



# Matlab Guide & Interface de Aquisição

**Rafael A. Mariano**



# Interface Gráfica ( *GUI* )

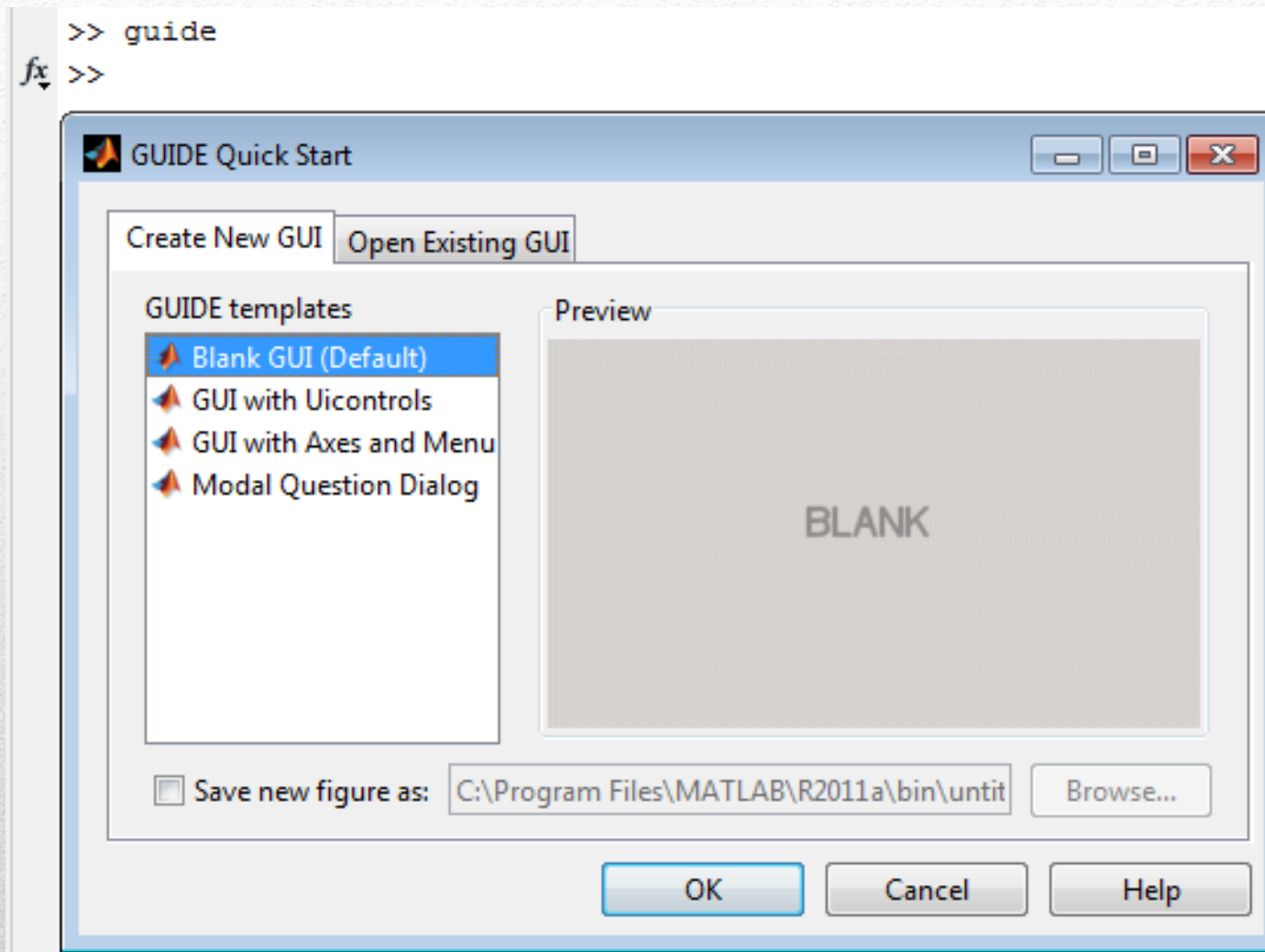
## *" Grafical User Interface "*

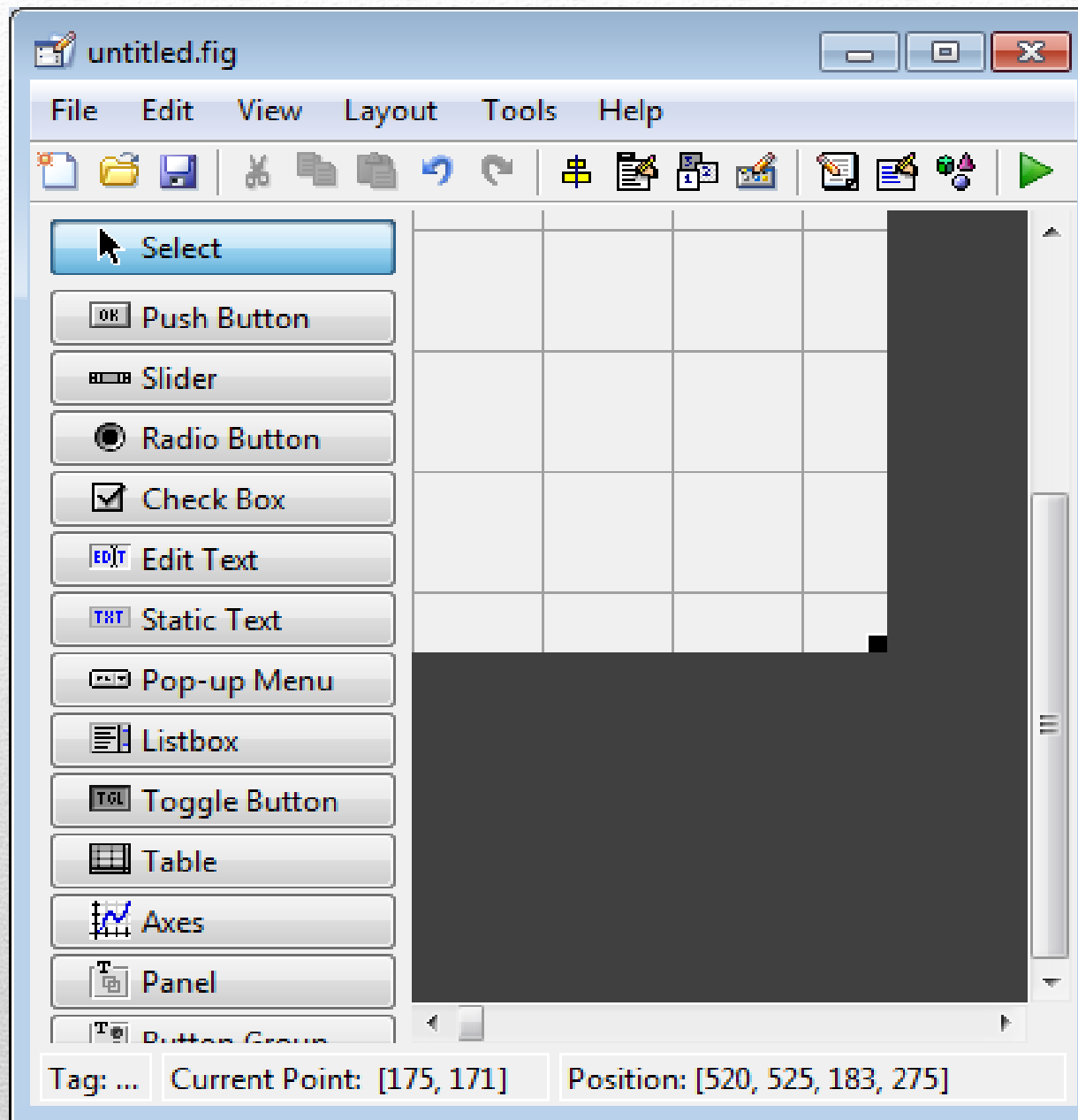
