



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Curso de Graduação em Engenharia Elétrica

LÉLIA VAN DER LINDEN

**DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA PARA O  
ENSINO DE TRANSISTORES BIPOLARES DE JUNÇÃO NO  
LABORATÓRIO DE DISPOSITIVOS ELETRÔNICOS**

Campina Grande, Paraíba  
Agosto de 2018

LÉLIA VAN DER LINDEN

DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA PARA O  
ENSINO DE TRANSISTORES BIPOLARES DE JUNÇÃO NO  
LABORATÓRIO DE DISPOSITIVOS ELETRÔNICOS

*Trabalho de Conclusão de Curso submetido à  
Unidade Acadêmica de Engenharia Elétrica da  
Universidade Federal de Campina Grande  
como parte dos requisitos necessários para a  
obtenção do grau de Bacharel em Ciências no  
Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento da Informação

Orientador:

Gutemberg Gonçalves dos Santos Júnior, D. Sc.

Campina Grande, Paraíba  
Agosto de 2018

LÉLIA VAN DER LINDEN

DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA PARA O  
ENSINO DE TRANSISTORES BIPOLARES DE JUNÇÃO NO  
LABORATÓRIO DE DISPOSITIVOS ELETRÔNICOS

*Trabalho de Conclusão de Curso submetido à Unidade  
Acadêmica de Engenharia Elétrica da Universidade  
Federal de Campina Grande como parte dos requisitos  
necessários para a obtenção do grau de Bacharel em  
Ciências no Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento de Informação

Aprovado em        /        /

**Professor Avaliador**  
Universidade Federal de Campina Grande  
Avaliador

**Gutemberg Gonçalves dos Santos Júnior, D. Sc.**  
Universidade Federal de Campina Grande  
Orientador, UFCG

Dedico este trabalho aos meus pais.

## AGRADECIMENTOS

Agradeço à minha família, em primeiro lugar, em especial meus pais que sempre sonharam com uma filha engenheira.

Agradeço também ao meu namorado, Guilherme Fagner Sales Cavalcante, pelo apoio, companheirismo e compreensão durante a graduação.

Agradeço aos meus amigos do curso e fora dele.

Agradeço ao meu orientador Gutemberg Gonçalves dos Santos Júnior.

*Nos dias quotidianos*

*É que se passam*

*Os anos*

*Millôr Fernandes.*

## RESUMO

Este trabalho de conclusão de curso (TCC) apresenta o projeto de uma interface gráfica para o ensino de Transistores Bipolares de Junção (TBJ) no âmbito do ensino da disciplina de Laboratório de Dispositivos Eletrônicos do curso de graduação em Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG). A interface foi elaborada a partir do ambiente GUIDE (Graphical User Interface Development Environment) do MATLAB seguindo o guia de experimentos referente ao TBJ do Laboratório de Dispositivos Eletrônicos a partir de simulações de transistores reais no software Cadence OrCAD. O objetivo desta interface é fornecer uma ferramenta que acompanhe diretamente os experimentos realizados em aulas práticas, facilitando o entendimento do aluno e permitindo que ele teste e refaça o experimento remotamente.

**Palavras-chave:** GUI, GUIDE, MATLAB, transistor bipolar de junção (TBJ), interface, eletrônica, ensino da eletrônica.

## ABSTRACT

This undergraduate thesis presents a graphical interface for the teaching of Bipolar Junction Transistors on the extent of lecturing the Electronical Devices Laboratory class of the Federal University of Campina Grande's Electrical Engineering course. The interface was elaborated through MATLAB's GUIDE (Graphical User Interface Development Environment), according to the experiments guide related to the Electronical Devices Laboratory's BJT, coming from the simulation of real transistors on the Cadence OrCAD software. This interface goal is to provide a tool that directly accompanies the experiments performed on practical classes, favoring the student's understanding and allowing him to remotely test and redo the experiment.

**Keywords:** GUI, GUIDE, MATLAB, bipolar junction transistor (TBJ), interface, electronics, electronics lecturing.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Foto do primeiro mouse.....	12
Figura 2 - Apple Lisa e sua interface.....	13
Figura 3 - Transistores do tipo Bipolar de Junção encontrados no mercado.....	15
Figura 4 - Simbologia do TBJ.....	16
Figura 5 - O ambiente GUIDE.....	18
Figura 6 - Parâmetros do bloco " <i>Solver Configuration</i> " no Simulink.....	21
Figura 7 - Interface do Menu.....	22
Figura 8 - O menu na aplicação GUIDE.....	23
Figura 9 - Interface do Experimento 1 (Parte 1).....	25
Figura 10 - Exemplo de uma simulação de gerador de curva característica do TBJ no Simulink disponível no MATLAB e modificada para o transistor utilizado no experimento.....	26
Figura 11 - Circuito simulado para a parte 1 do Experimento 1.....	27
Figura 12 - Interface do Experimento 1 (Parte 2).....	29
Figura 13 - Circuito simulado para a parte 2 do Experimento 1.....	30
Figura 14 - Mensagem ao usuário do tipo "box" indicando resposta correta.....	32
Figura 15 - Interface do Experimento 1 (Parte 3).....	33
Figura 16 - Circuito simulado para a parte 3 do Experimento 1.....	33
Figura 17 - Interface do Experimento 2 (Parte 1).....	36
Figura 18 - Circuito simulado para a parte 1 do Experimento 2.....	37

## LISTA DE ABREVIATURAS E SIGLAS

DEE	Departamento de Engenharia Elétrica
CAD	Desenho Assistido por Computador
GUIDE	Graphical User Interface Development Environment
IBM	International Business Machines
SPICE	Simulated Program with Integrated Circuits Emphasis
TBJ	Transistor Bipolar de Junção
UFCG	Universidade Federal de Campina Grande

# SUMÁRIO

1	Introdução.....	12
2	Conceitos.....	15
2.1	O Transistor Bipolar de Junção.....	15
2.2	MATLAB.....	17
2.3	GUIDE.....	18
3	Desenvolvimento da Interface.....	19
3.1	Menu.....	22
3.2	Experimento 1 – Aplicações Do Transistor TBJ.....	24
3.2.1	Parte 1.....	24
3.2.2	Parte 2.....	28
3.2.3	Parte 3.....	32
3.3	Experimento 2 – O TBJ como Amplificador.....	35
3.3.1	Parte 1.....	35
4	Conclusão.....	39

# 1 INTRODUÇÃO

A forma como os humanos interagem com os computadores mostrou sempre uma importância para a área de engenharia de *software*. Criar uma composição visual, aliada com comportamento temporal comandado por comando externos e internos, como dispositivos de apontamento e ícones gráficos, é um estudo contínuo que teve seu início na década de 60. Em 1963, Ivan Sutherland desenvolveu o primeiro programa de Desenho Assistido por Computador (CAD), *Sketchpad*, onde uma caneta era utilizada para criar e manipular objetos em desenhos de engenharia.

Entretanto, a ideia de uma interface gráfica em computadores surgiu décadas antes de se ter tecnologia disponível para implementá-la. Um dos primeiros a imaginar essa possibilidade foi o engenheiro e político americano Vannevar Bush no início do século XX. Bush imaginou uma máquina de informações chamada “*Memex*”, em que indivíduos poderiam guardar de forma virtual todos os seus livros, arquivos e recordações de forma mecanizada, para que estes dados sejam facilmente acessados.

O “*Memex*” indicou o nascimento dos *hyperlinks* e se tornou inspiração para os pesquisadores do *Stanford Research Institute* que em 1968, liderado pelo engenheiro Douglas Engelbat, produziu o primeiro protótipo de um periférico: o mouse (Figura 1).

No dia em que o mouse foi apresentado, Douglas Engelbat foi além e mostrou vários conjuntos de periféricos que incluíam o *hypertext*, comunicação por rede e a primeira videoconferência da história.

Figura 1 - Foto do primeiro mouse.



Fonte: Doug Engelbart Institute (2011).

Entretanto, interfaces como conhecemos hoje em dia só começaram a ser desenvolvidas com o surgimento da linguagem de programação e ambiente de desenvolvimento *Smalltalk* no começo da década de 70, no qual janelas móveis, menus, barras de rolagem, caixas de diálogos, botões e ícones foram introduzidos.

Já no final da década de 70, com a contratação de ex-funcionários da Xerox, a Apple Computer, liderada por Steve Jobs e Jef Raskin continuaram desenvolvendo ideias de interface a partir de times criados para o desenvolvimento da Apple Lisa e Macintosh.

O Apple Lisa (Figura 2) foi um computador do tipo *desktop* lançado em 1983 e um dos primeiros a oferecer uma interface gráfica ao usuário (GUI), baseada em ícones indicativos de documento ou aplicação. A equipe foi a primeira a criar menus do tipo *pull-down*, caixas de marcação do tipo *checkboxes* e atalhos de teclado além de conceitos como o duplo clique e lixeira.

Figura 2 - Apple Lisa e sua interface.



Fonte: <https://fossbytes.com/> (2018).

Pode-se citar também como exemplos de computadores que utilizaram GUIs de forma pioneira na década de 80 e 90 o VisiOn da VisiCorp com interface produzida pela IBM, Amiga da Commodore e finalmente o Windows 2.0 da Microsoft, consolidando o mercado com o lançamento do Windows 95.

