



UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
COLEGIADO DO CURSO DE ENG. ELÉTRICA



João Pedro Caires Ferreira

SIMULOAD: Simulador de Curvas de Carga

Orientador: Prof. Dr. André Luiz de Carvalho Valente

Salvador-Ba - Brasil

14 de Julho de 2023

João Pedro Caires Ferreira

SIMULOAD: Simulador de Curvas de Carga

Projeto apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal da Bahia como parte dos requisitos para a obtenção do grau de Engenheiro(a) Eletricista.

Orientador: Prof. Dr. André Luiz de Carvalho Valente

Salvador-Ba - Brasil

14 de Julho de 2023

João Pedro Caires Ferreira

SIMULOAD: Simulador de Curvas de Carga

Projeto apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal da Bahia como parte dos requisitos para a obtenção do grau de Engenheiro(a) Eletricista.

Trabalho aprovado. Salvador-Ba – Brasil, 14 de Julho de 2023:

Orientador(a): Prof. Dr. André Luiz de Carvalho Valente

Prof. Dr. Fernando Augusto Moreira

Prof. Dr. Karcus Day Rosário de Assis

**Salvador-Ba – Brasil
14 de Julho de 2023**

Este trabalho é dedicado a todos que influenciaram de alguma forma na minha carreira e hábitos. Hoje caminho em direção ao melhor futuro que posso ter e não chegaria nele sozinho. Desejo o melhor para vocês e espero ter retribuído de alguma forma.

Agradecimentos

Agradeço ao meu pai, Antônio Geraldo Ferreira, engenheiro eletricista, referência nacional no Conselho Regional de Engenharia e Agronomia da Bahia – CREA-BA e minha primeira imagem sobre excelência. Agradeço à minha mãe Cinara e à minha irmã Beatriz por todo o suporte durante esse período, e o quanto a nossa relação se fortaleceu durante esse período é reflexo disso.

Agradeço à minha namorada Naiara por fazer parte de todas as decisões mais importantes que tomei durante esses 7 anos de curso, que me guiaram para finalizá-lo extremamente satisfeito com a minha trajetória. Agradeço também aos meus sogros Regina e Rogério por terem me acolhido como um filho. Agradeço aos meus padrinhos e tios, ao meu cunhado e aos meus primos por todo o apoio.

Agradeço aos colegas e amigos do curso. Isso é o que tem de mais valioso na universidade. Agradeço a Miguel Damásio, um colega que se tornou um irmão. Agradeço a Caléo Meneses, minha dupla do curso em diversos momentos e que também ajudou toda a nossa turma várias vezes. Agradeço a Henrique Poleselo e Luiz Gustavo, amigos que levo para a vida e desejo o melhor sempre. Agradeço a Amanda Borges pelo nosso contato na empresa júnior, por me fazer enxergar meus principais defeitos e trabalhar nisso, me tornando uma pessoa e profissional melhor. Agradeço aos colegas e amigos Leonardo Pedreira, Paula Garcia, Caio França e Bruno Sales, vejo um futuro brilhante para cada um de vocês. Agradeço a Mariana Guimarães por toda a orientação e auxílio nessa reta final.

Agradeço às lideranças que tive durante todos os meus estágios, tive a sorte de aprender e evoluir muito com profissionais excelentes em todas as experiências que tive. Agradeço também aos professores do curso que realmente se preocupam com o futuro dos alunos, somos muito gratos por termos vocês no departamento.

“Liberdade não é fazer o
que se quer, mas querer
o que se faz.”

(Jean-Paul Sartre)

Resumo

O presente trabalho discorre sobre o desenvolvimento de uma solução didática para aprimorar o aprendizado dos estudantes de Engenharia Elétrica no assunto de curvas de carga no contexto da distribuição de energia elétrica.

A solução foi idealizada como um software, nomeado de Simuload, com interface visual amigável capaz de modelar e simular curvas de carga e curvas características de um transformador para análise e estudo. O software utilizou como base a PPH 2019 – Pesquisa de Posse e Hábitos de Uso de Equipamentos Elétricos na Classe Residencial 2019, adotando suas premissas, sua forma de representar o consumo dos equipamentos, e também seus dados como registros padrões prontos para uso no Simuload.

Os requisitos funcionais e o propósito do software guiaram as escolhas dos recursos e ferramentas para o seu desenvolvimento. A linguagem escolhida foi Python com um banco de dados relacional SQLite e interface gráfica com PyQt5, utilizando o Qt Designer para construção de telas. Além disso, foram utilizados Git e Github para versionamento de código, Visual Studio Code como ferramenta de desenvolvimento, e os módulos Matplotlib, Numpy, PyInstaller e csv. O Simuload também possui um algoritmo de aleatoriedade que permite simular curvas de carga em intervalos menores a partir do mapeamento de hora em hora.

Foi feito um estudo de caso real com uma dupla de estudantes que utilizou o Simuload para a realização de um trabalho de simulação de curvas de carga na disciplina Laboratório Integrado VI no semestre 2023.1 do curso de Engenharia Elétrica da UFBA. Durante o seu uso foram apontados pontos de melhoria e algumas desvantagens do software, mas a sua praticidade e interface amigável o tornam uma alternativa mais vantajosa diante dos métodos tradicionais utilizados para a realização do trabalho. Esta foi a conclusão a partir de uma avaliação qualitativa do uso do Simuload durante o estudo de caso.

Palavras-chave: Simuload, simulação de curvas de carga, distribuição de energia elétrica, software didático, software de simulação.

Abstract

The present study discusses the development of a didactic solution to enhance the learning of Electrical Engineering students in the topic of load curves in the context of electrical distribution. The solution was conceived as a software, named Simuload, with a user-friendly visual interface capable of modeling and simulating load curves and transformer characteristic curves for analysis and study. The software was based on the PPH 2019 – *Pesquisa de Posse e Hábitos de Uso de Equipamentos Elétricos na Classe Residencial 2019*, adopting its premises, its way of representing equipment consumption, and also its data as default records ready for use in Simuload.

The functional requirements and the purpose of the software guided the choices of resources and tools for its development. The chosen language was Python with a relational SQLite database and a graphical interface with PyQt5, using Qt Designer for screen building. In addition, Git and Github were used for code versioning, Visual Studio Code as a development tool, and the modules Matplotlib, Numpy, PyInstaller, and csv. Simuload also has a randomness algorithm that allows simulating load curves at smaller intervals based on hourly mapping.

A real case study was conducted with a pair of students who used Simuload to perform a load curve simulation project in the Integrated Laboratory VI course in the 2023.1 semester of the Electrical Engineering program at UFBA. During its use, some improvement points and disadvantages of the software were identified, but its practicality and user-friendly interface make it a more advantageous alternative compared to traditional methods used for the project execution. This was the conclusion obtained from a qualitative evaluation of the use of Simuload during the case study.

Keywords: Simuload, load curve simulation, electrical distribution, didactic software, simulation software.

Lista de Figuras

<i>Fig. 1 Gráfico UML do Banco de Dados desenhado na ferramenta Miro.</i>	8
<i>Fig. 2 Código fonte do Simuload no editor de código VSCode.</i>	9
<i>Fig. 3 Interface do software Qt Designer e edição da tela de novo equipamento do Simuload.</i>	11
<i>Fig. 4 Visualização da tabela de CargaEquipamento e seu script de criação na extensão SQLite no VSCode.</i>	12
<i>Fig. 5 Pastas e executável que compõem a versão de distribuição para Windows do Simuload.</i>	13
<i>Fig. 6 Dashboard de dados de consumo de lâmpadas na PPH 2019 para a região Nordeste.</i>	15
<i>Fig. 7 Submissão de alterações no código utilizando o Git através do terminal embutido no VSCode.</i>	17
<i>Fig. 8 Mapeamento de consumo dos equipamentos e cargas para simulação de curvas no Google Sheets.</i>	20
<i>Fig. 9 Cálculo das curvas de carga a partir de consumidores residencial, comercial, industrial e iluminação no Google Sheets.</i>	21
<i>Fig. 10 Curva de simulação da demanda dos estabelecimentos pela curva característica do transformador no Google Sheets.</i>	21
<i>Fig. 11 Janela principal do Simuload.</i>	22
<i>Fig. 12 Janela de equipamentos do Simuload.</i>	23
<i>Fig. 13 Janela de criação de equipamento do Simuload.</i>	23
<i>Fig. 14 Janela de cargas do Simuload.</i>	24
<i>Fig. 15 Janela de criação de carga do Simuload.</i>	25
<i>Fig. 16 Janela de criação de curva do Simuload.</i>	26
<i>Fig. 17 Janela de transformadores do Simuload.</i>	27
<i>Fig. 18 Janela de criação de transformador do Simuload.</i>	27
<i>Fig. 19 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 1 hora.</i>	29
<i>Fig. 20 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 30 minutos.</i>	29

<i>Fig. 21 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 15 minutos.....</i>	<i>30</i>
<i>Fig. 22 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 5 minutos.....</i>	<i>30</i>

Lista de Abreviaturas e Siglas

PPH 2019 – Pesquisa de Posse e Hábitos de Uso de Equipamentos Elétricos na Classe Residencial 2019

Procel – Programa Nacional de Conservação de Energia Elétrica

UML – Unified Modeling Language

VSCode – Visual Studio Code

CSV – Comma-separated values

MME – Ministério de Minas e Energia

Sumário

Capítulo 1: Introdução.....	1
1.1 Simuload.....	2
Capítulo 2: Objetivos.....	3
2.1 Objetivo geral.....	3
2.2 Objetivos específicos.....	4
Capítulo 3: Metodologia.....	5
3.1 Desenvolvimento do software.....	5
3.2 Validação.....	6
Capítulo 4: Especificações.....	7
4.1 Banco de dados.....	7
4.2 Módulos e ferramentas.....	8
4.2.1 Ferramenta de edição de código.....	9
4.2.2 Ferramenta de interface visual.....	10
4.2.3 Ferramentas matemáticas.....	11
4.2.4 Manipulação de dados.....	12
4.2.5 Distribuição do software.....	13
4.3 Mapeamento de dados.....	14
4.4 Algoritmo de aleatoriedade.....	15
4.5 Versionamento de código.....	17
Capítulo 5: Estudo de Caso.....	19
5.1 Simulando com outros softwares.....	19
5.2 Aplicação do Simuload.....	22
5.3 Resultados e comparações.....	31

Capítulo 6: Conclusão.....	33
6.1 Objetivos e metodologia.....	33
6.2 Considerações finais.....	34
Referências.....	36
Apêndice A – Código do algoritmo de aleatoriedade.....	38
Apêndice B – Repositório do Simuload.....	40
Apêndice C – Anexo I. Tabelas da norma SM04.14-01.001.....	41

Capítulo 1: Introdução

Sabe-se que no atual cenário brasileiro no que diz respeito ao consumo de energia elétrica, há uma grande necessidade de estudar formas efetivas de aproveitamento dos recursos e suas distribuições. As formas de coletar, mensurar, analisar e utilizar os dados referentes ao consumo de energia são de extrema importância para o país e portanto devem ser abordadas de forma crítica, sobretudo dentro da universidade no curso de engenharia elétrica.

Em uma perspectiva mais geral, o Programa Nacional de Conservação de Energia Elétrica – Procel, foi criado pelo governo com intuito de promover o uso eficiente da energia elétrica e combater o seu desperdício. O programa possui ações que auxiliam na eficiência dos bens e serviços, bem como possibilitam a postergação de investimentos no setor elétrico, reduzindo assim também os possíveis impactos ambientais. Tais ações são fundamentais para a manutenção de um sistema de distribuição eficiente e que vise maior economia e uso consciente dos nossos recursos.

Nessa conjuntura é importante salientar que a simulação de curvas de carga é uma atividade fundamental no planejamento da distribuição de energia elétrica, permitindo que as empresas do setor controlem a demanda de energia de forma eficiente e segura. Com a simulação de curvas de carga, é possível determinar o perfil de consumo de uma determinada rede de distribuição, identificar pontos críticos que podem gerar sobrecargas e avaliar o desempenho da rede em situações de pico de demanda. Ou seja, é parte intrínseca na construção de um planejamento energético eficiente, independente da sua dimensão ou âmbito.

Este é um campo de atuação dentro da engenharia elétrica que permite uma infinidade de possibilidades para o profissional da área. Portanto, é necessário que o estudante, ainda na universidade, tenha contato com ferramentas que contribuam de forma eficiente na sua formação, fazendo com que ele tenha domínio tanto na execução quanto na análise dos dados coletados. A simulação de curvas de carga é uma atividade cujos seus resultados e análises podem ser exploradas tanto no setor público, quanto privado, o que permite ampliar de maneira ainda mais significativa as possibilidades de atuação dentro do mercado de trabalho.

1.1 Simuload

O software Simuload, produto desenvolvido como objeto de estudo deste trabalho, foi pensado e elaborado para ser uma ferramenta importante no aprendizado da simulação de curvas de carga em redes de distribuição de baixa tensão, pois oferece uma maneira eficiente de simular curvas de carga, evitando excesso de trabalho manual e consequentemente conectando o estudante com a parte prática. Durante o estudo de caso feito com o Simuload, foi observada a importância do software no desenvolvimento das habilidades técnicas dos estudantes, que são de extrema importância no campo de atuação da Engenharia Elétrica.

O objetivo deste trabalho de conclusão de curso é apresentar o software Simuload, sua metodologia de desenvolvimento, sua eficácia como ferramenta didática para simulação de curvas de carga em redes de distribuição de baixa tensão, bem como apresentar os resultados obtidos através do experimento. Para isso, o software foi utilizado por uma dupla de alunos em um estudo de caso real, que realizou uma análise dos resultados obtidos através do uso do Simuload na simulação de curvas de carga em uma rede de distribuição de energia elétrica, comparando-os com outros métodos mais tradicionais utilizados para simulação de curvas de carga.

O trabalho está estruturado da seguinte forma: após esta introdução, será apresentada uma revisão bibliográfica sobre a simulação de curvas de carga em redes de distribuição de energia elétrica, seguida pela descrição detalhada do software Simuload e sua metodologia de desenvolvimento. Com esta pesquisa, espera-se contribuir para o desenvolvimento de ferramentas didáticas eficientes para o ensino de simulação de curvas de carga em redes de distribuição de baixa tensão, promovendo uma formação mais completa e qualificada para os estudantes de Engenharia Elétrica.

