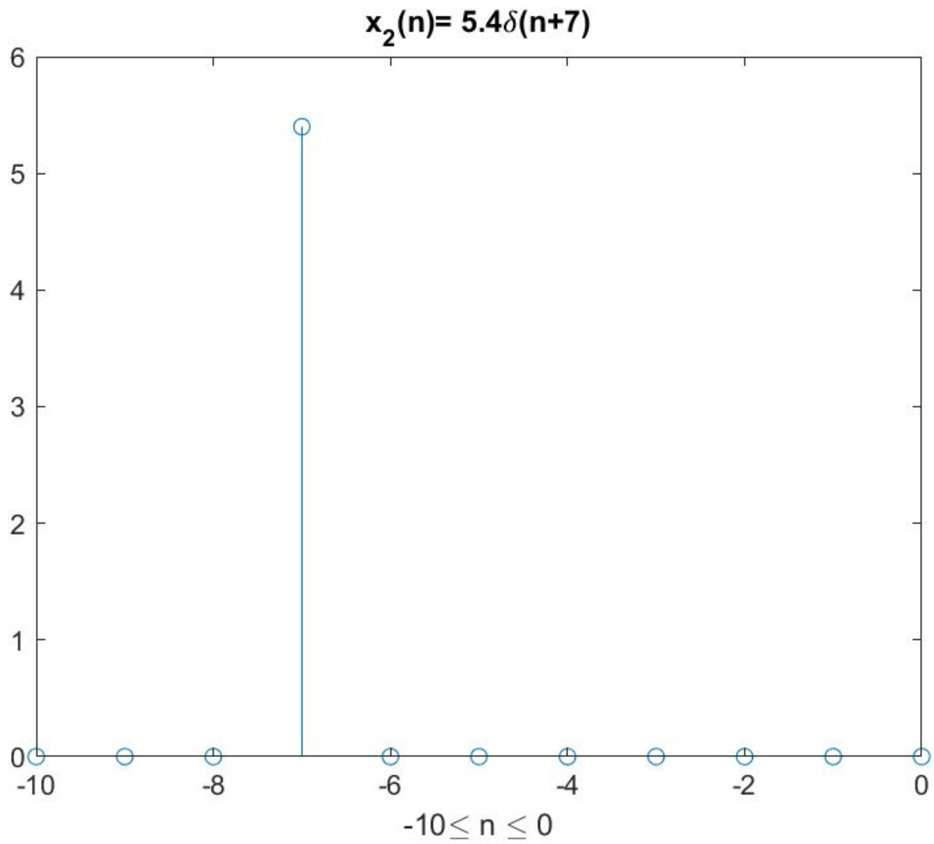
$$x_2(n) = 5.4\delta(n+7) \qquad -10 \le n \le 0$$

```
% a) Impulsos

% x1)
rX1 = -10:20;
dX1 = (rX1-5) == 0;
x1 = 0.8 .* dX1;
figure; stem(rX1,x1)
xlabel('-10\leq n \leq 20')
title('x_1(n)= 0.8\delta(n-5)')

% x2)
rX2 = -10:0;
dX2 = (rX2+7) == 0;
x2 = 5.4 .* dX2;
figure; stem(rX2,x2)
xlabel('-10\leq n \leq 0')
title('x_2(n)= 5.4\delta(n+7)')
```


x1(n)= 0.8δ(n-5)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-5
0
5
10
15
20
-10≤ n ≤ 20


x2(n)= 5.4δ(n+7)
6
5
4
3
2
1
0
-10
-8
-6
-4
-2
0
-10≤ n ≤ 0

b) **Sinusoidales**

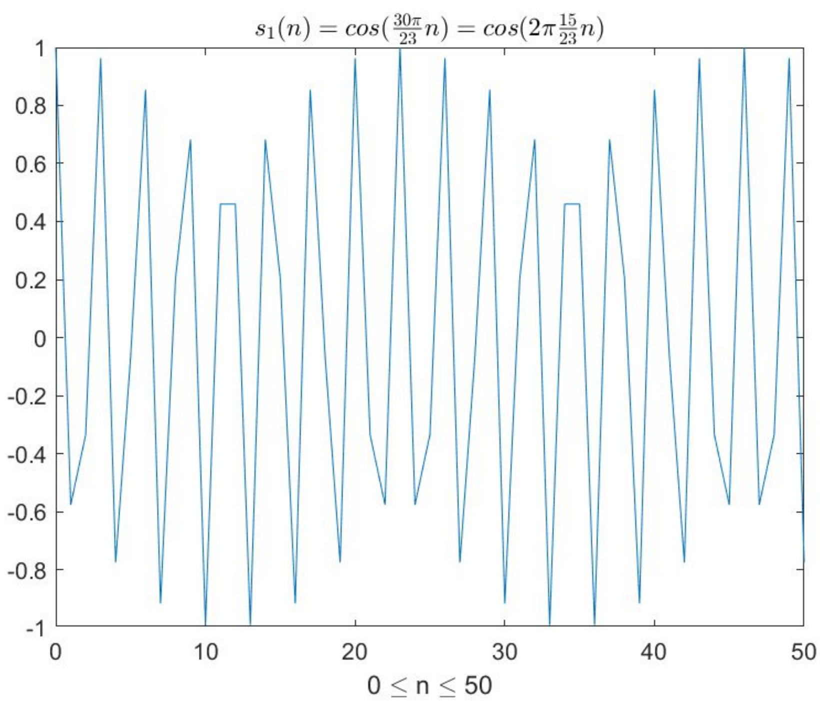
A medida que generas y visualizas las señales, completa la siguiente tabla (indicando las unidades).