# GUIDE

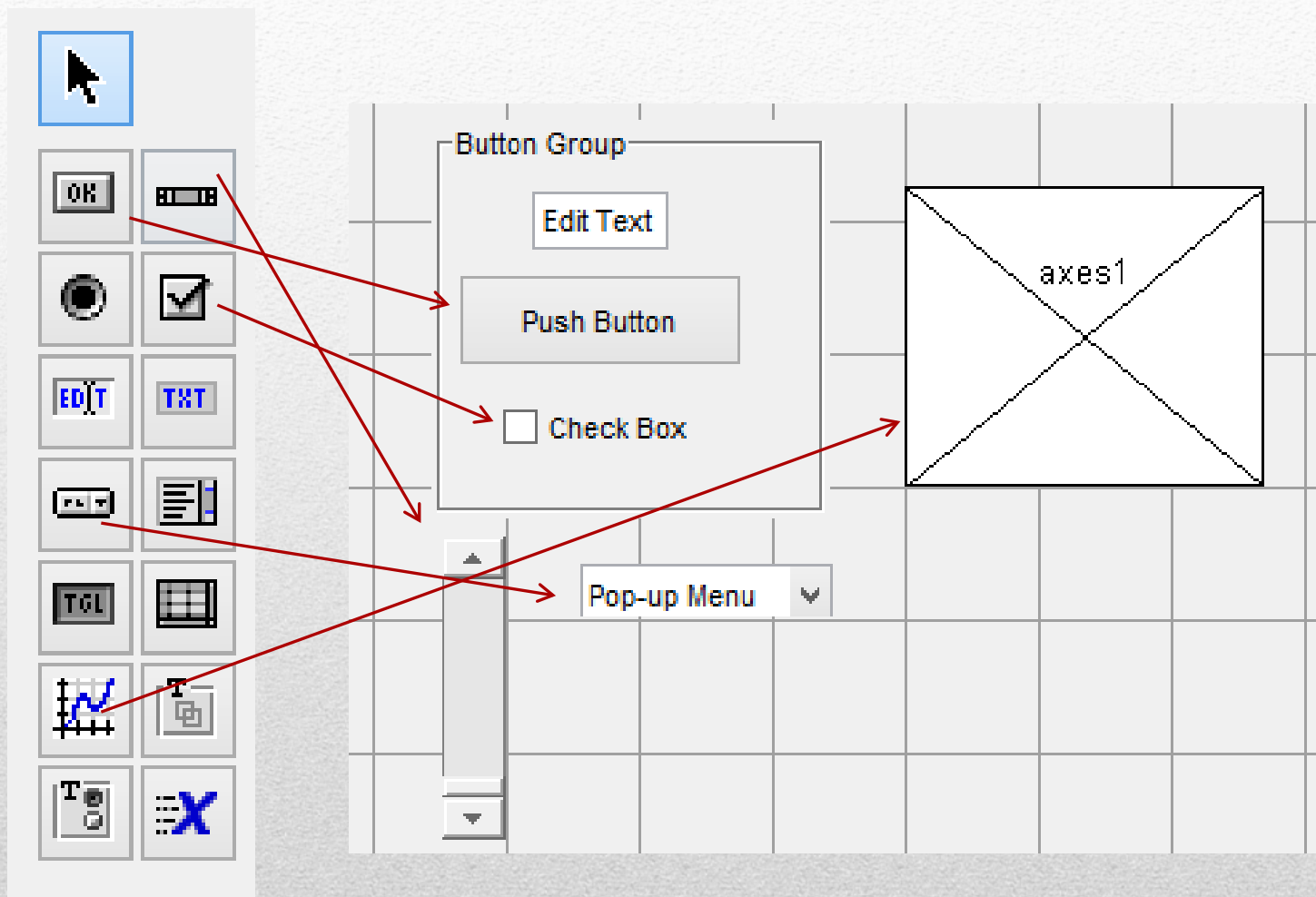


## *Graphical User Interface Development Environment*

<http://www.mathworks.com/discovery/matlab-gui.html>





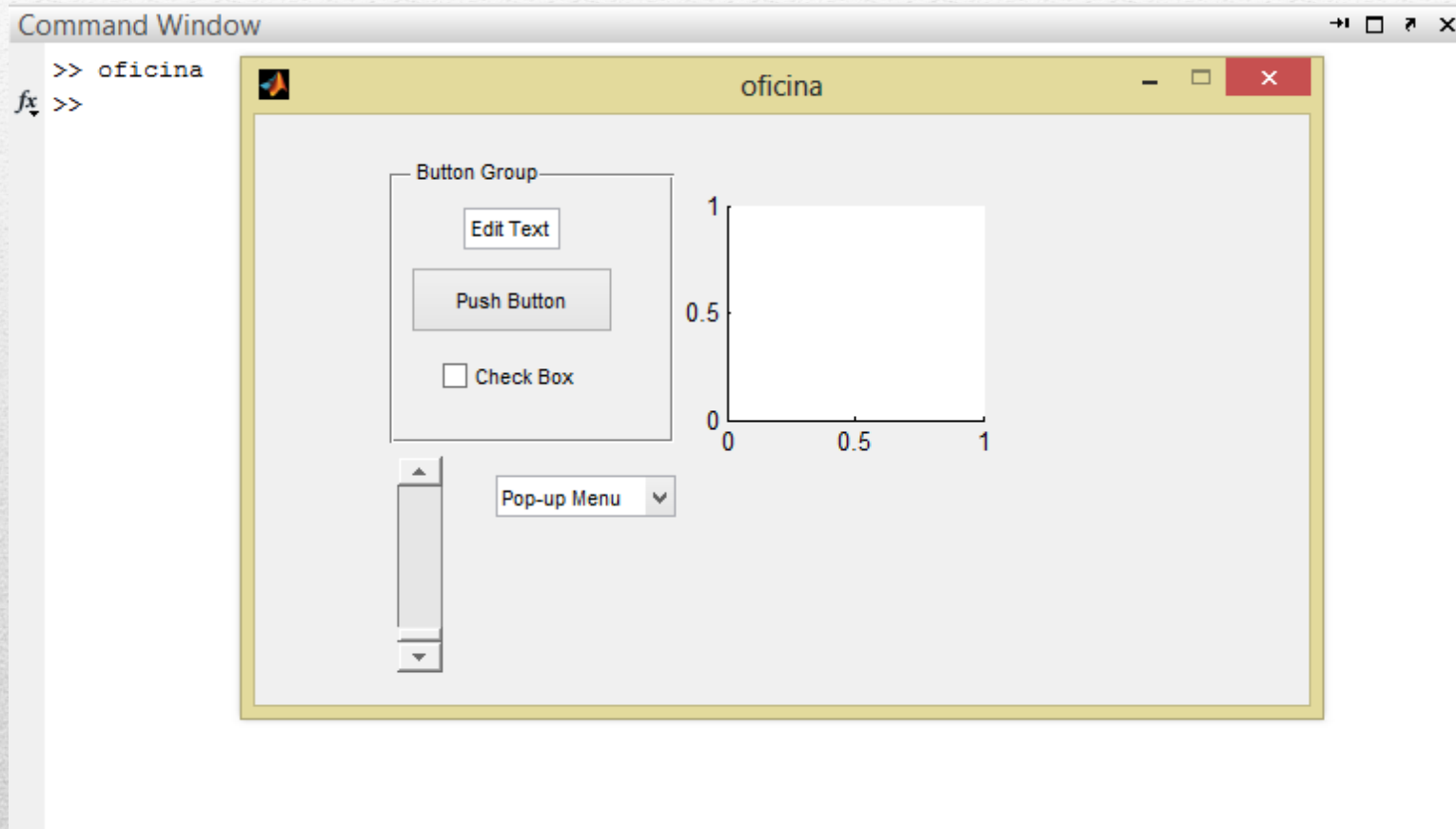


0 código por traz de cada elemento



