

Instituto  
Tecnológico y de  
Estudios Superiores  
de Occidente –  
ITESO



**ITESO**  
Universidad Jesuita  
de Guadalajara

Materia:    Sistemas de comunicaciones digitales  
Profesor:    Dr. Omar Longoria  
Fecha:       11/09/2022  
Autor(es):   Alexis Luna Delgado

## Tarea 04

### Ejercicio 1.

Utilizaremos el archivo de la “Lena” en escala de grises y recortando solamente una sección de la imagen. Estos bits serán los que usaremos para construir el código de línea. Para hacer esto es necesario:

1. Leer el archivo de “la Lena” en escala de grises como lo hicimos en clases pasadas y recortar la imagen para quedarse solamente con la sección del ojo. Aquí tendrás una matriz.
  - a. `load lena512.mat`
  - b. `imshow(uint8(lena512))`
  - c. `lenarec=lena512(252:284,318:350);`
  - d. `imshow(uint8(lenarec))`
2. Convertir estas muestras tipo entero a binario. Utilice la función `de2bi()` para hacer la conversión:
  - a. `b=de2bi(lenarec,8);`
  - b. `b=b';`
  - c. `bits=b(:); % Vector de bits concatenado`
3. Deduce y codifica el proceso inverso (bits a pixeles) para reconstruir la imagen.
4. Reconstruye y muestra la imagen



### Ejercicio 1

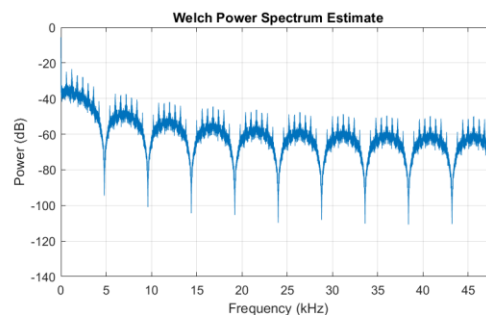
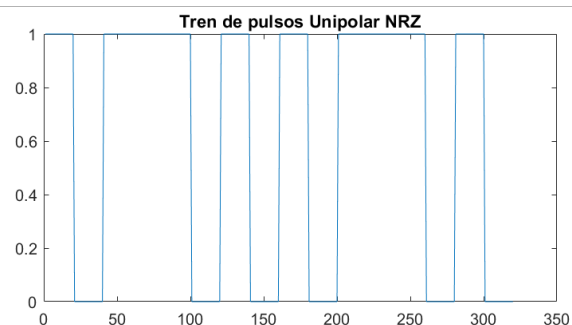
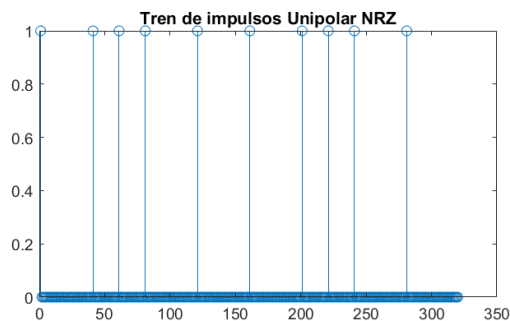
```
load lena512.mat
imshow(uint8(lena512))
lenarec=lena512(252:284,318:350);
imshow(uint8(lenarec))
b = de2bi(lenarec,8,'left-msb');
b=b';
bits=b(:); % Vector de bits concatenado
% Reconstruir la imagen
bitsM = reshape(bits,8,1089);
bitsM = bitsM';
decval = bi2de(bitsM,'left-msb');
lenaRS = reshape(decval, size(lenarec));
imshow(uint8(lenaRS))
```

## Ejercicio 2.

Diseñe pulsos para transmitir con los códigos de línea Unipolar NRZ, Polar NRZ, Polar Rz, Bipolar NRZ (AMI pulso completo) y Manchester, suponiendo que tenemos 20 muestras por pulso y  $F_s=96000$  Hz. Determine el mapeo de bit a amplitud de pulso en cada caso (la regla de codificación) y grafique el pulso base usando `wvtool()` que se aplicaría para cada código de línea.

