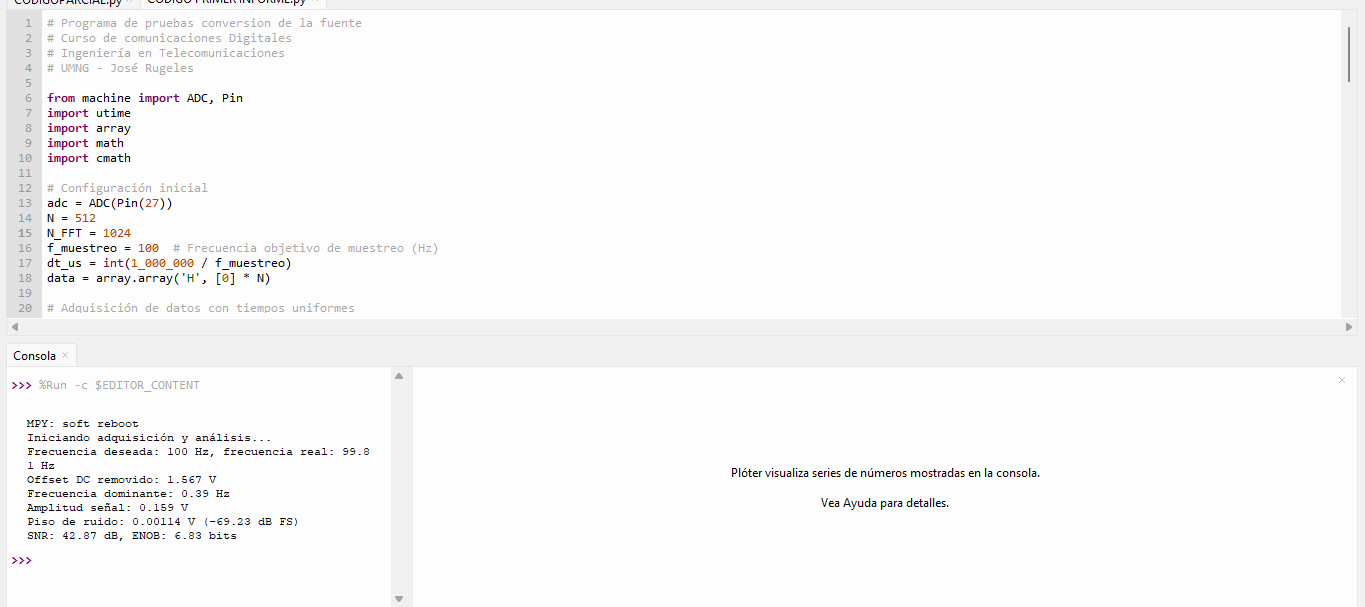
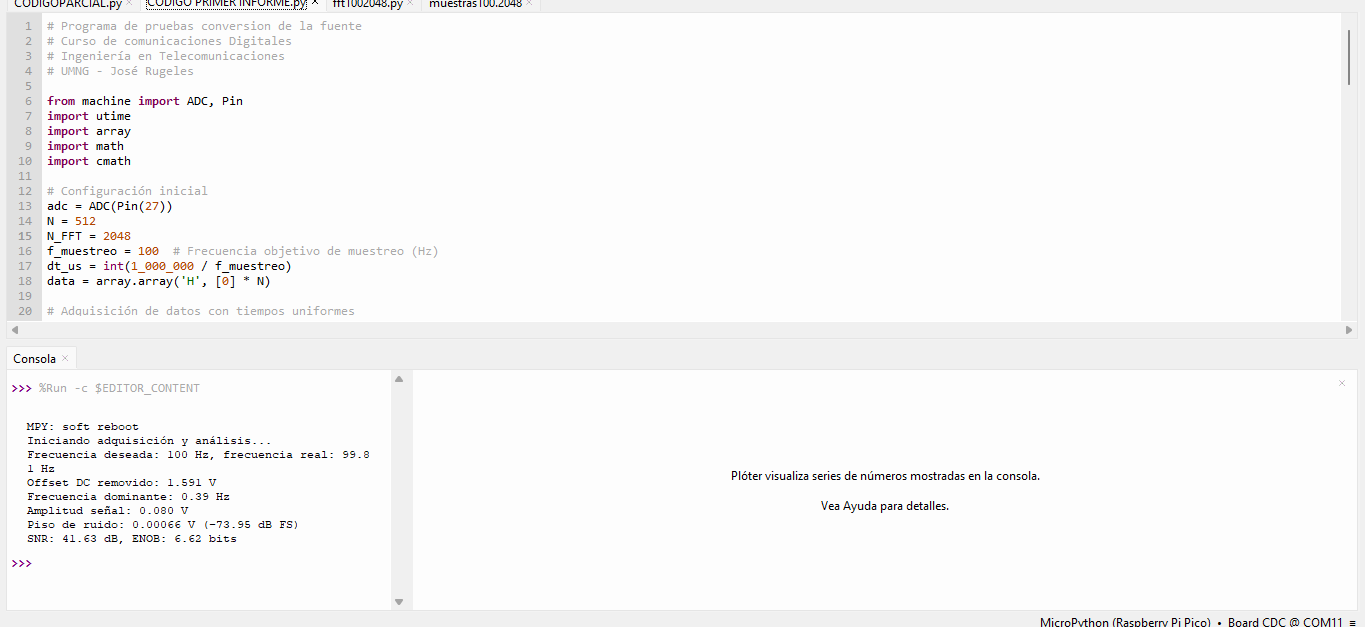
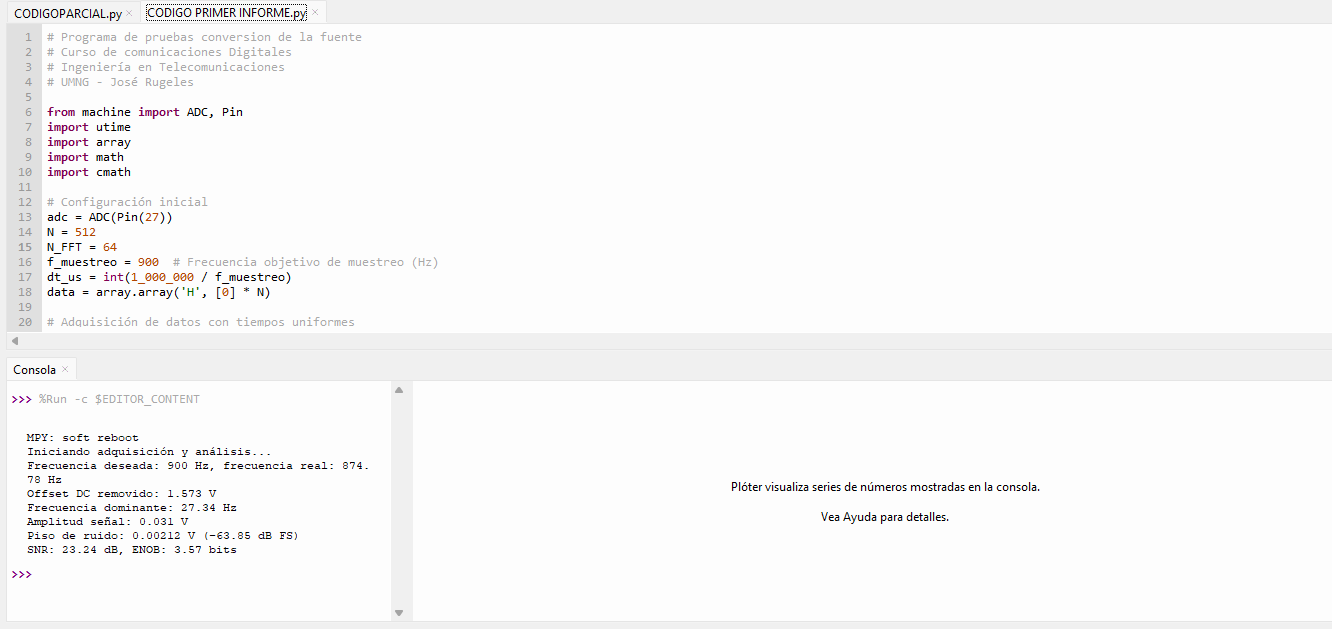
100 con 1024