Em quase 4 décadas, os computadores apresentaram rápidas e drásticas mudanças em processamento e capacidade de armazenamento. Entretanto, não se pode dizer o

mesmo sobre GUIs. Isso é um ponto positivo pois mostra a consolidação de um conceito eficaz, implementado por periféricos como mouse e teclado e interfaces gráficas internas, como menus, ícones e botões.

Na engenharia, o uso de ferramentas e *softwares* com interfaces gráficas é amplamente utilizado em níveis de graduação e pós-graduação nas universidades por todo o mundo. Os mais comuns são MATLAB, Simulink, LabView, OrCad, MultiSim e outros, permitindo que circuitos eletrônicos, modelos de controle, simulações SPICE e programação de dispositivos sejam realizados.

O visual e a interação com o usuário, promovem um melhor aprendizado ao aluno que busca entender assuntos vistos em teoria serem aplicados por meio de simulações como complemento de suas atividades práticas. Este é o princípio no qual este trabalho de conclusão de curso se baseou, no qual uma interface gráfica é desenvolvida no ambiente GUIDE do MATLAB para os experimentos do “Guia 02 – Transistor Bipolar de Junção” do Laboratório de Dispositivos Eletrônicos do curso de graduação em Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG).

O curso de graduação em Engenharia Elétrica UFCG apresenta um currículo composto por aulas teóricas e práticas, esta última chamada de laboratórios. De 19 disciplinas obrigatórias dos Conteúdos Profissionais Essenciais, 16 possuem suas respectivas disciplinas de laboratórios. Isso corresponde a quase 20% da carga horária desse grupo de matérias e 8.21% de seu conjunto com os conteúdos básicos.

Vale salientar também que dessas 16 disciplinas laboratoriais obrigatórias disponibilizadas pelo Departamento de Engenharia Elétrica (DEE), 15 possuem a carga horária de 15h cada, incluindo o Laboratório de Dispositivos, e 1 (Laboratório de Arquitetura e Sistemas Digitais) de 30h, o que corresponde, para o primeiro grupo, 1h semanal para o período escolar.

O Laboratório de Dispositivos Eletrônicos possui experimentos referidos em guias para os seguintes dispositivos: diodos semicondutores, MOSFETs e TBJs. Tais práticas são realizadas utilizando montagens feitas em protoboards com o auxílio de instrumentos de medição, como o osciloscópio e o multímetro, fontes de tensão contínua e gerador de funções.

Assim, este documento descreve os conceitos e atividades desenvolvidas para a criação de uma interface gráfica que segue o guia de experimentos utilizado com o objetivo de fornecer ao usuário da interface um melhor entendimento do que pode ser feito durante as montagens de tais práticas no Laboratório de Dispositivos Eletrônicos.

## 2 CONCEITOS

### 2.1 O TRANSISTOR BIPOLAR DE JUNÇÃO

O Transistor Bipolar de Junção (TBJ) é um dispositivo de três terminais concebido em utilizado em 1948 pela Bell Telephone Laboratories que possui aplicações em grande escala na eletrônica para fabricação de circuitos discretos e integrados. Na Figura 3 é mostrado alguns tipos de TBJs encontrados no mercado.

Figura 3 - Transistores do tipo Bipolar de Junção encontrados no mercado.



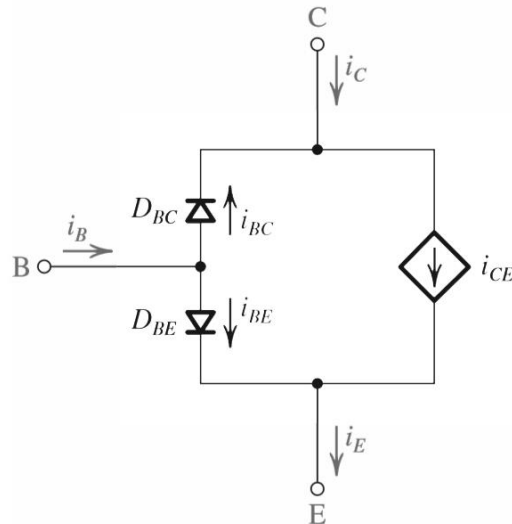
Fonte: <https://www.polytechnichub.com/> (2018).

Os TBJs são divididos por dois tipos, o NPN e o PNP que são baseados no tipo de dopagem dos três terminais do semicondutor, o coletor, emissor e base. Dependendo da polarização de cada terminal do transistor, o TBJ pode trabalhar em diversas regiões de operação, como a ativa, utilizado quando o transistor opera como amplificador, saturação e corte, ambas usadas para aplicações em transistores como circuitos lógicos.

O circuito de um TBJ pode ser pensado como dois diodos, por ser de junção P-N, e uma fonte de corrente controlada. Com isso, algumas equações foram desenvolvidas para introduzir um modelo matemático ao TBJ. Uma delas é o modelo, ou equações de

Ebers–Moll. O SPICE, e também o Simulink, utiliza uma forma deste modelo chamado “Forma de Transporte” do modelo de Ebers-Moll, ilustrado na Figura 4.

Figura 4 – Modelo de Transporte de Ebers-Moll.



Fonte: Adaptado de [2].

As correntes do diodo base-coletor ( $D_{BC}$ ) e do diodo base-emissor ( $D_{BE}$ ) são respectivamente:

$$i_{BC} = \frac{I_S}{\beta_R} (e^{\frac{v_{BC}}{n_R V_t}} - 1)$$

e

$$i_{BE} = \frac{I_S}{\beta_F} (e^{\frac{v_{BE}}{n_F V_t}} - 1)$$

Com  $n_R$  e  $n_F$  os coeficientes de emissão das junções base-coletor e base-emissor respectivamente,  $\beta_R$  e  $\beta_F$  os  $\beta$  direto e reverso do TBJ.

Assim, a corrente da fonte controlada  $i_{CE}$  é definida como:

$$i_{CE} = I_S (e^{\frac{v_{BE}}{n_F V_t}} - e^{\frac{v_{BC}}{n_R V_t}}) (1 - \frac{v_{BC}}{V_a})$$

No qual o termo  $(1 - \frac{v_{BC}}{V_a})$  é referente ao efeito Early em um TBJ com polarização direta e  $I_S$  é a corrente de saturação reversa.



Com isso, considerando  $n_R$  e  $n_F$  iguais a 1, o modelo de Ebers-Moll pode ser usado para descrever as seguintes correntes em qualquer região de operação:

$$\begin{aligned} i_c &= I_s \left[ \left( e^{\frac{v_{BE}}{V_t}} - e^{\frac{v_{BC}}{V_t}} \right) - \frac{1}{\beta_R} \left( e^{\frac{v_{BC}}{V_t}} - 1 \right) \right] \\ i_B &= I_s \left[ \frac{1}{\beta_F} \left( e^{\frac{v_{BE}}{V_t}} - 1 \right) + \frac{1}{\beta_R} \left( e^{\frac{v_{BC}}{V_t}} - 1 \right) \right] \\ i_c &= I_s \left[ \left( e^{\frac{v_{BE}}{V_t}} - e^{\frac{v_{BC}}{V_t}} \right) + \frac{1}{\beta_F} \left( e^{\frac{v_{BE}}{V_t}} - 1 \right) \right] \end{aligned}$$

Alguns modelos SPICE adicionam resistores que representam as resistências da base, coletor e emissor e dois capacitores não lineares para modelar a dinâmica de operação do transistor.

## 2.2 MATLAB

O MATLAB (acrônimo de MATrix LABoratory) é um *software* de computação numérico utilizado como ferramenta de soluções em cálculos matemáticos voltados para tecnologia científica e engenharia. O MATLAB é amplamente utilizado devido a sua popularidade, consagrada por modos de operação iterativos, programação simplificada, funções pré-estabelecidas, facilidades gráficas e um espaço de trabalho embutido com um depurador de informações.

