



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

EDUARDO MOREIRA COSTA

**DESENVOLVIMENTO DA INTERFACE GRÁFICA PARA UM EMULADOR DE
CARGAS MECÂNICAS PARA ENSAIOS DE MOTORES ELÉTRICOS**

FORTALEZA

2019

EDUARDO MOREIRA COSTA

**DESENVOLVIMENTO DA INTERFACE GRÁFICA PARA UM EMULADOR DE
CARGAS MECÂNICAS PARA ENSAIOS DE MOTORES ELÉTRICOS**

Trabalho de conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, com requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Paulo Peixoto Praça.

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- C871d Costa, Eduardo Moreira.
Desenvolvimento de uma interface gráfica para um emulador de cargas mecânicas para ensaios de motores elétricos / Eduardo Moreira Costa. – 2019.
29 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2019.
Orientação: Prof. Dr. Paulo Peixoto Praça.
1. Emulador de cargas mecânicas. 2. Lazarus. 3. RaspBerry Pi. I. Título.

CDD 621.3

EDUARDO MOREIRA COSTA

**DESENVOLVIMENTO DA INTERFACE GRÁFICA DE UM EMULADOR DE
CARGAS MECÂNICAS PARA ENSAIOS DE MOTORES ELÉTRICOS**

Trabalho de conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica do
Centro de Tecnologia da Universidade Federal
do Ceará, com requisito parcial à obtenção do
grau de bacharel em Engenharia Elétrica.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

Prof. Dr. Paulo Peixoto Praça (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Raphael Amaral da Câmara
Universidade Federal do Ceará (UFC)

Prof. Msc. Adriano Holanda Pereira (IFCE)
Instituto Federal do Ceará (IFCE)

A Todas as pessoas que de alguma forma
contribuíram para este momento.

AGRADECIMENTOS

Aos Professores. e servidores do departamento de engenharia elétrica da UFC por todo apoio dado durante a graduação.

Aos professores participantes da banca examinadora Dr. Paulo Peixoto Praça, Dr. Raphael Amaral da Câmara e ao Msc. Adriano Holanda Pereira, pelo tempo e pelas valiosas colaborações e sugestões.

Aos colegas feitos durante a graduação, pelas reflexões, críticas e sugestões recebidas.

O fusca te leva ao mesmo lugar que a BMW. Cachorro quente mata a fome da mesma forma que caviar, a água de qualquer praia é salgada como a de Fernando de Noronha. Você só não aproveita sua vida porque quer ter a vida dos outros. Esquecemos que as oportunidades vão embora e só nos resta gratidão pelo que temos. Felicidade é um estado e não um bem material. (Alexandre Magno Abrão)

RESUMO

O presente trabalho apresenta o desenvolvimento de uma interface gráfica amigável para um sistema de emulação de cargas mecânicas. Este emulador é utilizado para ensaios de motores de indução trifásicos de até 10 cv. A Interface Humano-Máquina (IHM) é elaborada no ambiente de desenvolvimento integrado Lazarus[®], rodando em um *Raspberry Pi*[®] e faz uso da biblioteca PascalSCADA[®] e de um *script* em *Python* para a leitura de velocidade do motor. O sistema é desenvolvido sob o conceito de *software* SCADA (*Supervisory Control and Data Acquisition*) alinhando-se assim a filosofia de reprodução de um ambiente industrial do laboratório onde este projeto foi desenvolvido, tem uma interface intuitiva, capaz de exibir e armazenar as grandezas elétricas e mecânicas do motor durante os ensaios para uma posterior análise. O trabalho inicia com uma curta exposição sobre a importância dos estudos em eficiência energética e do motor de indução trifásico (MIT) para a indústria, passa então para a descrição da bancada utilizada e dos requisitos da IHM. Na sequência apresenta os softwares utilizados no desenvolvimento da IHM, e o projeto da interface propriamente dito. Em seguida, mostra o resultado final da interface gráfica, faz algumas análises e conclusões sobre o mesmo.

Palavras-chave: Emulador de carga mecânica, Lazarus, *RaspBerry Pi*.

ABSTRACT

The present term paper supports the development of a friendly graphical interface for a mechanical load emulation system. This emulator is used for testing three-phase induction motors up to 10 hp. The Human Machine Interface (HMI) is built on the Lazarus® integrated development environment, running on a Raspberry Pi® and makes use of the PascalSCADA® library and a Python script for engine speed reading. The developed system under the concept of SCADA (Supervisory Control and Data Acquisition) software, thus aligning the philosophy of reproduction of an industrial laboratory environment where this project was developed. The developed system has an intuitive interface, capable of displaying and storing the electrical and mechanical quantities of the motor during the tests for further analysis. The work begins with a brief presentation on the importance of energy efficiency studies and the three-phase induction motor (MIT) for industry, then goes on to describe the bench used and the requirements that the HMI needs to have. Following is the software used in the development of HMI, and the interface design itself. Subsequently, it shows the final result of the graphical interface, makes some analysis and conclusions about it.

Keywords: Mechanical load emulation, Lazarus, *RaspBerry Pi*.

