

IDENTIFICAÇÃO

Dia da Semana	Data	Hora	Grupo	Nº Alunos			
5.ª Feira	12/11/2020	9:00-11:00	31	89910	–	94192	–

FIGURA PARTE I, Situação I

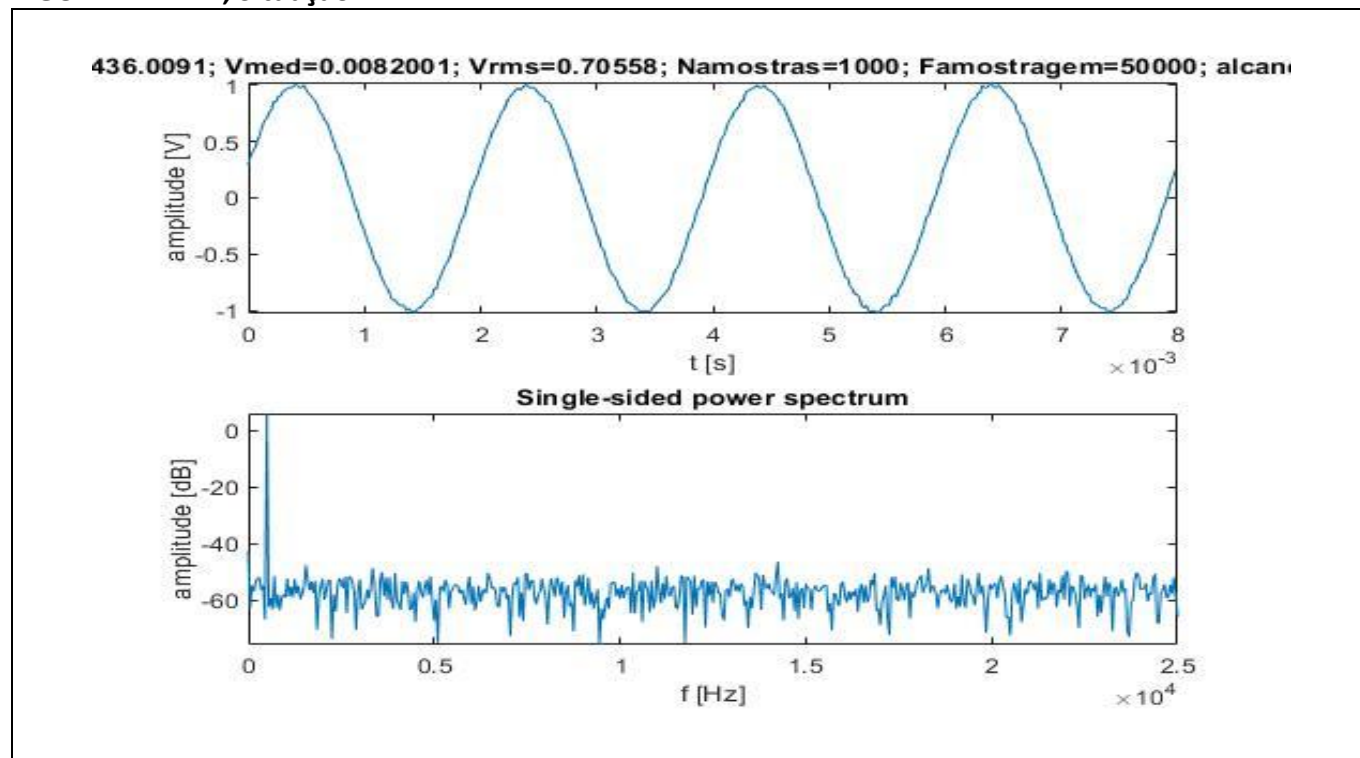


FIGURA PARTE I, Situação II