Capítulo 2: **Objetivos**

O capítulo de objetivos tem como propósito definir de forma clara e concisa os resultados almejados com o desenvolvimento e utilização do software Simuload. Neste capítulo, serão apresentados os objetivos gerais e específicos do trabalho, proporcionando uma visão abrangente das metas a serem alcançadas. Os objetivos estabelecidos serão fundamentais para direcionar o desenvolvimento do software e guiar a análise e avaliação dos resultados obtidos.

2.1 Objetivo geral

O objetivo geral deste trabalho é apresentar o software Simuload e sua metodologia de desenvolvimento, bem como avaliar sua eficácia como ferramenta didática para a simulação de curvas de carga em redes de distribuição de baixa tensão. Para isso, foram realizadas simulações de curvas de carga em uma rede de distribuição de energia elétrica, utilizando o Simuload como ferramenta principal, a fim de demonstrar sua capacidade de simulação de uma forma didática.

A criação do software Simuload foi motivada pela necessidade da existência de uma ferramenta didática que pudesse auxiliar na simulação de curvas de carga em redes de distribuição de baixa tensão, a ser utilizada sobretudo por estudantes de Engenharia Elétrica. Sua interface foi desenvolvida de maneira que se apresentasse de forma intuitiva e de fácil utilização para os usuários no momento da simulação das curvas de carga, permitindo ao usuário otimizar o trabalho e consequentemente concluir a tarefa de forma mais rápida e eficiente.

Por meio da utilização do Simuload, espera-se que seja possível contribuir para o desenvolvimento de habilidades técnicas importantes para a atuação na área de Engenharia Elétrica. Dentre essas habilidades pode-se destacar: a capacidade de simular e analisar curvas de carga em redes de distribuição de energia elétrica e a capacidade de compreender e aplicar conceitos relacionados à gestão de demanda de energia elétrica, o que é de extrema importância no campo prático de atuação, pois através dos dados obtidos com a simulação o profissional desta área deve ser capaz de planejar maneiras eficientes de lidar com as demandas de consumo.

Com isso, espera-se também que este trabalho possa contribuir para a disseminação do conhecimento sobre a simulação de curvas de carga em redes de distribuição de energia elétrica, bem como para o aprimoramento da formação de estudantes e profissionais da área de Engenharia Elétrica.

Por meio da utilização do Simuload, espera-se que seja possível contribuir com o desenvolvimento de habilidades técnicas importantes para a atuação dos profissionais na área de Engenharia Elétrica. Por conseguinte, espera-se que este trabalho possa contribuir para a disseminação do conhecimento sobre a simulação de curvas de carga em redes de distribuição de energia elétrica, bem como sua importância para o aprimoramento da formação de estudantes e profissionais da área de Engenharia Elétrica, sobretudo no que diz respeito ao campo prático de atuação.

O objetivo geral é, portanto, avaliar a eficiência e usabilidade do software Simuload para facilitar o processo de estudo e aprendizagem do assunto de curvas de carga, e os objetivos específicos visam a identificação de limitações e proposição de melhorias para aprimorar o software.

2.2 Objetivos específicos

Os objetivos específicos deste trabalho incluem:

- Avaliação da eficiência e usabilidade do software Simuload para a simulação de curvas de carga;
- Identificação de possíveis limitações ou dificuldades no uso do software;
- Proposição de melhorias e aprimoramentos para torná-lo ainda mais didático e útil para fins de aprendizagem.

Para tanto, serão realizadas simulações de curvas de carga utilizando o software Simuload em diferentes cenários, e os resultados obtidos serão utilizados para avaliar a eficiência do software em facilitar o processo de elaboração e estudo de curvas de carga. Além disso, serão realizadas análises para identificar possíveis limitações ou dificuldades no uso do software, de forma a propor melhorias e aprimoramentos que possam torná-lo ainda mais didático e útil para fins de aprendizagem.

Capítulo 3: Metodologia

Neste capítulo será descrito como o software Simuload foi desenvolvido, incluindo linguagem de programação, ferramentas utilizadas e mapeamento do banco de dados. Além disso, será apresentado o processo de validação da ferramenta, que consiste em avaliar a sua eficácia em simular curvas de carga e atender às necessidades dos estudantes. O Simuload é uma ferramenta desenvolvida para facilitar o processo de elaboração, estudo e aprendizagem de curvas de carga, e a sua metodologia de desenvolvimento é fundamental para garantir a qualidade e eficácia da ferramenta.

3.1 Desenvolvimento do software

Durante o processo de desenvolvimento do Simuload foram estabelecidos alguns requisitos funcionais com o objetivo de tornar a ferramenta prática e completa para o uso dos estudantes. Esses requisitos foram:

1. Desenvolver botões, telas e gráficos de curva totalmente funcionais, permitindo a criação de simulações de curva de carga de redes de distribuição de baixa tensão de forma simples e intuitiva;
2. Criação de um banco de dados funcional com integridade dos dados das entidades, garantindo a correta representação dos dados utilizados nas simulações de curva de carga;
3. Assegurar a integridade dos dados nas curvas simuladas, verificando a precisão das simulações e a consistência dos resultados obtidos;
4. Incluir modelos de exemplo padrões que permitam ao usuário realizar simulações com dados de referência;
5. Calcular a faixa de carregamento do transformador, possibilitando a verificação das limitações de carga em um determinado transformador;
6. Permitir a exportação das simulações para um arquivo CSV, para que os resultados possam ser facilmente analisados e compartilhados.

Esses seis itens foram pensados levando em consideração a experiência vivida nas matérias Laboratório Integrado VI e Transmissão e Distribuição de Energia Elétrica. Ambas fazem parte da grade curricular do sétimo semestre do curso de Engenharia Elétrica da UFBA e são de extrema importância na formação profissional. Visando maior aproveitamento e crescimento dos estudantes nessa fase crucial do curso, todo o conteúdo e trocas vivenciadas nesses dois componentes curriculares serviram de insumos para o processo de desenvolvimento do Simuload.

3.2 Validação

Após a etapa de desenvolvimento e criação, se fez necessária a validação da efetividade da ferramenta. Para tanto, foi realizada uma avaliação qualitativa do seu uso por uma dupla de estudantes no contexto da disciplina Laboratório Integrado VI. Os critérios avaliados incluíram a facilidade de uso, a clareza dos resultados gerados, a compreensão da interface e a adequação dos modelos de exemplo. A avaliação teve como objetivo identificar pontos fortes e fracos da ferramenta e fornecer subsídios para possíveis melhorias. Os resultados desta avaliação serão discutidos de maneira mais abrangente no capítulo cinco, no qual será abordado o estudo de caso de maneira mais detalhada.

Foi estabelecido um diálogo constante com a dupla de estudantes que foi escolhida para utilizar o software em simulações de curvas de carga, possibilitando maior compreensão e aproximação das reais necessidades dos estudantes durante todo o processo de desenvolvimento. Dessa maneira, houve uma aproximação das funcionalidades do software com as principais necessidades dos estudantes, o que tornou o Simuload uma ferramenta bastante didática e eficaz no que se propõe. Essa abordagem colaborativa foi de extrema importância na criação do software, sobretudo por contribuir com maior efetividade no alcance dos objetivos específicos e consequentemente na no objetivo geral do trabalho.

Capítulo 4: Especificações

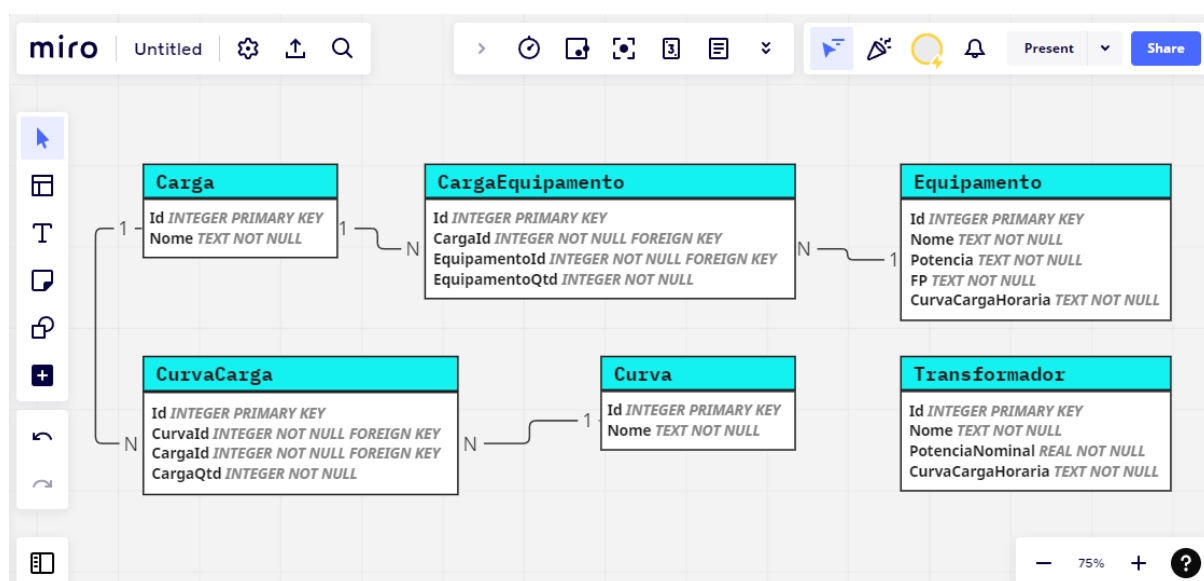
Nesta seção, são apresentadas as especificações técnicas do software Simuload, desenvolvido em Python (PYTHON SOFTWARE FOUNDATION, 2001), abrangendo desde os módulos e ferramentas utilizados até o mapeamento dos dados e o algoritmo de aleatoriedade implementado. Serão discutidas as decisões tomadas em relação às funcionalidades, os requisitos matemáticos e estatísticos, bem como a estrutura do banco de dados utilizado. A seção de especificação é fundamental para compreender as escolhas técnicas feitas ao longo do desenvolvimento do Simuload, garantindo a transparência e a compreensão das decisões tomadas para construir uma ferramenta robusta e eficiente.

4.1 Banco de dados

O Simuload utiliza um banco de dados relacional que inclui quatro entidades: *Equipamento*, *Carga*, *Curva* e *Transformador*. A entidade *Equipamento* representa os equipamentos que compõem uma carga em watts. Já a entidade *Carga* representa um estabelecimento consumidor em watts, podendo ser uma casa, um galpão, um mercado, iluminação pública, entre outros. A entidade *Curva* representa a distribuição temporal do consumo de várias cargas e, conseqüentemente, o consumo de energia elétrica numa região atendida por um transformador. Por fim, a entidade *Transformador* representa o fornecimento na curva característica de um transformador que atende uma região.

As entidades no banco de dados são relacionadas entre si. A entidade *Equipamento* tem uma relação de muitos para muitos com a entidade *Carga*, e para isso existe uma tabela associativa *CargaEquipamento* que relaciona quais *Equipamentos* fazem parte de quais *Cargas*. A entidade *Carga* tem uma relação de muitos para muitos com a entidade *Curva*, e para isso também existe uma tabela associativa *CurvaCarga* que relaciona quais *Curvas* possuem quais *Cargas*. A entidade *Transformador* é independente e possui uma estrutura de demanda distribuída ao longo do tempo, assim como o equipamento. Essas relações permitem que os dados sejam organizados e recuperados de forma eficiente no Simuload e estão representadas no gráfico UML – *Unified Modeling Language* (MIRO, 2023) abaixo, desenhado através da ferramenta Miro (MIRO, 2023).

Fig. 1 Gráfico UML do Banco de Dados desenhado na ferramenta Miro.



4.2 Módulos e ferramentas

Na fase de planejamento e desenvolvimento do Simuload, foram tomadas decisões estratégicas em relação aos módulos e ferramentas a serem utilizados. Para garantir a eficiência e o funcionamento adequado do sistema, uma pesquisa detalhada foi realizada para identificar os pacotes e ferramentas mais adequados às necessidades do projeto.

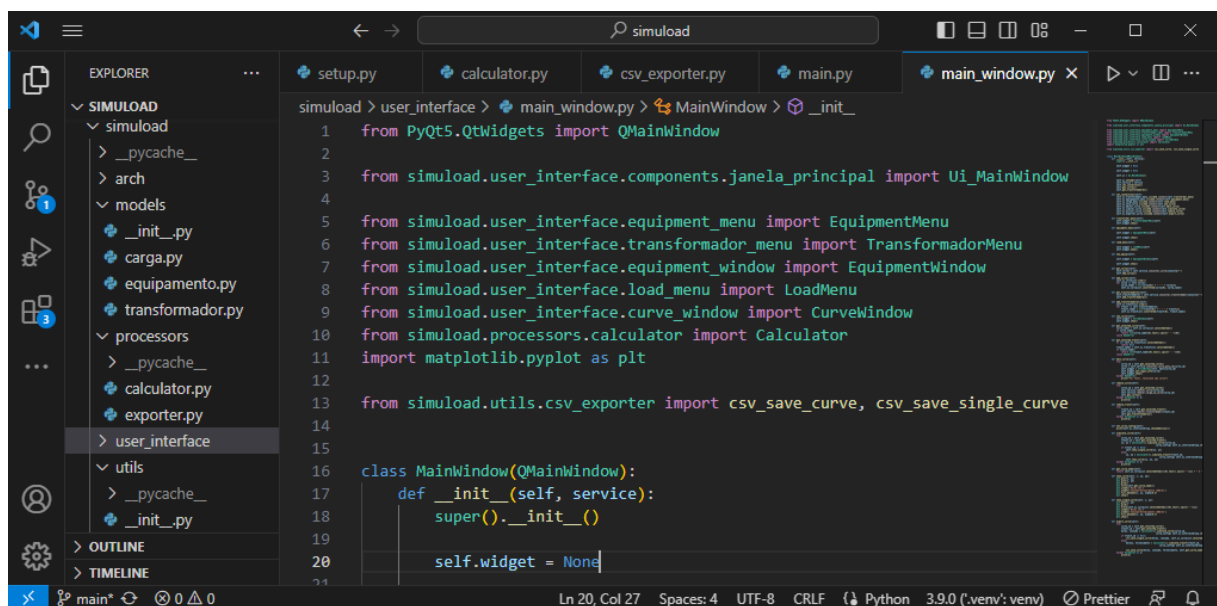
Durante essa pesquisa, diversos critérios foram considerados, como as funcionalidades oferecidas pelos módulos, a facilidade de uso, a popularidade no meio de desenvolvimento Python e a disponibilidade de material de apoio, como documentação e comunidades de suporte. Esses critérios foram cruciais para selecionar as melhores opções que atendessem às demandas específicas do Simuload.