LISTA DE ILUSTRAÇÕES

Figura 1 – Gráficos genéricos do tipos de cargas que se pretende emular.....	19
Figura 2 – Bancada do emulador de carga.	20
Figura 3 – Emulador de cargas mecânicas	20
Figura 4 – Diagrama de blocos do emulador de carga mecânica.	23
Figura 5 – O IDE do Lazarus.	26
Figura 6 – Versão Final da IHM.....	28

LISTA DE GRÁFICOS

Gráfico 1 – Percentual de consumo de energia no Brasil por fonte	15
Gráfico 2 – Participação setorial no consumo de eletricidade	16

LISTA DE ABREVIATURAS E SIGLAS

PROCEL	Programa Nacional de Conservação da Energia Elétrica
IHM	Interface humano máquina
MIT	Motor de indução trifásico
MME	Ministério de Minas e Energias
EPE.	Empresa de Pesquisa Energética
LAMOTRIZ	Laboratório de sistemas motrizes
USB	<i>Universal Serial Bus</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
OS	Sistema Operacional
GPL	<i>General Public Licence</i>
IDE	<i>Integrated Development Environment</i>
RAD	<i>Rapid Application Development</i>

LISTA DE SÍMBOLOS

°C	Graus Celsius
°	Graus
A/D	Analógico Digital
D/A	Digital Analógico
cc	Corrente contínua
mm	Milímetros

SUMÁRIO

1	INTRODUÇÃO	15
2	O emulador de carga mecânica	18
2.1	Definição	18
2.1.1	Objetivos do emulador de carga mecânica	18
2.2	A bancada	19
2.3	Detalhamento do emulador	20
2.3.1	Freio Eletromagnético	21
2.3.2	Conversor cc-cc <i>chopper</i>	21
2.3.3	Arduino	21
2.3.4	Conversor de frequência e <i>Soft starter</i>	21
2.3.5	Medidor Multigrandezas e Conversor USB-i-485	22
2.3.6	Sensor de temperatura	22
2.3.7	Sensor de Velocidade	22
2.3.8	Raspberry Pi	22
2.4	Diagrama do sistema emulador de carga mecânica	23
3	A IHM	23
3.1	Os requisitos da IHM	24
3.2	O Sistema Operacional Raspbian	24
3.3	Lazarus	24
3.4	Python	27
4	Resultados	28
5	Conclusões	29
	Referências	30

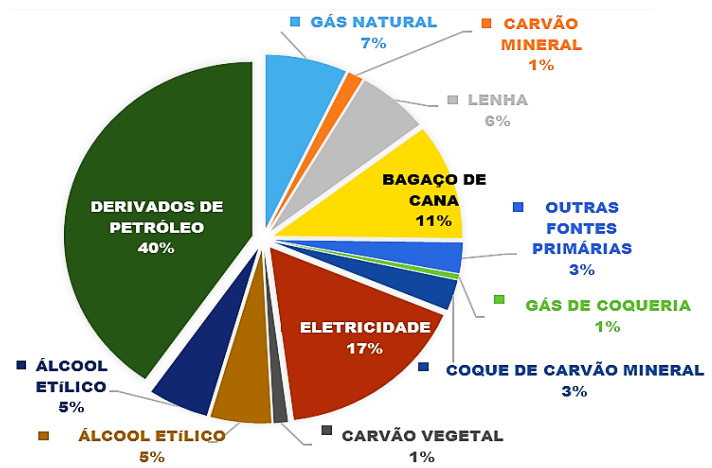
1 INTRODUÇÃO

Existe uma forte correlação entre desenvolvimento econômico e tecnológico e consumo de energia. O desenvolvimento econômico e tecnológico, já contribuiu para crises de desabastecimentos e impactos ambientais negativos, como exemplo no Brasil, pode-se citar a expansão de termo elétricas devido ao racionamento de energia elétrica ocorrido no ano de 2001. Fatos semelhantes chamaram a atenção para a escassez dos recursos naturais, impulsionando a conscientização da importância da preservação do meio ambiente.

Surge então o conceito de eficiência energética visando à redução dos impactos ambientais através do desenvolvimento de sistemas mais eficientes, capazes de realizar os mesmos trabalhos com um menor consumo de energia. É seguindo esta linha de pensamento que, em 1985 o Brasil avançou um passo importante na área com a criação Programa Nacional de Conservação da Energia Elétrica (PROCEL) pelos Ministérios de Minas e Energia e da Indústria e Comércio. Devido aos resultados do programa e a situação favorável dentro e fora do país, em 1991 o PROCEL foi transformando em um programa de governo. O PROCEL atua em diversos setores da sociedade (educação, indústria, poder público e comércio) visando à eficiência da produção e do consumo de energia elétrica com o objetivo de eliminar os desperdícios e reduzir os custos. (BRASIL 2006)

Segundo o Balanço Energético Nacional (BEN 2018) (EPE 2018) a energia elétrica corresponde a aproximadamente 17% do total da energia consumida no país, conforme mostra a Gráfico 1.

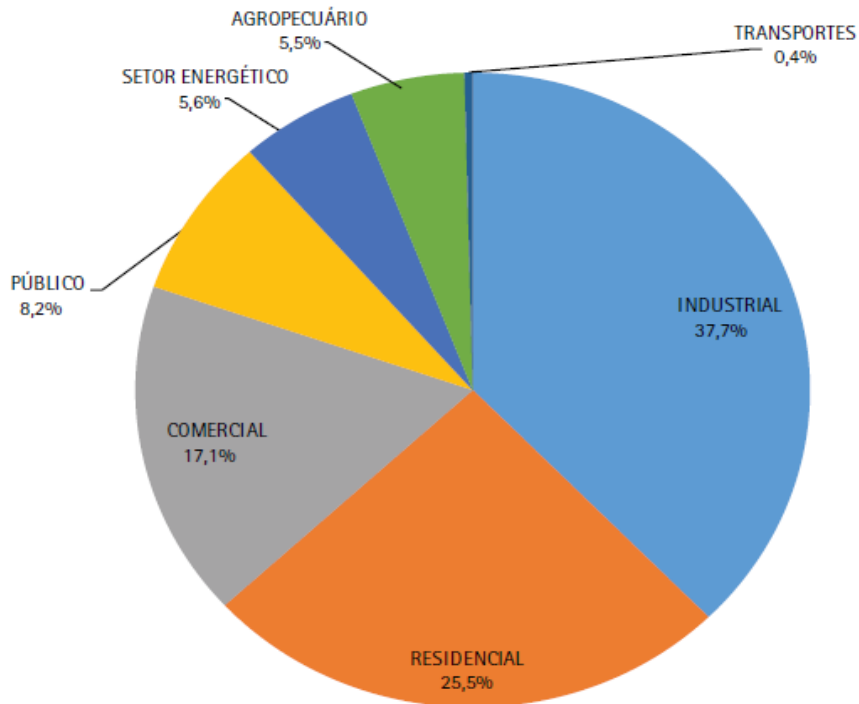
Gráfico 1 – Percentual de consumo de energia no Brasil por fonte.