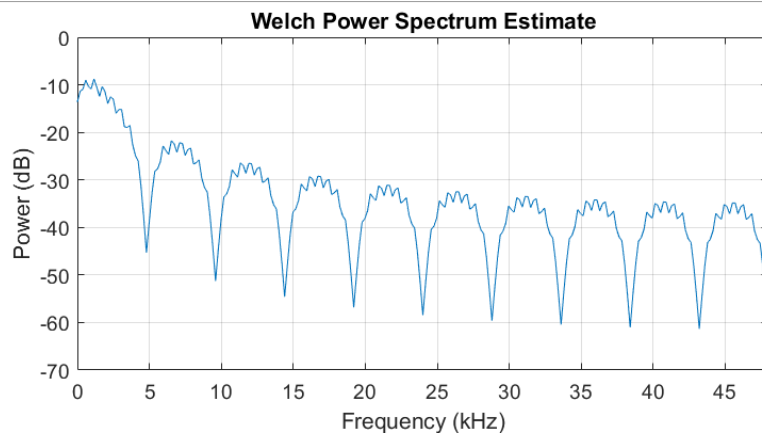
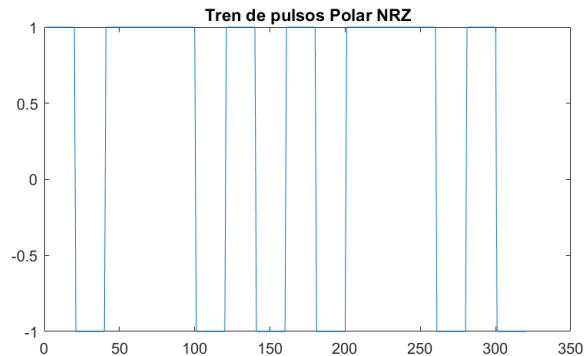
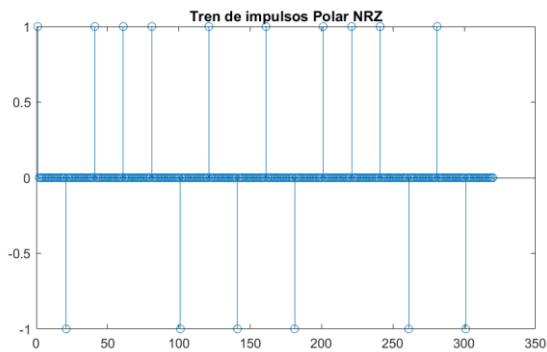
### Ejercicio 2: Unipolar NRZ

```
Fs= 96000;
mp= 20;
%No_bits= 8712;
Baud_rate= Fs/mp %Symbols per second
Bit_rate=Baud_rate % bits/s
Ts=1/Fs;
% The bit rate is Rb= Rs= Fs / mp, because 1 bit= 1 symbol and every symbol
has mp
% samples per bit
%pnrz = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]; % Pulse type with mp
samples
pnrz =ones(1,mp); % Similar to Arduino pulses with 5V
%wvtool(pnrz)
s = zeros(1,numel(bits)*mp);
s(1:mp:end) = bits; % Impulse Train
stem(s(1:mp*16))
title('Tren de impulsos Unipolar NRZ');
xUNRZ = conv(pnrz,s); % Pulse Train
plot(xUNRZ(1:mp*16))
title('Tren de pulsos Unipolar NRZ');
pwelch(xUNRZ,[],[],[],Fs,'power'); % PSD of Unipolar NRZ
%wvtool(xUNRZ)
```