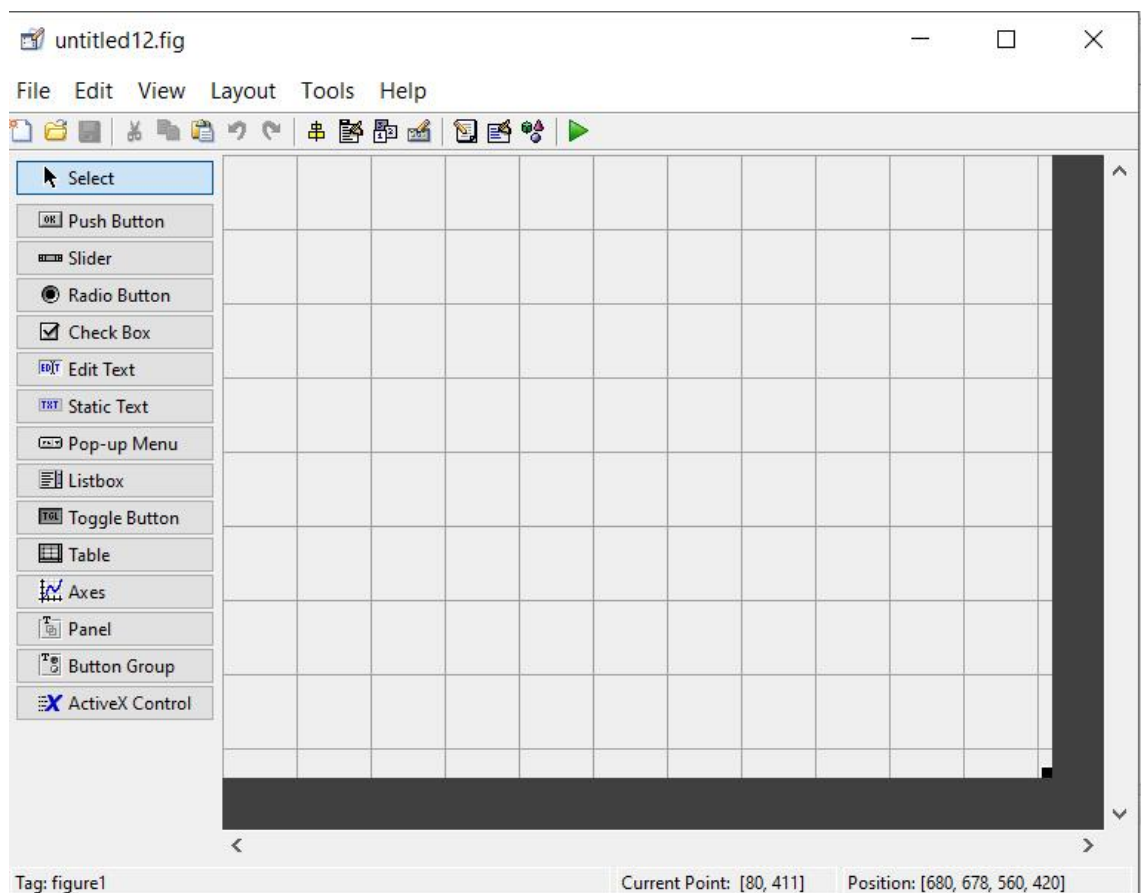
O software é atualmente desenvolvido pela *The Mathworks, Inc*, empresa fundada em 1984 nos Estados Unidos. Porém, foi criado em 1970 por Cleve Moler, na Universidade do Novo México e espalhado rapidamente pelas comunidades acadêmicas de outras universidades.

## 2.3 GUIDE

As GUIs são também conhecidas como interfaces gráficas ao usuário no português. Elas permitem o controle simplificado por meio de periféricos à aplicativos e documentos em um software, simplificando a interação do usuário com a aplicação.

O GUIDE é um ambiente de desenvolvimento de interface gráfica ao usuário que permite a criação de aplicativos personalizados ao projetar graficamente uma interface proposta. Este ambiente é mostrado na Figura 5 a partir de um projeto em branco, no qual é possível observar que ele oferece uma criação manual e simples de interfaces.

Figura 5 - O ambiente GUIDE.



Fonte: Autor (2018).

O projeto de interface é feito de maneira simples, basta selecionar as ferramentas de interface do menu do lado esquerdo da Figura 5 e arrastar para a figura de interface

quadriculada ao lado, montando o aplicativo com a forma requerida para o usuário. A partir disso, o GUIDE gera automaticamente o código MATLAB que permite que as interações entre os blocos de interface selecionados no editor de design sejam feitas de acordo com um comportamento selecionado.

Algumas dessas ferramentas de design foram bastante utilizadas no projeto descrito por este documento. Entre elas o *edit box*, caixas de textos que podem ser alteradas conforme o código se desenvolve, *push button*, ou botões, que quando selecionados permite a execução de certos comandos e o *pop-up menu*, uma espécie de menu embutido para a seleção discreta de escolhas.

Assim, o código MATLAB gerado é dividido em funções relacionadas às ferramentas de interface selecionada. Geralmente são criados dois tipos de função, a *createFcn* e o *callback*. A primeira está relacionada com a aparência da ferramenta, e a segunda indica ou consome as ações que estão sendo utilizadas com ela. Por exemplo, o código de uma função *callback* de um *push button* será realizado quando ele for pressionado. De mesma forma, é em uma função *callback* que dados podem ser armazenados de forma direta, utilizando a opção  *hObject*.

Com isso, foi utilizado o ambiente GUIDE do MATLAB para produzir uma interface gráfica que segue o guia de experimentos do Transistor Bipolar de Junção do Laboratório de Dispositivos Eletrônicos para auxiliar no seu entendimento.

### 3 DESENVOLVIMENTO DA INTERFACE

A interface desenvolvida teve como objetivo auxiliar no entendimento dos Transistores Bipolares de Junção durante o seu equivalente experimento na disciplina de Laboratório de Dispositivos Eletrônicos, e por isso, seguiu o guia de experimentos “Guia 02 – Transistor Bipolar de Junção”.

O documento seguido apresenta uma fundamentação teórica geral para os TBJs, como sua característica de encapsulamento, construção, modelos e guia para experimentos relacionados à sua identificação, aplicação nas regiões, funcionamento como amplificador e amplificador diferencial.

O primeiro experimento, “Identificação dos Terminais do Transistor Bipolar” não está presente na interface desenvolvida pois ela é realizada de forma presencial, portando um multímetro funcionando como um ohmímetro e um transistor, com o objetivo de identificar os terminais dos dispositivos de acordo com a sua polarização no equipamento de medição.

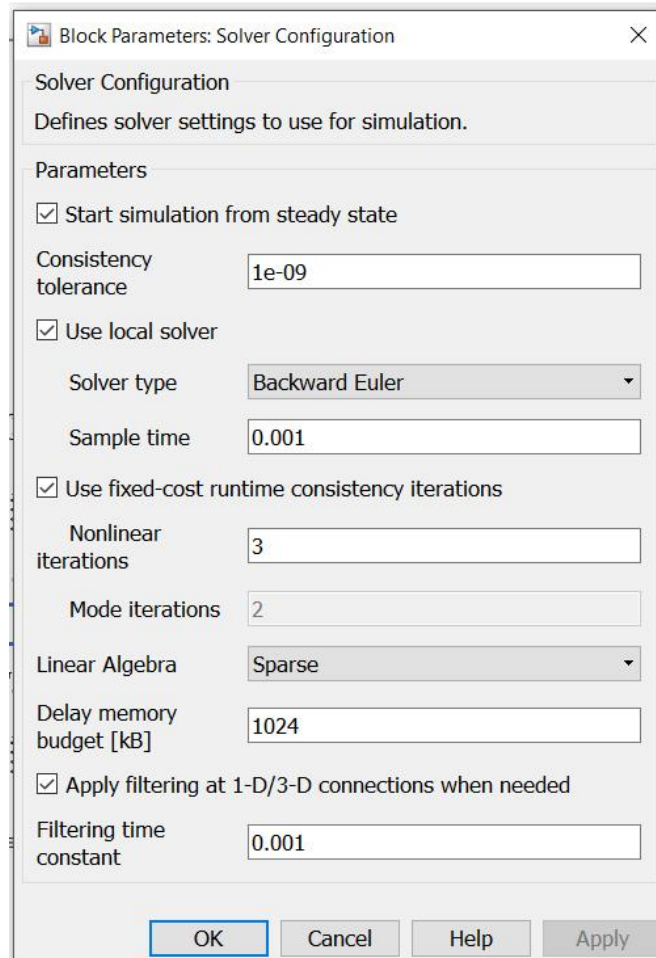
Nos demais experimentos, os circuitos foram simulados no Simulink a partir dos elementos da biblioteca Simscape, no qual utilizou o bloco *Solver Configuration* para utilizar o método numérico *Backward Euler* com *sample time* de 0.001, como descrito na Figura 6, e *time step* de 10s. Além disso, foi utilizado os blocos *To Workspace* da biblioteca padrão do Simulink para exportar as variáveis que foram simuladas por meio dos circuitos realizados. Para fazer a conexão “mecânica-virtual” do Simscape com o Simulink, foi necessário a implementação do bloco de conversão *Ps-Simulink Converter* a partir de sensores de tensão e corrente indicados.

As simulações dos circuitos no Simulink representam a abordagem escolhida para obter a caracterização dos experimentos pois ela apresenta característica confiável e simplificada de solução pois utilizam o modelo de transporte de Ebers-Moll e métodos numéricos para alcançar os resultados. Uma outra forma seria utilizar as equações do modelo e aplicar um método numérico no *software* MATLAB para encontrar um ponto de operação nestas equações intertravadas que o modelo fornece.

O transistor utilizado na simulação dos circuitos no Simulink foi o do bloco “NPN Bipolar Transistor” da biblioteca Simscape, que representa um transistor do tipo NPN modelado a partir das equações de Ebers-Moll. Para tornar a simulação o mais próximo do ideal, dados de *datasheets* do transistor BC548B de vários fabricantes foram analisados e aplicados nos parâmetros do transistor simulado.

Foi considerado também a utilização de um modelo SPICE do transistor tipo NPN disponível na biblioteca Simscape, porém, houve uma incompatibilidade de grandezas físicas dos modelos SPICE disponíveis com o do Simulink. Porém, o modelo NPN utilizado se mostrou bastante satisfatório em termos de exatidão dos resultados.