Fonte: adaptado de EPE(2018)

O BEN 2018 mostra que dentre todos os setores da economia, a indústria é o maior consumidor de energia elétrica do país com cerca de 37,7% do total. O Gráfico 2 apresenta a composição do consumo de energia elétrica no Brasil por setores da economia.

Gráfico 1 – Participação setorial no consumo de eletricidade



Fonte: EPE (2018)

Segundo o Ministério de Minas e Energia (MME), os motores elétricos de indução trifásicos (MITs) são responsáveis por consumir a maior parcela de energia elétrica no setor industrial. Os MITs ainda apresentam um grande potencial de conservação de energia, seja pela quantidade instalados ou por aplicações ineficientes. BRASIL (2019)

Sob este ponto de vista, em que o desenvolvimento de equipamentos mais eficientes se faz essencial, surge no departamento de engenharia elétrica da Universidade Federal do Ceará o laboratório de sistemas motrizes (LAMOTRIZ), capaz de ensaiar as principais plantas existentes em uma indústria. Dentre as plantas existentes no laboratório existe uma para ensaio de MIT, que utiliza como carga um freio eletromagnético descrito em PEREIRA (2006), esta bancada passou por uma primeira atualização onde foi proposto uma maneira de controlar a corrente aplicada ao freio e consequentemente o torque aplicado ao motor sob teste CORREIA et al. (2009), em 2018 esta bancada passou por uma nova atualização, que foi o desenvolvimento de um sistema capaz de emular os principais tipos de cargas mecânicas tracionadas por um MIT como descrito em MORI (2018). O presente trabalho tem como

objetivo mostrar o desenvolvimento da interface gráfica para o emulador de carga mecânica proposto por MORI (2018) utilizando o conceito de software SCADA para possibilitar a integração futura entre todas as bancadas do LAMOTRIZ.

2 O emulador de carga mecânica

Os primeiros passos no desenvolvimento de uma IHM é conhecer as necessidades tanto do processo quanto do usuário final. Esse capítulo faz uma descrição do emulador de carga proposto por MORI (2018).

2.1 Definição

Emulador de carga mecânica é todo sistema capaz de reproduzir o comportamento do conjugado em função da velocidade mecânica para cargas reais como bombas de água, ventiladores, esteiras, veículos automotores entre outros.

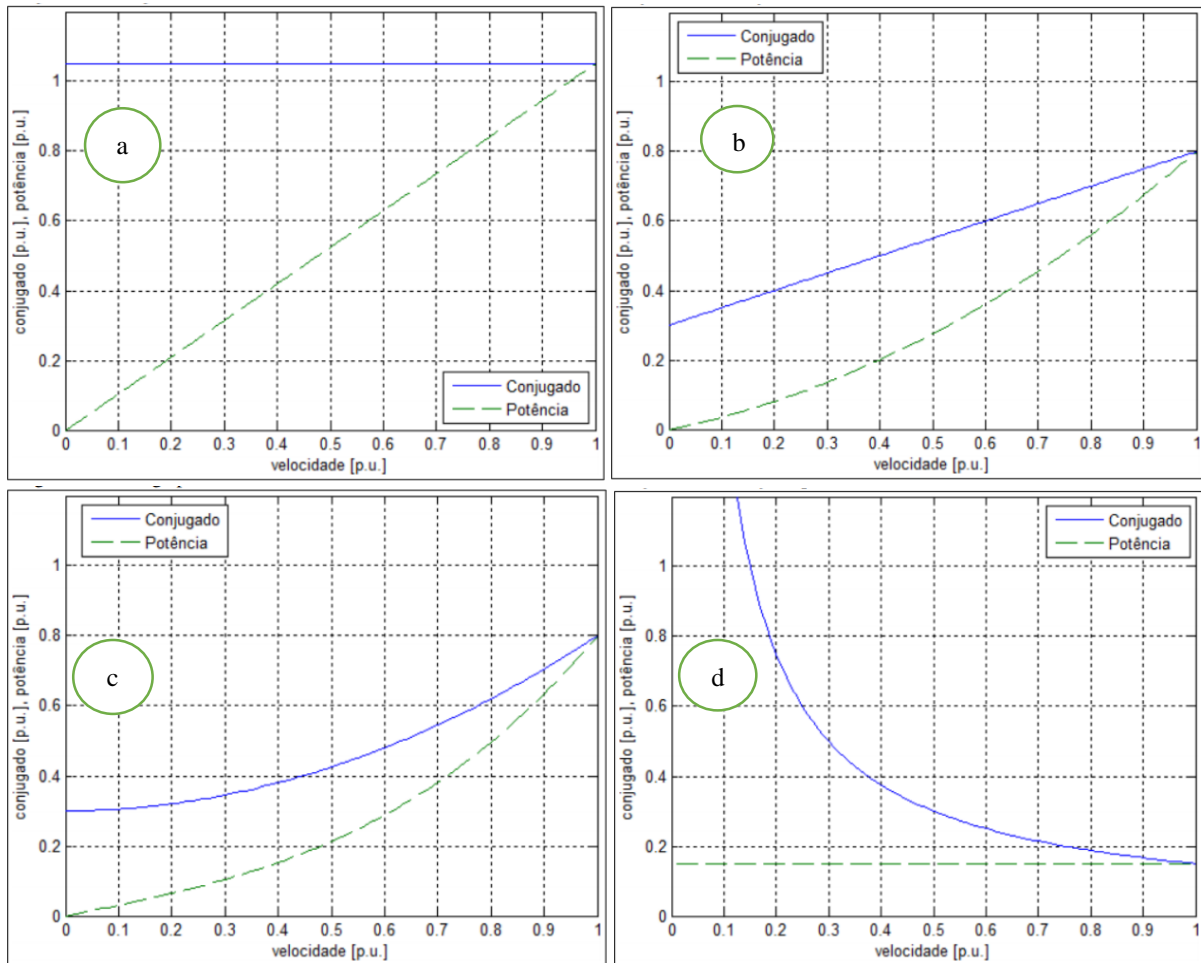
2.1.1 Objetivos do emulador de carga mecânica