A escolha criteriosa dos módulos e ferramentas utilizados é fundamental para o sucesso do projeto, pois eles fornecem recursos e funcionalidades essenciais para a implementação das diversas funcionalidades do software. Através dessa abordagem, foi possível garantir a qualidade, a robustez e a efetividade do Simuload, proporcionando uma experiência aprimorada ao usuário e facilitando o desenvolvimento das atividades propostas.

4.2.1 Ferramenta de edição de código.

Dentre as várias ferramentas disponíveis para a edição de código, optou-se por utilizar o VSCode – *Visual Studio Code* (MICROSOFT, 2023) como o principal editor de código para o desenvolvimento do Simuload. O VSCode foi selecionado devido à sua ampla popularidade, reconhecido como um dos principais editores de código para desenvolvedores Python, oferecendo uma experiência de edição altamente produtiva.

Fig. 2 Código fonte do Simuload no editor de código VSCode.



A escolha do VSCode baseou-se em diversos fatores. Em primeiro lugar, sua interface intuitiva e amigável proporciona uma curva de aprendizado suave, permitindo aos desenvolvedores se familiarizarem rapidamente com o editor. Além disso, o VSCode possui suporte para a linguagem Python, o que facilita a escrita, depuração e execução de código Python diretamente no editor.

Um dos principais pontos fortes do VSCode é a sua extensibilidade. Através da instalação de extensões, é possível expandir ainda mais as funcionalidades do editor, personalizando-o de acordo com as necessidades do Simuload. No caso do desenvolvimento em Python, existem diversas extensões úteis disponíveis, como integração com sistemas de controle de versão, suporte a testes automatizados, formatação de código, *linting* e autocompletar, que contribuem para um fluxo de trabalho eficiente e eficaz.

Portanto, a escolha do VSCode como o editor de código para o desenvolvimento do Simuload foi embasada em sua popularidade, facilidade de uso específica para códigos

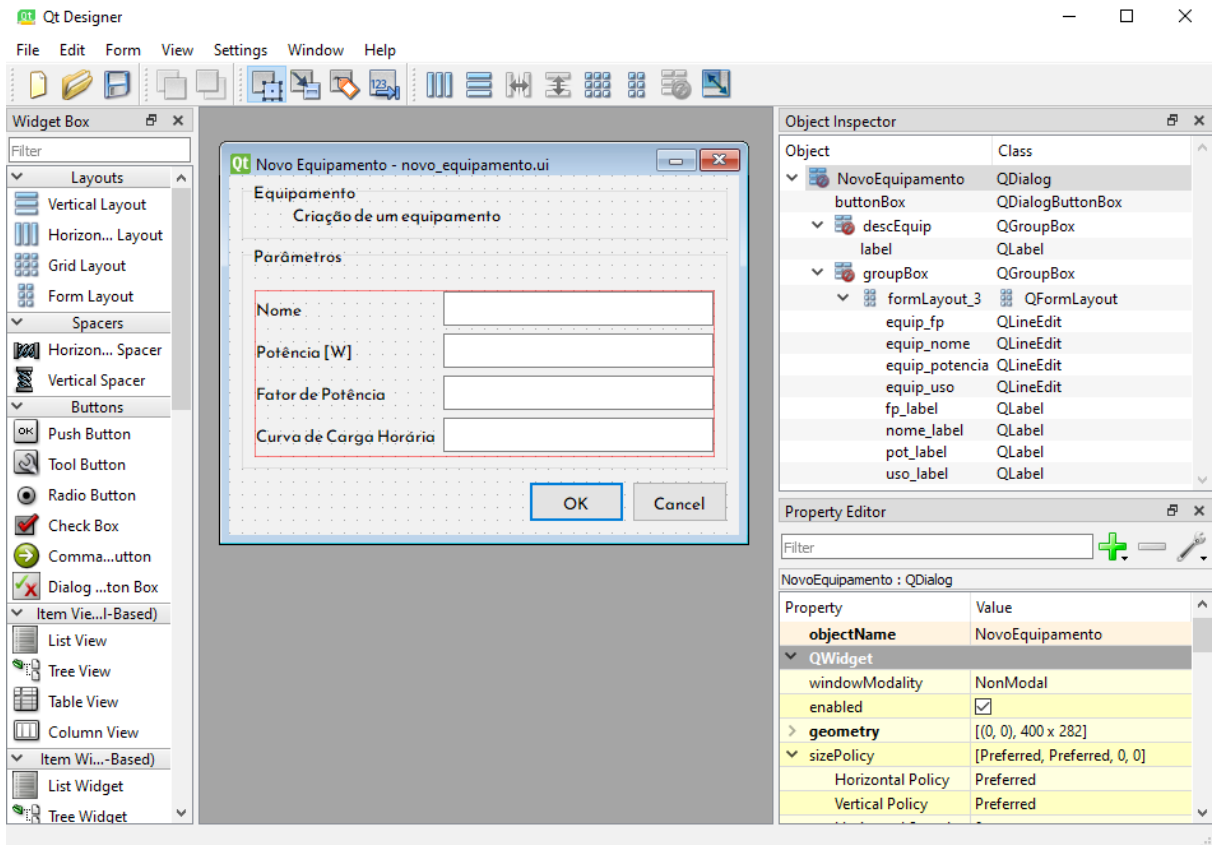
Python e a ampla disponibilidade de extensões úteis que auxiliam no desenvolvimento do software. Essa escolha contribui para uma experiência de programação agradável e produtiva, garantindo um ambiente de trabalho adequado para o desenvolvimento do Simuload.

4.2.2 *Ferramenta de interface visual.*

O processo de desenvolvimento do software Simuload envolveu diversas etapas, dentre as quais se destacam a escolha das ferramentas e tecnologias a serem utilizadas. Para a criação da interface gráfica, optou-se pela ferramenta PyQt5 (RIVERBANK COMPUTING LIMITED, 2023), que oferece uma vasta gama de recursos para desenvolvimento de interfaces gráficas em Python, incluindo *widgets*, diálogos e estilos personalizados. A escolha da linguagem de programação Python se deu pelo seu caráter de código aberto, grande comunidade de desenvolvedores, além de sua facilidade de uso e versatilidade.

Através do software Qt Designer (THE QT COMPANY, 2023), é possível desenhar as telas no formato *.ui*, definindo todas as suas estruturas e nome das variáveis onde serão inseridos os valores de texto. Com os arquivos *.ui* em mãos, o PyQt5 é responsável por transformá-los em código Python, possibilitando a integração das telas com o sistema.

Fig. 3 Interface do software Qt Designer e edição da tela de novo equipamento do Simuload.



4.2.3 Ferramentas matemáticas.

Para a simulação de gráficos, foi escolhida a biblioteca Matplotlib (THE MATPLOTLIB DEVELOPMENT TEAM, 2012), que oferece uma ampla variedade de gráficos e recursos de visualização de dados em Python, além de ser fácil de usar e ter uma boa documentação. O Matplotlib foi utilizado para a criação dos gráficos de curvas de carga simulados pelo software, permitindo a visualização das informações de maneira clara e eficiente.

Os cálculos realizados pelo software foram feitos diretamente pelo Python, utilizando as bibliotecas padrão do pacote, o que permitiu uma implementação mais simples e direta. Para obter os valores distribuídos em intervalos menores de simulação, foi utilizada a biblioteca Numpy, que oferece uma variedade de recursos para processamento de dados numéricos em Python, incluindo a interpolação de dados, que permitiu a obtenção de valores intermediários a partir dos dados de entrada.

4.2.4 Manipulação de dados.

No âmbito das especificações técnicas, uma decisão relevante diz respeito ao armazenamento dos dados das simulações realizadas. Nesse sentido, optou-se por utilizar o banco de dados em memória SQLite (HWACI – APPLIED SOFTWARE RESEARCH, 2023), uma solução confiável e eficiente para a persistência de dados estruturados no ambiente Python. A escolha do SQLite se deve à sua facilidade de uso e à compatibilidade com diferentes sistemas operacionais, garantindo a portabilidade e a acessibilidade dos dados armazenados.

O VSCode também possui uma extensão de visualização do banco de dados SQLite, exibindo os registros de cada tabela, suas colunas e restrições, *scripts* de criação das tabelas e possibilidade de consultas personalizadas no banco. A interface da extensão no VSCode pode ser visualizada na figura abaixo.

Fig. 4 Visualização da tabela de CargaEquipamento e seu script de criação na extensão SQLite no VSCode.

The screenshot displays the SQLite extension interface in VSCode. The top bar shows the database name 'simuload.db' and the selected table 'CargaEquipamento'. Below the table name, there are tabs for 'Schema', 'Query Editor', 'Auto Reload', 'Find', and 'Other Tools...'. The 'Schema' tab is active, showing the SQL script for creating the 'CargaEquipamento' table. The script defines four columns: 'CargaEquipamentoId' (INTEGER PRIMARY KEY), 'CargaId' (INTEGER NOT NULL), 'EquipamentoId' (INTEGER NOT NULL), and 'EquipamentoQtd' (INTEGER NOT NULL). It also includes two foreign key constraints: 'FOREIGN KEY (CargaId) REFERENCES Cargas (CargaId)' and 'FOREIGN KEY (EquipamentoId) REFERENCES Equipamentos (EquipamentoId)'. Below the script, there are buttons for '+ Index' and '+ Trigger'. The bottom section shows a table view of the data for 'CargaEquipamento'. The table has four columns: 'CargaEquipamentoId', 'CargaId', 'EquipamentoId', and 'EquipamentoQtd'. The data is as follows:

	CargaEquipamentoId INTEGER PRIMARY KEY	CargaId INTEGER NOT NULL REFERENCES "Cargas" ("CargaId")	EquipamentoId INTEGER NOT NULL REFERENCES "Equipamentos" ("EquipamentoId")	EquipamentoQtd INTEGER NOT NULL
1	63	2	2	1
2	64	2	10	2
3	65	2	3	25
4	66	2	5	2
5	67	2	6	1
6	68	2	8	1
7	69	2	10	2

Além disso, foi necessário considerar a exportação dos dados dos gráficos em formato CSV, visando a facilitar a análise e o compartilhamento dessas informações. Para realizar essa tarefa, fez-se uso do módulo *csv* nativo do Python, que oferece funcionalidades específicas para manipulação desse tipo de arquivo. Adicionalmente, foram explorados os métodos utilitários do módulo *os*, permitindo a criação de pastas no sistema operacional para uma organização adequada dos arquivos CSV gerados.

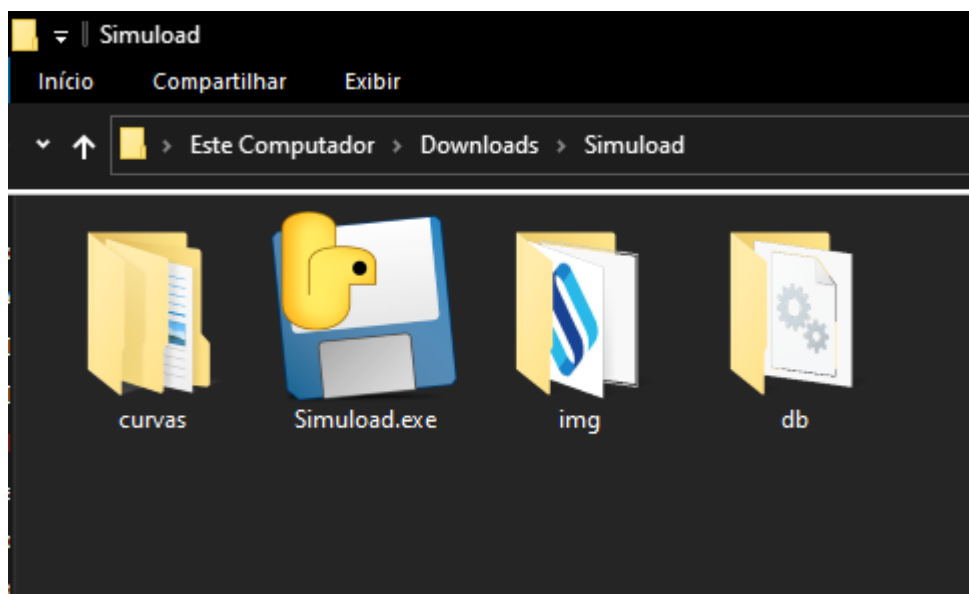
Essas decisões técnicas foram fundamentais para garantir a integridade dos dados, a eficiência no armazenamento e a praticidade na exportação dos resultados gerados pelo Simuload. Ao adotar o SQLite como banco de dados e o módulo csv do Python para exportação de dados em formato CSV, o projeto assegura uma abordagem robusta e adaptável, facilitando o acesso e a utilização das informações geradas pela ferramenta.

4.2.5 Distribuição do software.

Para gerar um executável do software, foi utilizada a ferramenta PyInstaller (CORTESE et al., 2023), que permite empacotar o código-fonte e todas as suas dependências em um único arquivo executável, facilitando a distribuição do software para usuários finais. Como o PyInstaller gera os arquivos através do próprio sistema operacional utilizado, as versões de distribuição do Simuload foram geradas em máquinas com sistemas operacionais diferentes, Linux e Windows, para permitir o seu uso nos dois sistemas.

O executável é capaz de armazenar as curvas exportadas e reconhecer o arquivo do banco de dados na sua própria pasta, além das imagens de logo do software, sendo essa a estrutura dos arquivos compactados de distribuição para download. Abaixo, temos uma imagem da estrutura de pastas e arquivos da versão para o sistema operacional Windows do Simuload.

Fig. 5 Pastas e executável que compõem a versão de distribuição para Windows do Simuload.