Figura 6 - Parâmetros do bloco "Solver Configuration" no Simulink.



Fonte: Autor (2018).

Já os experimentos no qual necessitam de fonte AC para ser realizado, o conjunto MATLAB-GUIDE-Simulink mostrou-se um obstáculo para a resolução de iterações no escopo de visualização de gráficos. O bloco de fonte de tensão AC disponível na biblioteca "Simscape" do Simulink é resolvido a partir do *time step* da simulação, ou seja, para frequências altas de 1kHz, como pedido no experimento, é necessário um *time step* grande o suficiente para prolongar a simulação em vários minutos. Uma solução encontrada para esse problema foi aumentar o *sample time* do método numérico mostrado no bloco "Solver Configuration", mas ainda assim, esse tempo não foi reduzido para um aceitável e comprometeu os resultados finais da simulação. Até então não se encontrou uma solução para esta limitação, porém outras alternativas estão sendo testadas.

### 3.1 MENU

Interfaces gráficas são utilizadas principalmente para manter a organização de dados, facilitando o caminho das escolhas dos usuários. Por este motivo, um menu contendo um guia virtual para os experimentos foi desenvolvido de forma simples e direta, mostrado na Figura 7, no qual o utilizador poderá fazer a escolha de qual experimento deseja visualizar ou simplesmente fechar a aplicação.

Figura 7 - Interface do Menu.



Fonte: Autor (2018).

No ambiente GUIDE da Figura 8, o menu foi confeccionado a partir de um bloco do tipo *Axes* de índice 1 para a inserção da imagem de fundo criada previamente utilizando o *software* Adobe Photoshop, um bloco do tipo *Pop-up Menu* para a escolha dos experimentos e um *Push Button* para fechar a aplicação.

No código do MATLAB, a imagem de fundo, por estar sendo utilizada a todo momento, é chamada na função *OpeningFcn* do menu por meio do seguinte comando:

```
axes(handles.axes1)
imshow('menu.jpg');
```

A escolha do experimento a ser realizado e de fechar a aplicação está presente na funções do tipo *callback* do *pop-up menu* e do *push button* respectivamente.

Pop-up menu Callback

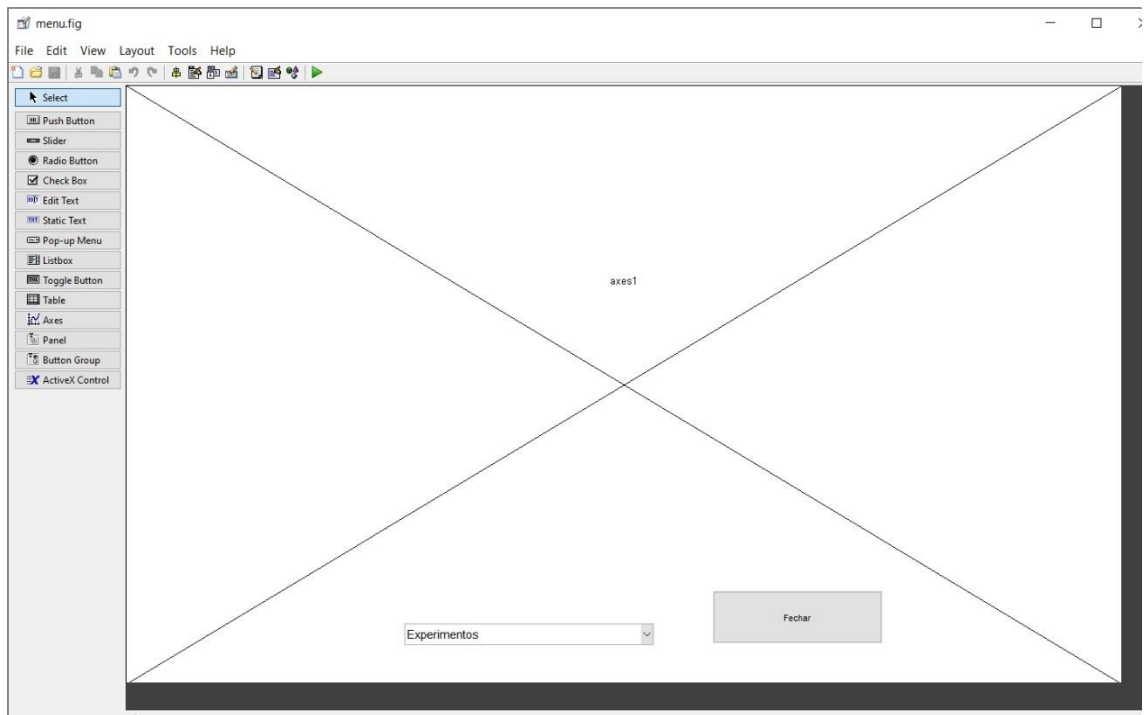
```
% Opções do Menu
contents = cellstr(get(hObject,'String'));
exp_popupmenu = contents(get(hObject,'Value'));
if(strcmp(exp_popupmenu,'Experimento 1'))
    set(exp1,'Visible','on')
    set(menu,'Visible','off')
elseif(strcmp(exp_popupmenu,'Experimento 2'))

end
```

Push Button Callback

```
delete(get(0,'Children'));
```

Figura 8 - O menu na aplicação GUIDE.



Fonte: Autor (2018).

## 3.2 EXPERIMENTO 1 – APLICAÇÕES DO TRANSISTOR TBJ

O experimento 1, seguindo o guia do laboratório de dispositivos eletrônicos, é dividido em três partes e quatro circuitos são montados. A primeira parte consiste em variar o potenciômetro do circuito e verificar os valores de tensão e corrente nos terminais do transistor, verificando a região em que ele está atuando, o valor do  $\beta$  correspondente e sua curva característica. Já na segunda e terceira parte, três circuitos são montados com o objetivo de observar os seus valores de saída ao variar as entradas de 0V e 5V e identificar a função lógica implementada.

O algoritmo implementado para fazer a alteração das partes é a mesma utilizada no menu, ou seja, quando um experimento for selecionado, sua interface passará a ser visível e todas as outras invisíveis. Tal código foi implementado nas funções do tipo *callback* dos *pop-up menus*, ou seja, realizados apenas quando há mudanças nesse bloco.

```
% Opções do Menu
contents = cellstr(get(hObject, 'String'));
menu_exp1 = contents(get(hObject, 'Value'));
if(strcmp(menu_exp1, 'Parte 1'))
    set(exp1, 'Visible', 'on')
    set(exp1_2, 'Visible', 'off')
    set(exp1_3, 'Visible', 'off')

elseif(strcmp(menu_exp1, 'Parte 2'))
    set(exp1, 'Visible', 'off')
    set(exp1_2, 'Visible', 'on')
    set(exp1_3, 'Visible', 'off')

elseif(strcmp(menu_exp1, 'Parte 3'))
    set(exp1, 'Visible', 'off')
    set(exp1_2, 'Visible', 'off')
    set(exp1_3, 'Visible', 'on')
end
```

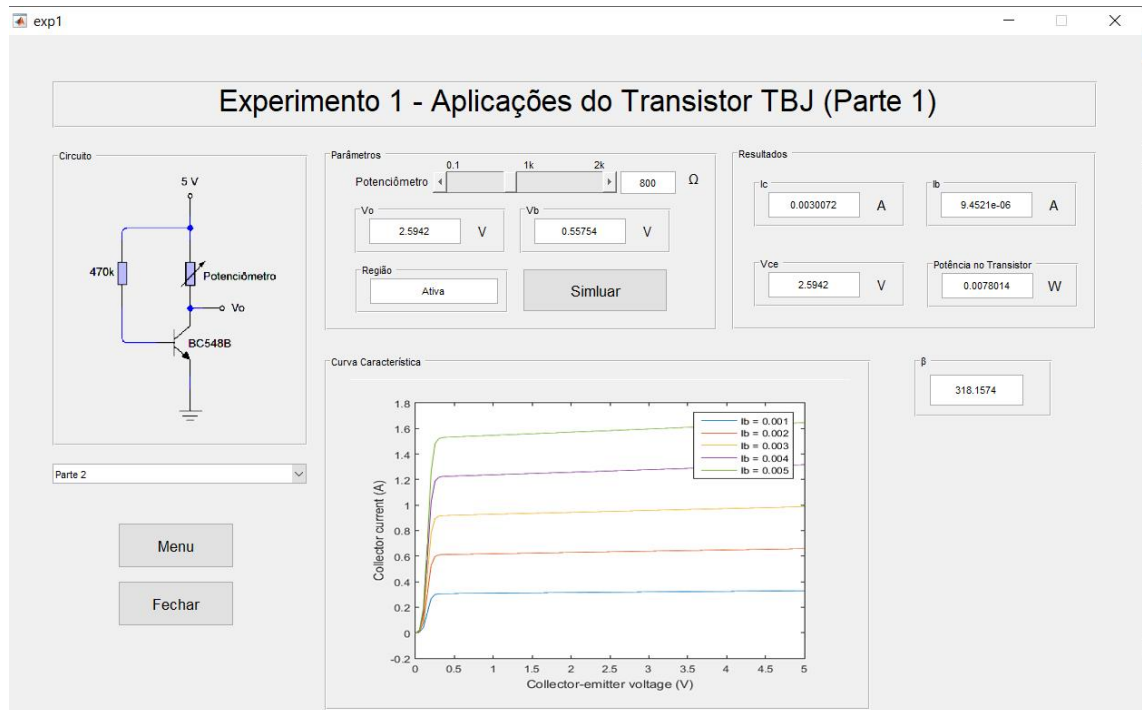
### 3.2.1 PARTE 1

A interface da parte 1 mostrada na Figura 9 foi realizada de forma simples para o melhor entendimento do usuário. A figura do circuito, extraída do guia de experimentos seguido, foi inserida para auxiliar no entendimento da montagem do experimento. Além disso, um pequeno menu de navegação foi adicionado embaixo desta imagem para que o



utilizador da interface possa navegar entre os experimentos e a parte a qual deseja realizar no experimento 1.