O emulador de carga tem como objetivo reproduzir os quartos principais perfis de carga encontrados na indústria que são:

- a) Carga com conjugado constante: Cargas cujo o conjugado não é alterado com a variação da velocidade. Como exemplos tem-se compressores e bombas a pistão e esteiras transportadoras de fluxo contínuo.
- b) Carga de conjugado linear: Cargas cujo o conjugado é alterado linearmente com a variação da velocidade. Como exemplos tem-se bombas a vácuo e sistemas de acoplamento hidráulico ou eletromagnético.
- c) Carga de conjugado parabólico: Cargas cujo o conjugado é alterado com o quadrado da velocidade. Como exemplos tem-se bombas centrífugas e ventiladores.
- d) Carga de conjugado hiperbólico: Cargas cujo o conjugado é alterado com o inverso da variação da velocidade. Como exemplos tem-se bobinadeiras e máquinas de sonda e perfuração de petróleo.

Na Figura 1 tem-se os gráficos de conjugado e potência para cada tipo de carga que se pretende emular. A linha contínua representa o conjugado e a linha tracejada representa a potência mecânica.

Figura 1 – Gráficos genéricos do tipos de cargas que se pretende emular. a) carga com conjugado constante, b) carga com conjugado linear, c) carga com conjugado parabólico e d) carga com conjugado hiperbólico.

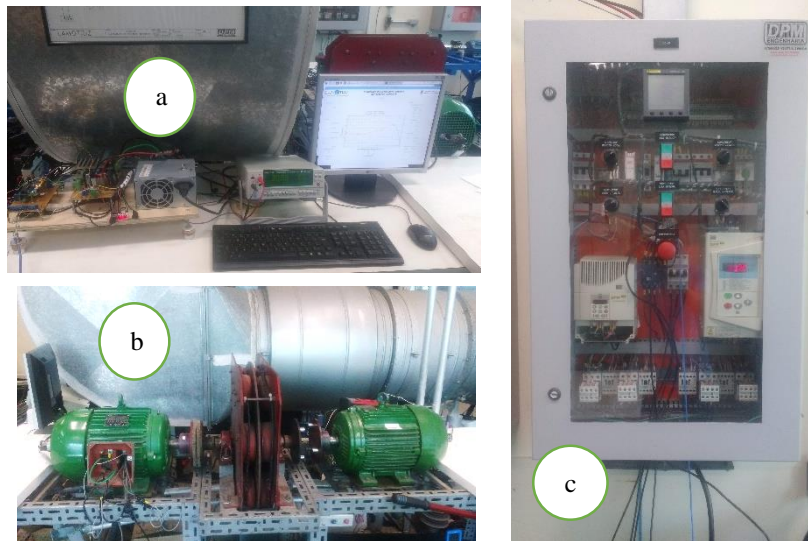


Fonte: adaptado de Weg (2008)

2.2 A bancada

A bancada do emulador de carga é composta por um freio eletromagnético, um conversor cc-cc chopper, um Arduino UNO, um Raspberry Pi II, um conversor de frequência, uma soft start, um medidor multigrandezas, um sensor de temperatura, um sensor de velocidade e um conversor USB-i-485, dois ventiladores e dois MITs. A Figura 2 mostra as partes que compõem a bancada.

