



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SANTA CATARINA-CÂMPUS ITAJAÍ

GUSTAVO HENRIQUE SILVA ZANETTI

**PLUGIN PARA SOFTWARE DE DESENHO CAD PARA
AUTOMAÇÃO DE DESENHOS DE ESQUEMAS UNIFILARES E
MULTIFILARES**

ITAJAÍ-SC
2024



GUSTAVO HENRIQUE SILVA ZANETTI

**PLUGIN PARA SOFTWARE DE DESENHO CAD PARA
AUTOMAÇÃO DE DESENHOS DE ESQUEMAS UNIFILARES E
MULTIFILARES**

Monografia apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina-Câmpus Itajaí ligado ao Ministério da Educação, como requisito parcial da disciplina Trabalho de conclusão de curso - TCC conclusão do curso de Bacharelado em Engenharia Elétrica.

Área de concentração: Programação.

Disciplina: Trabalho de conclusão do curso - TCC

Orientador: Prof. Sergio A. B. Petrovcic, Dr. Eng.

ITAJAÍ-SC
2024

Ficha de Identificação da Obra
Sistema de Bibliotecas Integradas do IFSC - Campus Itajaí

Zanetti, Gustavo Henrique Silva
Z28p *Plugin* para software de desenho CAD para automação de desenhos de esquemas unifilares e multifilares / Gustavo Henrique Silva Zanetti ; orientador, Sergio A. B. Petrovic. -- 2024.
86 f.

Trabalho de Conclusão de Curso (Graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Campus Itajaí, Graduação em Engenharia Elétrica, Itajaí, 2024.

Inclui bibliografia.

1. Engenharia Elétrica. 2. Instalações elétricas. 3. Automação. I. Petrovic, Sergio A. B. II. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina – Graduação em Engenharia Elétrica. III. Título.

CDD 621.3192

PLUGIN PARA SOFTWARE DE DESENHO CAD PARA AUTOMAÇÃO DE DESENHOS DE ESQUEMAS UNIFILARES E MULTIFILARES

GUSTAVO HENRIQUE SILVA ZANETTI

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletricista e aprovado na sua forma final pela banca examinadora do curso de engenharia elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Itajaí, 26 de AGOSTO de 2024

BANCA EXAMINADORA:

Dr. Sergio Augusto Bitencourt Petrovic

Instituto Federal de Santa Catarina – IFSC

Dra. Ana Elisa Ferreira Schmidt

Instituto Federal de Santa Catarina – IFSC

Dr. Saimon Miranda Fagundes

Instituto Federal de Santa Catarina - IFSC

RESUMO

No contexto de projetos de instalações elétricas, a criação de esquemas unifilares é uma etapa crucial, porém demorada e sujeita a erros manuais. Para otimizar esse processo, foi desenvolvido um plugin para CAD na disciplina de Projeto Integrador III do curso de Engenharia Elétrica do IFSC - Campus Itajaí. Este plugin automatiza a geração de esquemas unifilares a partir do quadro de cargas, permitindo que o projetista se concentre na planilha do quadro de cargas e automatize a geração do esquema, reduzindo erros e aumentando a eficiência. Por meio de testes de usabilidade, métodos foram aplicados para avaliar o programa, recebendo uma excelente avaliação dos usuários e atingindo os objetivos estabelecidos. Além disso, indicadores de desempenho mostraram que o tempo de execução do projeto com o *plugin* foi mais de 100 vezes mais rápido comparado ao método convencional. A análise de erros revelou que o *plugin* não gerou erros na geração dos esquemas, enquanto o método convencional apresentou erros. O trabalho também incluiu melhorias em comparação à versão inicial, com a adição de novas funcionalidades e melhorias no *layout*.

Palavras-chave: esquema unifilar, *plugin* AutoCAD, automação, programação.

ABSTRACT

In the context of electrical installation projects, creating single-line diagrams is a crucial but time-consuming and error-prone step. To optimize this process, a CAD plugin was developed in the course “Projeto Integrador III” of the Electrical Engineering program at IFSC - Campus Itajaí. This plugin automates the generation of single-line diagrams from the load panel, allowing the designer to focus on the load panel spreadsheet and automate the diagram generation, reducing errors and increasing efficiency. Usability tests were conducted, and methods were applied to evaluate the program, receiving excellent user ratings and achieving established objectives. Additionally, performance indicators showed that project execution time with the plugin was over 100 times faster compared to the conventional method. Error analysis revealed that the plugin did not generate errors in diagram generation, while the conventional method had errors. The work also included improvements compared to the initial version, with the addition of new features and layout enhancements.

Keywords: single-line scheme, plugin, AutoCAD, programming.