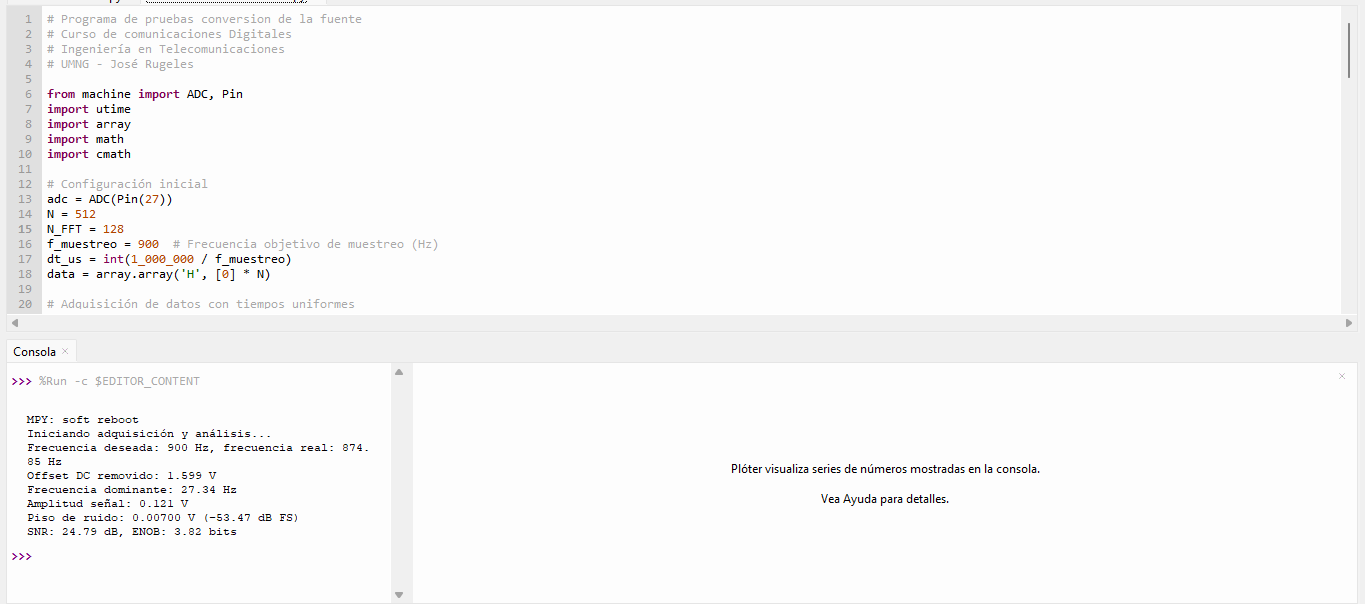
100con2048



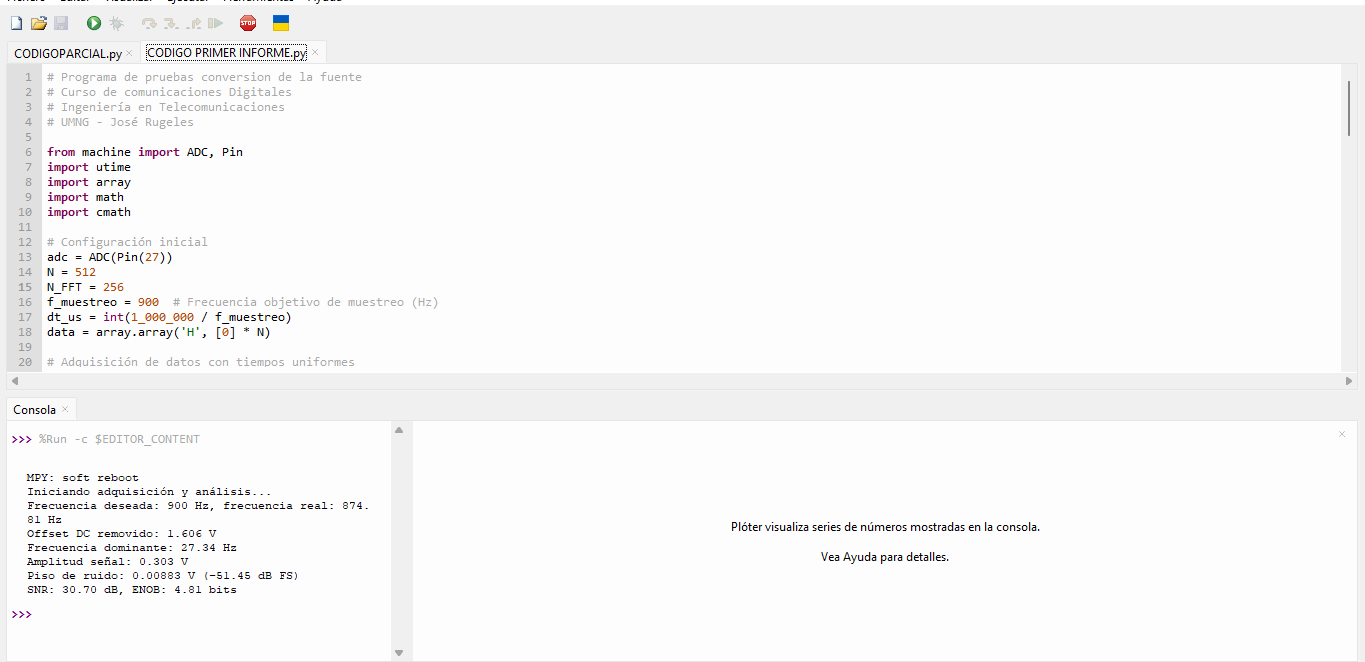