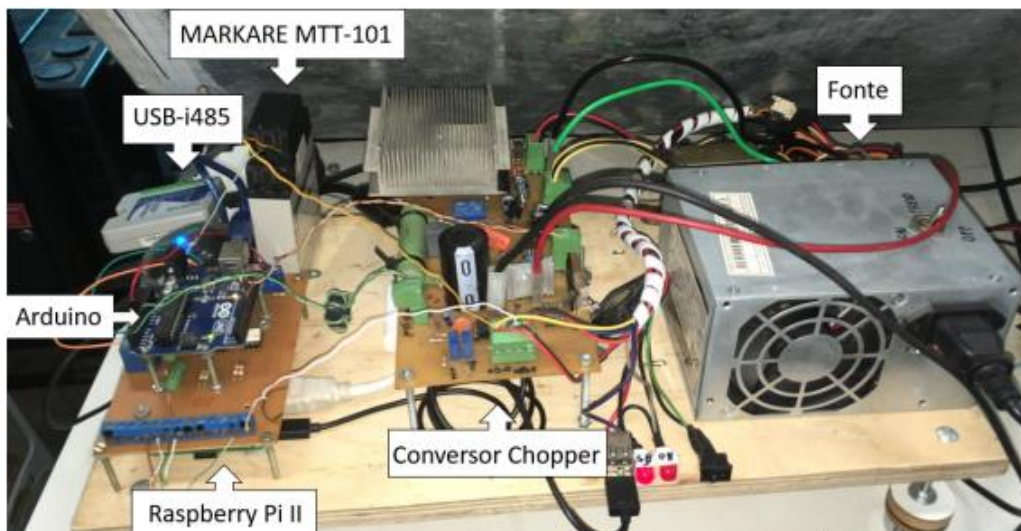
Figura 2 – Bancada do emulador de carga. a) Bancada de operação, b) Bancada com o freio e os motores ser ensaiados, e c) quadro de comando dos motores com a soft start, o conversor, as proteções do motor e o medidor multigrandezas



Fonte: o autor

Na Figura 3 o sistema montado por MORI (2018).

Figura 3 – Emulador de cargas mecânicas



Fonte: (MORI, 2018)

2.3 Detalhamento do emulador

Na seção anterior foi descrito os componentes que formam o emulador de carga mecânica, agora será descrito a função de cada componente e como eles interagem entre si.

2.3.1 Freio Eletromagnético

Responsável por impor torque ao motor, funciona baseado na lei de indução magnética de Faraday/Lenz que diz que a variação de fluxo magnético em um material condutor faz surgir uma força eletromotriz com o intuito de opor-se à variação de fluxo.

Ao circular uma corrente elétrica contínua nas bobinas do freio, surge um campo magnético que envolve o disco. O disco é acoplado ao eixo do motor de forma direta.

Quando o motor impõe movimento ao disco, este último, passa a ver o campo se deslocando em sentido contrário a seu movimento, pela lei de Faraday/Lenz aparece no disco uma força eletromotriz. Como o disco é feito de material condutor elétrico, a tensão induzida faz com que passe a circular corrente elétrica no disco, essa corrente induzida no disco, faz surgir um campo magnético, que por sua vez interage com o campo gerado nas bobinas fazendo aparecer forças no disco que se opõem ao movimento do mesmo, impondo assim torque ao eixo do motor. As correntes induzidas no disco são conhecidas como correntes de Foucault. (HAYT JUNIOR; BUCK, 2013)

2.3.2 Conversor cc-cc *chopper*

O conversor utilizado é do tipo *Chopper* que reduz a tensão de saída em relação à tensão de entrada proporcionalmente ao ciclo de trabalho da chave. Como a carga tem uma indutância significativa, a corrente elétrica se mantém com baixa oscilação a cada chaveamento. (MORI, 2018)

2.3.3 Arduino

Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, inicialmente a ideia era utilizar apenas a *Raspberry Pi*, mas como ela não tem conversor analógico digital (A/D), optou-se em utilizar um Arduino, pois o mesmo já possui conversores A/D, a troca de informações com a *Raspberry Pi* se torna mais fácil, pois ocorre por comunicação serial e o principal, já estava disponível no laboratório. O Arduino é utilizado para acionar o conversor, fazer o controle de corrente do freio e ler a temperatura do motor.

2.3.4 Conversor de frequência e *Soft starter*

São equipamentos de acionamento de MITs. A *soft starter* apenas faz a partida do motor enquanto o conversor permite a variação de velocidade do motor.

2.3.5 Medidor Multigrandezas e Conversor USB-i-485

O medidor multigrandezas é responsável por medir as grandezas elétricas do motor tais como tensão, corrente, potência, fator de potência e taxa de distorção harmônica. É possível conectá-lo a uma rede industrial 485 utilizando o protocolo MODBUS.

O conversor usb-i-485 é o drive de conexão da Raspberry com o barramento 485.

2.3.6 Sensor transdutor de temperatura

O sensor de temperatura fixado no interior do motor é um PT 100 com faixa de medição que varia de 0 °C a 600 °C. Ele recebe o nome de transdutor porque a temperatura é convertida em sinal de tensão na escala de milivolts, esse sinal passa por um transmissor, o MAR-KARE MTT-101, que converte essa escala de milivolts em uma escala padronizada, linear de 0 V a 10 V, para que a temperatura possa ser lida com o Arduino.

2.3.7 Sensor de Velocidade

O sensor utilizado para medir velocidade é um *encoder* incremental com resolução de cinco pulsos por revolução, com dois canais defasados de 90 ° e um sinal de passagem por zero.

2.3.8 O Raspberry Pi

O Raspberry Pi é um computador de placa única com dimensões bem reduzidas, 84 mm de largura e 54 mm de comprimento aproximadamente, que utiliza prioritariamente um cartão microSD como armazenamento. Possui todas as vantagens de um computador normal como interfaceamento com periféricos via USB (*universal serial bus*), porta para conexão com a internet, saída de vídeo e de áudio, conector para câmera digital e com a vantagem de ter disponível portas digitais programáveis de entrada e saída.

O Raspberry Pi é responsável pela interface entre o operador e o emulador de carga, pela leitura da velocidade do motor sob teste e da leitura das grandezas elétricas junto ao medidor multigrandezas via rede 485 e protocolo MODBUS.

3 A IHM

3.1 Os requisitos da IHM

Após conhecer a proposta do emulador é possível definir alguns pontos obrigatórios da IHM:

- a) Um campo para que o usuário informe qual o perfil de carga a ser emulado.
- b) Um campo onde se possa monitorar as grandezas elétricas do motor.
- c) Um campo para as grandezas mecânicas.