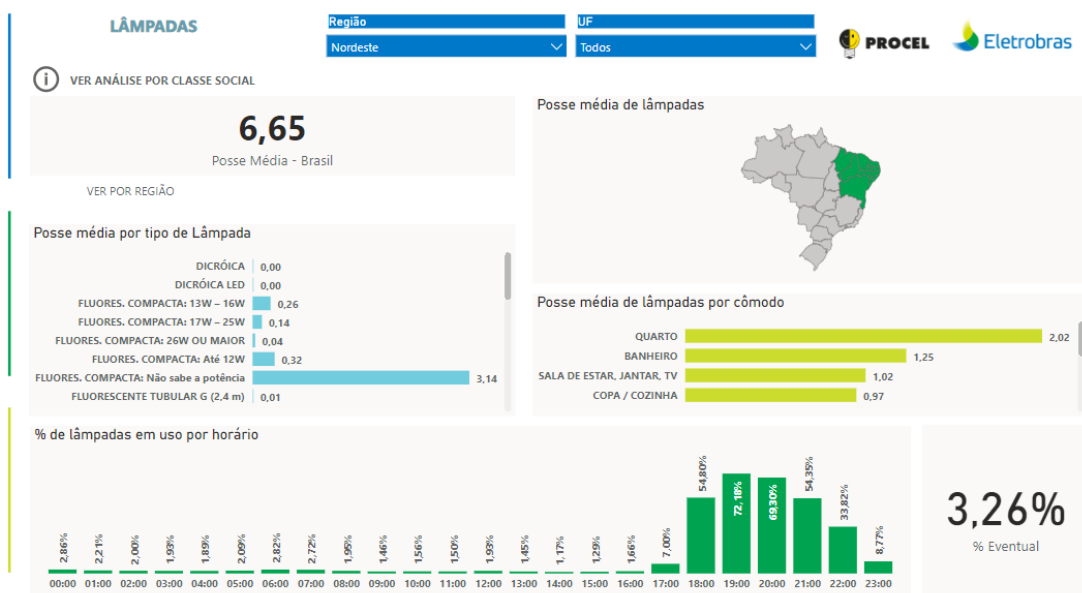
4.3 Mapeamento de dados

Para a coleta de dados utilizados na simulação, foram utilizadas informações retiradas da Pesquisa de Posse e Hábitos de Uso de Equipamentos Elétricos na Classe Residencial 2019 – PPH 2019 (MINISTÉRIO DE MINAS E ENERGIA, 2019) do Programa Nacional de Conservação de Energia Elétrica – Procel. O Procel é um programa de governo coordenado pelo Ministério de Minas e Energia – MME e executado pela Eletrobras, cujo objetivo é promover a conservação de energia elétrica. A PPH 2019 é uma pesquisa declaratória que traça um perfil da posse e hábitos de consumo de equipamentos elétricos e nos setores residencial, comercial e industrial, com o intuito de avaliar o mercado de eficiência energética nas cinco regiões do Brasil.

Os dados utilizados na simulação foram obtidos a partir da PPH 2019, que foi realizada em diferentes regiões do país, o que permitiu a obtenção de um conjunto diversificado de dados de consumo de energia elétrica. As informações foram coletadas através de questionários que perguntavam sobre a posse e uso de diferentes equipamentos elétricos, bem como sobre os hábitos de consumo de energia elétrica em residências, estabelecimentos comerciais e industriais.

Os dados coletados foram utilizados para a definição das características dos equipamentos e cargas presentes na simulação do Simuload, tais como a potência, fator de potência e distribuição horária do uso percentual. Esses dados foram importantes para que as simulações realizadas pelo Simuload fossem o mais próximas possíveis da realidade, permitindo assim que os resultados obtidos a partir das simulações fossem mais precisos e confiáveis. Foi escolhido o filtro da região Nordeste em todos os dados mapeados para o sistema se aproximar da realidade da UFBA. O filtro por Estado escolhendo a Bahia não foi utilizado porque nem todos os equipamentos possuem os seus dados disponíveis a partir dessa opção. Na figura abaixo podemos visualizar uma das telas do PPH 2019 com o filtro da região Nordeste para o equipamento de lâmpada, os dados de curva de carga e outras informações de uso que mapeadas pelo Procel.

Fig. 6 Dashboard de dados de consumo de lâmpadas na PPH 2019 para a região Nordeste.



4.4 Algoritmo de aleatoriedade

Dentro da metodologia adotada neste trabalho de conclusão de curso, foi desenvolvido um algoritmo de aleatoriedade para representar as curvas de carga em intervalos de tempo menores. Esse algoritmo desempenha um papel essencial no aprimoramento da representação das curvas de carga no sistema Simuload, visando proporcionar uma experiência mais detalhada e realista aos usuários. Isso permite que, a partir de vetores de apenas 24 pontos nos equipamentos, o usuário tenha acesso a simulações com resolução maior sem precisar preencher centenas de pontos para isso.

O algoritmo, implementado em Python, segue uma sequência de passos para a geração de pontos aleatórios que serão adicionados aos vetores de 24 pontos, representando o consumo hora a hora de um equipamento ou a curva de carga horária de um transformador.

Inicialmente, é realizado o cálculo do número de pontos necessários para a interpolação, com base no intervalo desejado. Por exemplo, se o intervalo for de 30 minutos, serão necessários 48 pontos (24 horas divididas em intervalos de 30 minutos). Para esse caso de 48 pontos, cada subintervalo (intervalo de pontos por hora) possui 2 pontos. Em seguida, é criado um vetor chamado que contém os pontos de intervalo entre 0 e 23, representando as horas do dia.

Dentro de um *loop* principal, percorre-se cada ponto do vetor de interpolação. Para cada ponto, são realizadas as seguintes etapas:

1. O valor atual do vetor de consumo é atribuído a uma variável que representa a média do subintervalo atual. Esse valor é calculado a partir do vetor de consumo original, que contém os valores de consumo em intervalos de 1 hora.
2. Se o intervalo de interpolação for menor do que 1 hora (por exemplo, 30 minutos), entra-se em um *loop* adicional para gerar os pontos aleatórios. Caso contrário, pula-se esta etapa.
3. Uma *seed* única é calculada com base na soma de todos os valores do vetor de consumo original. A *seed* é um número fixo que, quando definido, garante que os valores aleatórios gerados sejam os mesmos para todas as simulações de uma mesma curva.
4. Um número aleatório é gerado dentro de um intervalo predefinido, representando a variação percentual permitida em relação ao valor-base. Neste caso, a variação é definida em até 40% do valor-base, positiva ou negativa.
5. Dependendo do índice do ponto no *loop*, o sinal do valor aleatório gerado é invertido para alternar entre valores positivos e negativos.
6. O valor aleatório gerado é adicionado ao valor médio multiplicado pela porcentagem de variação e dividido por 100. Isso resulta em um valor ajustado que representa o consumo aleatório no subintervalo.
7. Um novo valor médio é acumulado a partir dos valores gerados, pois cada valor contribui para a média total do subintervalo.
8. É calculado um fator de normalização com base na média original e na nova média acumulada. Esse fator é utilizado posteriormente para garantir que os valores não ultrapassem a média original.
9. Os valores do subintervalo são multiplicados pelo fator de normalização para ajustá-los à média original. Se algum valor exceder a média original, ele é substituído pela média e o valor excedente é adicionado ao próximo ponto, mantendo a consistência do consumo total do subintervalo. Caso seja o último ponto do subintervalo, o valor excedente é adicionado ao ponto anterior.
10. O *loop* continua para o próximo subintervalo até que todos os pontos do vetor de interpolação sejam processados.

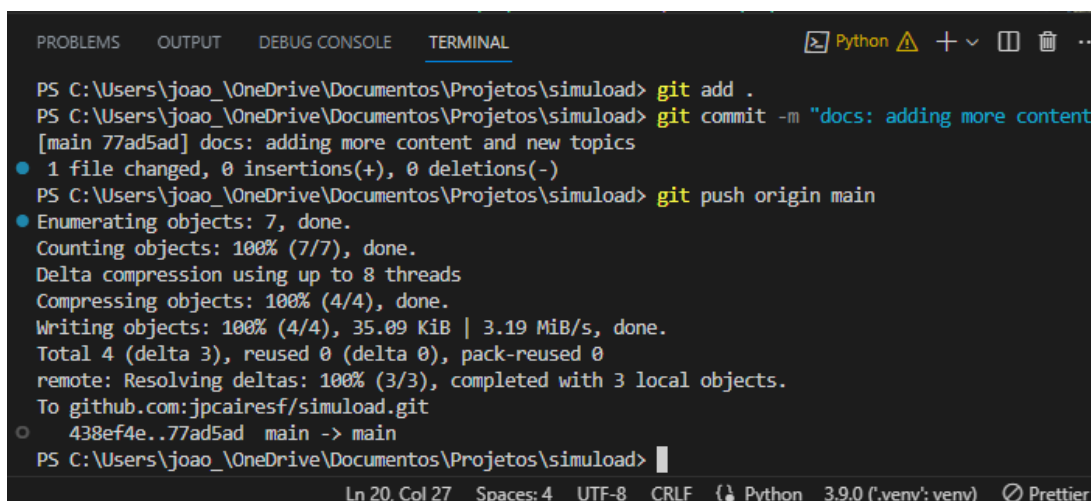
Ao final do algoritmo, é retornado um vetor contendo os pontos interpolados da curva de carga, representados em quilowatts. Esse vetor é utilizado para visualizar a curva de carga com maior resolução, refletindo o consumo detalhado dos equipamentos ao longo do tempo. Em algumas simulações, tem-se a impressão de que a curva de carga com intervalos menores possuem magnitude menor que a curva simulada para o intervalo de 1 hora. Isso acontece por conta da premissa de que o valor aleatório deve ser sempre menor ou igual ao valor da curva. A implementação completa do algoritmo pode ser encontrada no apêndice.

4.5 Versionamento de código

No desenvolvimento do Simuload, foi adotado o sistema de controle de versão Git (TORVALDS; HAMANO, 2005) junto a plataforma de hospedagem GitHub (GITHUB, 2023) para gerenciar o versionamento do código fonte e facilitar a disponibilização das versões de distribuição do software.

O Git é um sistema de controle de versão distribuído amplamente utilizado na indústria de desenvolvimento de software. Ele permite controlar o histórico de alterações do código fonte, rastrear e registrar as modificações realizadas ao longo do tempo. Com o Git, é possível criar ramificações para desenvolver novos recursos ou corrigir problemas sem afetar a versão principal do software. Com o Git instalado, é possível executar seus comandos no terminal da máquina utilizada para o desenvolvimento. Além disso, o VSCode possui um terminal embutido em sua plataforma que permite a execução desses comandos, como demonstrado na figura abaixo.