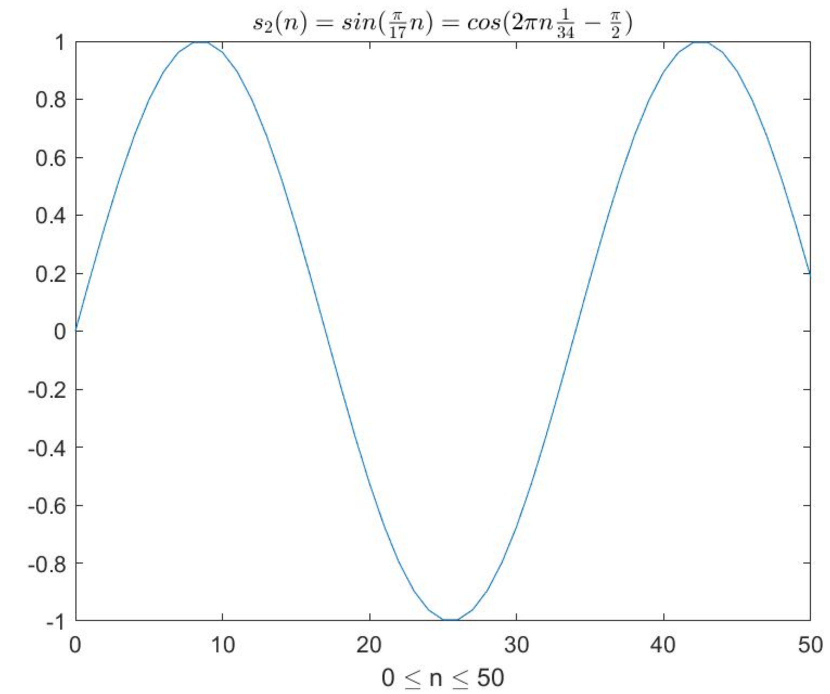
Señal	Amplitud **A**	Frecuencia **F**	Fase ϕ	Periodo **N**
$s_1(n)$	1	15/23	0	23
$s_2(n)$	1	1/34	-π/2	34
$s_3(n)$	0.8	1/2	π/2	2

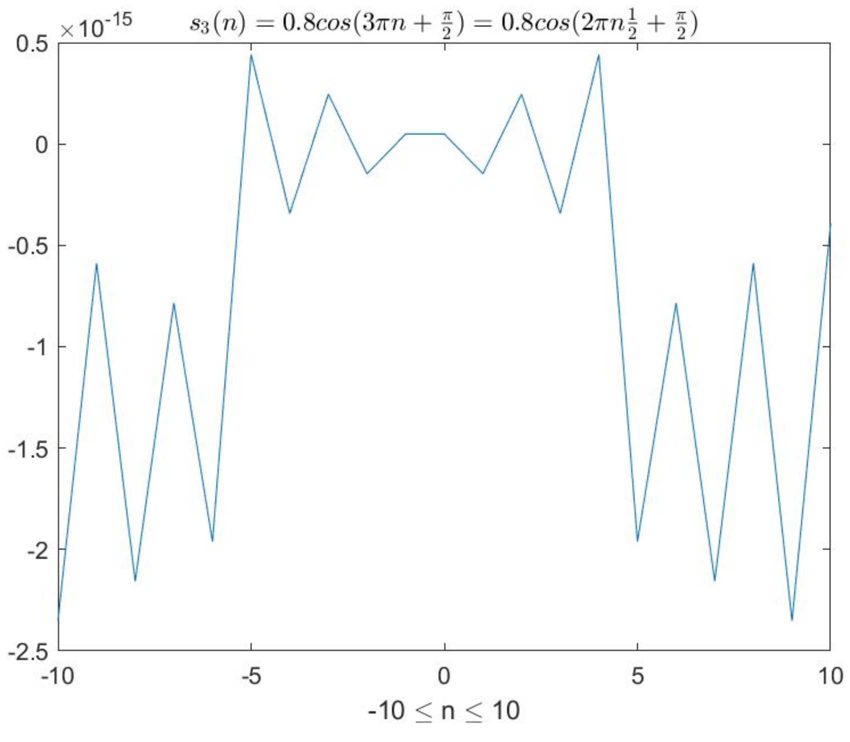
```
%b) Sinusoidales
n1 = 0:50;
s1 = cos(2*pi*(15/23)*n1);
figure;
plot(n1,s1);
xlabel('0 \leq n \leq 50');
title('$s_1(n)=cos(\frac{30\pi}{23}n)=cos(2\pi \frac{15}{23}n)$','interpreter','latex');

n2 = 0:50;
s2 = cos((2*pi*1/34)*n2 - pi/2);
figure; plot(n2,s2);
xlabel('0 \leq n \leq 50');
title('$s_2(n)=sin(\frac{\pi}{17}n)=cos(2\pi n\frac{1}{34} - \frac{\pi}{2})$','interpreter','latex');

n3 = -10:10;
s3 = 0.8 * cos(2*pi*(1/2)*n3 + pi/2);
figure; plot(n3,s3);
xlabel('-10 \leq n \leq 10');
title('$s_3(n)=0.8cos(3\pi n + \frac{\pi}{2}) = 0.8cos(2\pi n \frac{1}{2} + \frac{\pi}{2})$','interpreter','latex');
```