3.2 O Sistema operacional Raspbian

O Raspberry, como dito anteriormente é um computador com dimensões reduzidas, e como todo computador precisa de um sistema operacional (OS). Existem algumas opções disponíveis como o Windows IoT Core da Microsoft, o UBUNTU core, Raspbian, entre outros. O Raspbian foi o escolhido para esse projeto por ter maior suporte na internet devido ao fato de ser o OS oficial dos desenvolvedores do Raspberry Pi. (RASPBERRY PI FOUNDATION, 2017).

3.3 O Lazarus

Após a definição do OS, vem a escolha da plataforma onde a IHM será desenvolvida. Neste ponto existem muitas opções como Processing, Qt creator IDE, Eclipse, Lazarus, dentre outras.

O Lazarus foi o escolhido pelos seguintes motivos:

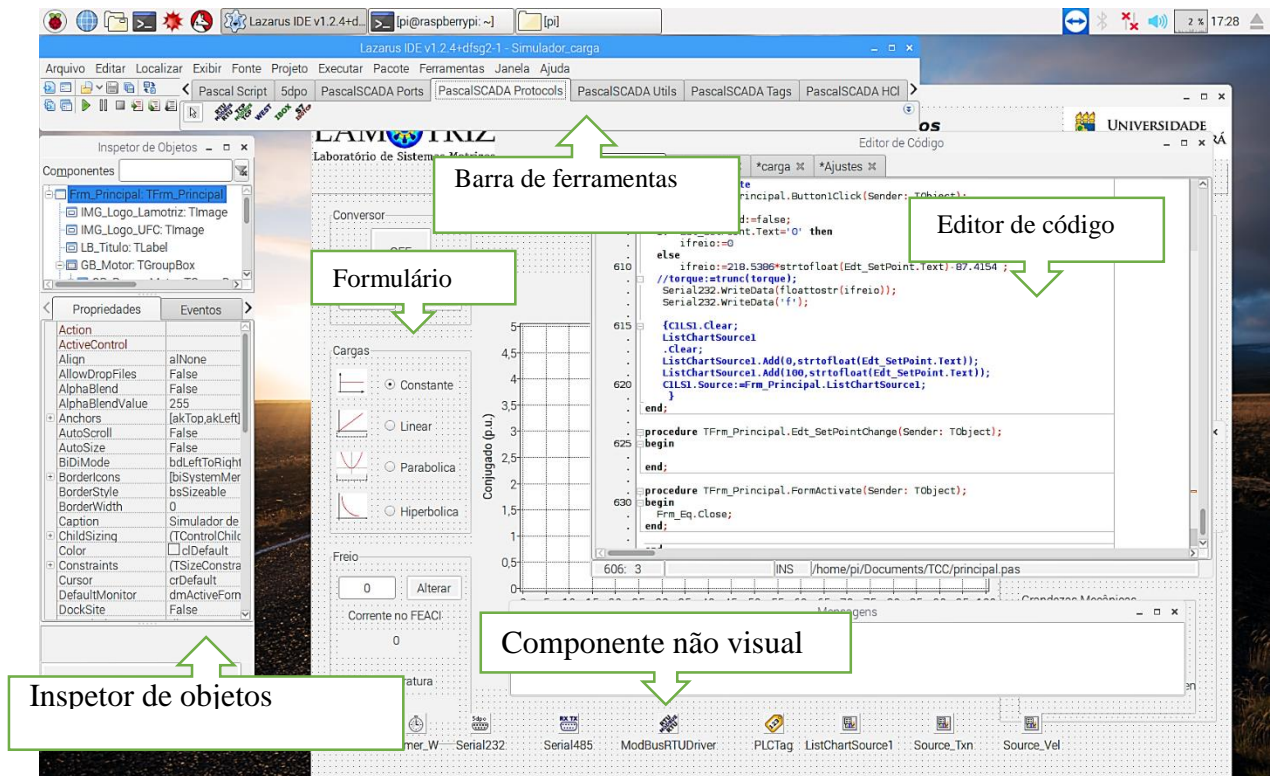
- O fato de ser multiplataforma, possibilita compilar aplicativos para executarem em Windows, Linux, Mac OS X ou outra plataforma compatível.
- É baseado em um modelo RAD (*Rapid Application Development*) ou em português, desenvolvimento rápido de aplicação, já traz consigo vários componentes e métodos prontos.
- Possui um *plugin* chamado PascalSCADA que permite que se crie aplicações IHM/SCADA (PASCALSCADA, 2017) como as existentes na indústria. Alinhando-se assim a filosofia de reprodução de um ambiente industrial do LAMOTRIZ, laboratório onde esse projeto foi desenvolvido.

O Lazarus é um ambiente de desenvolvimento integrado (IDE) de código fonte aberto para o compilador Free Pascal, desenvolvido sob a licença GPL (*General Public Licence*). (SANTOS, 2011)

Na Figura 5 é apresentada a IDE do Lazarus. Em destaque tem-se:

- A janela do editor de código, onde se localiza os códigos do aplicativo.
- A barra de ferramentas onde encontra-se além das opções de configuração do IDE a paleta de componentes que são utilizados para a configuração do aplicativo, tais como botões, caixa de texto, componentes para comunicação entre outros, alguns componentes são visuais e outros não. Os visuais são aqueles que ficam visíveis durante a execução do programa como botões e caixa de texto, já os não visuais só são visíveis enquanto o programa está sendo editado como por exemplo temporizador e componentes de comunicação.
- Um Formulário onde fica a IHM propriamente dita, uma IHM pode ser composta por diversos formulários.
- O inspetor de objetos, onde se tem todos os objetos e formulários do aplicativo com suas propriedades e eventos. (SOURCEFORGE, 2017)
- O objeto responsável pela comunicação MODBUS que só podem ser vistos em modo de programação, é um componente não visual.