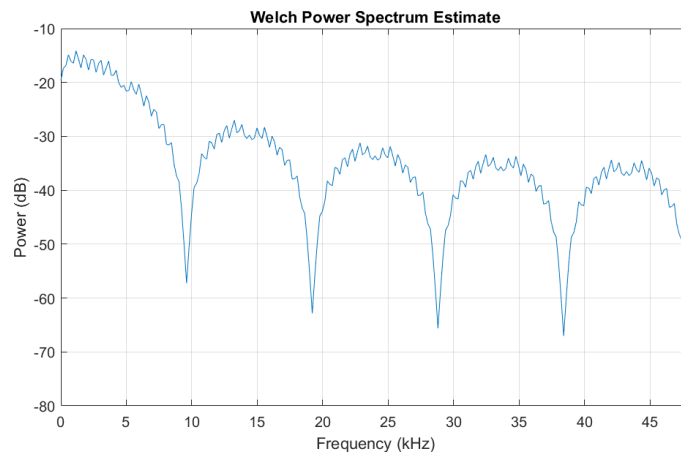
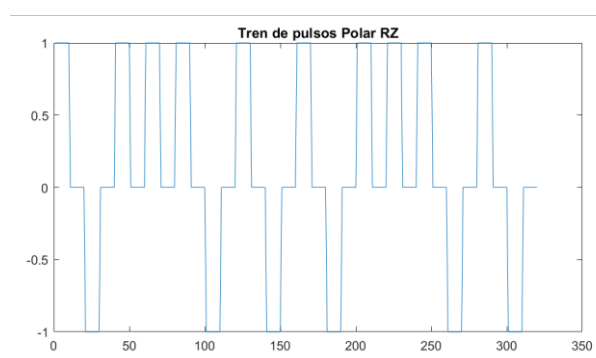
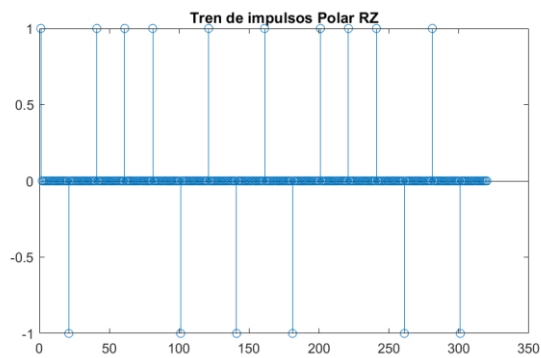
## Ejercicio 2: Polar NRZ

```
%Baud_rate= Fs/mp %Symbols per second
%Bit_rate=Baud_rate % bits/s
pnrz=ones(1,mp);
s1=bits;
s1(s1==0)=-1;
s=zeros(1,numel(s1)*mp);
s(1:mp:end)=s1; %Impulse train
stem(s(1:mp*16))
title('Tren de impulsos Polar NRZ');
xPNRZ=conv(pnrz,s); %Pulse train
plot(xPNRZ(1:mp*16))
title('Tren de pulsos Polar NRZ');
pwelch(xPNRZ,500,300,500,Fs,'power');
%%wvtool(xPNRZ)
```



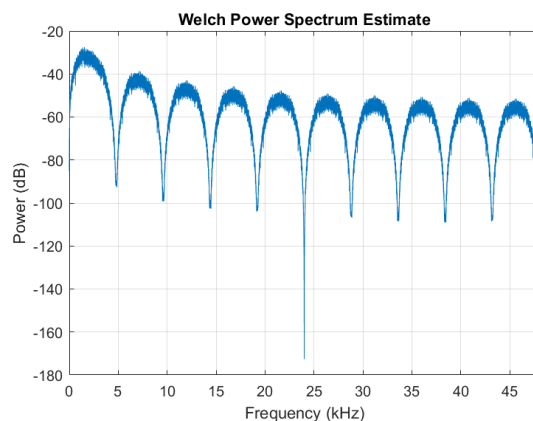
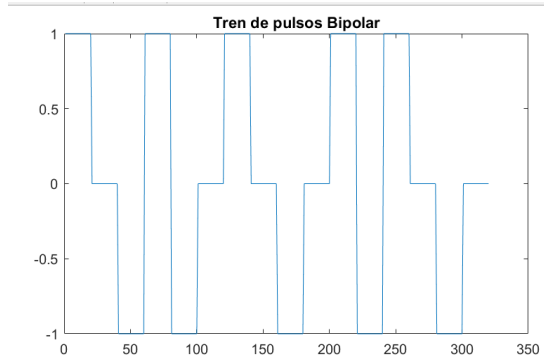
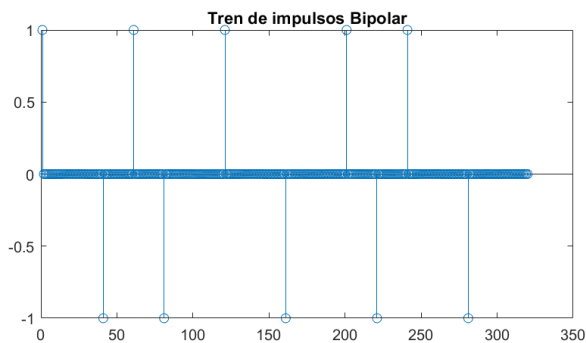
## Ejercicio 2: Polar RZ

```
%Baud_rate= Fs/mp %Symbols per second
%Bit_rate=Baud_rate % bits/s
pprz=[ones(1,mp/2),zeros(1,mp/2)];
s1=int8(bits);
s1(bits==0)=-1;
s=zeros(1,numel(s1)*mp);
s(1:mp:end)=s1;
stem(s(1:mp*16))
title('Tren de impulsos Polar RZ');
xprz=conv(pprz,s);
plot(xprz(1:mp*16)); % Show 16 symbols
title('Tren de pulsos Polar RZ');
pwelch(xprz,500,300,500,Fs,'power');
%%wvtool(xprz)
```