Fig. 7 Submissão de alterações no código utilizando o Git através do terminal embutido no VSCode.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + - [ ] [ ] ...
PS C:\Users\joao_OneDrive\Documentos\Projetos\simuload> git add .
PS C:\Users\joao_OneDrive\Documentos\Projetos\simuload> git commit -m "docs: adding more content
[main 77ad5ad] docs: adding more content and new topics
• 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\joao_OneDrive\Documentos\Projetos\simuload> git push origin main
• Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 35.09 KiB | 3.19 MiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:jpcairesf/simuload.git
• 438ef4e..77ad5ad main -> main
PS C:\Users\joao_OneDrive\Documentos\Projetos\simuload> |
Ln 20, Col 27 Spaces: 4 UTF-8 CRLF Python 3.9.0 (.venv: venv) Prettier
```

O GitHub é uma plataforma web que oferece recursos de hospedagem de repositórios Git. Por meio do GitHub, é possível armazenar de forma segura o código fonte do Simuload em um repositório centralizado. Além disso, ele fornece uma interface amigável para visualizar as alterações feitas no código ao longo do tempo. No Apêndice B deste trabalho, são fornecidos os links para acessar o repositório do Simuload no GitHub com o código fonte do software disponível para consulta e download, bem como os requisitos do sistema, instruções detalhadas de instalação e os procedimentos passo a passo.

Capítulo 5: Estudo de Caso

A ferramenta Simuload foi utilizada por uma dupla de alunos em um estudo de caso real, visando mapear uma região com suas respectivas cargas e equipamentos elétricos. Essa atividade é realizada durante a matéria – Laboratório Integrado VI, que faz parte do sétimo semestre na grade do curso de Engenharia Elétrica da UFBA, cursada no semestre letivo de 2023.1. Tal atividade consiste em simular a curva de carga da região e escolher o transformador adequado para atender as condições de demanda de energia elétrica. Essa é uma atividade de suma importância na formação dos estudantes, pois a partir dela é possível ter insumos que nortearão a necessidade elétrica de um determinado local.

5.1 Simulando com outros softwares

Para simular os gráficos de curvas de cargas sem o Simuload, geralmente é utilizado um software de planilhas como o Google Sheets (GOOGLE, 2023). Estes são os recursos comumente usados atualmente pelos estudantes durante o componente curricular Laboratório Integrado VI. No entanto, tal processo acaba sendo muito manual e trabalhoso para o preenchimento dos dados. No intervalo de 5 minutos, por exemplo, são praticamente 300 células de valores preenchidos por intervalo para diversos equipamentos.

Fig. 8 Mapeamento de consumo dos equipamentos e cargas para simulação de curvas no Google Sheets.

The screenshot shows a Google Sheet titled "Curva de carga" with a menu bar (Arquivo, Editar, Ver, Inserir, Formatar, Dados, Ferramentas, Extensões, Ajuda) and a toolbar. The sheet contains a table with the following data:

	A	B	C	D	E	F	G	H
1		Ferro	Chuveiro	Iluminação	Tomadas	Residência 50m	Residência 100m	Residência 115m
243	20:05	0	698	100	102	3234	6162	7774
244	20:10	0	1150	100	100	3650	7000	9050
245	20:15	0	1180	100	100	3680	7060	9140
246	20:20	0	1350	100	100	3850	7400	9650
247	20:25	0	1680	100	101	4198	8093	10680
248	20:30	0	1990	100	100	4490	8680	11570
249	20:35	0	2000	100	100	4500	8700	11600
250	20:40	0	1960	100	100	4460	8620	11480
251	20:45	0	1820	100	103	4374	8439	11180
252	20:50	0	910	100	100	3410	6520	8330
253	20:55	0	345	100	100	2845	5390	6635
254	21:00	211	0	100	100	2711	4911	5811
255	21:05	314	0	100	100	2814	5014	5914
256	21:10	680	0	100	102	3216	5446	6360
257	21:15	910	0	100	101	3428	5643	6550
258	21:20	1000	0	100	100	3500	5700	6600
259	21:25	1000	0	100	100	3500	5700	6600
260	21:30	1000	0	100	100	3500	5700	6600
261	21:35	1000	0	100	98	3464	5634	6520
262	21:40	780	0	100	97	3226	5381	6260
263	21:45	342	0	100	99	2824	5009	5902
264	21:50	120	0	100	100	2620	4820	5720
265	21:55	11	0	100	85	2241	4216	5011
266	22:00	0	0	100	78	1888	3778	4488

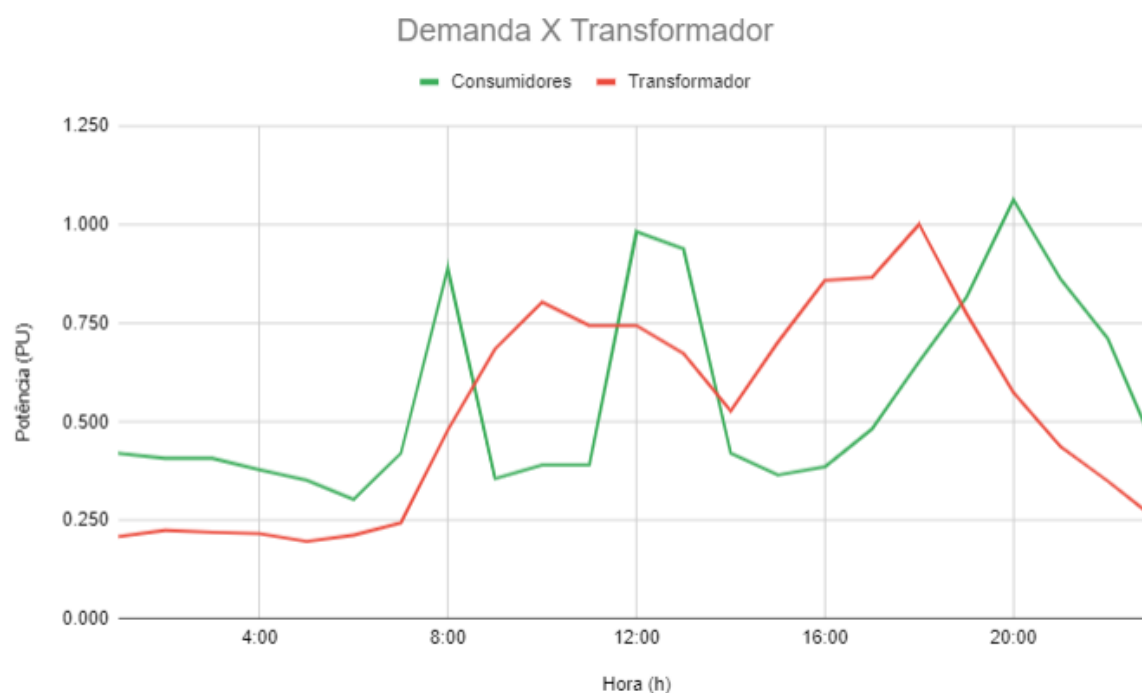
At the bottom of the sheet, there are filters for time intervals (5min, 15min, 1h) and categories (industrial, comercial, iluminacao, transformadores).

Com os dados de consumo dos equipamentos em mãos, é combinada a quantidade de cada equipamento para compor as residências e estabelecimentos, que por sua vez serão combinados para compor uma região para ser atendida pelos transformadores. O consumo dos estabelecimentos são calculados a partir desses valores e utilizados na simulação, comparando com as curvas do transformador e ajustando as quantidades para adequar o consumo conforme mostrado nas figuras abaixo.

Fig. 9 Cálculo das curvas de carga a partir de consumidores residencial, comercial, industrial e iluminação no Google Sheets.

	A	B	C	D	E	F	G	H
1			Tipo 1	Tipo 2	Tipo 3	Demanda	Demanda PU	
2		0:00	0.260	0.269	0.366	31700	0.423	
3		1:00	0.209	0.210	0.270	31540	0.421	
4		2:00	0.225	0.230	0.231	30570	0.408	
5		3:00	0.220	0.226	0.218	30570	0.408	
6		4:00	0.217	0.222	0.216	28380	0.378	
7		5:00	0.197	0.202	0.219	26445	0.353	
8		6:00	0.213	0.222	0.233	22740	0.303	
9		7:00	0.244	0.259	0.322	31520	0.420	
10		8:00	0.480	0.480	0.370	66646	0.889	
11		9:00	0.685	0.650	0.313	26730	0.356	
12		10:00	0.803	0.759	0.296	29290	0.391	
13		11:00	0.744	0.707	0.343	29290	0.391	
14		12:00	0.744	0.720	0.418	73620	0.982	
15		13:00	0.673	0.650	0.429	70390	0.939	
16		14:00	0.528	0.511	0.380	31535	0.420	
17		15:00	0.701	0.678	0.351	27420	0.366	
18		16:00	0.858	0.822	0.399	28940	0.386	
19		17:00	0.866	0.841	0.045	36178	0.482	
20		18:00	1.000	1.000	0.660	49000	0.653	
21		19:00	0.775	0.811	1.000	61158	0.815	
22		20:00	0.575	0.602	0.915	79720	1.063	
23		21:00	0.437	0.461	0.797	64670	0.862	
24		22:00	0.350	0.333	0.705	53352	0.711	
25		23:00	0.256	0.265	0.579	32620	0.435	
26								
27			Residencia 50 m²	Residencial 100 m²	Residencial 115 m²	Comercial	Industrial	Iluminação
28	Quantidade	4	3	0	1	0	2	Somatório Demandas
29	Demanda máxima	5,730	9,065	12448	16500	24500	6000	78615
30	Demanda total	22920	27195	0	16500	0	12000	

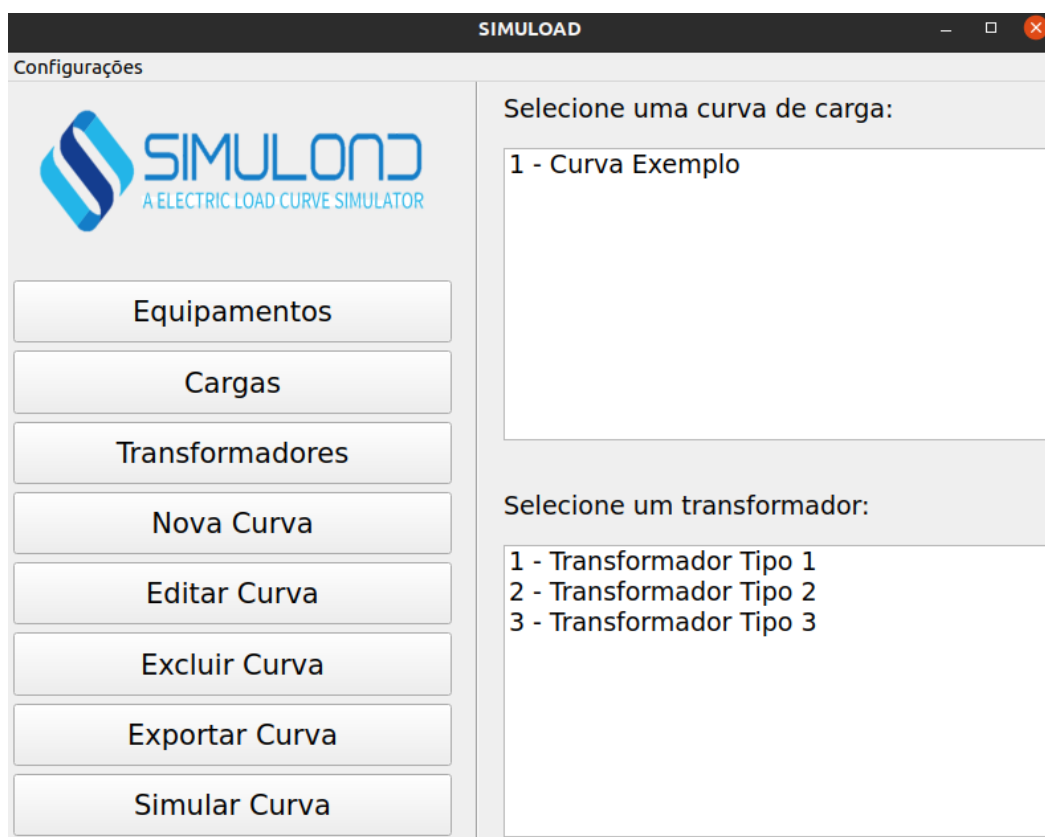
Fig. 10 Curva de simulação da demanda dos estabelecimentos pela curva característica do transformador no Google Sheets.



5.2 Aplicação do Simuload

Realizando a simulação com o Simuload, é possível utilizar os equipamentos e cargas pré-definidos no banco de dados ou customizar novas opções. Dentre os registros pré-definidos, estão os equipamentos e suas curvas de carga horária, mapeadas pela PPH 2019, e potência, mapeadas pela norma SM04.14-01.001 em seu anexo I (NEOENERGIA COELBA, 2014, p. 21) que pode ser encontrado no Apêndice C. Na tela principal é possível visualizar os comandos de criação de componentes ou simulações de curvas.

Fig. 11 Janela principal do Simuload.



Os alunos iniciaram a atividade adicionando os equipamentos na ferramenta, sendo adicionar, editar ou excluir um equipamento, tarefas possíveis no menu equipamentos. No campo *Curva de Carga Horária*, podemos definir a distribuição de uso em 24 horas do equipamento seguindo o padrão de números entre 0 e 1 nos colchetes:

[1 1 1 1 1 1 1 1 1 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.2 0.2 0.2 0.2]

Após adicionar os equipamentos, foram definidas a potência, o fator de potência e a distribuição horária de uso de cada equipamento. No campo *Potência* deve ser inserido um valor numérico que pode ser decimal utilizando um ponto no lugar da vírgula. O campo *Fator de Potência* também pode ser decimal mas deve variar entre 0 e 1.

Fig. 12 Janela de equipamentos do Simuload.

Equipamentos

Procurar

- 1 - Freezer
- 2 - Refrigerador Frost-free
- 3 - PPH 2019 - Lâmpada
- 4 - PPH 2019 - Ar-Condicionado
- 5 - PPH 2019 - Televisor
- 6 - PPH 2019 - Máquina de Lavar Roupas
- 7 - PPH 2019 - Chuveiro Elétrico
- 8 - PPH 2019 - Computador
- 9 - PPH 2019 - Ferro Elétrico a Seco
- 10 - PPH 2019 - Notebook
- 11 - PPH 2019 - Ventilador de Teto
- 12 - Ar-Condicionado Comercial

Excluir Criar Novo Editar

Fig. 13 Janela de criação de equipamento do Simuload.

Novo Equipamento

Equipamento

Criação de um equipamento

Parâmetros

Nome: Notebook

Potência [W]: 250

Fator de Potência: 0.98

Curva de Carga Horária: 0 0 0 0 0 0.2 0.3 0.9 1 1 1 1 0.7 0.6

Cancel OK