Figura 5 – O IDE do Lazarus com destaque para a barra de ferramentas, a janela de edição de códigos, um formulário da IHM composto por componentes visuais e não visuais e o inspetor de objetos.



Fonte: o autor

A criação de uma IHM no Lazarus é muito simples e rápida, basta ir na barra de ferramentas, na paleta de componentes, escolher o componente e clicar no formulário para que o componente escolhido apareça no formulário. Para posicionar o componente é só clicar com o botão esquerdo do mouse, segurar e arrastar. Editar as propriedades de um componente também é simples, é só selecionar o componente no formulário e ir na janela do inspetor de objetos na aba propriedades. É possível editar todo *layout* do formulário sem que o programador escreva uma única linha de código.

Após configurar o *layout* do formulário, chega a hora de escrever algum código, dizer o que acontece quando o usuário interagir com os componentes.

Todos os componentes têm eventos associados a eles, tais como ao ser clicado (*OnClick*) ou ao passar o mouse sobre (*OnMouseMove*) dentre outros muitos. Ao selecionar o evento, a janela de edição de código toma o foco no IDE já na posição onde se deve digitar o código do evento selecionado.

3.4 A linguagem de programação Python

A medição da velocidade feita no Lazarus apresentava inconsistências, para contornar este problema optou-se por fazer uma codificação em Python.

O Python é uma linguagem interpretativa, nativa do Raspbian e muito semelhante a linguagem C, fatores esses que motivaram sua escolha para resolver o problema da leitura da velocidade. (PYTHON SOFTWARE FOUNDATION, 2018)

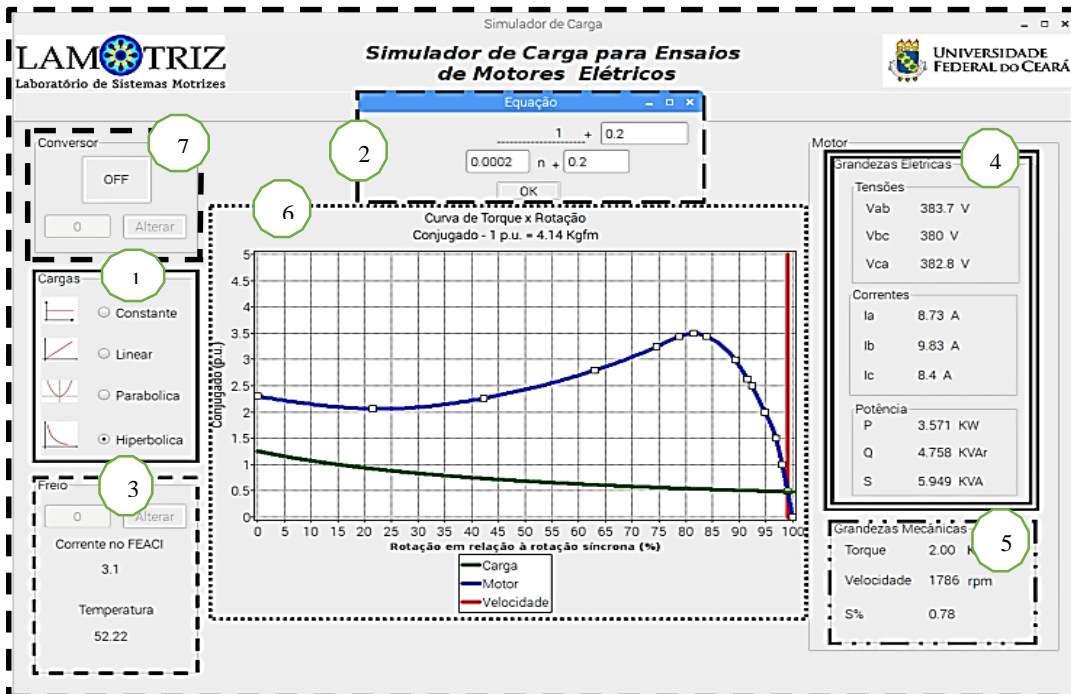
Ao iniciar a aplicação, a mesma iniciava o script em segundo plano. O script calculava a velocidade e enviava a aplicação de forma transparente ao usuário. Alterando um arquivo de texto é possível alterar parâmetros do script em tempo de execução, tais como resolução do *encoder* em caso de substituição do mesmo, e parâmetros do filtro aplicado na leitura de velocidade.

Posteriormente descobriu-se que o conversor utilizado para acionar o motor causava ruídos na leitura de velocidade. Isso devido ao Raspberry trabalha com a tensão de 3,3 V e a fonte de alimentação do mesmo não ser isolada.

4 Resultados

A Figura 6 mostra a versão final da IHM construída.

Figura 6 – Versão Final da IHM



Fonte: adaptado de MORI (2018)

1. Campo para seleção do tipo de carga
2. Janela para edição da equação correspondente a carga selecionada, com exceção da carga constante que é alterada em 3.
3. Campo para inserção do valor da carga quando a mesma for constante e visualização da corrente aplicada ao freio e da temperatura do motor.
4. Campo com as grandezas elétricas do motor, Tensão aplicada, corrente e potências.
5. Campo com as grandezas mecânicas do motor (conjugado, velocidade e escorregamento).
6. Gráfico do ensaio.