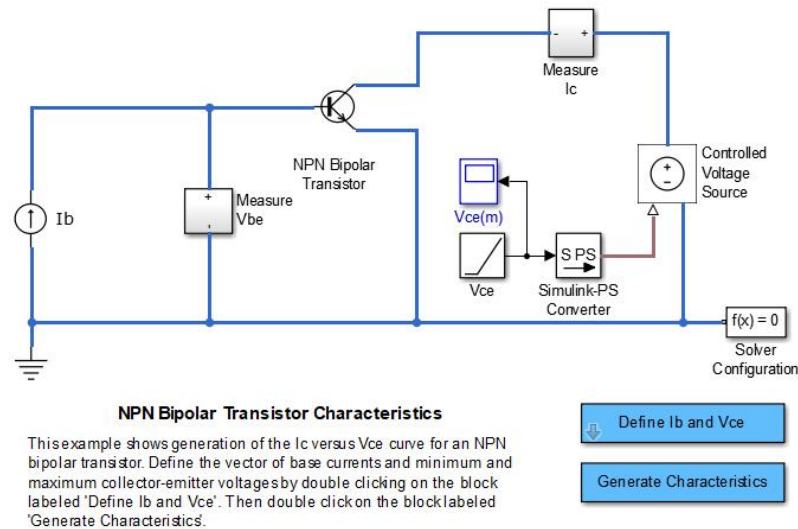
Figura 9 - Interface do Experimento 1 (Parte 1).



Fonte: Autor (2018).

A curva característica I-V (corrente no coletor-tensão coletor emissor) foi extraída a partir do exemplo “*elec\_bipolar\_npn*” do Simulink mostrado na Figura 10, modificando o bloco *NPN Bipolar Transistor* para um modelo próximo do real utilizando os datasheets estudados.

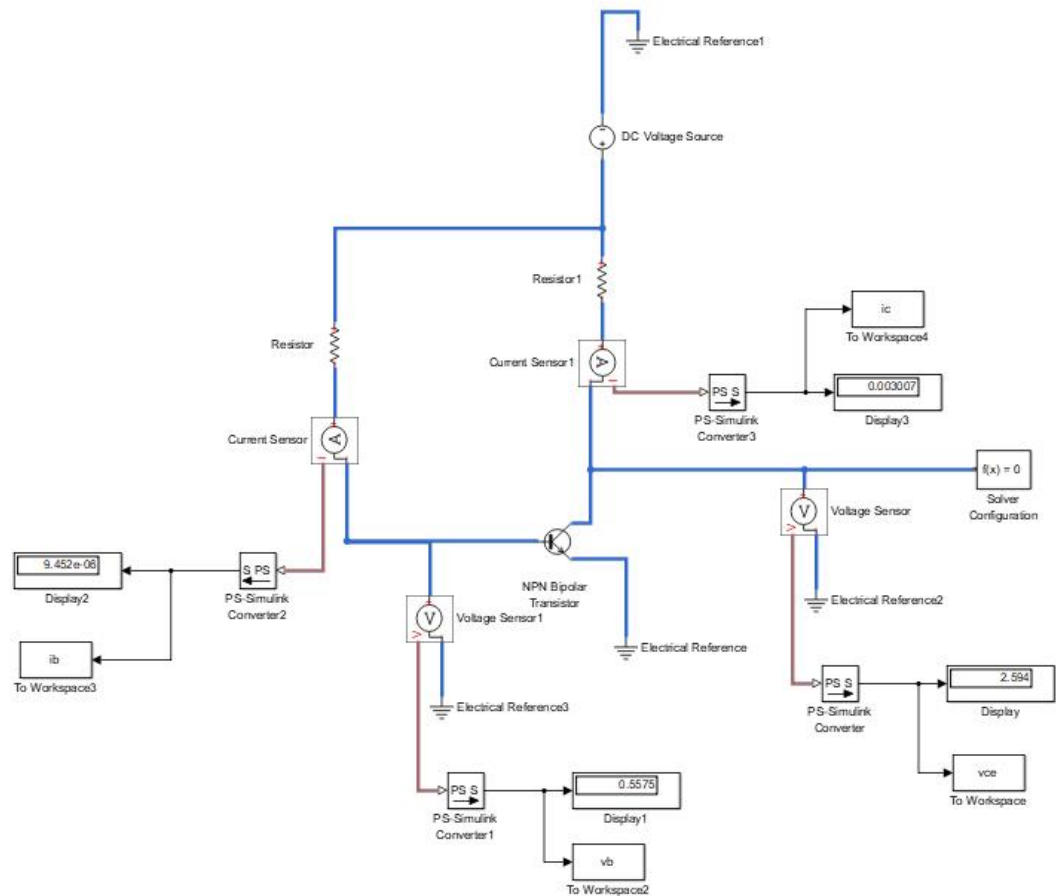
Figura 10 - Exemplo de uma simulação de gerador de curva característica do TBJ no Simulink disponível no MATLAB e modificada para o transistor utilizado no experimento.



Fonte: Autor (2018).

Para obter os valores requeridos no experimento do circuito estudado na Figura 9, uma simulação no Simulink mostrada na Figura 11 foi realizada utilizando-se parâmetros reais para o transistor do bloco *NPN Bipolar Transistor*, blocos exportadores de variáveis para o *workspace* e parâmetros de blocos oriundos do *workspace*. Por exemplo, o bloco *Resistor 1* tem como valor uma variável controlada por um *slider* na interface ao passo em que blocos *To Workspace* exportam as variáveis  $i_c$ ,  $v_{ce}$ ,  $v_b$  e  $i_b$  para o *workspace* do MATLAB.

Figura 11 - Circuito simulado para a parte 1 do Experimento 1.



Fonte: Autor (2018).

Para utilizar as variáveis manipuladas na interface, elas foram declaradas como globais na função *callback* do botão “Simular”, indicando que só estarão disponíveis quando o botão for pressionado na interface. Nesta mesma função, os cálculos de  $I_b$ ,  $I_c$ ,  $\beta$ , potência no transistor e a determinação da região atuante no transistor foram determinadas e podem ser conferidas nos seguintes comandos comentados:

```
global pot vce ic ib vb w beta
```

```
%--- Insere os valores das variáveis nas caixas de exibição do tipo  
"Edit Box"
```

```

set(handles.vceval, 'String', num2str(vce));
set(handles.voval, 'String', num2str(vce));
set(handles.icval, 'String', num2str(ic));
set(handles.ibval, 'String', num2str(ib));
set(handles.vbval, 'String', num2str(vb));

%--- Cálculo de beta e inserção do valor na Edit Box correspondente

beta = ic/ib;

set(handles.betaval, 'String', num2str(beta));

%Recebe o valor do potenciômetro do slider e envia para o simulink

pot = get(handles.slider1, 'Value');

% calcula a potência com os valores de ic e vce

w = ic*vce;
set(handles.wval, 'String', num2str(w)); % imprime o valor da potência
calculado

%--- Determina a região

if vce < vb
    set(handles.regiaoval, 'String', 'Saturação');
else
    set(handles.regiaoval, 'String', 'Ativa');
end

%--- Fim do determina a região

```

Já para estabelecer a conexão entre o Simulink, GUIDE e MATLAB, a seguinte linha foi adicionada na função callback do botão “Simular”:

```
sim('experimento1'); % circuito no simulink
```