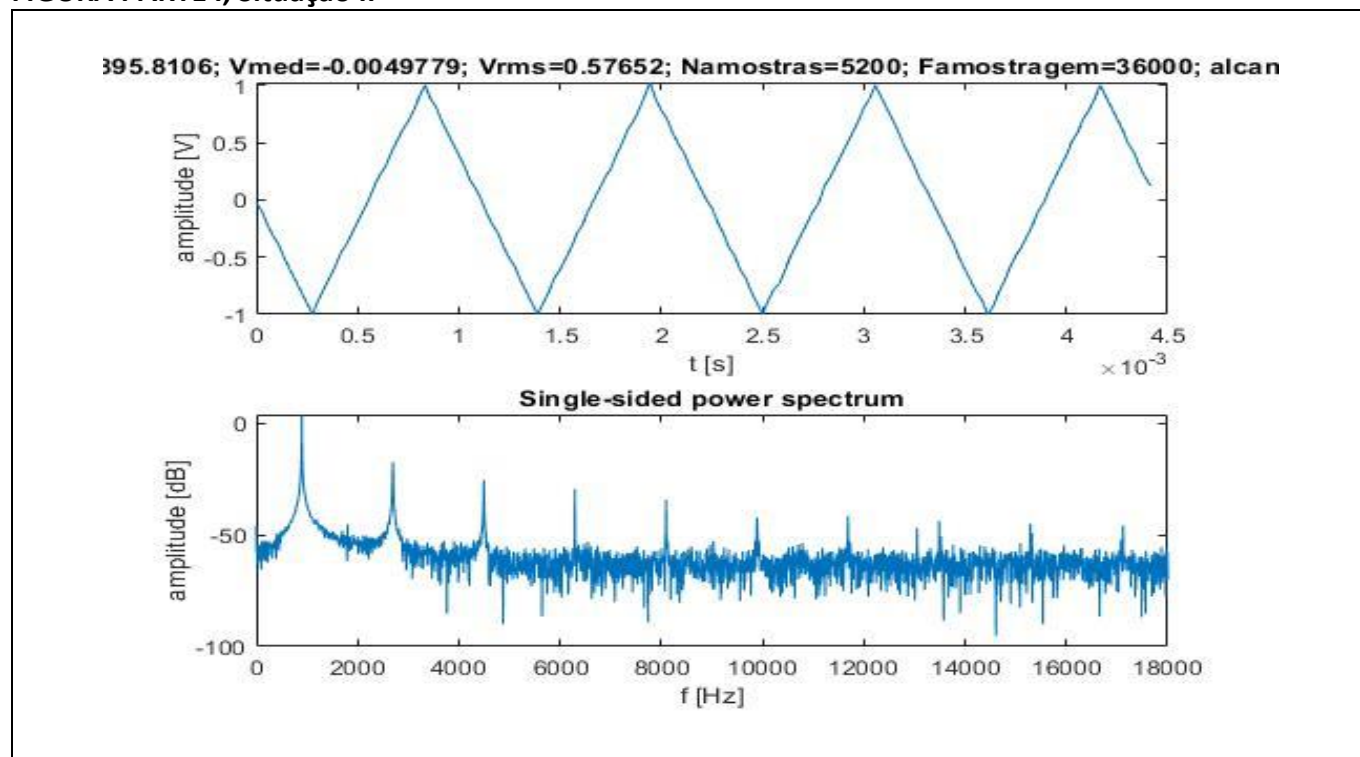


FIGURA PARTE I, Situação III

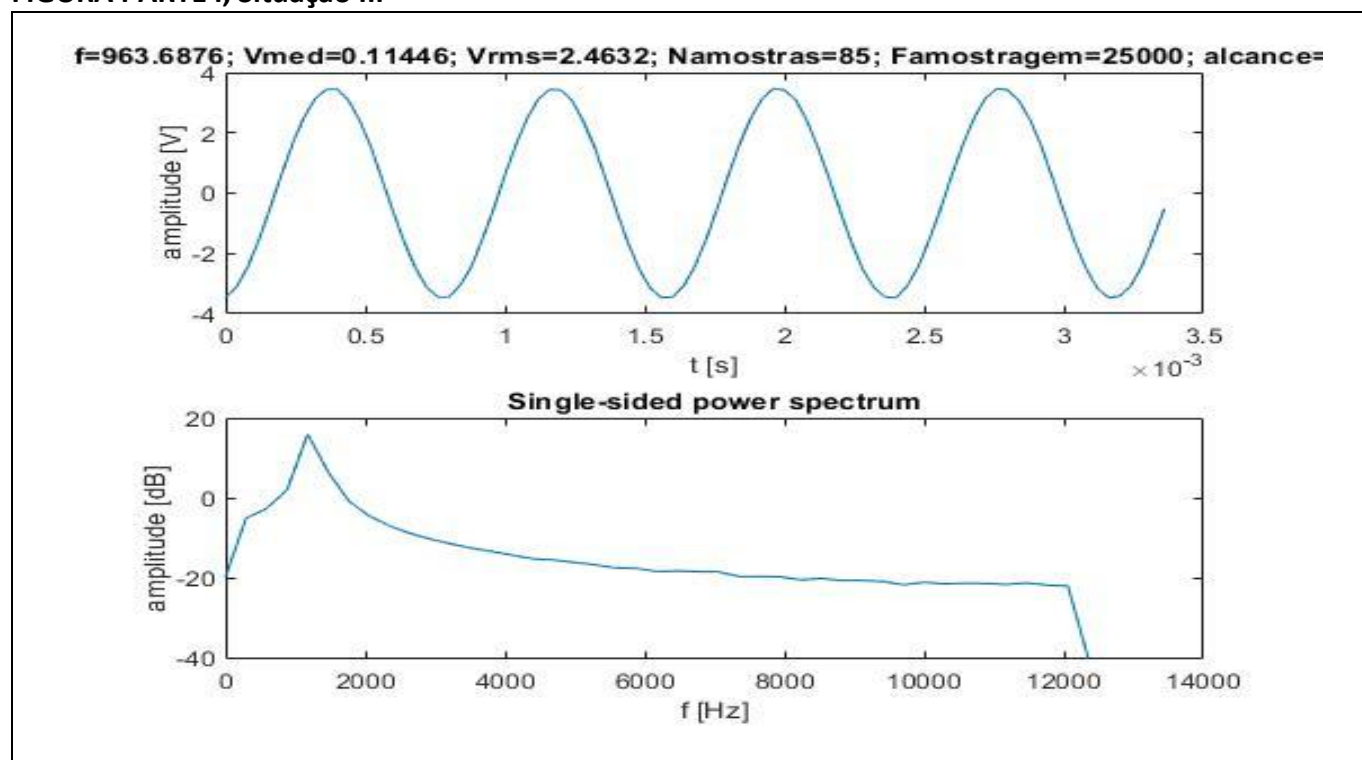


FIGURA PARTE I, Situação IV

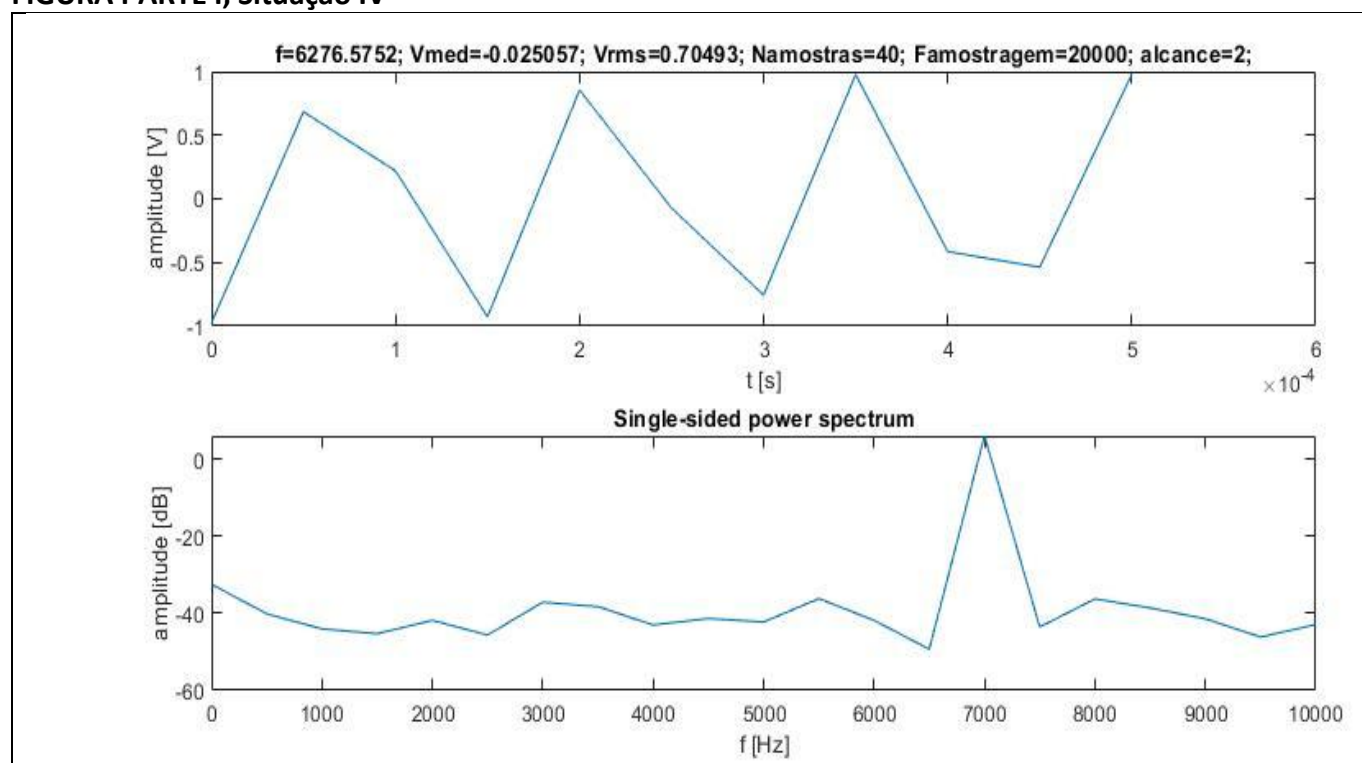
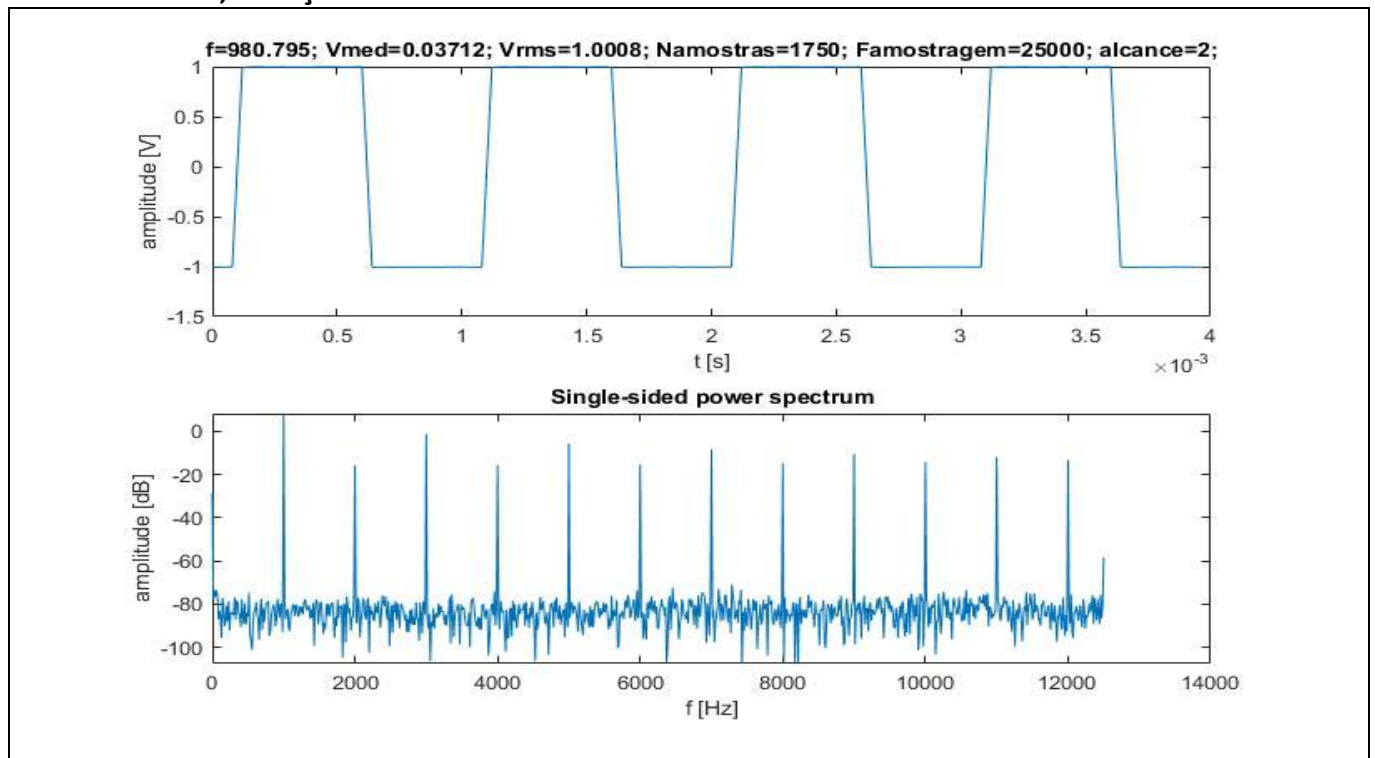