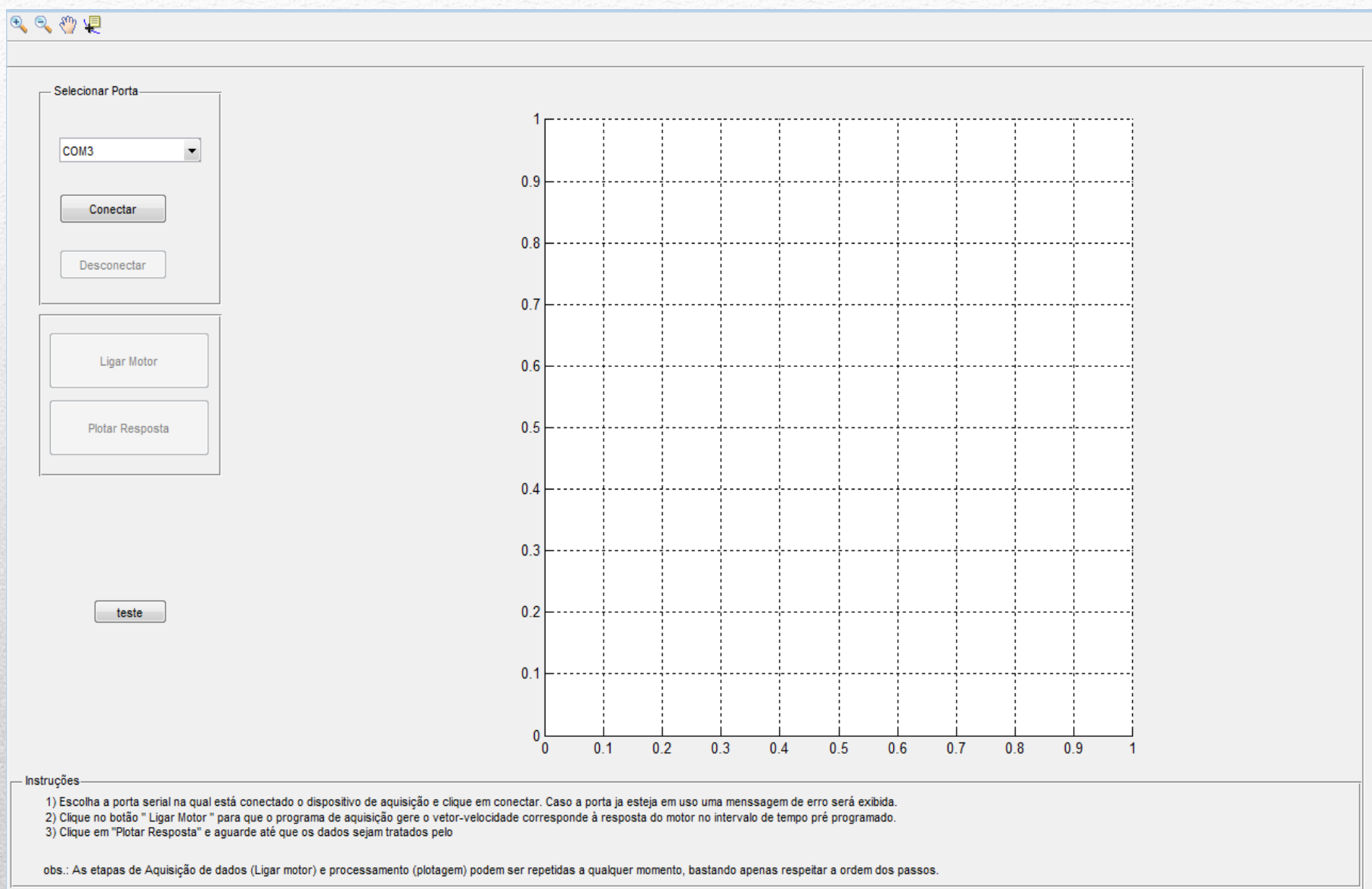
```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

# Chamando a interface





# Interface de Aquisição



Selecionar Porta

COM3

Conectar

Desconectar

Ligar Motor

Plotar Resposta