5 Conclusões

O objetivo deste trabalho foi alcançado com o desenvolvimento de uma interface humano máquina bastante intuitiva para o emulador de carga, com a capacidade de armazenar e exportar as medições para uma posterior análise.

O Lazarus junto com o *plugin* PascalSCADA mostrou-se uma ferramenta para construção IHM, seja uma simples ou com o conceito de sistemas SCADA, permite uma construção rápida de aplicativos com suporte aos mais famosos bancos de dados, muitos componentes prontos, e um vasto suporte online. Mostrou também ter uma limitação quando se precisa trabalhar com tempos em escala menor do que segundos como no caso da leitura de velocidade.

A Raspberry Pi utilizada mostrou alguns problemas que foram baixa imunidade a ruídos e ausência de conversores A/D e D/A, mas mesmo assim uma ótima solução para embarcar sistemas.

Como sugestões para trabalhos futuros ficam:

- Melhorar o sistema de armazenamento de dados.
- Implementar mais funcionalidades na IHM.
- Utilizar o PascalSCADA para fazer o sistema supervisor do laboratório LAMOTRIZ.

REFERÊNCIAS

- BRASIL. Procelinfo. Eletrobras (Ed.). **O Programa**. 2006. Disponível em: <<http://www.procelinfo.com.br/main.asp?ViewID={A84BD56D-D750-477C-8E20-2BF2D94B4EE2}>>. Acesso em: 17 abr. 2018.
- EPE [Empresa de Pesquisa Energética] Balanço Energético Nacional (BEN) 2018: Ano base 2017. Disponível em <http://epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-303/topico-419/BEN2018__Int.pdf>. Acesso em 17 abr. 2018.
- BRASIL. MME. . MME abre consulta pública sobre eficiência de Motores Elétricos Trifásicos. Disponível em: <http://www.mme.gov.br/web/guest/pagina-inicial?p_p_id=101&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&_101_struts_action=%2Fasset_publisher%2Fview_content&_101_returnToFullPageURL=%2F&_101_assetEntryId=4436237&_101_type=content&_101_urlTitle=mme-abre-consulta-publica-sobre-eficiencia-de-motores-eletricos-trifasicos&redirect=http%3A%2F%2Fwww.mme.gov.br%2Fweb%2Fguest%2Fpagina-inicial%3Fp_p_id%3D3%26p_p_lifecycle%3D0%26p_p_state%3Dmaximized%26p_p_mode%3Dview%26_3_groupId%3D0%26_3_keywords%3Dmotores%2Bel%25C3%25A9tricos%2Bconsumo%2Bde%2Benergia%26_3_struts_action%3D%252Fsearch%252Fsearch%26_3_redirect%3D%252F&inheritRedirect=true>. Acesso em: 05 ago. 2019.
- PEREIRA, Adriano Holanda. FREIO ELETROMAGNÉTICO PARA ENSAIOS DE MOTORES ELÉTRICOS DE INDUÇÃO. 2006. 108 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Engenharia Elétrica, Universidade Federal do Ceará, Fortaleza, 2006. Disponível em: <http://repositorio.ufc.br/bitstream/riufc/18699/1/2006_dis_ahpereira.pdf>. Acesso em: 23 mar. 2017.
- CORREIA, Wilkley B et al. IDENTIFICAÇÃO NÃO-LINEAR BASEADA EM INFERÊNCIA FUZZY NA MODELAGEM DE UM FREIO ELETROMAGNÉTICO. In: SBAI, 2009, Brasília. **Anais**. Brasília: Sbai, 2009. p. 1-6. Disponível em: <<https://fei.edu.br/sbai/SBAI2009/artigos/SISTEMAS%20NEBULOSOS/55809.pdf>>. Acesso em: 14 abr. 2018.
- MORI, Marcio Massao Nishide. Emulador de Cargas Mecânicas com Freio Eletromagnético para Ensaios de Motores Elétricos de Indução. 2018. 113 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Engenharia Elétrica, Universidade Federal do Ceará, Fortaleza, 2018. Disponível em: <<http://www.repositorio.ufc.br/handle/riufc/38800>>. Acesso em: 3 mar. 2019.
- WEG. Modulo1 Comandos e Proteção. 2008. Disponível em: <[https://docente.ifsc.edu.br/rafael.grebogi/MaterialDidatico/Eletromecanica/EletricidadeIndustrial\(OLD\)/M1-ComandoeProteç~ao.pdf](https://docente.ifsc.edu.br/rafael.grebogi/MaterialDidatico/Eletromecanica/EletricidadeIndustrial(OLD)/M1-ComandoeProteç~ao.pdf)>. Acesso em: 25 mai. 2017.
- HAYT JUNIOR, William H.; BUCK, John A.. Eletromagnetismo. 8. ed. Porto Alegre: Mcgraw Hill Brasil, 2013. (8563308769, 9788563308764).
- RASPBERRY PI FOUNDATION (Uk) (Ed.). Raspberry Pi. Disponível em: <<https://www.raspberrypi.org/>>. Acesso em: 05 nov. 2017.

SOURCEFORGE. **Lazarus**. Disponível em: <<https://www.lazarus-ide.org/>>. Acesso em: 05 nov. 2017.

SANTOS, Jean Patrick Figueiredo dos. Desenvolvendo Aplicativos com Lazarus Linux & Windows. [s.l.]: Clube de Autores, 2011. 346 p.

PASCALSCADA. PascalSCADA. Disponível em: <<http://www.pascalcada.com/pb/>>. Acesso em: 5 nov. 2017.

PYTHON SOFTWARE FOUNDATION. PYTHON. Disponível em: <<https://www.python.org/>>. Acesso em: 11 jan. 2018.