Em seguida, eles adicionaram as cargas, que poderiam ser estabelecimentos residenciais, comerciais ou até mesmo iluminação pública. No menu cargas é possível juntar diversas configurações de equipamentos para construir a carga desejada. Na tela de criação é possível adicionar equipamentos unitários (>) ou em lote (>>) selecionando o número desejado logo abaixo. Também é possível remover equipamentos selecionados à direita.

Fig. 14 Janela de cargas do Simuload.

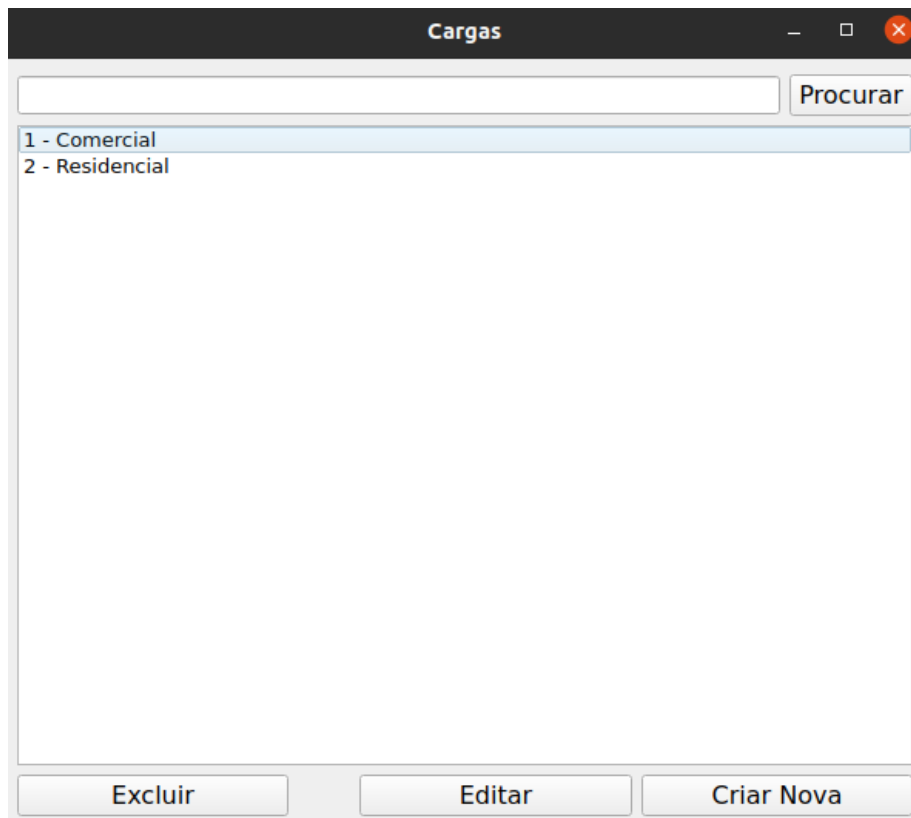
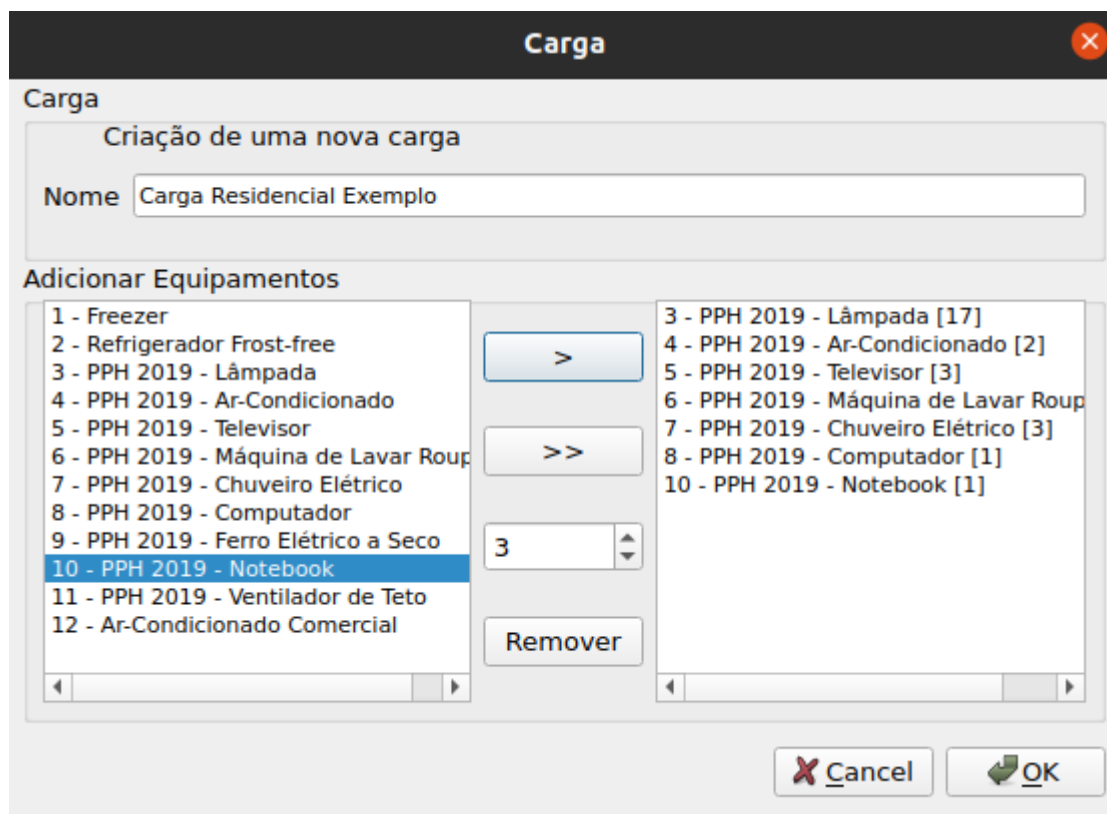


Fig. 15 Janela de criação de carga do Simuload.



A carga era alimentada pelos equipamentos adicionados anteriormente. Após a adição das cargas, os alunos criaram uma curva, que representa a distribuição temporal do consumo de energia elétrica das várias cargas. Essa curva representa o consumo de energia elétrica numa região atendida por um transformador.

Na tela principal é possível editar as configurações para a simulação da curva e criar curvas baseadas nas cargas existentes. Da mesma forma que a janela de novas cargas, as curvas utilizam a estrutura de adição individual (>) ou em lote (>>) das cargas, sendo possível escolher a quantidade do lote adicionado pela numeração abaixo.

Fig. 16 Janela de criação de curva do Simuload.

Adicionar Curva

Curva

Criação de uma nova curva para simular

Nome: Curva Exemplo 2

Adicionar Cargas

Lista de Cargas	Operações	Lista de Cargas Selecionadas	
1 - Comercial	> >> 13 Remover	1 - Comercial [7]	
2 - Residencial		2 - Residencial [13]	

Cancel OK

Os transformadores característicos de 75 kVA dos tipos 1, 2 e 3, utilizados no projeto da disciplina Laboratório Integrado VI, já estão registrados por padrão no Simuload e prontos para uso nas simulações. No menu transformadores é possível criar e configurar outros transformadores com a mesma estrutura de distribuição de 24 horas dos equipamentos no campo Curva de Carga Horária:

[1 1 1 1 1 1 1 1 1 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.2 0.2 0.2 0.2]

Fig. 17 Janela de transformadores do Simuload.

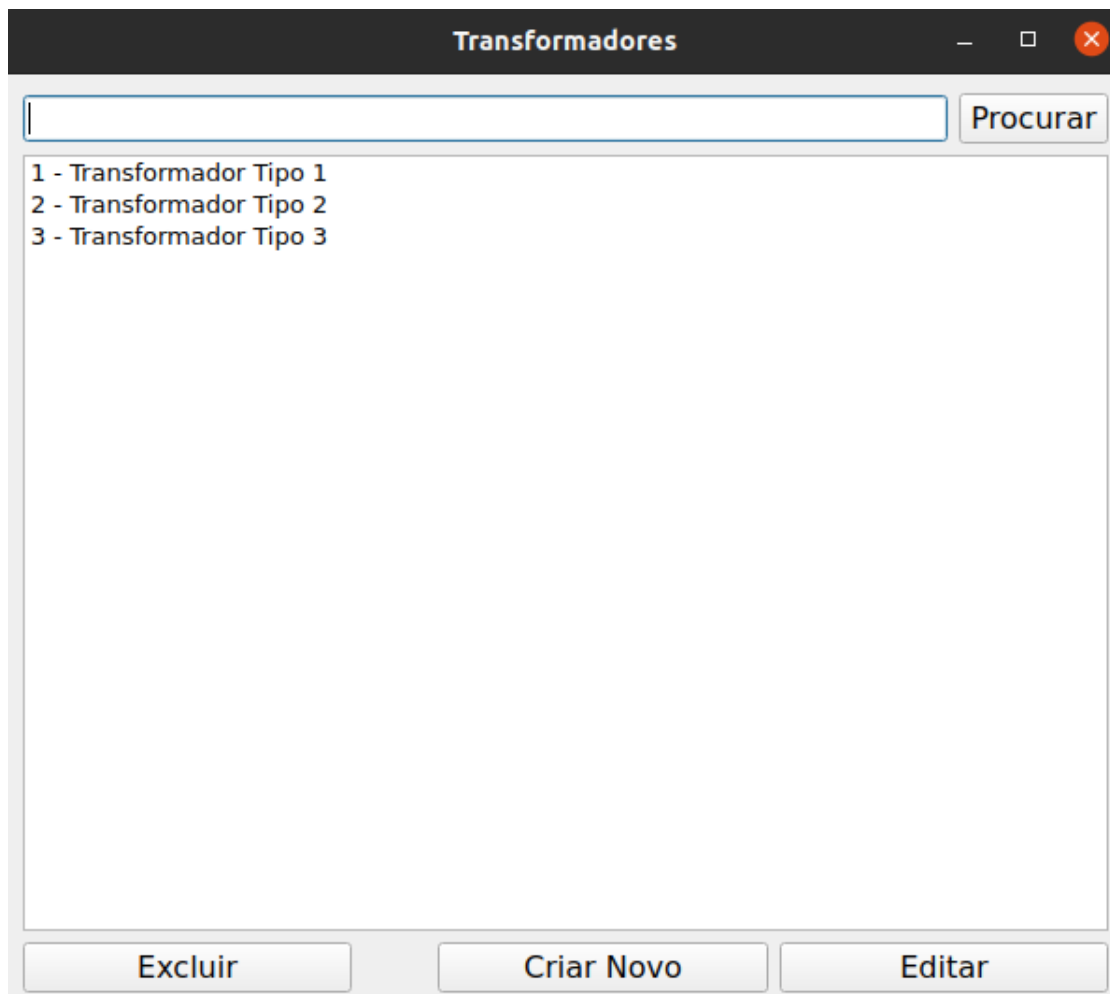


Fig. 18 Janela de criação de transformador do Simuload.

Novo Transformador

Transformador

Criação de um transformador

Parâmetros

Nome: Transformador Exemplo

Potência Nominal [VA]: 75000

Curva de Carga Horária: 0.95 0.8 0.82 0.87 0.89 0.88 0.8 0.

Cancel OK

Para realizar a simulação, basta clicar na opção *Simular Curva* na janela principal. É possível selecionar uma curva de carga individual ou uma curva de carga, junto com um

transformador. Se a simulação ou exportação não funcionarem, é provável que algum formato esteja incorreto no campo *Curva de Carga Horária* de um equipamento ou transformador.

Durante a simulação, a curva de carga é representada pela linha azul, enquanto a do transformador é representada pela linha laranja. A simulação oferece várias opções utilitárias padrão fornecidas pelo pacote Matplotlib. É possível mover a curva, aplicar zoom, desfazer ações, editar bordas e espaçamentos, configurar as características da curva e salvá-la como imagem.

No gráfico, o eixo Y representa o consumo/fornecimento da curva e do transformador em quilowatt-hora, enquanto o eixo X representa as horas. Antes de realizar a simulação, é possível selecionar o intervalo desejado nas configurações. As opções disponíveis são: 1 hora, 30 minutos, 15 minutos e 5 minutos, sendo que o intervalo padrão é de 1 hora. A seguir, apresentamos as simulações realizadas com intervalo de 1 hora e, respectivamente, intervalos de 30 minutos, 15 minutos e 5 minutos. Essas simulações utilizam as curvas registradas por padrão no Simuload.

Fig. 19 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 1 hora.

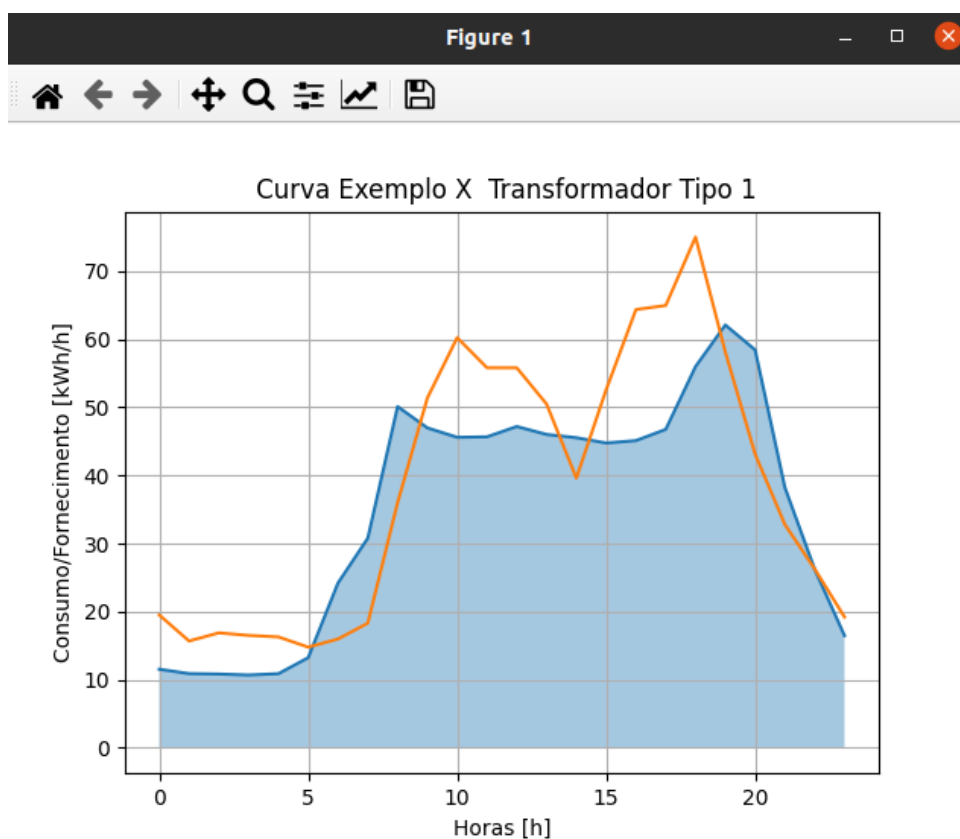


Fig. 20 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 30 minutos.

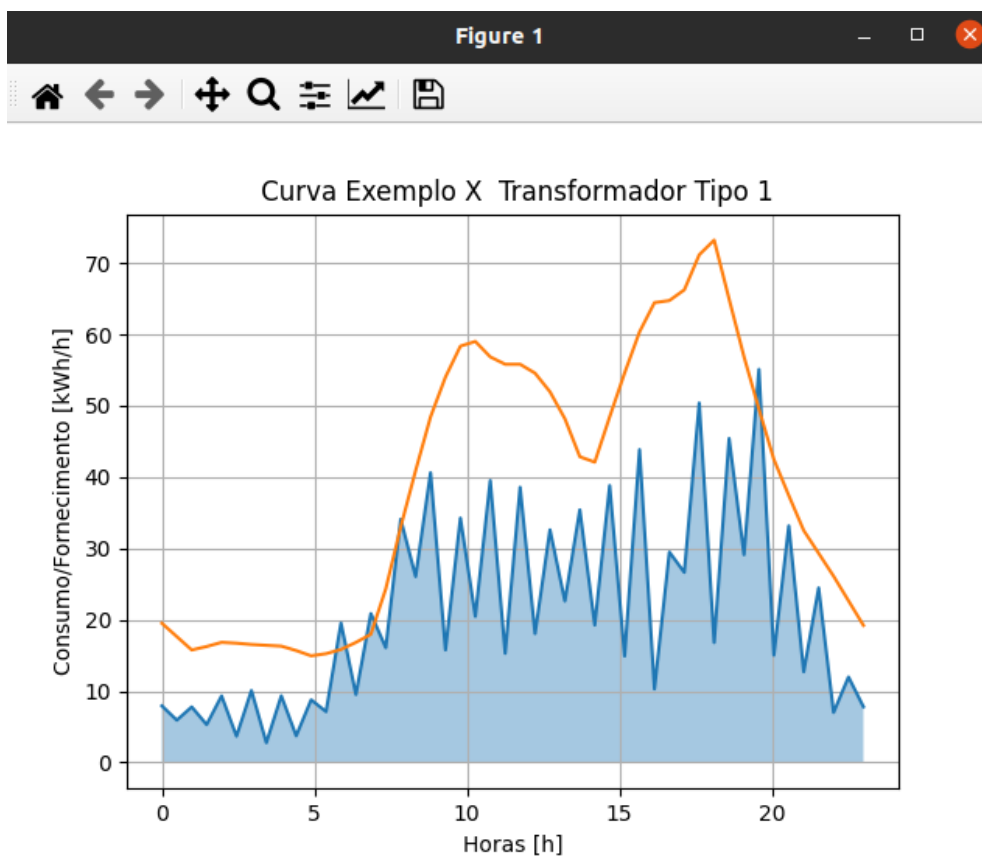


Fig. 21 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 15 minutos.

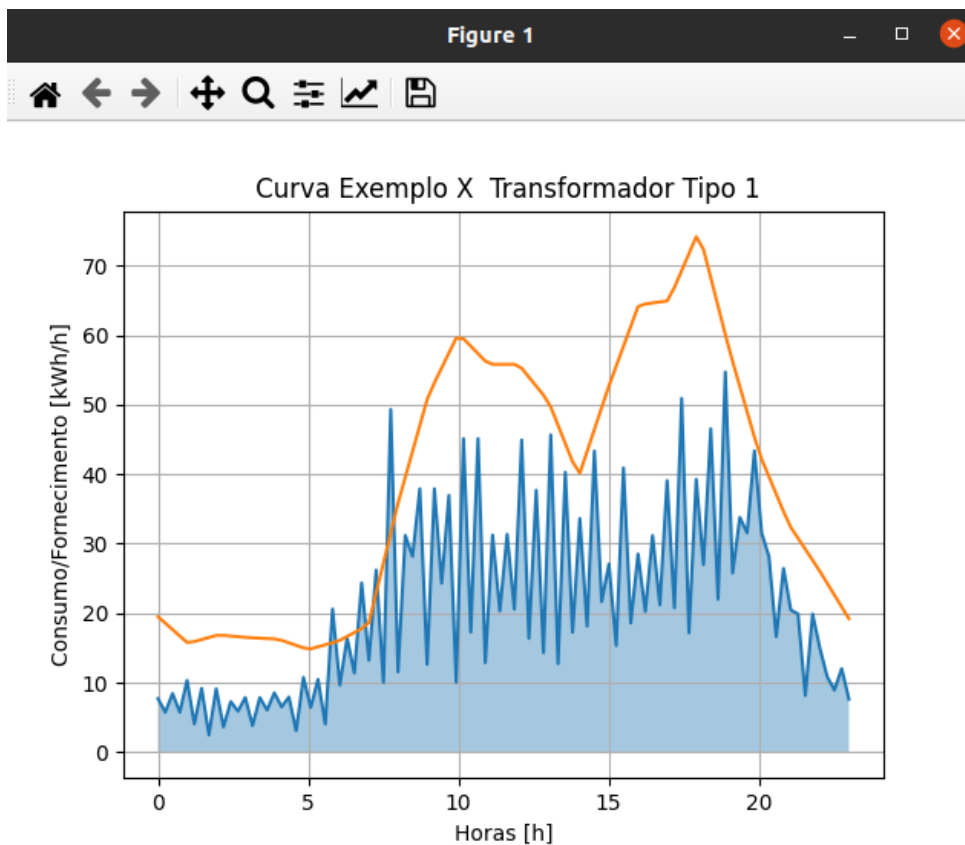
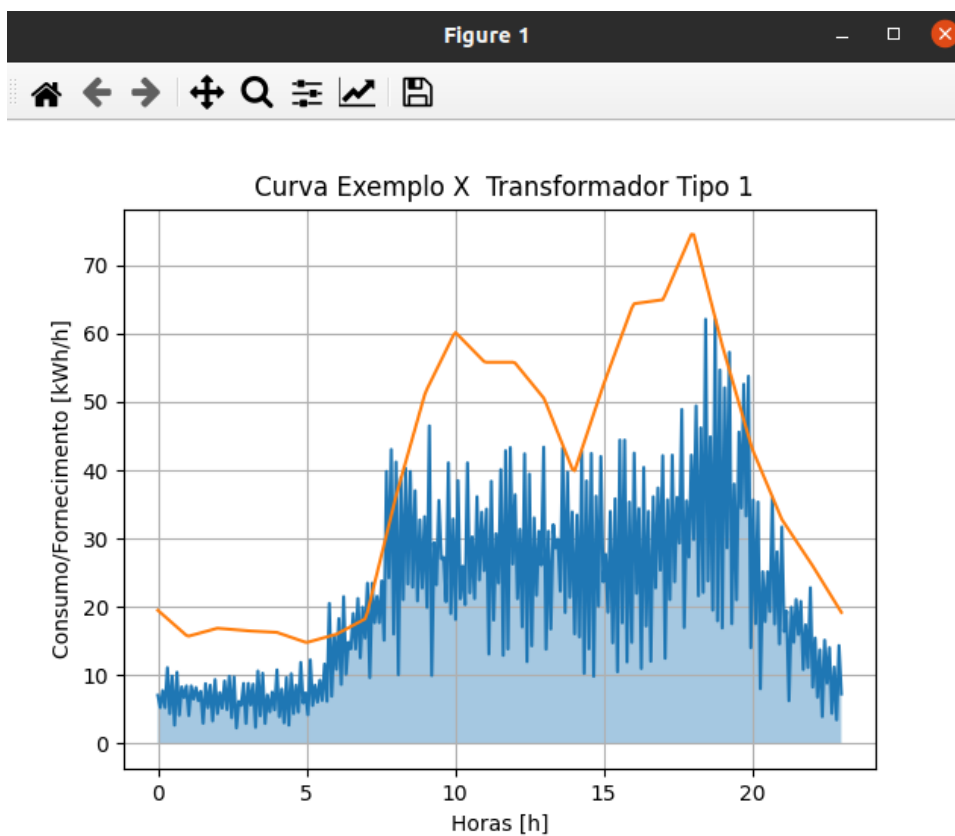


Fig. 22 Gráfico da curva padrão e transformador do tipo 1 registrados no Simuload no intervalo de simulação de 5 minutos.



Seguindo as etapas de utilização do Simuload, os alunos conseguiram adaptar os equipamentos, cargas e curvas para atender aos critérios do projeto em questão. O objetivo é ajustar a curva de carga para que o consumo esteja entre 100% e 130% da capacidade do transformador.

Para uma manipulação e avaliação mais precisa dos dados, foi implementada a funcionalidade de exportar curva. Essa opção gera um arquivo CSV com base na curva e transformador selecionados, considerando o intervalo escolhido nas configurações. O arquivo CSV é salvo na pasta “*curvas*”, localizada no mesmo diretório do Simuload. Cada simulação cria uma nova pasta com o nome da curva e transformador utilizados. Dentro dessa pasta, há o arquivo CSV com os dados de simulação, além do nome do arquivo que indica a data e horário em que foi exportado.

5.3 Resultados e comparações

A utilização do Simuload se mostrou uma alternativa mais prática e eficiente quando comparada com a maneira comumente utilizada pelos alunos, na qual consiste em fazer as simulações manualmente, hora a hora, inserindo valores de vários equipamentos em vários estabelecimentos e calculando tudo por planilhas. Todo esse trabalho aumenta o tempo total destinado a essa atividade e pode levar a uma maior possibilidade de erro, pois os cálculos são feitos de forma manual pelos estudantes. Com o Simuload foi possível fazer todo esse cadastramento, análise e escolha do transformador adequado de forma mais prática e sem comprometer o processo de aprendizagem proposto no componente curricular. Além disso, o próprio trabalho acaba exigindo mais tempo no registro de valores e fórmulas do que na análise das curvas de carga e adequação ao transformador.

Em contrapartida, o Simuload apresenta algumas limitações e pontos a serem melhorados. Uma das principais desvantagens é a falta de exibição de erros, o que pode dificultar a utilização para novos usuários em certas etapas. Embora o software se esforce para evitar erros e preenchimentos incorretos nos campos, a estrutura dos campos de *Curva de Carga Horária* pode ser problemática. Caso algum valor seja preenchido incorretamente, a simulação não será executada e a localização do erro não será informada ao usuário. Por esse motivo, os usuários optaram por preencher e revisar esses valores em um editor de texto antes de inseri-los no Simuload. Exibir os erros corretamente ou construir uma estrutura que valide esses dados antes de serem inseridos seria uma melhoria valiosa.

Outra limitação do software é que ele se baseia no modelo de dados da PPH 2019, projetado principalmente para simulações com intervalo de 1 hora. Os intervalos de 30, 15 e 5 minutos são gerados por meio de um algoritmo de aleatoriedade, o que pode não refletir com precisão o funcionamento real de alguns equipamentos. Além disso, pode dar a impressão de que o consumo é menor nos intervalos menores devido aos valores aleatórios serem menores ou iguais aos da referência (gráfico com intervalo de 1 hora). Como resultado, os valores da simulação em outros intervalos não podem ser ajustados e, embora o software busque fornecer uma representação geral, pode haver discrepâncias em relação à realidade operacional.

Ao utilizar o Simuload, os alunos puderam ajustar as cargas, equipamentos e a curva para atender às condições do transformador, de forma mais ágil e eficiente. A ferramenta se mostrou uma alternativa prática e útil para a realização da atividade de Laboratório Integrado VI, proporcionando uma experiência mais enriquecedora e produtiva para os alunos.

Capítulo 6: Conclusão

O Simuload, projeto desenvolvido e escolhido como trabalho de conclusão do curso de Engenharia Elétrica da UFBA, destaca uma necessidade iminente de utilizar a tecnologia a favor do estudante, a fim de uma aprendizagem mais prática e eficiente. É fundamental que sejam desenvolvidas ferramentas didáticas que possam auxiliar no desenvolvimento de conhecimentos e habilidades para análise e planejamento de sistemas elétricos.

A simulação de curvas de carga em redes de distribuição de energia elétrica é uma atividade de extrema importância no campo de atuação do engenheiro eletricista, pois é responsável por embasar o planejamento da distribuição de energia elétrica, permitindo o controle eficiente e seguro da demanda de energia, a identificação de pontos críticos e a avaliação do desempenho da rede em situações de pico de demanda.