¿Observas algo extraño en **s_3(n)**? ¿Cómo puede explicarse? ***Al visualizar la gráfica, podemos observar que la señal es simétrica y que los valores de N entre -1 y 0 tienen un valor en la función de 0. Esto puede ser porque la frecuencia es de 1/2 , y está en el máximo de oscilación [F = 0.5].***

c) **Exponenciales complejas** (P1_2_exponenciales_complejas.m)

A medida que generas los armónicos de orden 4, 5 y 11, rellena la siguiente tabla.

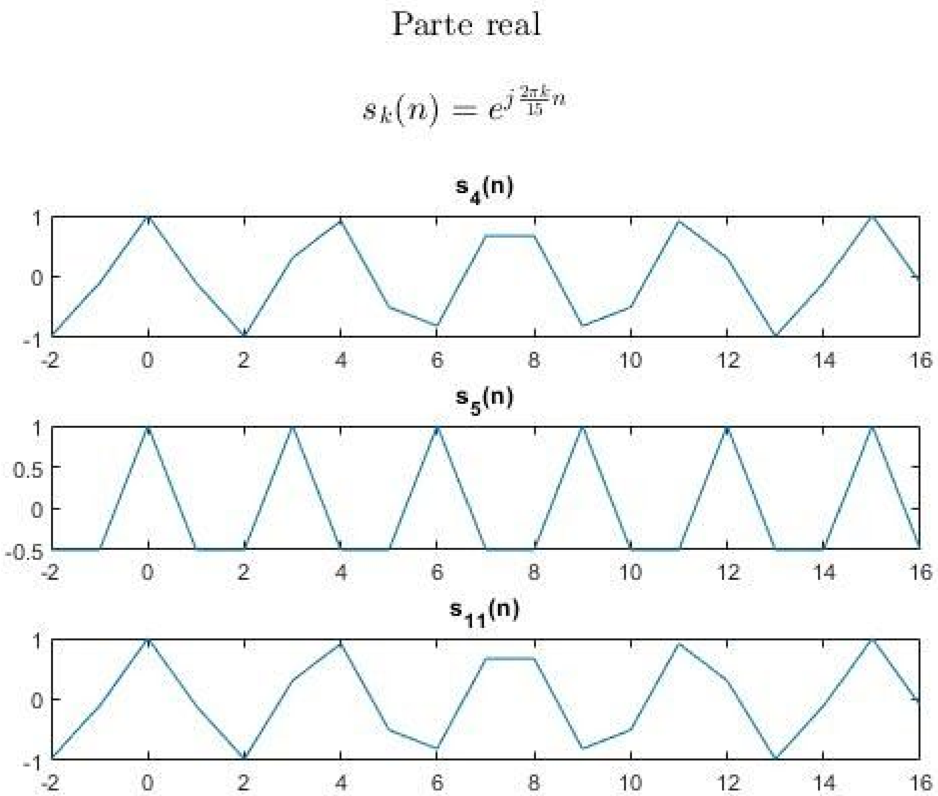
Señal	Frecuencia **F**	Periodo **N**
$S_4(n)$	4/15	15
$S_5(n)$	1/3	3
$S_{11}(n)$	11/15	15