900 con 64



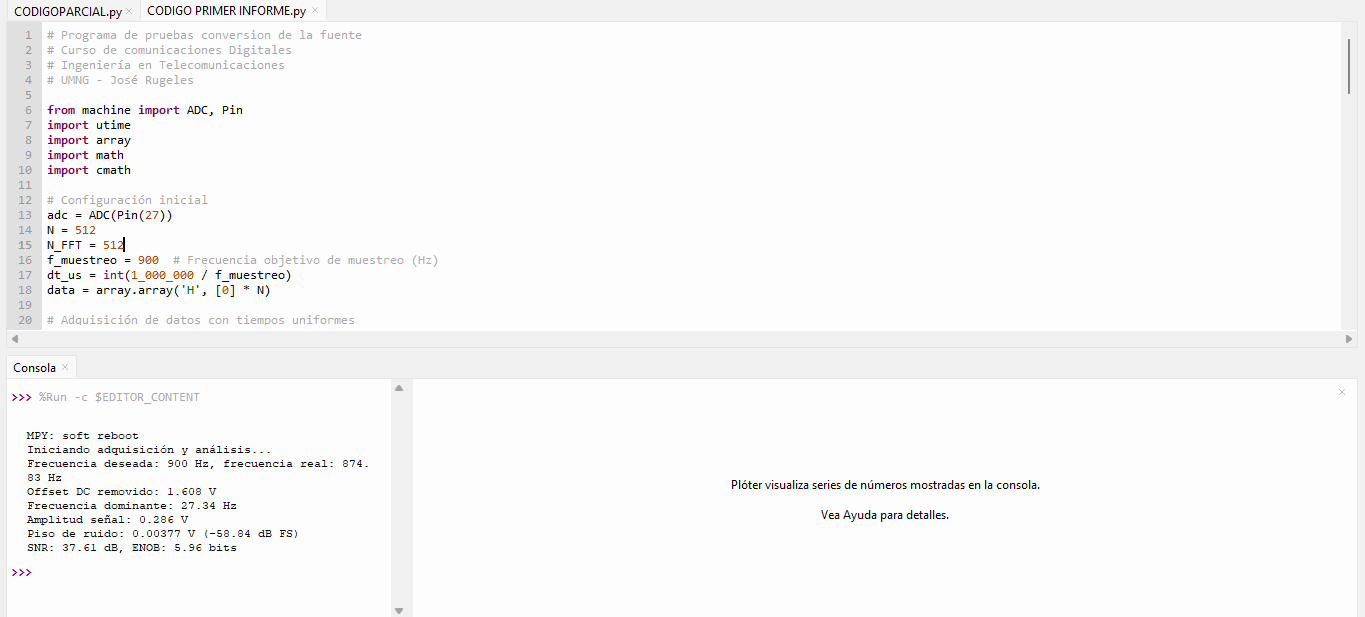
900con128



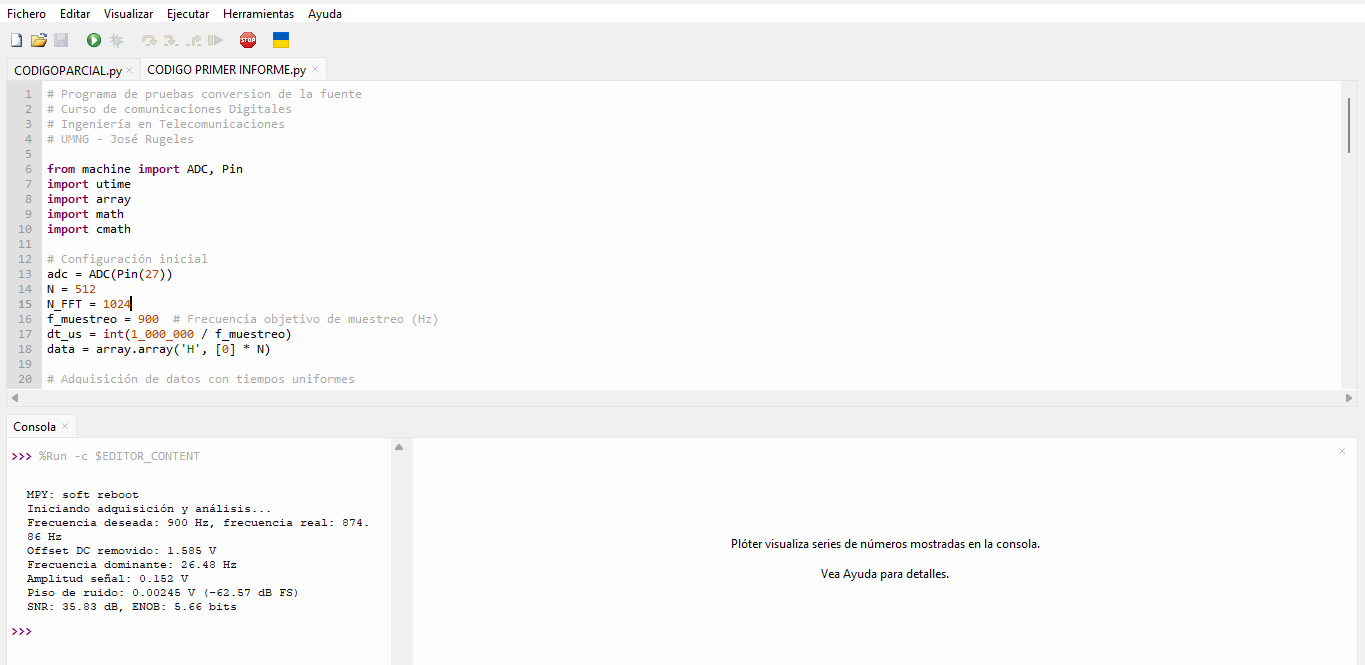