FIGURA PARTE I, Situação V



CÓDIGO PARTE1

(sugestão, fazer *paste as text* para manter tamanho e formatação)

```
%% Instrumentacao e Medidas - Ano Letivo 2020/2021
%% Laboratorio 2 - Aquisicao de Sinais - DAQ_parte1

%% Autores
%% Joao Pedro Fino n_89910
%% Duarte Oliveira n_94192

clear;
close all;

%numero de amostras
n_am = 1750;

%frequencia de amostragem
f_am = 25000;

%desfasamento em frequencia
dF = f_am/n_am;

%desfasamento no tempo
dT = 1/f_am;

%alcance
alcance = 2;

%para gravar no workspace
savefile='log_DAQ_partela_situacao1';

% %utilizar dados do ficheiro gravado
% m = matfile('log_DAQ_partela_situacao1.mat');
% aux = m.data;
% data = aux;

%% Funcoes para o AI

Duration = n_am/f_am;
AI = daq.createSession('ni');
AI.Rate = f_am;
AI.DurationInSeconds = Duration;
ai0 = AI.addAnalogInputChannel('Dev1', 'ai0', 'Voltage');

%Alcance de tensao pedido
```

```

ai0.Range = [-alcance, alcance]
AI.NumberOfScans = n_am;
data = startForeground(AI);

%% Espectro

S = abs(fft(data)/n_am);
G_final = S(1:n_am/2+1);
G_final(2:end-1) = 2*G_final(2:end-1);

%% Calculo das Harmonicas

[val, index] = max(G_final);
index = index - 1;
freq = index*dF;

harmonic = zeros([3 11]);
for n = 1:11

    if index*n < length(G_final)
        harmonic(1, n) = index*dF*n;
        harmonic(2, n) = G_final(index*n+1);
        harmonic(3, n) = db(G_final(index*n+1));
    end
end

%% Estimativa da frequencia

sum_d = 0;
sum_n = 0;
est = index-1;

for n = est-1:est+1

    sum_n = sum_n + ( G_final(n)*((n)*dF) );
    sum_d = sum_d + G_final(n);
end
f_estimated = sum_n/sum_d;
Ts = floor(n_am/(f_am/f_estimated));
npontos = floor(f_am/f_estimated*Ts);

%% Espectro Estimado

S_estimado = abs(fft(data(1:npontos))/n_am);
G_estim_final = S_estimado(1:n_am/2+1);
G_estim_final(2:end-1) = 2*G_final(2:end-1);

%% Estimativa Valor Médio

average = 0;
for n = 1:npontos
    average = average + data(n);
end

%tensao media
%soma dos valores medidos/ n_valores medidos
vmedEST = average / npontos;

%% Valor Eficaz de RMS estimado

vrms=0;
N = length(data);
for n=1:N
    vrms = vrms + power(data(n) , 2 );
end

%valor eficaz = raiz da soma dos valores medidos ao quadrado/n_medicoes
vrms=vrms/N;
vrms=sqrt(vrms);

%% THD - Total Harmonical Distortion

thd = (sqrt(sum(power(harmonic(2, 2:length(harmonic)),2)))/harmonic(2,1));

%% Representacao Grafica

t = 0:dT:4/freq-dT;
f = 0:dF:(f_am)/2;