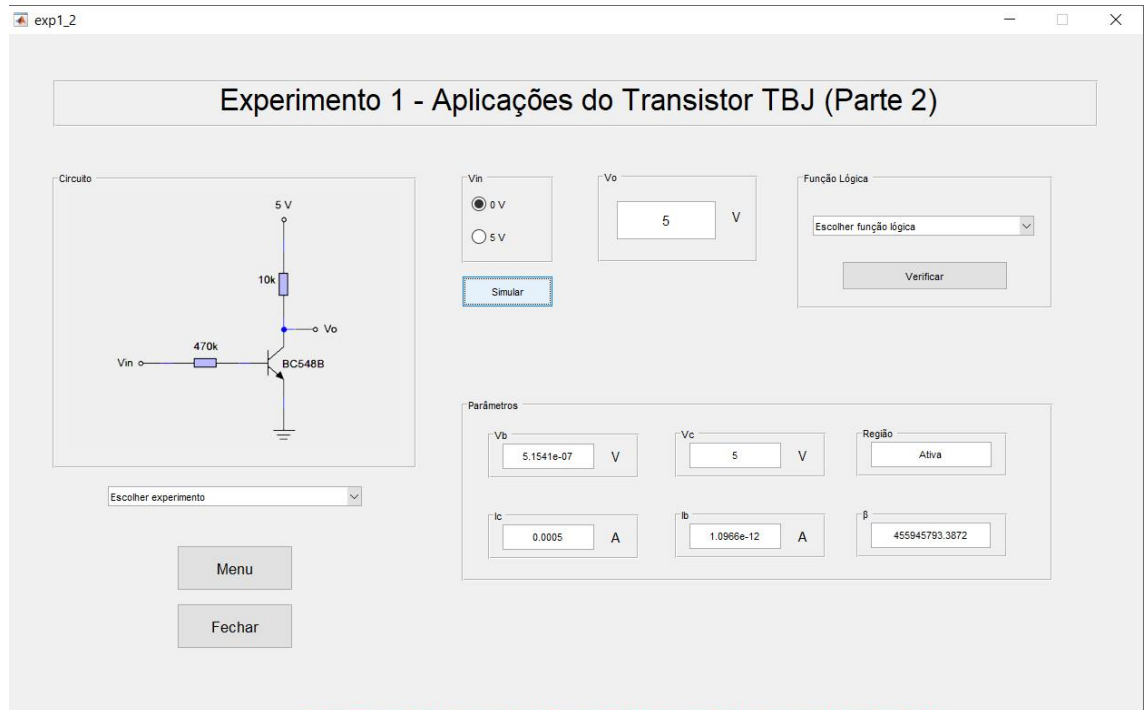
### 3.2.2 PARTE 2

A parte 2 é referente ao circuito simulado presente na Figura 12 da sua interface criada, semelhante a parte 1, na qual contém o mesmo menu de navegação. Este experimento é acessado a partir do menu do tipo *pop-up* presente em todas as partes da interface do Experimento 1.

A parte 2 do Experimento 1 tem como objetivo identificar a função lógica a qual o circuito representa, bem como identificar a região atuante e seus parâmetros indicados pelos blocos do tipo *Edit Box*, como Vc e Vb e atualizados na função *callback* do *push*

button “Simular” mostrado abaixo. As correntes  $i_c$  e  $i_b$  são calculadas a partir da Lei de Ohm.

Figura 12 – Interface do Experimento 1 (Parte 2).



Fonte: Autor (2018).

```

global vb vo ic ib beta

ic = vo/10000;

ib = vb/470000;

set(handles.voval, 'String', num2str(vo));
set(handles.vcval, 'String', num2str(vc));
set(handles.icval, 'String', num2str(ic));
set(handles.ibval, 'String', num2str(ib));
set(handles.vbval, 'String', num2str(vb));

beta = ic/ib;

set(handles.betaval, 'String', num2str(beta));

%--- Determina a região

if vo < vb
    set(handles.regiaoval, 'String', 'Saturação');
else

```

```

set(handles.regiaoval,'String','Ativa');
end

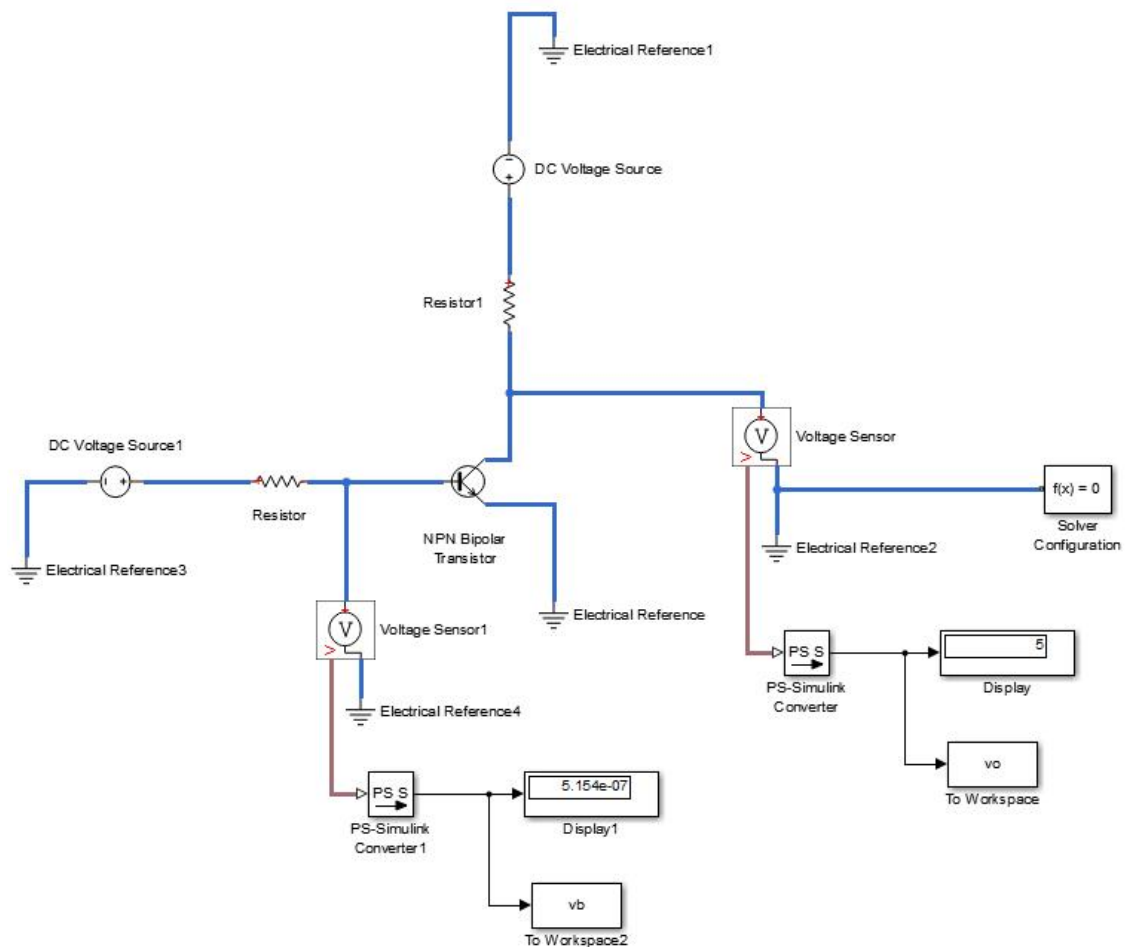
%--- Fim do determina a região

sim('experimento1_2'); % circuito no simulink

```

A mesma abordagem da parte 1 foi inserida para simular o circuito no Simulink. Entretanto, o parâmetro “*Constant Voltage*” da fonte de tensão DC “DC Voltage Source 1” da Figura 13 foi alterado para “*vin*”, uma variável global do workspace do MATLAB, controlada pelos botões do tipo *Radio Button* disponível no GUIDE e representada dentro do *panel Vin* da interface da Figura 12. De mesma forma, as variáveis *vo* e *vb*, também declaradas como globais, foram exportadas para o *workspace* a partir do bloco *To Workspace*.

Figura 13 - Circuito simulado para a parte 2 do Experimento 1.



Fonte: Autor (2018).

Os botões do tipo *radio button* dentro de um *button group* permitem a seleção de apenas um botão por vez pela aplicação do GUIDE. Portanto, o código abaixo foi utilizado na sua função *uibuttongroup3\_SelectionChangedFcn* do *button group* utilizado. Nele, é comparado a escolha do usuário a parte de modificações feita dentro no grupo de botões.

```
A = get(hObject, 'String');
global vin
switch A
    case '0 V'
        vin = 0;
    case '5 V'
        vin = 5;
end
```

Para a indicação da porta lógica utilizada, pensou-se em uma forma didática que aumentasse a interação do usuário com a interface criada. Para isso, testou-se previamente o circuito da parte 2 e determinou-se que a função lógica representada pelo circuito em questão é a *Not*. Então, foi adicionado um *pop-up menu* na aplicação GUIDE com as opções das seguintes funções lógicas: *and*, *or*, *xor*, *nand*, *nor*, *xnor* e *not*. Na função *callback* do *push button* “Verificar” foi adicionado os comandos abaixo, no qual a *string* pertencente a escolha do usuário no *pop-up menu* de nome “menulogica” é testada e verificada a partir de mensagens do tipo *box* que são mostradas ao usuário na interface indicando se a escolha foi correta ou não. Tal mensagem do tipo *box* é apresentada na Figura 14.

```
contents = cellstr(get(handles.menulogica, 'String'));
verif = contents(get(handles.menulogica, 'Value'));

if(strcmp(verif, 'AND'))
    msgbox('Resposta errada! Tente novamente.');
```

```
elseif(strcmp(verif, 'OR'))
    msgbox('Resposta errada! Tente novamente.');
```

```
elseif(strcmp(verif, 'XOR'))
    msgbox('Resposta errada! Tente novamente.');
```

```
elseif(strcmp(verif, 'NAND'))
    msgbox('Resposta errada! Tente novamente.');
```

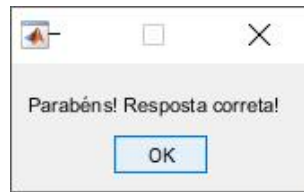
```
elseif(strcmp(verif,'NOR'))
    msgbox('Resposta errada! Tente novamente.');
```

```
elseif(strcmp(verif,'XNOR'))
    msgbox('Resposta errada! Tente novamente.');
```

```
elseif(strcmp(verif,'NOT'))
    msgbox('Parabéns! Resposta correta!');
```

```
end
```

Figura 14 - Mensagem ao usuário do tipo "box" indicando resposta correta.