900con256



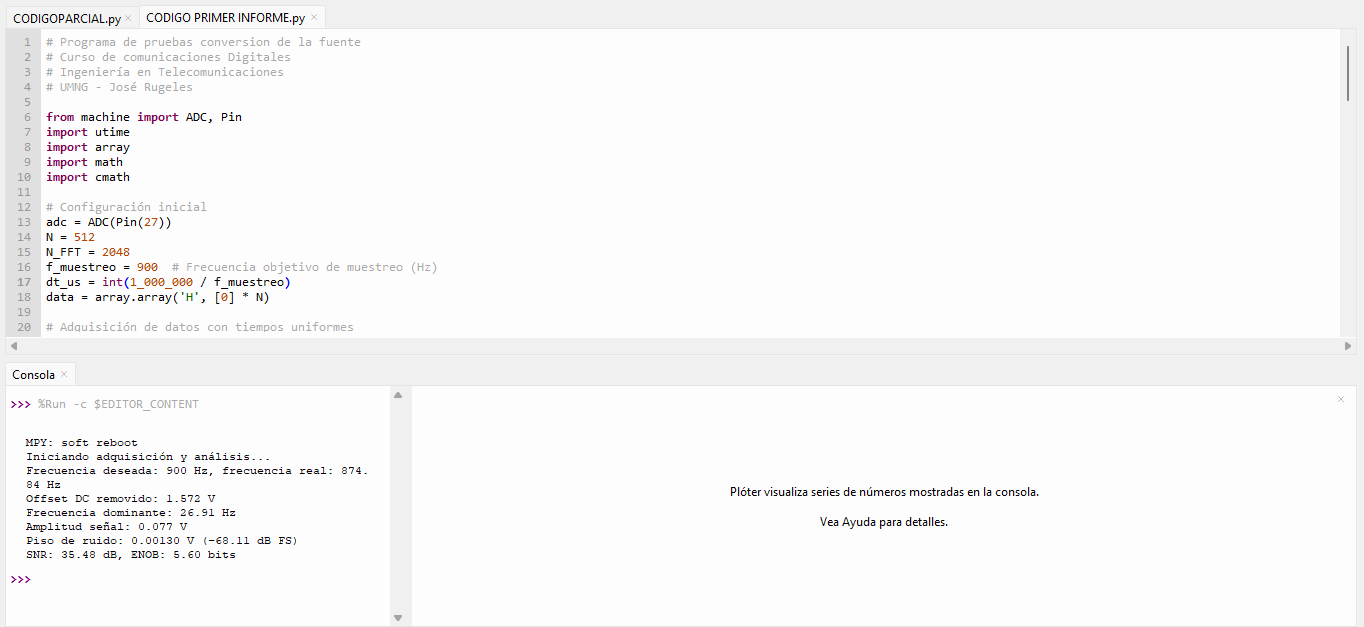
900con512