```

```

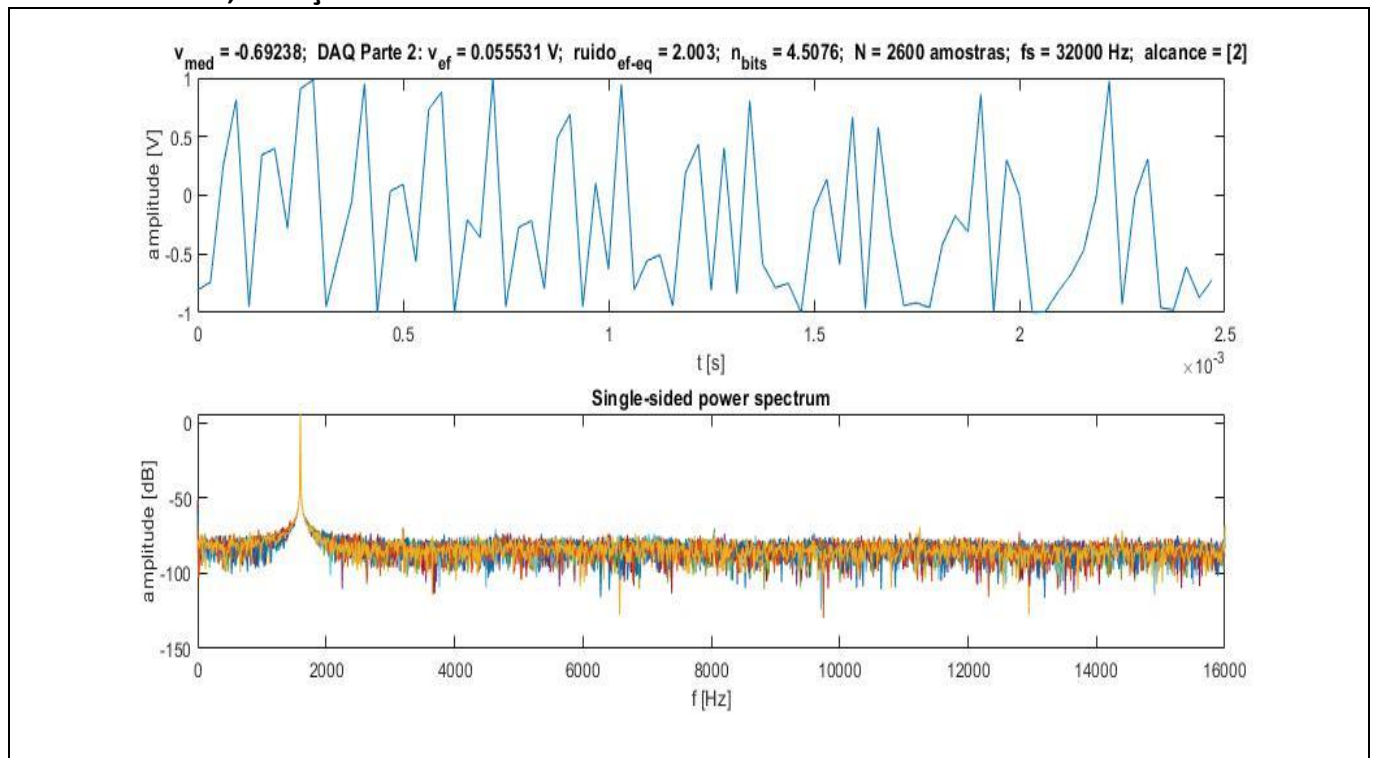
figure(1);
subplot(2,1,1);
plot( t, data(1:length(t)) );
text = ['f=' , num2str(f_estimated) ,'; Vmed=' , num2str(vmedEST) ,'; Vrms=' , num2str(vrms) ,';
Namostras=' ,num2str(n_am) ,'; Famostragem=' ,num2str(f_am), ' ; alcance=' ,num2str(alcance) ,';'];
title( text );
ylabel('amplitude [V]');
xlabel('t [s]');

subplot(2,1,2);
semilogx(f, db(G_estim_final));
plot(f, db(G_estim_final));
ylabel('amplitude [dB]');
xlabel('f [Hz]');
title('Single-sided power spectrum');

harmonic(1,:)
harmonic(2,:)
harmonic(3,:)
save(savefile);

```

FIGURA PARTE II, Situação VI



CÓDIGO PARTE2

(sugestão, fazer *paste as text* para manter tamanho e formatação)

```
%% Instrumentacao e Medidas - Ano Letivo 2020/2021
%% Laboratorio 2 - Aquisicao de Sinais - DAQ_parte2

%% Autores
%% Joao Pedro Fino n_89910
%% Duarte Oliveira n_94192

clear;
close all;

%numero de amostras
n_am = 2600;

%frequencia de amostragem
f_am = 32000;

%desfasamento em frequencia
dF = f_am/n_am;

%desfasamento no tempo
dT = 1/f_am;

%alcance
alcance = 2;

%para gravar no workspace
savefile='log_DAQ_parte2_situacao1';

%numero de repeticoes
nReps = 10;

% m = matfile('log_DAQ_parte2_situacao1.mat'); %utilização dados do ficheiro
%
% data(1,:) = m.data;
% data(2,:) = m.data;

%% Usando o AI

Duration = n_am/f_am;
AI = daq.createSession('ni');
AI.Rate = f_am;
AI.DurationInSeconds = Duration;
ai0 = AI.addAnalogInputChannel('Dev1', 'ai0', 'Voltage');

%Alcance de tensao
```

```

ai0.Range = [-alcance, alcance];
AI.NumberOfScans = n_am;
%data = startForeground(AI);
%[data, t]= startForeground(AI); % iniciar aquisição

for (i = 1:nReps)
    [datax, tx]= startForeground(AI); % iniciar aquisição
    data(i,:) = datax;
    t(i,:) = tx;
end

%% Espectro

for (i = 1:nReps)
    S(i,:) = abs(fft(data(i,:))/n_am);
    G_final(i,:) = S(i, 1:n_am/2+1);
    G_final(i, 2:end-1) = 2*G_final(i, 2:end-1);
end

G_fin_med = mean(G_final.',2);

%% Calculo das Harmonicas

[val, index] = max(G_fin_med);
index = index - 1;
freq = index*dF;

harmonic = zeros([3 11]);
for n = 1:11

    if index*n < length(G_fin_med)
        harmonic(1, n) = index*dF*n;
        harmonic(2, n) = G_fin_med(index*n+1);
        harmonic(3, n) = db(G_fin_med(index*n+1));
    end
end;

%% Estimativa da frequencia

est = index-1;
sumn = 0;
sumd = 0;

for n = est-1:est+1

    sumn = sumn + ( G_fin_med(n)*((n)*dF) );
    sumd = sumd + G_fin_med(n);
end
f_estimated = sumn/sumd;
Ts = floor(n_am/(f_am/f_estimated) );
npontos = floor(f_am/f_estimated*Ts);

%% Espectro Estimado

for (i = 1:nReps)
    S_estimado(i,:) = abs(fft(data(i,1:npontos))/n_am);
    G_estim_final(i,:) = S_estimado(i, 1:n_am/2+1);
    G_estim_final(i, 2:end-1) = 2*G_final(i, 2:end-1);
end
G_fin_med_estim = mean(G_estim_final.',2);

%% Valor Medio

avg = 0
for (i = 1:npontos)
    avg = avg + data (n);
end
%tensao media
vmedEST = avg / npontos;

%% Valor Eficaz RMS estimado

vrms=0;
N=length(G_fin_med_estim);
for n=1:N
    vrms = vrms + power( G_fin_med_estim(n) , 2 );
end

vrms=vrms/N;
%Valor eficaz