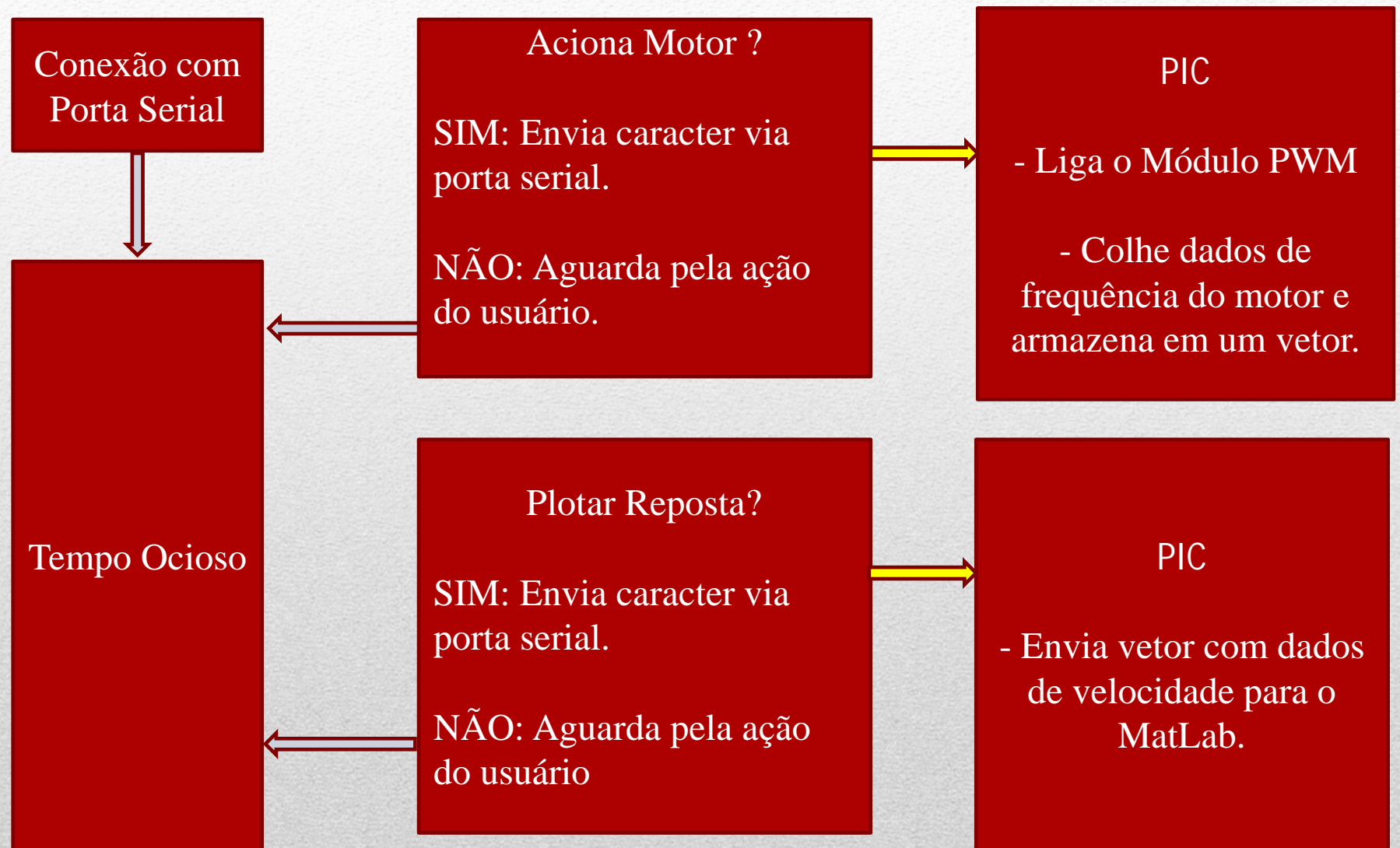
teste

Instruções

- 1) Escolha a porta serial na qual está conectado o dispositivo de aquisição e clique em conectar. Caso a porta já esteja em uso uma mensagem de erro será exibida.
- 2) Clique no botão "Ligar Motor" para que o programa de aquisição gere o vetor-velocidade corresponde à resposta do motor no intervalo de tempo pré programado.
- 3) Clique em "Plotar Resposta" e aguarde até que os dados sejam tratados pelo

obs.: As etapas de Aquisição de dados (Ligar motor) e processamento (plotagem) podem ser repetidas a qualquer momento, bastando apenas respeitar a ordem dos passos.

## Funcionamento da interface



Ligar Motor