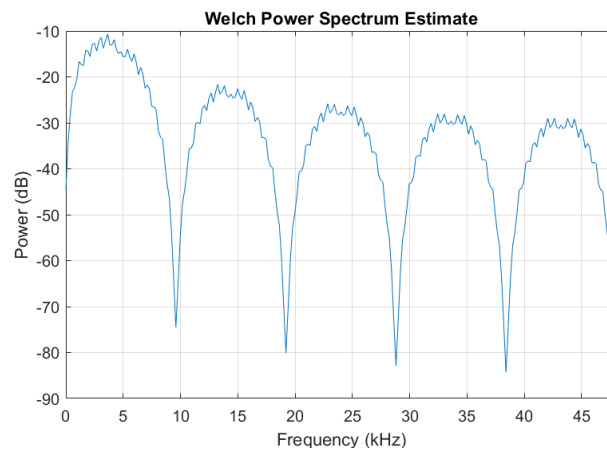
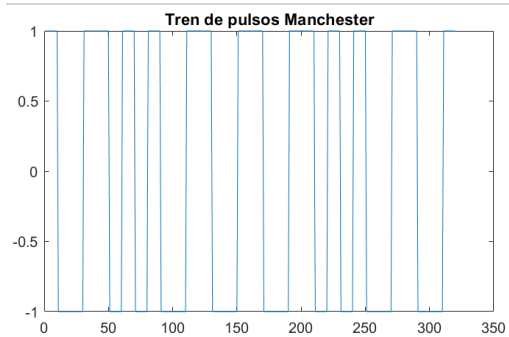
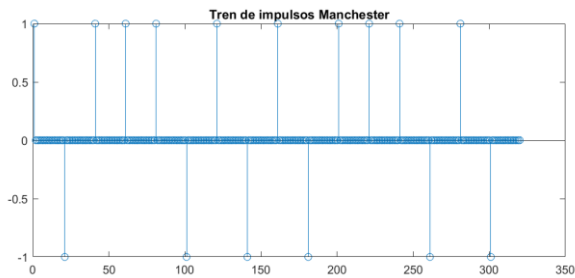
## Ejercicio 2: Bipolar NRZ

```
%Baud_rate= Fs/mp %Symbols per second
%Bit_rate=Baud_rate % bits/s
pAMI = [ones(1,mp),zeros(1,mp)];
s1 = bits;
flag = boolean(0);
for k= 1:numel(s1);
    if (s1(k)==1)
        if (flag == 1)
            s1(k)=-1;
        end
        flag = ~flag;
    end
end
s=zeros(1,numel(s1)*mp);
s(1:mp:end)=s1;
stem(s(1:mp*16))
title('Tren de impulsos Bipolar');
xAMI=conv(pAMI,s);
plot(xAMI(1:mp*16)); % Show 16 symbols
title('Tren de pulsos Bipolar');
pwelch(xAMI, [],[],[],Fs,'power');
%%wvtool(xAMI)
```



## Ejercicio 2: Manchester

```
%Baud_rate= Fs/mp %Symbols per second
%Bit_rate=Baud_rate % bits/s
pm=[ones(1,mp/2),-1*ones(1,mp/2)];
s1=bits;
s1(s1==0)=-1;
s=zeros(1,numel(bits)*mp);
s(1:mp:end)=s1;
stem(s(1:mp*16))
title('Tren de impulsos Manchester');
xm=conv(pm,s);
plot(xm(1:mp*16)); % Show 16 symbols
title('Tren de pulsos Manchester');
pwelch(xm,500,300,500,Fs,'power');
%%wvtool(xm)
```



### Ejercicio 3.

Genere las señales transmitidas (tren de pulsos) para los cuatro códigos de línea y grafique en el dominio del tiempo solamente el tren de pulsos para cada código de línea, correspondiente a los primeros 16 bits, tanto con la función plot() como con stem(). Conteste:

¿Cuánto tiempo ( $T_s$ ) hay entre muestra y muestra?

$$T_s = 1/F_s = 1/96000 = 1.041666666666667e-5$$

¿Cuánto tiempo ( $T_p$ ) hay entre pulsos transmitidos?

$$R_s = F_s/m_p = 4800$$

$$T_p = 1/R_s = 1/4800 = 2.083e-4$$

### Ejercicio 4.

A.) Calcule el Baud rate y el Bit rate

B.) Grafique la densidad espectral de potencia y estime el ancho de banda de la señal generada con cada código de línea.

C.) Compare con los espectros graficados en la figura del libro de Couch y, si hay diferencias, explíquelas.

Al comparar las gráficas del libro de Couch con los espectros graficados podemos ver que no existen diferencias notables.