Portanto, pode-se concluir que o software Simuload é uma ferramenta valiosa para a área de engenharia elétrica, visto que, através da sua capacidade de simular curvas de carga em sistemas elétricos de forma didática, ele pode ser utilizado como ferramenta de estudo para análise e planejamento de redes de distribuição de energia.

6.1 Objetivos e metodologia

O objetivo geral deste trabalho consistiu em apresentar o software e avaliar sua eficácia como uma ferramenta de simulação de curvas de carga em redes de distribuição de baixa tensão, com o intuito principal de contribuir para o desenvolvimento de habilidades técnicas dos estudantes de Engenharia Elétrica.

Dentre os objetivos específicos incluem avaliar a eficiência e usabilidade do software, identificar possíveis limitações e propor melhorias. A metodologia de desenvolvimento do Simuload foi baseada nas principais necessidades dos estudantes no que diz respeito a simulação de curvas de carga, bem como na necessidade de apresentar uma ferramenta de fácil utilização, para que fosse o mais intuitiva e didática possível. Assim, o processo de validação pós criação da ferramenta foi feito por meio de avaliação qualitativa do seu uso por estudantes em uma simulação real e apresentou-se como uma ferramenta eficiente e didática para esse fim.

No que diz respeito a parte mais técnica do software, foram adotadas diversas ferramentas e tecnologias para diferentes aspectos do desenvolvimento do Simuload. A ferramenta PyQt5 foi escolhida para a criação da interface gráfica, a biblioteca Matplotlib foi utilizada para a simulação de gráficos, o SQLite foi escolhido como banco de dados para armazenamento de dados estruturados, já o PyInstaller foi utilizado para gerar um executável do software facilitando sua distribuição. Todo o seu versionamento e disponibilização foi feito utilizando as ferramentas Git e Github.

Essas decisões técnicas e ferramentas foram fundamentais para garantir a facilidade de uso, eficiência no armazenamento de dados, geração de gráficos e distribuição do software. O Simuload se beneficia dessas escolhas, proporcionando uma experiência mais proveitosa e confiável aos estudantes, o que foi desde o início um dos seus principais objetivos, visto que, não haveria total aproveitamento se os mesmos não conseguissem se adaptar e utilizar de forma correta.

6.2 Considerações finais

O Simuload é uma ferramenta pensada sobretudo para auxiliar os estudantes de engenharia elétrica e não buscou apresentar-se como opção para o ambiente profissional de forma geral. No entanto, o intuito com a criação do software foi também estender as possibilidades e o debate no que diz respeito aos recursos tecnológicos existentes atualmente. É preciso pensar na área de simulação de curvas de carga de forma mais ampla e abrangente, não só no âmbito profissional, mas sobretudo no âmbito acadêmico, pois é na formação que o estudante tem a possibilidade de ter seus primeiros contatos com o esse tipo de análise.

Neste aspecto, o Simuload apresenta-se como uma ferramenta que traduz a necessidade de intercâmbio entre a tecnologia e a prática, trazendo diversas vantagens, tais como: eficiência e produtividade, pois facilita o processo de mapeamento de dados, análise e geração de gráficos, permitindo que os resultados sejam obtidos e avaliados de forma mais rápida e precisa; compreensão dos dados, pois oferecem recursos avançados de visualização de dados, permitindo que as informações sejam apresentadas de forma clara e intuitiva; otimização, pois é possível realizar análises preditivas, modelando diferentes cenários e simulações para prever comportamentos futuros.

Em resumo, o uso da tecnologia, no caso específico deste trabalho: o Simuload, oferece vantagens significativas, melhorando a eficiência, a compreensão dos dados e possibilitando aos estudantes uma experiência de aprendizado mais proveitosa. Levando em consideração o cenário atual, é imprescindível estabelecer diálogos e abrir espaço para ferramentas capazes de trazer benefícios no aprendizado dos estudantes.

O Simuload apresenta algumas desvantagens por conta das premissas adotadas no seu desenvolvimento, baseado no PPH 2019, e pontos de melhoria que não foram implementados. Apesar disso, ele foi considerado pelos estudantes uma alternativa mais prática para realizar as simulações de curvas de carga. Mesmo sendo necessária maior atenção dos usuários no momento do preenchimento de alguns campos, este não é um fator que compromete o resultado final quando todas as informações são inseridas da maneira correta. Um bom ponto de melhoria para esse cenário seria implementar a exibição de erros ou uma estrutura que valide esses campos antes de registrá-los no software.

O Simuload apresenta-se de maneira mais eficiente nas simulações com intervalo de 1 hora, podendo sofrer alterações no resultado final quando usado em outros intervalos temporários diferentes. Isso é decorrente da adoção dos modelos da PPH 2019 como base para seu desenvolvimento e do seu algoritmo de aleatoriedade que não é totalmente preciso, apesar de buscar ser o mais próximo da realidade. Esta é uma desvantagem que merece maior atenção, pois pode comprometer o resultado final do processo, portanto, é indicado que o software seja utilizado respeitando essa limitação. Os gráficos de intervalos menores que 1 hora são estimativas geradas para fins didáticos, para resultados mais precisos é necessário um mapeamento de dados mais minucioso, o que comprometia o processo de aprendizagem em outras formas de modelagem e simulação das curvas de carga. Visto esse cenário, o Simuload também possui uma funcionalidade de exportação dos dados para um arquivo CSV, que permite a sua manipulação de forma mais precisa.

A partir da experiência dos estudantes e das análises dos resultados obtidos no estudo de caso, foi possível compreender todas as vantagens e desvantagens do Simuload. O intuito principal foi proporcionar aos estudantes uma maneira mais prática de realizar as simulações de curvas de carga, reiterando a importância do pensamento analítico em todo o processo. Poder examinar o Simuload na prática foi extremamente enriquecedor e desafiador, pois ao mesmo tempo em que a ferramenta se mostrou eficiente na atividade proposta, foi possível também colher insumos para o aprimoramento da mesma, o que certamente o torna um projeto bastante relevante a se evoluir em trabalhos futuros.

Referências

PYTHON SOFTWARE FOUNDATION. **Python**. [S. l.], 2023. Disponível em: <https://www.python.org/>. Acesso em: 7 jul. 2023.

MIRO. **Miro**. [S. l.], 2023. Disponível em: <https://miro.com/pt/>. Acesso em: 7 jul. 2023.

MICROSOFT. **Visual Studio Code**. 1.80. [S. l.], 2023. Disponível em: <https://code.visualstudio.com/>. Acesso em: 7 jul. 2023.

RIVERBANK COMPUTING LIMITED. **PyQt5**. 5.15.9. [S. l.], 2023. Disponível em: <https://pypi.org/project/PyQt5/>. Acesso em: 7 jul. 2023.

THE QT COMPANY. **Qt Designer**. 5.15. [S. l.], 2023. Disponível em: <https://doc.qt.io/qt-5.15/>. Acesso em: 7 jul. 2023.

THE MATPLOTLIB DEVELOPMENT TEAM. **Matplotlib**. 3.7.2. [S. l.], 2023. Disponível em: <https://matplotlib.org/>. Acesso em: 7 jul. 2023.

HWACI – APPLIED SOFTWARE RESEARCH (coord.). **SQLite**. 3.42.0. [S. l.], 2023. Disponível em: <https://www.sqlite.org/index.html>. Acesso em: 7 jul. 2023.

CORTESI, David; BAJO, Giovanni; CABAN, William; MCMILLAN, Gordon. **PyInstaller**. 5.13.0. [S. l.], 2023. Disponível em: <https://pyinstaller.org/en/stable/>. Acesso em: 7 jul. 2023.

MINISTÉRIO DE MINAS E ENERGIA (Brasil). Programa Nacional de Conservação de Energia Elétrica. **PPH 2019**: Pesquisa de Posse e Hábitos de Uso de Equipamentos Elétricos na Classe Residencial 2019. [S. l.], 2019. Disponível em: <https://q.eletronbras.com/pt/Paginas/PPH-2019.aspx>. Acesso em: 7 jul. 2023.

TORVALDS, Linus; HAMANO, Junio. **Git**. [S. l.], 2005. Disponível em: <https://git-scm.com/>. Acesso em: 7 jul. 2023.

GITHUB. **GitHub**. [S. l.], 2023. Disponível em: <https://github.com/>. Acesso em: 7 jul. 2023.

GOOGLE. **Google Sheets**. [S. l.], 2023. Disponível em: <https://docs.google.com/spreadsheets/>. Acesso em: 7 jul. 2023.

NEOENERGIA COELBA. SM04.14-01.001, 12ª Edição, de 3 de agosto de 2014. **Fornecimento de Energia Elétrica em Tensão Secundária de Distribuição a Edificações Individuais**, [S. l.], p. 21-22, 3 ago. 2014. Disponível em: <https://servicos.neoenergiacoelba.com.br/comercial-industrial/Documents/Normas%20e%20Tarifas/SM04.14-01.001.pdf>. Acesso em: 7 jul. 2023.

Apêndice A – Código do algoritmo de aleatoriedade