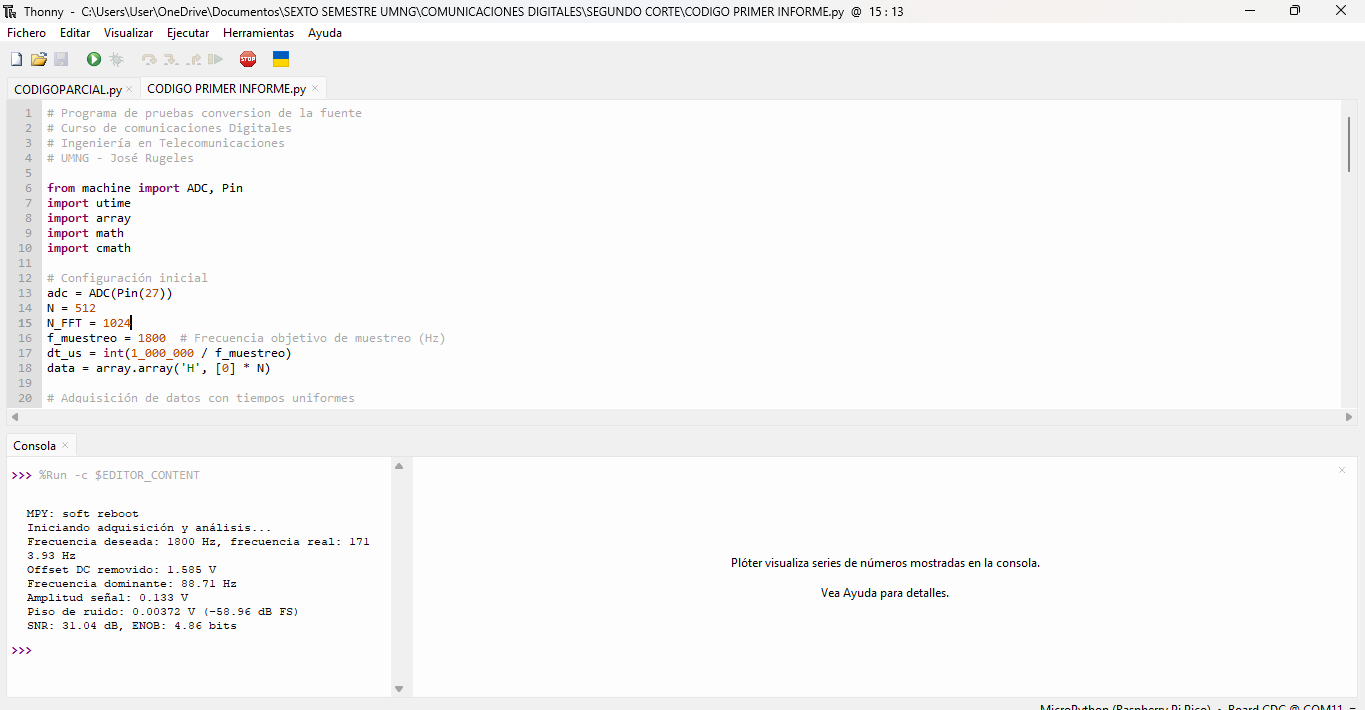
900con1024



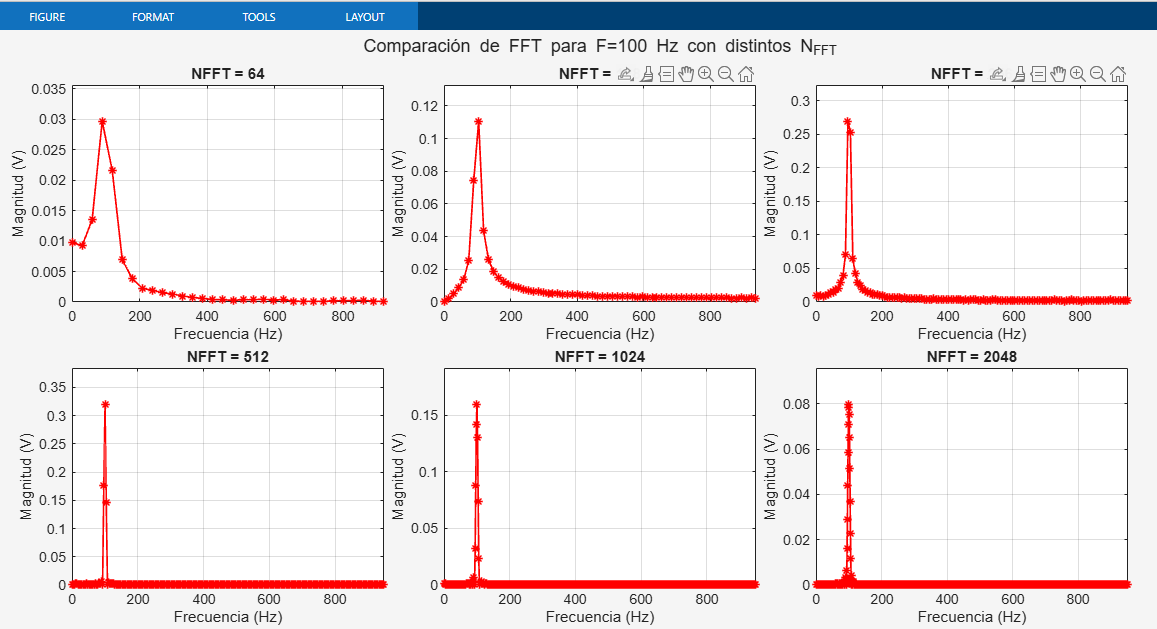