```
% --- Executes on button press in pushbutton_ligamotor.
function pushbutton_ligamotor_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_ligamotor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.pushbutton1, 'Enable', 'off');
set(handles.pushbutton2, 'Enable', 'on');
set(handles.popupmenu1, 'Enable', 'off');
set(handles.pushbutton_ligamotor, 'Enable', 'on');

TxText = 'a';
fprintf(handles.serialCom, TxText);

set(handles.pushbutton1, 'Enable', 'off');
set(handles.pushbutton2, 'Enable', 'on');
set(handles.popupmenu1, 'Enable', 'off');
set(handles.pushbutton_ligamotor, 'Enable', 'on');

h = msgbox('Gerando vetor velocidade', 'Status');
```

Plotar Resposta



```
- function pushbutton_recebe_Callback(hObject, eventdata, handles)

    TxText = 'b';
    TxText2 = '';
    fprintf(handles.serialCom, TxText);
    fprintf(handles.serialCom, TxText2);
    flushinput(handles.serialCom);
    flushoutput(handles.serialCom);

    RxText = fscanf(handles.serialCom, '%5d', [500 1])

    fclose(handles.serialCom);
    flushinput(handles.serialCom);
    flushoutput(handles.serialCom);
    delete(handles.serialCom); |
```

Plotar Resposta



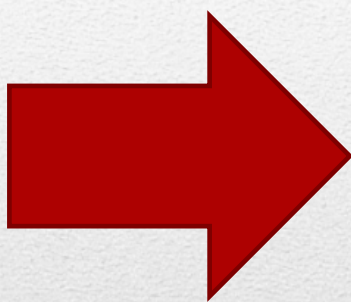
*Continuação do código...*

```
y = RxText/1024*2*pi
t = 0:0.001:0.499;
t1 = t';
u = 12*ones(size(t1));