Fonte: Autor (2018).

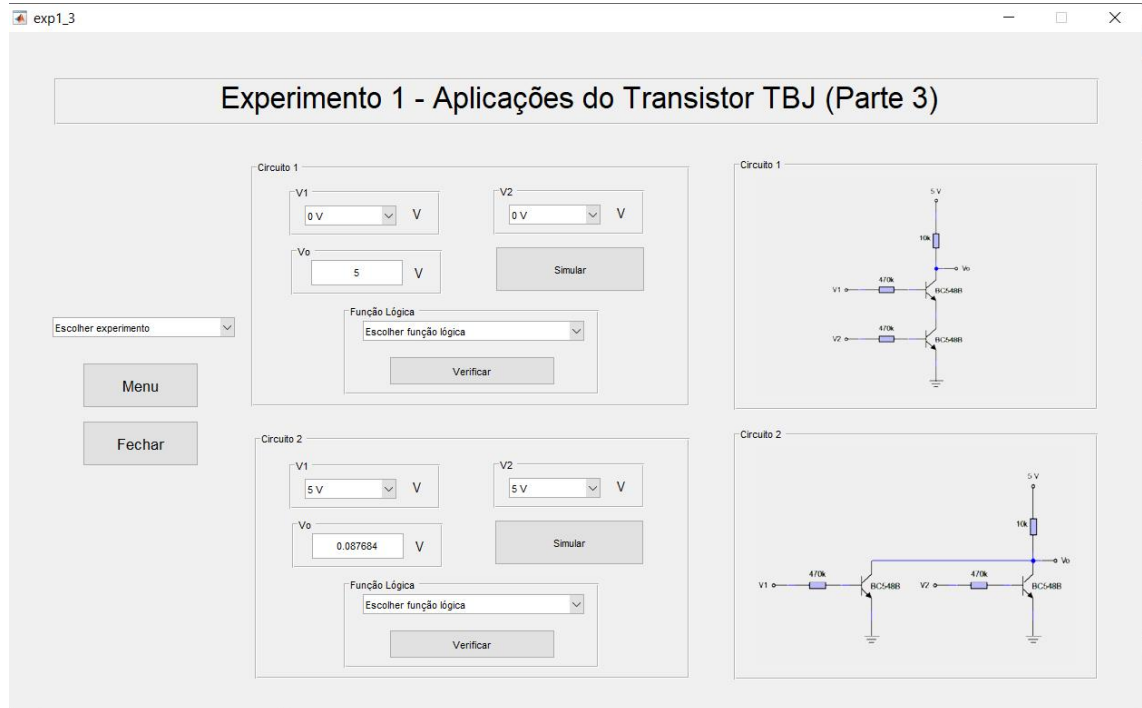
### 3.2.3 PARTE 3

A parte 3 estuda as saídas de dois circuitos que estão inseridos na interface do experimento da Figura 15. Seguindo o guia de experimentos, cada um dos os circuitos realizam uma função lógica diferente, na qual o usuário da interface deverá descobrir ao fazer testes com os níveis lógicos de entrada representados pelas variáveis importadas pela simulação da Figura 16 do Simulink a partir do *workspace* do MATLAB.

Para cada circuito, *panels* diferentes foram adicionados no ambiente GUIDE para que a escolha das entradas fosse selecionada separadamente e os valores das tensões de saída  $V_o$  fossem mostrados nos *edit boxes* correspondentes a partir dos valores das variáveis  $v_{11}$ ,  $v_{21}$ ,  $v_{12}$  e  $v_{22}$  exportadas para a simulação dos circuitos no Simulink, no qual correspondem aos blocos de *DC Voltage Source* 1,2,5 e 4 respectivamente da Figura 16.

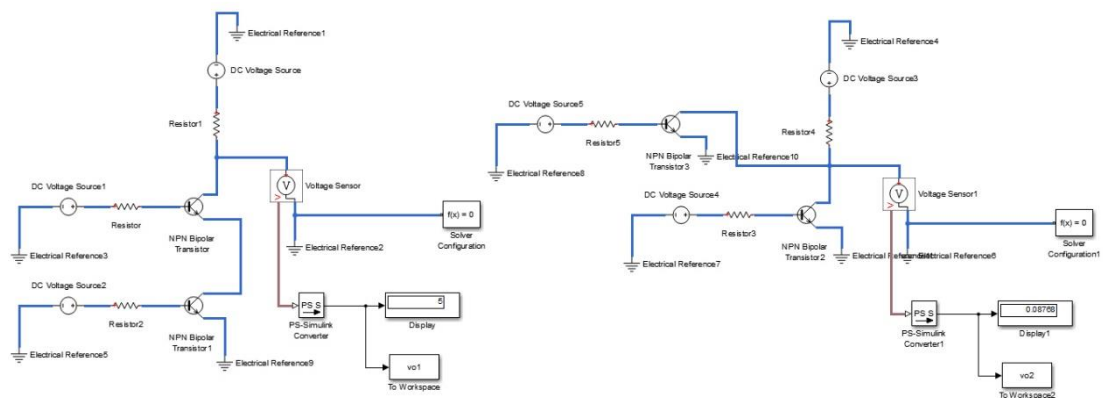


Figura 15 - Interface do Experimento 1 (Parte 3).



Fonte: Autor (2018).

Figura 16 - Circuito simulado para a parte 3 do Experimento 1.



Fonte: Autor (2018).

Dessa forma, cada *push button* “Simular” do circuito correspondente aplica comandos que exportam as variáveis para o Simulink a partir da escolha do usuário em

um *pop-up menu* na interface e imprime os valores de saídas simulados nos *edit boxes* correspondentes quando o botão é pressionado. Como exemplo, a função *callback* do botão “Simular” para o circuito 1 é mostrado abaixo, no qual a escolha das tensões  $v_{11}$  e  $v_{21}$  a serem exportadas para o Simulink é feita a partir de dois *pop-up menus* com as opções de 0V e 5V. Para o circuito 2, o código utilizado é similar.

```

global v11 v21 vo1

% Para vo1
contents = cellstr(get(handles.v11val, 'String'));
ver11 = contents(get(handles.v11val, 'Value'));
if(strcmp(ver11, '0 V'))
    v11 = 0;
else
    v11 = 5;
end

contents = cellstr(get(handles.v21val, 'String'));
ver21 = contents(get(handles.v21val, 'Value'));
if(strcmp(ver21, '0 V'))
    v21 = 0;
else
    v21 = 5;
end

% Imprime o valor de vo1 simulado no correspondente
edit box
set(handles.vo1val, 'String', num2str(vo1));

% circuito no simulink
sim('experimento1_3');

```

Para a verificação dos circuitos, eles foram previamente analisados e indicados que implementam a função lógica *nand* para o circuito 1 e a função lógica *nor* para o circuito 2. De maneira similar utilizada na parte 2 do Experimento 1, um *pop-up menu* permite que o usuário faça a escolha de resposta para qual função lógica ele determinou após as análises feita na interface, e que também possa conferi-la por meio de mensagens do tipo *box*.

### 3.3 EXPERIMENTO 2 – O TBJ COMO AMPLIFICADOR

O experimento 2 do guia de experimentos do Laboratório de Dispositivos Eletrônicos seguido é dividido em duas partes e aborda montagens do TBJ que estudam o comportamento do transistor como um amplificador.