```

```

vrms=sqrt(vrms);

%% Ruido Eficaz

%Estimativa do ruido eficaz equivalente

somatorio = 0;
for (i = 1:(npontos/2-1))
    somatorio = (somatorio + 10^(20*log10(G_fin_med_estim(i))/10));
end
n_rms = sqrt(somatorio);

%% Numero de bits da placa

%potencia da fundamental
pfundamental = 20*log(max(2*G_fin_med_estim));

%potência total do sinal restante
ptotal_rest = 20*log(sum(2*G_fin_med_estim)) - 20*log(max(2*G_fin_med_estim));

% SINAD (Signal-to-noise and distortion ratio)
Sinad = pfundamental + ptotal_rest;

% ENOB (Effective Number Of Bits)
n_bits = (Sinad - 1.76)/6.02;

%% Representacao grafica
%comando subplot

t = 0:dT:4/freq-dT;
f = 0:dF:(f_am)/2;

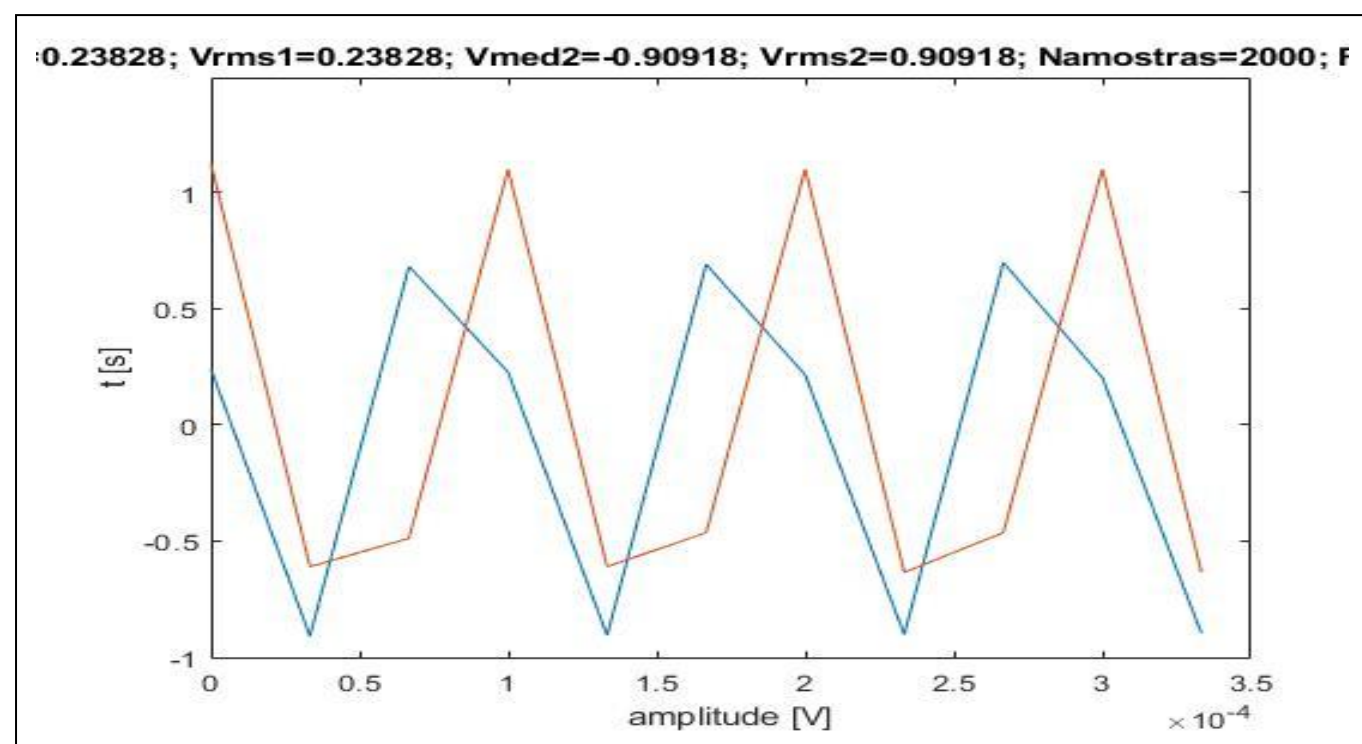
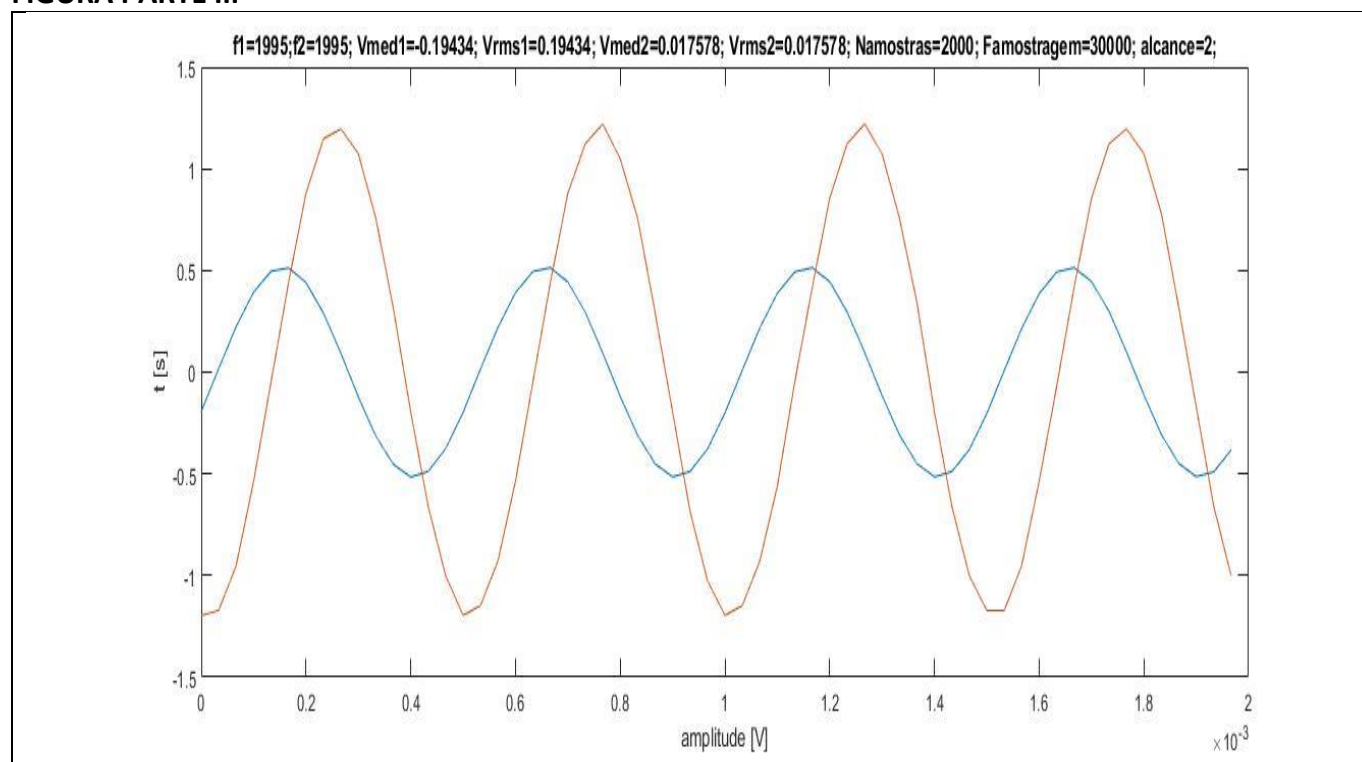
figure(1);
subplot(2,1,1);
plot( t, data(1:length(t)) );
text = ['v_m_e_d = ',num2str(vmedEST) ,'; DAQ Parte 2: v_e_f = ',num2str(vrms),' V; ruido_e_f_-_e_q = ',num2str(n_rms),'; n_b_i_t_s = ',num2str(n_bits),'; N = ',num2str(n_am),' amostras; fs = ',num2str(f_am),' Hz; alcance = ',num2str(alcance),'];'];
title( text );
ylabel('amplitude [V]');
xlabel('t [s]');

subplot(2,1,2);
semilogx(f, db(G_estim_final));
plot( f, db(G_estim_final));
ylabel('amplitude [dB]');
xlabel('f [Hz]');
title('Single-sided power spectrum');

% harmonic(1,:)
% harmonic(2,:)
% harmonic(3,:)
save(savefile);