```
def interpolador_curva(self, consumo_curva, intervalo = 1):
    values = int(24 * (60/intervalo))
    intervalo_interp = np.linspace(0, 23, values)

    # cria um vetor vazio com a quantidade de valores e define o subintervalo
    # subintervalo por ser de 5, 15 ou 30 minutos
    consumo_curva_interp = [None] * values
    sub_intervalo = int(values/24)
    count = 0

    # fixa uma seed para que o resultado seja o mesmo em todas as simulações
    seed = int(np.sum(consumo_curva))
    percentual_var = 40
    base_avg = 100 - percentual_var

    while (count < values):
        # valor no vetor de 24h é a média do sub intervalo
        consumo_curva_interp[count] = consumo_curva[int(count/sub_intervalo)]
        avg = consumo_curva_interp[count]

        if(values > 24):
            new_avg = 0
            for i in range(sub_intervalo):
                # setar uma seed única e gerar um valor de 0% a 40%
                np.random.seed(seed + count + i)
                rgen = np.random.randint(0, percentual_var)

                # comutar o sinal do valor gerado rgen
                if (count + i) % 2 != 0:
                    rgen = rgen*(-1)

                # soma o valor gerado a partir de meia média e acumula uma nova média
                consumo_curva_interp[count+i] = avg*(base_avg+rgen)*0.01
                new_avg += consumo_curva_interp[count+i]
                i+=1

            # calcular fator do que falta para a média original
            normalization = 100/base_avg
            fator = avg/(normalization*new_avg/sub_intervalo)

            # multiplicar todos os valores do sub intervalo pelo fator
            j = 0
            while (j < sub_intervalo):
                # se o valor exceder a média:
                # o ponto será igual à média e salvar o valor excedente
                if (consumo_curva_interp[count+j] * fator) > avg:
                    consumo_curva_interp[count+j] = avg
                    exceed = consumo_curva_interp[count+j] * fator - avg

                # se ainda houver próximo valor, adicionar o excedente ao próximo
                # e contar duas iterações
                if (j + 1) < sub_intervalo:
                    consumo_curva_interp[count+j+1] = consumo_curva_interp[count+j+1] *
fator + exceed

                j += 2
```

```

        # se não, adicionar ao anterior e contar uma iteração (fim do laço)
        else:
            consumo_curva_interp[count+j-1] = consumo_curva_interp[count+j-1] *
fator + exceed
            j += 1

        # se não exceder a média, seguir a fórmula do fator
        else:
            consumo_curva_interp[count+j] = consumo_curva_interp[count+j] * fator
            j += 1
        # a iteração acontece pra cada sub intervalo entre uma hora e outra
        count += sub_intervalo

# retorna valor em kW
return intervalo_interp, np.array(consumo_curva_interp)/1000

```

Apêndice B – Repositório do Simuload

O código do Simuload está hospedado num repositório no Github que pode ser acessado no seguinte link: <https://github.com/jpcairesf/simuload>. Nele, você encontrará todo o código fonte, histórico do versionamento de código e alterações, e um guia explicativo na página inicial. O guia contém instruções para download do código fonte e das versões executáveis de distribuição para Windows e Linux, além de um passo a passo de como utilizar o software.

Apêndice C – Anexo I. Tabelas da norma SM04.14-01.001



Norma
Fornecimento de Energia Elétrica em Tensão Secundária de Distribuição a Edificações Individuais

ANEXO I. TABELAS
Tabela 1 - Potências Média dos Aparelhos Eletrodomésticos

Potência Padronizada para Eletrodomésticos					
Item	Descrição	Potência	Item	Descrição	Potência
1	Amplificador de potência eletrônico	200 W	72	Liquidificador industrial	1000 W
2	Aparelho de raios X pequeno	3.500 W	73	Liquidificador residencial	320 W
3	Aparelho de raios X grande	7.000 W	74	Lixadeira grande	1000 W
4	Aparelho de ultrassonografia	500 W	75	Lixadeira pequena	850 W
5	Aquecedor de água até 400 litros	2000.W	76	Maquina de chope	900 W
6	Aquecedor de água por passagem	6.000 W	77	Maquina de cortar cabelo	200 W
7	Aquecedor de ambiente	1000.W	78	Máquina de costura	100 W
8	Aspirador de pó comercial	2240 W	79	Máquina de lavar louças	1200 W
9	Aspirador de pó residencial	750 W	80	Máquina de lavar c/ aquecimento	1500 W
10	Assadeira pequena	500 W	81	Maquina de lavar s/ aquecimento	400 W
11	Assadeira grande	1000 W	82	Maquina de secar roupas (grande)	3500 W
12	Balcão frigorífico pequeno	500 W	83	Maquina de secar roupas (média)	1100 W
13	Balcão frigorífico grande	1000 W	84	Máquina de solda pequena	1000 W
14	Hidromassagem sem aquecedor	600 W	85	Maquina de sorvete	2200 W
15	Hidromassagem com aquecedor	6.600 W	86	Máquina de xérox	1500 W
16	Batedeira de bolo	100 W	87	Micro computador	100 W
17	Bebedouro	200 W	88	Moedor de carne	320 W
18	Betoneira	1000 W	89	Moinho para diversos grãos	600 W
19	Bomba de combustível	740 W	90	Motor monofásico de até 1/2 cv	370 W
20	Cadeira de dentista	190 W	91	Motor monofásico de 3/4 cv	550 W
21	Cafeteira elétrica para uso comercial	1200 W	92	Motor monofásico de 1,0 cv	750 W
22	Cafeteira elétrica p/ uso doméstico	750 W	93	Motor monofásico de 1,5 cv	1100 W
23	Carregador de bateria de oficina	1200 W	94	Motor monofásico de 2,0 cv	1500 W
24	Chuveiro elétrico 127 V	4.400 W	95	Motor monofásico de 3,0 cv	2200 W
25	Chuveiro elétrico 220 V	6.000 W	96	Motor monofásico de 4,0 cv	3000 W
26	Compressor pequeno	370 W	97	Motor monofásico de 5,0 cv	3700 W
27	Condicionador de ar até 7.500 Btu	755 W	98	Motor monofásico de 7,5 cv	5500 W
28	Condicionador de ar de 9.000 Btu	968 W	99	Motor monofásico de 10,0 cv	7500 W
29	Condicionador de ar de 10.000 Btu	1031 W	100	Motor trifásico até 1/2 ou 0,50 cv	370 W
30	Condicionador de ar de 12.000 Btu	1.204 W	101	Motor trifásico de 3/4 ou 0,75 cv	550 W
31	Condicionador de ar de 18.000 Btu	2.000 W	102	Motor trifásico de 1,0 cv	750 W
32	Condicionador de ar de 21.000 Btu	2.354 W	103	Motor trifásico de 1,5 cv	1100 W
33	Condicionador de ar de 30.000 Btu	3.800 W	104	Motor trifásico de 2,0 cv	1500 W
34	Conjunto de som profissional	500 W	105	Motor trifásico de 3,0 cv	2200 W
35	Conjunto de som residencial	100 W	106	Motor trifásico de 4,0 cv	3000 W
36	Copiadora xérox (grande)	2.500 W	107	Motor trifásico de 5,0 cv	3700 W
37	Copiadora xérox (pequena)	1.500 W	108	Motor trifásico de 6,0 cv	4500 W
38	Cortador de grama	1.600 W	109	Motor trifásico de 7,5 cv	5500 W
39	Digital Video Disco (DVD)	50 W	110	Motor trifásico de 10,0 cv	7500 W
40	Ebulidor elétrico	1000 W	111	Motor trifásico de 12,5 cv	9200 W
41	Enceradeira residencial	400 W	112	Motor trifásico de 15,0 cv	11000 W
42	Espremedor de frutas comercial	500 W	113	Motor trifásico de 20,0 cv	15500 W
43	Espremedor de frutas residencial	200 W	114	Motor trifásico de 25,0 cv	18500 W
44	Esterilizador	1000 W	115	Motor trifásico de 30,0 cv	22000 W
45	Estufa de dentista	1000 W	116	Multiprocessador	420 W
46	Exaustor para fogão	100 W	117	Outros equipamentos	
47	Fatiador para frios	740 W	118	Pipoqueira	110 W

Potência Padronizada para Eletrodomésticos					
Item	Descrição	Potência	Item	Descrição	Potência
48	Fax	240 W	119	Prancha alisadora para cabelo	1000 W
49	Ferro de passar roupas	1000 W	120	Projeto / retro projeto	210 W
50	Ferro de solda grande	600 W	121	Refletor para iluminação diversa	500 W
51	Ferro de solda médio	400 W	122	Registradora elétrica	100 W
52	Ferro de solda pequeno	100 W	123	Sauna comercial	12000 W
53	Fogão comum c/ acendedor elétrico	90 W	124	Sauna residencial	4.500 W
54	Fogão elétrico (potência por boca)	1500 W	125	Scanner	1.250 W
55	Forno de microondas	1200 W	126	Secador de cabelos	1000 W
56	Forrageira	1200 W	127	Serra de carne	1000 W
57	Freezer horizontal	280 W	128	Serra elétrica	1000 W
58	Freezer vertical	200 W	129	Serra tico tico	240 W
59	Frigobar	100 W	130	Som modular (por módulo)	50 W
60	Fritadeira média	1500 W	131	Sugar	200 W
61	Furadeira pequena	350 W	132	Televisor de até 20 polegadas	150 W
62	Geladeira de uma porta	110 W	133	Televisor maior que 20 polegadas	200 W
63	Geladeira duplex	250 W	134	Torneira elétrica	2500 W
64	Geladeira frost-free	500 W	135	Torno de bancada	1820 W
65	Grelha elétrica grande	1500 W	136	Torradeira elétrica	1000 W
66	Grelha elétrica pequena	500 W	137	Touca térmica	700 W
67	Hidromassagem com aquecedor	6600 W	138	Vaporizador	300 W
68	Hidromassagem sem aquecedor	660 W	139	Ventilador de teto	120 W
69	Impressora comum	90 W	140	Ventilador pequeno	65 W
70	Impressora laser	900 W	141	Ventilador grande 50 cm	250 W
71	Lâmpadas		142	Vibrador para concreto	1000 W

OBS: Os valores acima estabelecidos são estimados, devido às diferenças entre fabricantes, modelos, estado de conservação, etc, mas devem ser utilizados nos cálculos da carga instalada, da demanda máxima e conseqüente definição do padrão a ser instalado; salvo o cliente apresente a nota fiscal de aquisição e folheto do fabricante do equipamento.

Tabela 2 - Potências Nominais de Condicionadores de Ar

Potências Nominais de Condicionadores de Ar			
Capacidade		Potencial Nominal	
Btu/h	Kj/h	Watt	VA
7.000	7385	708	903
7.500	7913	755	944
9.000	9495	968	1187
10.000	10550	1.031	1.448
12.000	12660	1.204	1.474
18.000	18990	1.917	2216
21.000	22155	2.354	2667
30.000		3.480	3.866

Nota:

1 - As potências nominais foram calculadas com base na tabela CAD-2008 do Programa Nacional de Conservação de Energia Elétrica - PROCEL do IMMETRO.