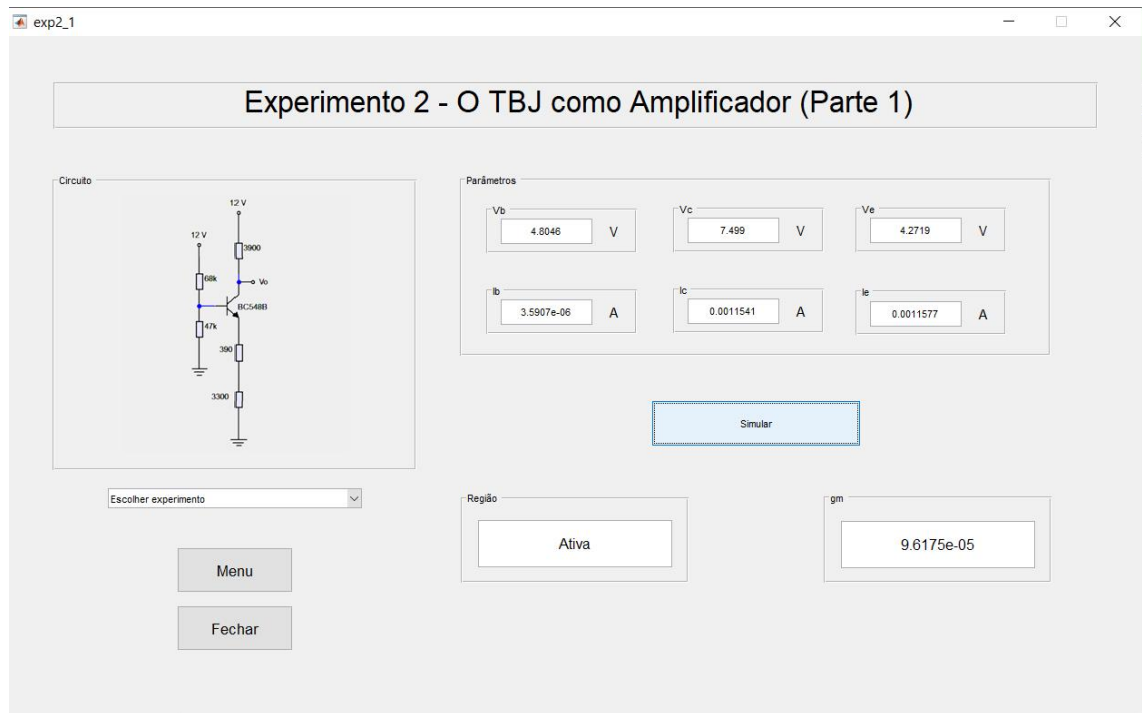
Como explicado anteriormente, este documento não irá tratar da parte 2 do experimento 2 devido às incompatibilidades de execução encontradas para a integração Simulink-MATLAB-GUIDE.

#### 3.3.1 PARTE 1

O objetivo da parte 1 do Experimento 2 é analisar um circuito mostrado na Figura 17 representando a interface criada, que utiliza um TBJ para implementar a função de amplificador utilizando tensões de entrada CC. A partir de uma simulação, os valores das tensões nos terminais do dispositivo são medidos e os valores das correntes correspondentes são calculados a partir da Lei de Ohm. Além disso, é constatado a região de operação do transistor e o valor de sua transcondutância.

De forma similar aos outros experimentos, a interface da parte 1 do Experimento 2 possui um menu de navegação logo abaixo da figura do circuito analisado, em que facilita a escolha de experimentos e caminhos seguidos pelo usuário.

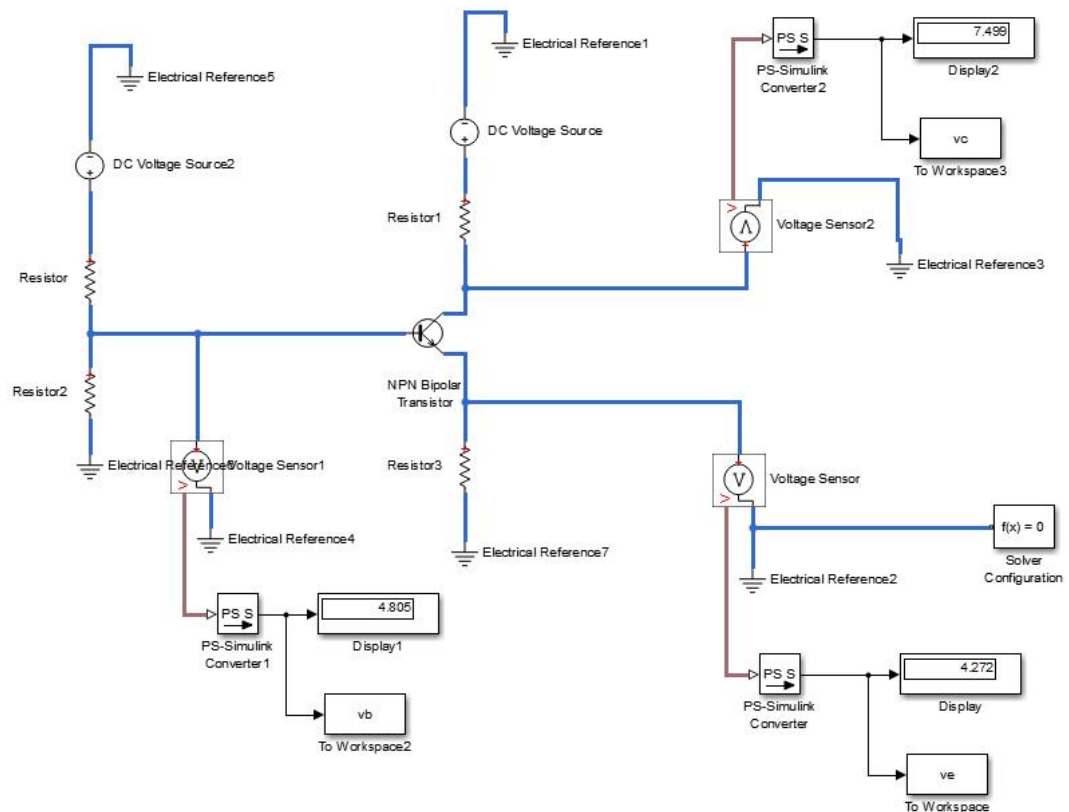
Figura 17 - Interface do Experimento 2 (Parte 1).



Fonte: Autor (2018).

Além disso, foi simulado um circuito no Simulink utilizando parâmetros de datasheet para um TBJ tipo NPN BC548B, como utilizado no experimento 1. O circuito simulado da Figura 18 possui parâmetros de entrada nas fontes DC pré-estabelecidos pelo guia de experimentos, e possui comunicação com o *workspace* para que valores possam ser exportados e mostrados em *edit boxes*. Como os valores são fixos, a região em que o transistor opera foi previamente determinada.

Figura 18 - Circuito simulado para a parte 1 do Experimento 2.



Fonte: Autor (2018).

Assim, o seguinte código foi implementado na função *callback* do *push button* “Simular”, para que os valores determinados por meio da simulação sejam importados no *workspace* e mostrados na interface criada.

```
global vb vc ve

set(handles.vcval, 'String', num2str(vc));
set(handles.vbval, 'String', num2str(vb));
set(handles.veval, 'String', num2str(ve));

ib = ((12-vb)/68000) - (vb/47000);

ic = (12-vc)/3900;

ie = ve / 3690;

set(handles.ibval, 'String', num2str(ib));
set(handles.icval, 'String', num2str(ic));
set(handles.ieval, 'String', num2str(ie));

%--- Determina a região
```

```
set(handles.regiaoval, 'String', 'Ativa');  
  
%--- Fim do determina a região  
  
%--- Determina gm  
  
gm = ic/12;  
  
set(handles.gmval, 'String', num2str(gm));  
  
%--- Fim do determina gm  
  
sim('experimento2_1'); % circuito no  
simulink
```

## 4 CONCLUSÃO

Neste documento transmitiram-se as diretrizes básicas para a elaboração de uma interface para o experimento de TBJ do Laboratório de Dispositivos Eletrônicos do curso de graduação de Engenharia Elétrica da Universidade Federal de Campina Grande.

Assim, é possível afirmar que além de ter adquirido novos conhecimentos, a aluna foi capaz de utilizar conhecimentos advindos de disciplinas da graduação do curso de Engenharia Elétrica, como o próprio Laboratório de Dispositivos Eletrônicos, além de outros que utilizam componentes e instrumentos de fonte e medição.

Foi possível concluir que a aplicação desenvolvida é capaz de ajudar no ensino e na aprendizagem de alunos e professores durante a Disciplina de Laboratório de Dispositivos Eletrônicos. Dessa forma, a interface produzida é de suma importância para a disciplina.

## Bibliografia

- [1] UFCG. "**Guia 02 - Transistores Bipolares de Junção**". Laboratório de Dispositivos Eletrônicos. [S.l.]: 2018.
- [2] SEDRA, S.; SMITH, K.. Microeletrônica. 4ª. Edição, Pearson Makron Books, São Paulo, Brasil, 2005.
- [3] Razavi, B. "**Fundamentos de Microeletrônica**". LTC. 1ª edição. 2010.
- [4] Boylestad, Robert e Nashelsky, Louis. "**Dispositivos Eletrônicos e Teoria de Circuitos**". Rio de Janeiro: Prentice-Hall do Brasil Ltda., 2001.
- [5] Malvino, Albert P. "**Eletrônica**". São Paulo: McGraw-Hill, 1995.
- [6] Ali H. Assi, Maitha H. Al Shamisi and Hassan A. N. Hejase. "**MATLAB GUI Application for Teaching Electronics**". United Arab Emirates University, United Arab Emirate, 2011.
- [7] Strle, Drago. "**Mixed-Signal Circuits Modelling and Simulations Using Matlab**". Intech Open, 2011
- [8] Azemi, Asad. "**Utilizing MATLAB in undergraduate electric circuits courses**". Frontiers in Education Conference, 1996. FIE '96. 26th Annual Conference., Proceedings of, Volume: 2, 1996.
- [9] Nikvand, Nima. "**MATLAB GUI (Graphical User Interface) Tutorial for Beginners**". University of Waterloo, 2010.
- [10] Alldatasheet, "**BC548B**", <https://www.alldatasheet.com/>, 2018