```


FIGURA PARTE III



CÓDIGO PARTE3

(sugestão, fazer *paste as text* para manter tamanho e formatação)

```
%% Instrumentacao e Medidas - Ano Letivo 2020/2021
%% Laboratorio 2 - Aquisicao de Sinais - DAQ_parte3

%% Autores
%% Joao Pedro Fino n_89910
%% Duarte Oliveira n_94192

clear;
close all;

%numero de amostras
n_am = 2000;

%frequencia de amostragem
f_am = 30000;

%desfasamento em frequencia
dF = f_am/n_am;

%desfasamento no tempo
dT = 1/f_am;

%alcance
alc = 2;

%para gravar no workspace
savefile='log_DAQ_parte3_situacao1';

Duration = n_am/f_am;

%% Usando o AI:

AI = daq.createSession('ni');
AI.Rate = f_am;
AI.DurationInSeconds = Duration;
ai0 = AI.addAnalogInputChannel('Dev1', 'ai0', 'Voltage');
ai1 = AI.addAnalogInputChannel('Dev1', 'ai1', 'Voltage');

%Alcance de tensao

ai0.Range = [-alc, alc];
%AI.NumberOfScans = 4096;
data = startForeground(AI);

%% Espectro

step1 = abs(fft(data(:, 1))/n_am);
step2 = step1(1:n_am/2+1);
step2(2:end-1) = 2*step2(2:end-1);

step4 = abs(fft(data(:, 2))/n_am);
step3 = step4(1:n_am/2+1);
step3(2:end-1) = 2*step3(2:end-1);

%% Frequencias
[val1, index1] = max(step2);
index1 = index1 - 1;
f1 = index1*dF;

[val2, index2] = max(step3);
index2 = index2 - 1;
f2 = index2*dF;

%% Tensoes medias e eficazes

vmed1=0;
vmed2=0;
vrms1=0;
vrms2=0;
N=length(data(1));

for n=1:N

    vmed1 = vmed1 + data(1, n);
    vrms1 = vrms1 + power( data(1, n) , 2 );
    vmed2 = vmed2 + data(2, n);
    vrms2 = vrms2 + power( data(2, n) , 2 );
```

```

end

vmed1=vmed1/N;
vrms1=vrms1/N;
vrms1=sqrt(vrms1);

vmed2=vmed2/N;
vrms2=vrms2/N;
vrms2=sqrt(vrms2);

%% Graficos

t1 = 0:dT:4/f1-dT;
t2 = 0:dT:4/f2-dT;

figure(1);

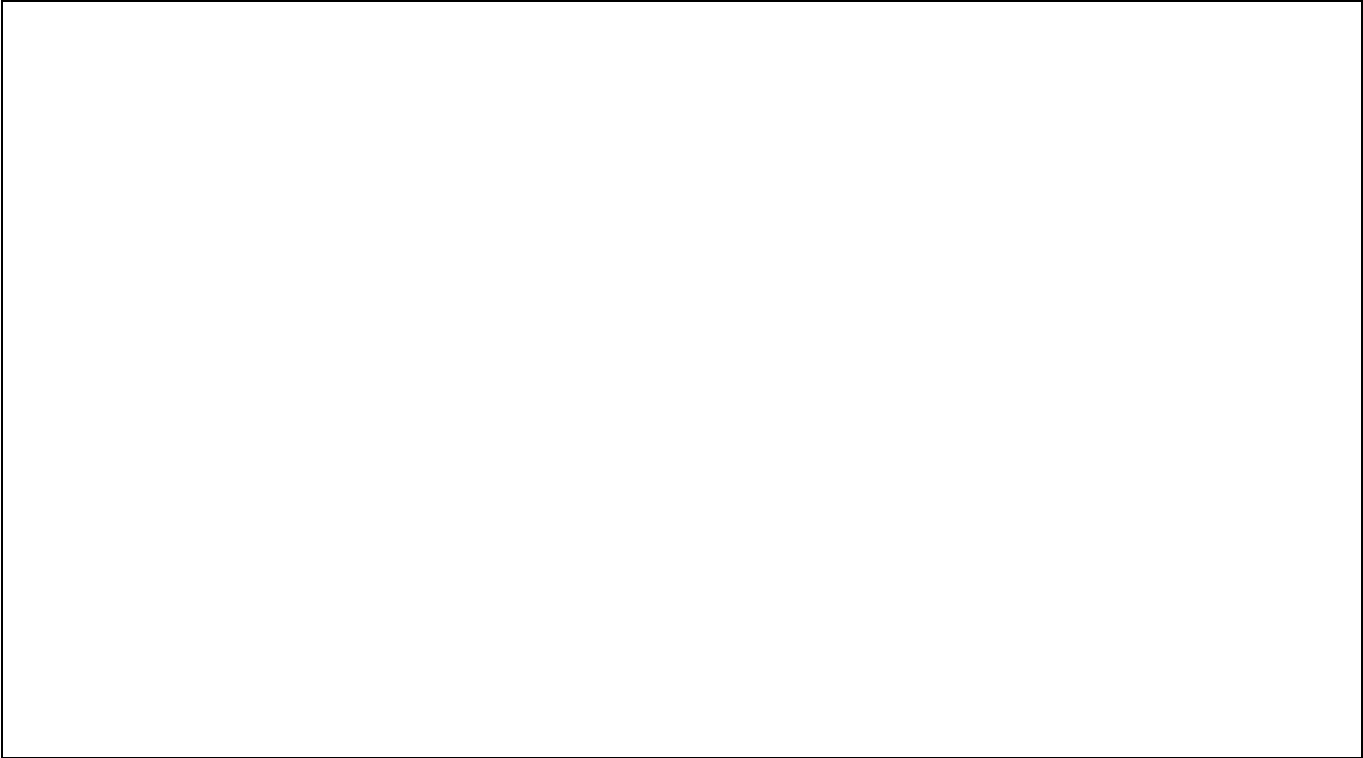
if length(t2) > length(t1)
    plot( t1, data(1:length(t1), 1), t1 , data(1:length(t1), 2 ));
    text = ['f1=' , num2str(f1) ,';f2=',num2str(f2),'; Vmed1=', num2str(vmed1) ,'; Vrms1=',
num2str(vrms1) ,'; Vmed2=', num2str(vmed2) ,'; Vrms2=', num2str(vrms2) ,'; Namostas=',num2str(n_am)
,'; Famostragem=',num2str(f_am), ' ; alcance=', num2str(alc) ,';'];
    title( text );
    xlabel('amplitude [V]');
    ylabel('t [s]');
else
    plot( t2, data(1:length(t2), 1), t2 , data(1:length(t2), 2));
    text = ['f1=' , num2str(f1) ,';f2=',num2str(f2),'; Vmed1=', num2str(vmed1) ,'; Vrms1=',
num2str(vrms1) ,'; Vmed2=', num2str(vmed2) ,'; Vrms2=', num2str(vrms2) ,'; Namostas=',num2str(n_am)
,'; Famostragem=',num2str(f_am), ' ; alcance=', num2str(alc) ,';'];
    title( text );
    xlabel('amplitude [V]');
    ylabel('t [s]');
end

save(savefile);

```

No fim da segunda sessão, devem converter este documento para PDF e submeter o PDF no fénix até 30 minutos depois do fim da segunda sessão. Não são aceites ficheiros por email.