900con2048

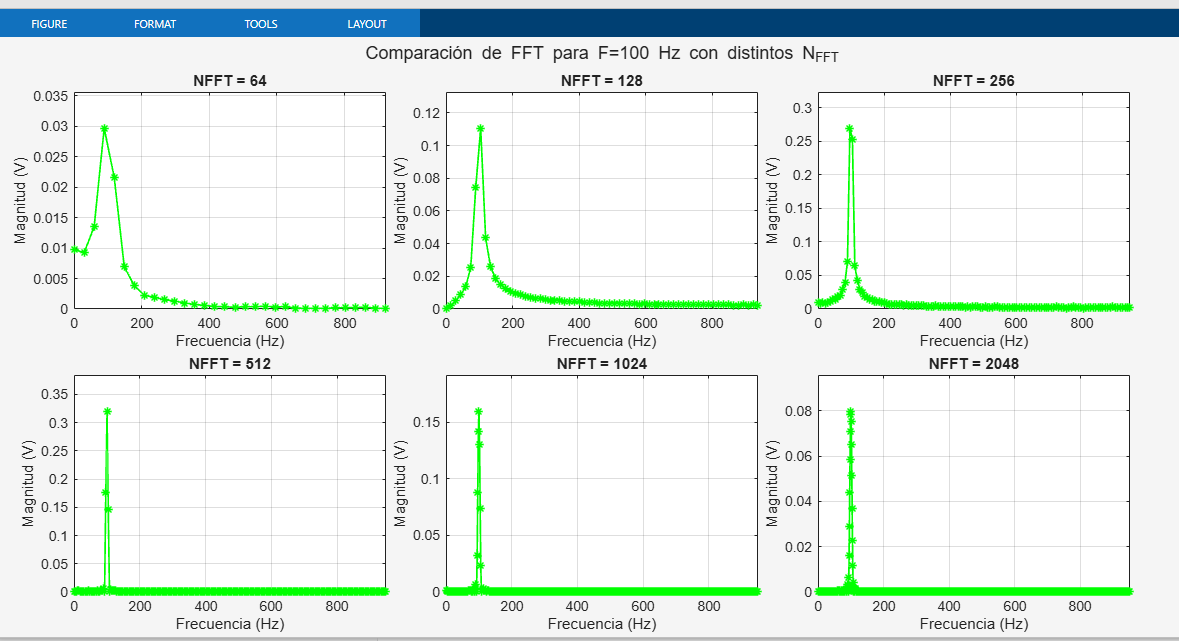
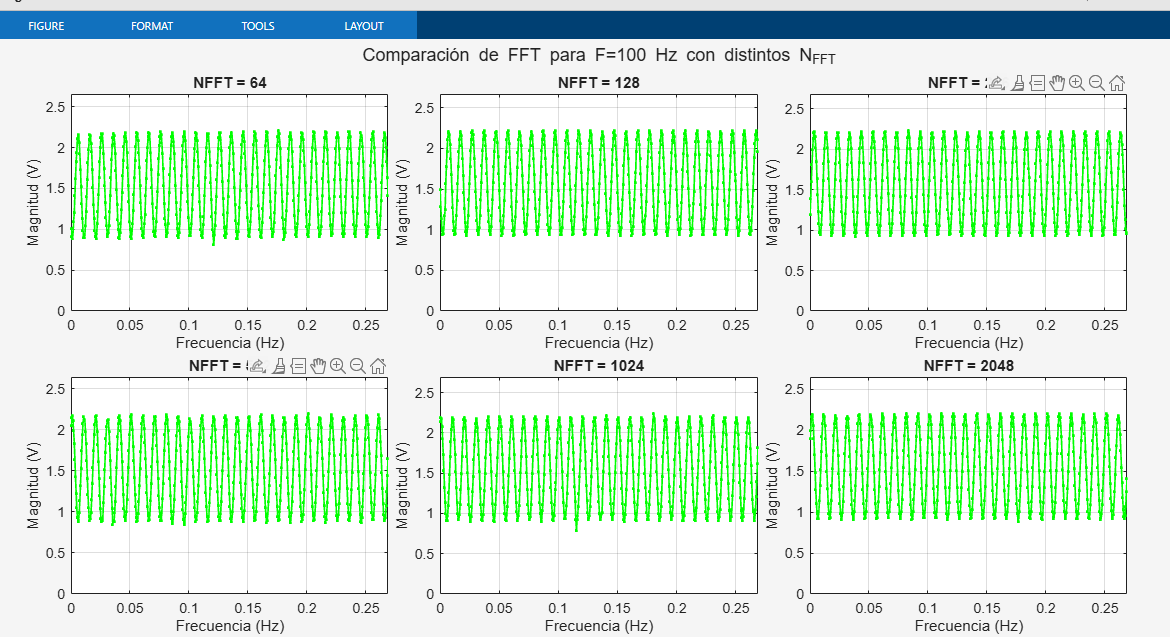