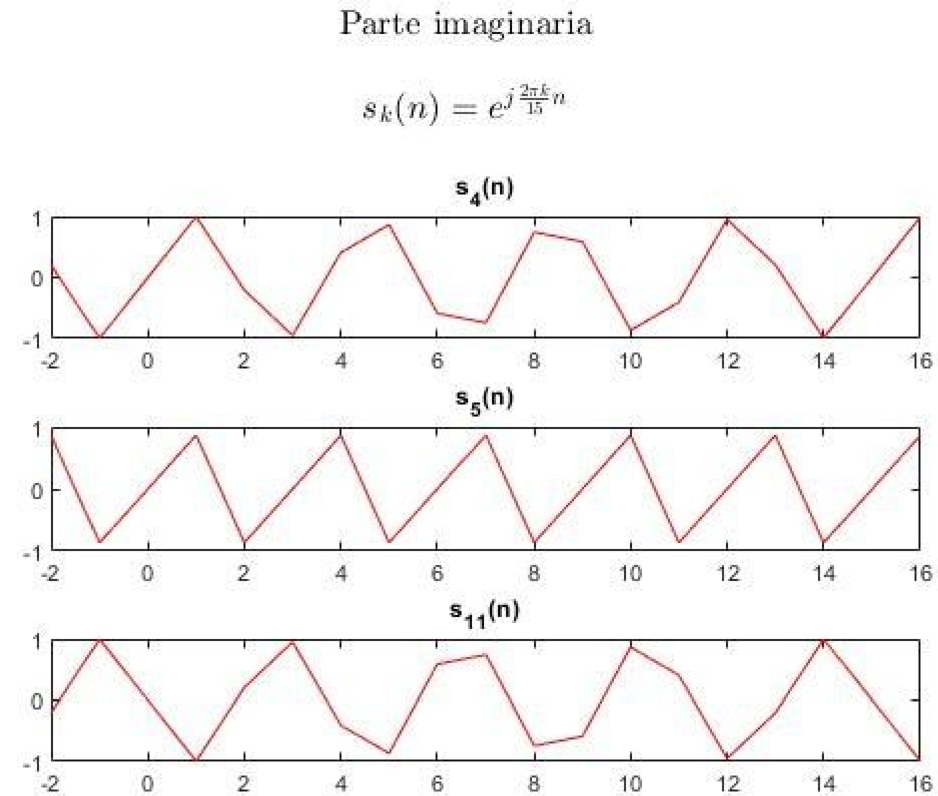
```
% Exponenciales Complejas
n = -2:16;
sk_4 = exp(1j*2*pi*4/15*n);
sk_5 = exp(1j*2*pi*5/15*n);
sk_11 = exp(1j*2*pi*11/15*n);

figure;
subplot(3,1,1);
plot(n,real(sk_4));
title('s_4(n)');
subplot(3,1,2);
plot(n,real(sk_5));
title('s_5(n)');
subplot(3,1,3);
plot(n,real(sk_11));
title('s_{11}(n)');
sgtitle({'Parte real';'';'$s_k(n)=e^{j\frac{2\pi k}{15}n}$'},'interpreter','latex');

figure;
subplot(3,1,1);
plot(n,imag(sk_4),'r');
title('s_4(n)');
subplot(3,1,2);
plot(n,imag(sk_5),'r');
title('s_5(n)');
subplot(3,1,3);
plot(n,imag(sk_11),'r');
title('s_{11}(n)');
sgtitle({'Parte imaginaria';'';'$s_k(n)=e^{j\frac{2\pi k}{15}n}$'},'interpreter','latex');
```

Adjunta las imágenes de la parte real y de la parte imaginaria en gráficas separadas de cada uno de estos armónicos. Etiqueta de manera adecuada las gráficas que se adjuntan.





¿Qué se observa al comparar dichas gráficas? ***Al visualizar las dos gráficas, podemos ver cómo , al comparar el armónico de 4 con el armónico de 11, las partes reales son iguales, pero, las partes imaginarias son opuestas.***

PROGRAMACIÓN EN MATLAB (P1_3_programacion_matlab.m)

a) Dado el código `prueba2.m` generado realiza la misma operación, pero prescindiendo ahora del comando `for`, esto es utilizando las posibilidades que ofrece MATLAB para operar con vectores.

```
% Prueba2.m en matlab:
n = 1:10; x = (n).^2; y = (n).^3
```

b) Dado un vector `v=-4:15`, genera un vector `w` del mismo tamaño en el que los valores mayores que 2 se sustituyen por -1 y los que son menores o iguales por 5.

```
% Vectores v y w:
v = -4:15; w = v; w(v > 2) = -1; w(v <= 2) = 5
```