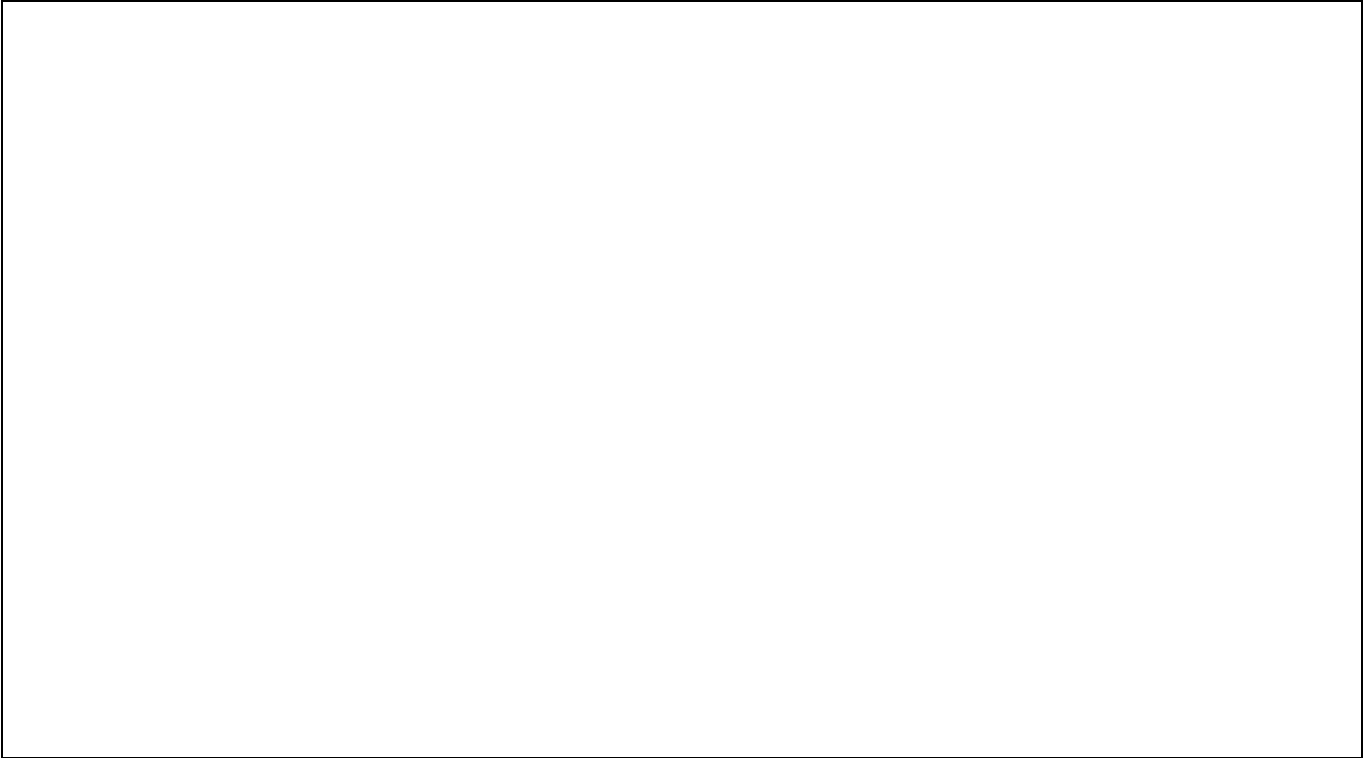
FIGURA FUNCIONALIDADE AVANÇADA 1



CÓDIGO FUNCIONALIDADE AVANÇADA 1



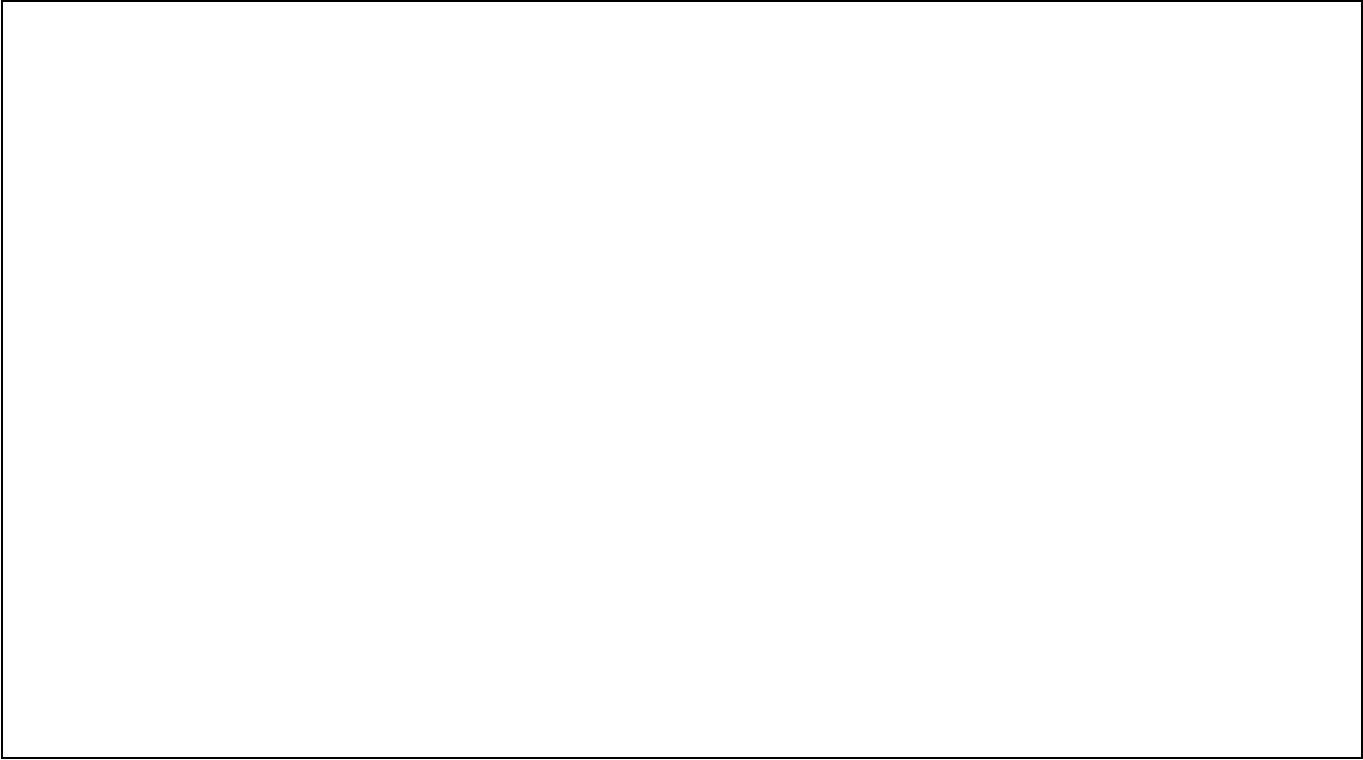
FIGURA FUNCIONALIDADE AVANÇADA 2



CÓDIGO FUNCIONALIDADE AVANÇADA 2