save velocidade y t u

plot(t,y,'b-');
title('Resposta do Motor');
axis([0 0.5 0 200])
xlabel('Tempo(s)');
ylabel('Velocidade (rad/s)');
legend('Resposta ao degrau');
grid
hold off
```

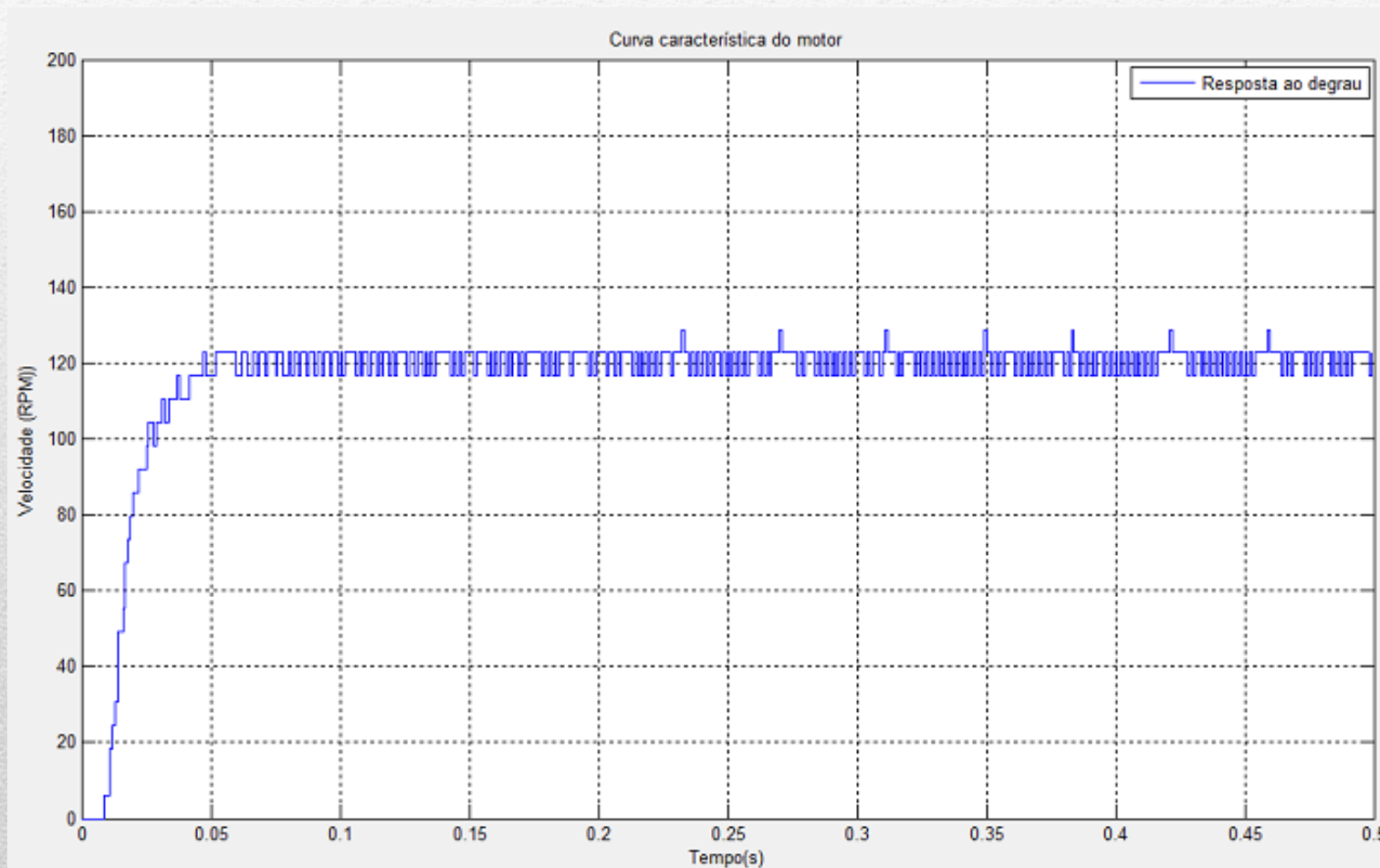
```
Command Window
y =
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    1000
    1000
    3000
    4000
    5000
    8000
    8000
    9000
    11000
    12000
    13000
    14000
    14000
    15000
    15000
    15000
    16000
    17000
    17000
    16000
    17000
    17000
```



```
Command Window
y =
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0
    6.1359
    6.1359
    18.4078
    24.5437
    30.6796
    49.0874
    49.0874
    55.2233
    67.4952
    73.6311
    79.7670
    85.9029
    85.9029
    92.0388
    92.0388
    92.0388
    98.1748
    104.3107
    104.3107
```



# Resultado esperado





# FIM

# Muito Obrigado !