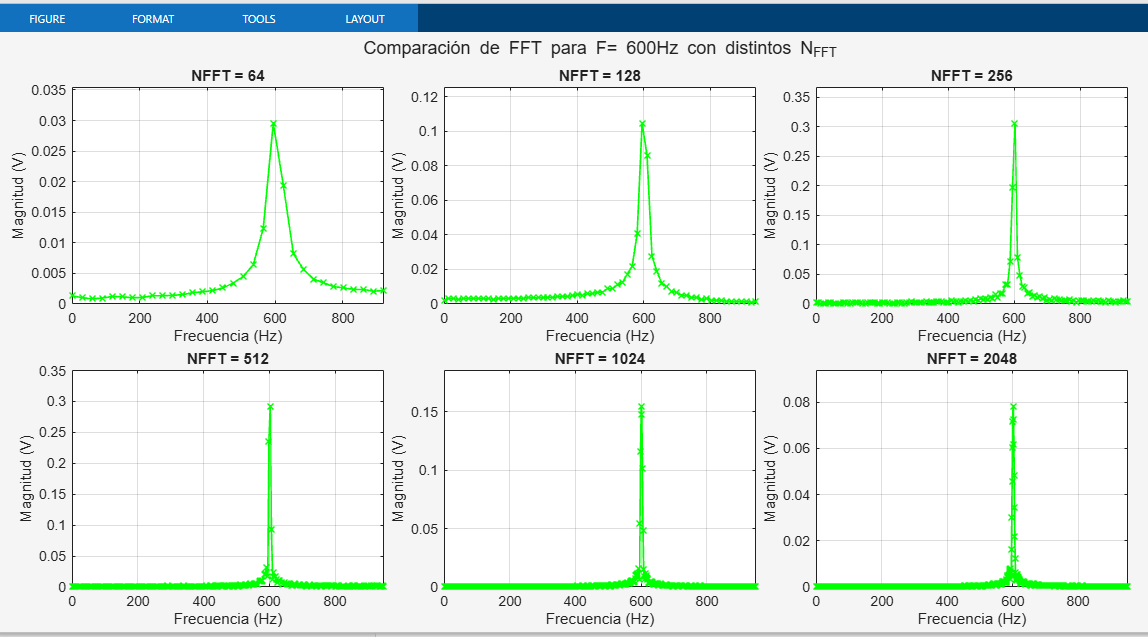
1800CON1024

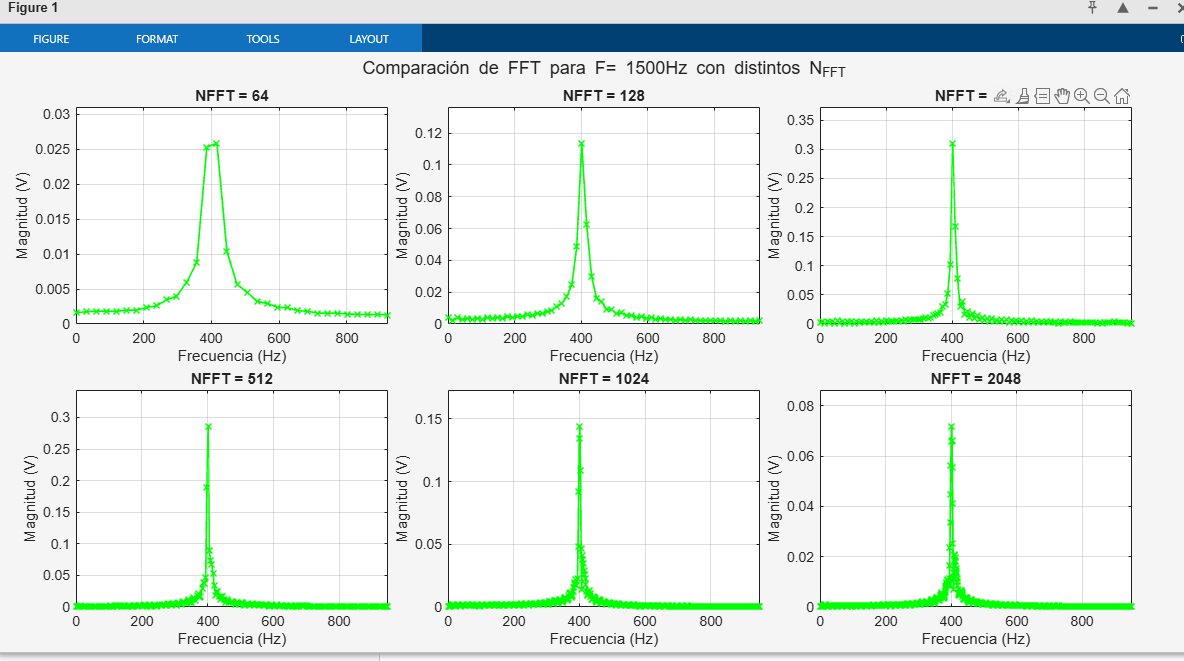


100 HZ

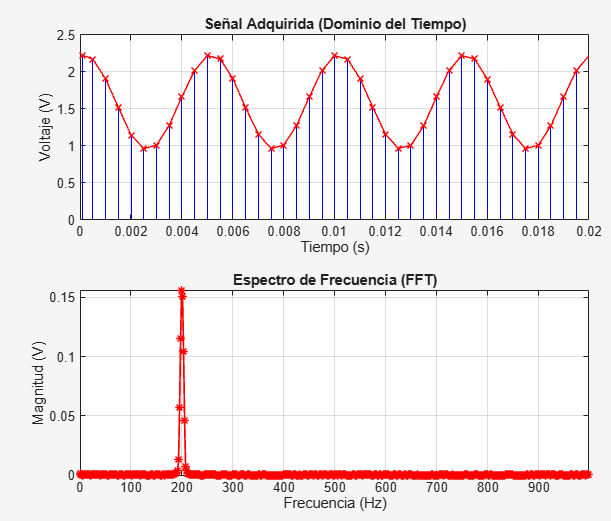






Grafica datos exportados Con el Otro código parte 2



**Código para graficar una sola señal en Matlab:**

% Script para visualizar datos desde MicroPython

% Archivos requeridos: 'muestras.txt' y 'fft.txt'

clc; clear; close all;

% --- Cargar datos de las muestras ---

muestras = readmatrix('muestras.txt'); % Cargar archivo (omite encabezado automáticamente)

t = muestras(:,1); % Tiempo (s)

V = muestras(:,2); % Voltaje (V)

% --- Graficar señal en el dominio del tiempo ---

subplot(2,1,1)

stem(t, V, 'b','Marker','none'); % stem para visualización discreta

hold on;

plot(t, V, 'r','Marker','x','LineWidth',1); % plot para suavizar

grid on;

xlabel('Tiempo (s)');

ylabel('Voltaje (V)');

title('Señal Adquirida (Dominio del Tiempo)');

xlim([0 0.02]);

% --- Cargar datos de la FFT ---

fft1 = readmatrix('fft.txt'); % Cargar archivo

f = fft1(:,1); % Frecuencia (Hz)

A = fft1(:,2); % Magnitud (V)

% --- Graficar espectro de frecuencia ---

subplot(2,1,2)

plot(f, A, 'r','Marker','\*','LineWidth',1.2);

grid on;

xlabel('Frecuencia (Hz)');

ylabel('Magnitud (V)');

title('Espectro de Frecuencia (FFT)');

xlim([0 max(f)]);

**Código para graficar las 6 señales con los diferentes N-FFT:**

clc; clear; close all;

% --- Archivos FFT a graficar ---

files = { ...

'fft100\_64.txt', ...

'fft100\_128.txt', ...

'fft100\_256.txt', ...

'fft100\_512.txt', ...

'fft100\_1024.txt', ...

'fft100\_2048.txt' };

nffts = [64 128 256 512 1024 2048]; % Para los títulos

% Layout 2x3

tiledlayout(2,3,'Padding','compact','TileSpacing','compact');

set(gcf,'Position',[100 100 1200 600]); % ventana más grande

for k = 1:numel(files)

nexttile;

if ~isfile(files{k})