FIGURA FUNCIONALIDADE AVANÇADA 3



CÓDIGO FUNCIONALIDADE AVANÇADA 3

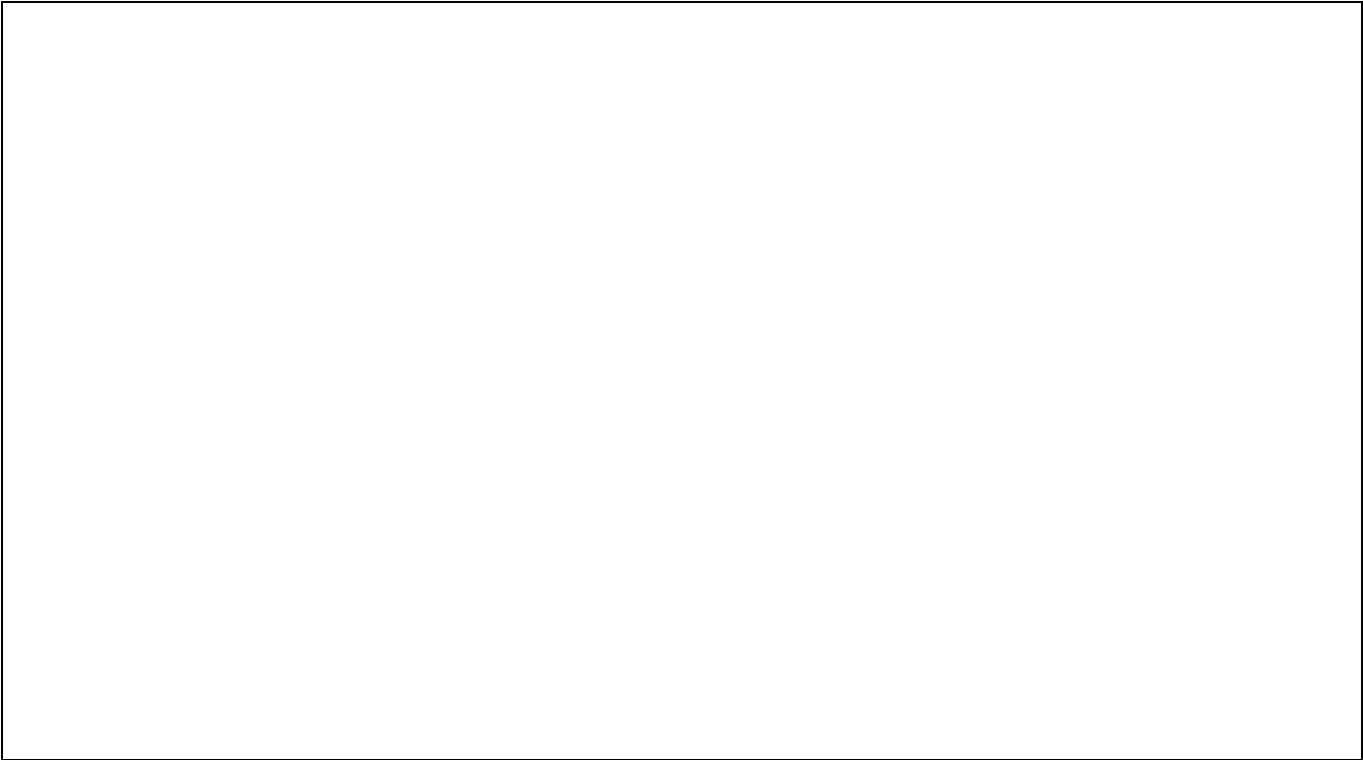
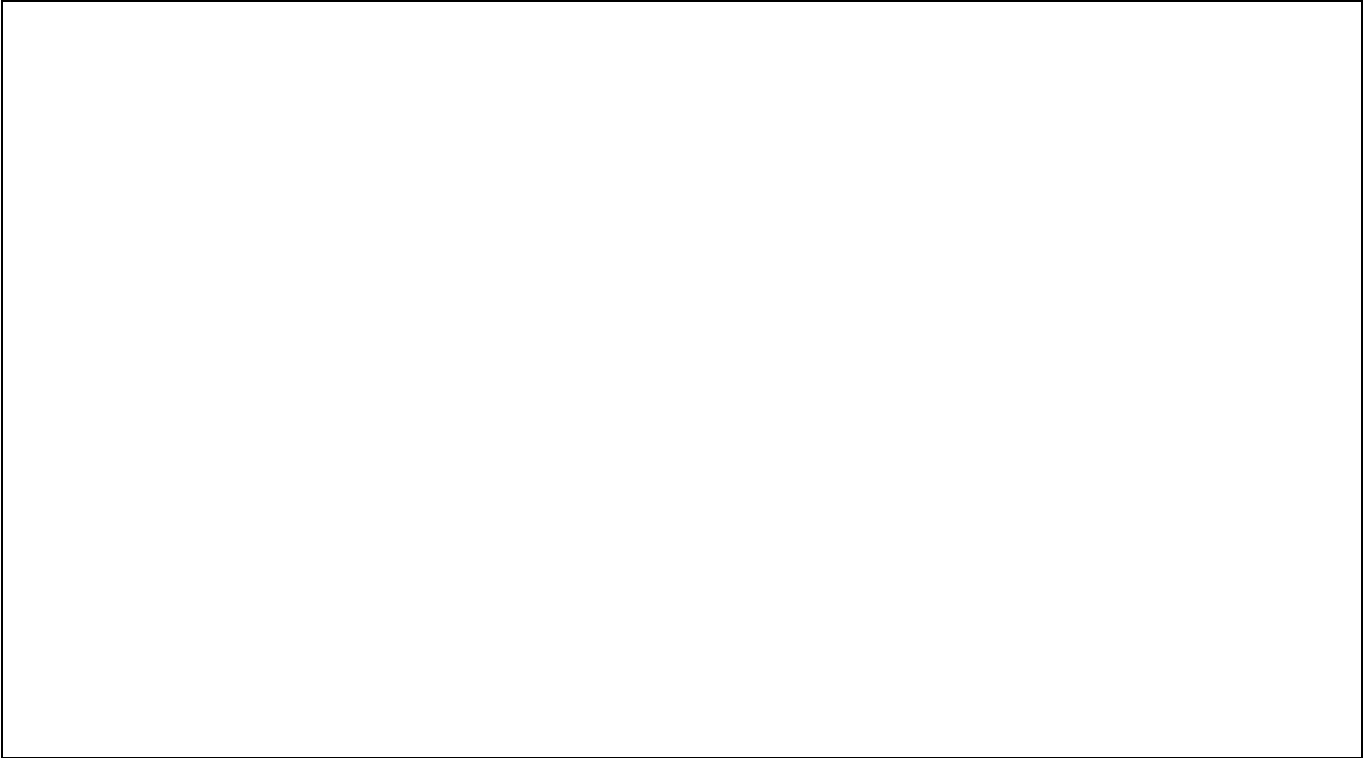


FIGURA FUNCIONALIDADE AVANÇADA 4



CÓDIGO FUNCIONALIDADE AVANÇADA 4