text(0.5,0.5,sprintf('No se encontró: %s',files{k}), ...

'HorizontalAlignment','center'); axis off;

continue;

end

% --- Leer archivo (ignorar encabezado) ---

fid = fopen(files{k}, 'r');

fgetl(fid); % descarta la primera línea

data = textscan(fid, '%f %f');

fclose(fid);

f = data{1}; % Frecuencia (Hz)

A = data{2}; % Magnitud (V)

% --- Graficar cada FFT ---

plot(f, A, 'g','Marker','\*','LineWidth',1.2); grid on;

xlabel('Frecuencia (Hz)');

ylabel('Magnitud (V)');

title(sprintf('NFFT = %d', nffts(k)));

% Ajustes de ejes

xlim([0 max(f)]); % todo el espectro calculado

ylim([0 max(A)\*1.2]); % autoescala en Y con margen

end

sgtitle('Comparación de FFT para F=100 Hz con distintos N\_{FFT}');

**código para Tonny, segundo punto**; **parte2\_codigo estable\_.py**

% Script para visualizar datos desde MicroPython

% Archivos requeridos: 'muestras.txt' y 'fft.txt'

clc; clear; close all;

% --- Cargar datos de las muestras ---

muestras = readmatrix('muestras.txt'); % Cargar archivo (omite encabezado automáticamente)

t = muestras(:,1); % Tiempo (s)

V = muestras(:,2); % Voltaje (V)

% --- Graficar señal en el dominio del tiempo ---

subplot(2,1,1)

stem(t, V, 'b','Marker','none'); % stem para visualización discreta

hold on;

plot(t, V, 'r','Marker','x','LineWidth',1); % plot para suavizar

grid on;

xlabel('Tiempo (s)');

ylabel('Voltaje (V)');

title('Señal Adquirida (Dominio del Tiempo)');

xlim([0 0.02]);

% --- Cargar datos de la FFT ---

fft1 = readmatrix('fft.txt'); % Cargar archivo

f = fft1(:,1); % Frecuencia (Hz)

A = fft1(:,2); % Magnitud (V)

% --- Graficar espectro de frecuencia ---

subplot(2,1,2)

plot(f, A, 'r','Marker','\*','LineWidth',1.2);

grid on;

xlabel('Frecuencia (Hz)');

ylabel('Magnitud (V)');

title('Espectro de Frecuencia (FFT)');

xlim([0 max(f)]);