LISTA DE FIGURAS

Figura 1 – Exemplo de um quadro de cargas.	7
Figura 2 – Exemplo de um esquema unifilar.	8
Figura 3 – Exemplo de um esquema multifilar.	9
Figura 4 – Disjuntor modelo físico e simbologia.	9
Figura 5 – Disjuntor residual modelo físico e simbologia - adaptado pelo autor. . .	10
Figura 6 – DPS modelo físico e simbologia - adaptado pelo autor	11
Figura 7 – Modelo do queijo suíço	12
Figura 8 – Interface do <i>plugin</i> - Implementação inicial projeto integrador.	22
Figura 9 – Mensagem de erro de entrada inválida - Demonstração.	23
Figura 10 – Janela de configurações - Implementação inicial projeto integrador. . .	24
Figura 11 – Desenho referência para o esquema unifilar (desenhado manualmente). . .	24
Figura 12 – Desenho gerado pelo <i>plugin</i> (desenho gerado automaticamente).	25
Figura 13 – Mensagem de inicialização do <i>plugin</i>	26
Figura 14 – <i>Ribbon tab</i> do <i>plugin</i>	26
Figura 15 – Interface do <i>plugin</i> - Implementação final menu geral.	27
Figura 16 – Interface do <i>plugin</i> - Implementação final menu de configurações. . . .	28
Figura 17 – Interface do <i>plugin</i> - Implementação lista de materiais.	28
Figura 18 – Interface do <i>plugin</i> - Implementação opções da lista de materiais. . . .	29
Figura 19 – Interface do <i>plugin</i> - Implementação quadro de cargas.	30
Figura 20 – Modelo de esquema de montagem.	30
Figura 21 – Interface do <i>plugin</i> - Guia montagem.	31
Figura 22 – Interface do <i>plugin</i> - Implementação menu sobre.	32
Figura 23 – Quadro de cargas utilizado em conjunto com o <i>plugin</i>	33
Figura 24 – Quadro de cargas - Mensagem de aviso para informação não padronizada. .	33
Figura 25 – Demonstração de projeto gerado pelo <i>plugin</i> , contendo - Esquema unifilar, lista de materiais, quadro de cargas simplificado e blocos para esquema de montagem.	33
Figura 26 – Relação entre benefício/custo e número de participantes.	34
Figura 27 – Pergunta 1 - Quadro de cargas - Sobre a facilidade de familiarização. . .	38
Figura 28 – Pergunta 2 - Quadro de cargas - Sobre a facilidade de navegação. . . .	38
Figura 29 – Pergunta 3 - Quadro de cargas - Sugestões de melhorias.	38
Figura 30 – Pergunta 4 - Quadro de cargas - Erros ou dificuldades durante o teste. . .	39
Figura 31 – Pergunta 1 - <i>Plugin</i> - Sobre a facilidade de familiarização.	39
Figura 32 – Pergunta 2 - <i>Plugin</i> - Sobre o resultado final da automação (esquema). .	39
Figura 33 – Pergunta 3 - <i>Plugin</i> - Sugestões de melhorias.	40
Figura 34 – Pergunta 4 - <i>Plugin</i> - Erros ou dificuldades durante o teste.	40

Figura 35 – Pergunta 1 - <i>Feedback</i> geral - Classificação de produtividade.	40
Figura 36 – Pergunta 2 - <i>Feedback</i> geral - Classificação de eficácia para redução de erros.	41
Figura 37 – Pergunta 3 - <i>Feedback</i> geral - Classificação de recomendação.	41
Figura 38 – Pergunta 4 - <i>Feedback</i> geral - Expectativa de tempo economizado. . . .	41
Figura 39 – Pergunta 5 - <i>Feedback</i> geral - Comentários livres.	41
Figura 40 – Fluxograma operacional do <i>plugin</i>	42

LISTA DE QUADROS

Quadro 1 – Etapas do desenvolvimento do programa.	20
Quadro 2 – Dados sobre os participantes no teste de usabilidade.	35
Quadro 3 – Compêndio itens do teste.	35
Quadro 4 – Roteiro do teste de usabilidade.	37

LISTA DE TABELAS

Tabela 1 – Métricas do código - Comparação entre a versão inicial e final.	43
Tabela 2 – Tempo de execução de projetos manualmente.	44
Tabela 3 – Tempo de execução de projetos pelo <i>plugin</i>	44
Tabela 4 – Indicador de desempenho - Comparação da média aritmética do tempo de execução do projeto entre métodos.	44
Tabela 5 – Erros gerados pelo método manual	45
Tabela 6 – Indicador de desempenho - Comparação da média aritmética de erros gerados na execução do projeto entre métodos.	45
Tabela 7 – Avaliações referentes ao quadro de cargas.	46
Tabela 8 – Avaliações referentes ao <i>plugin</i>	47
Tabela 9 – Avaliações referentes ao <i>feedback</i> geral.	47
Tabela 10 – Tempo estimado pelos participantes economizado com a utilização do <i>plugin</i>	48

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i>
ART	Anotação de Responsabilidade Técnica
BIM	<i>Building Information Management</i>
CAD	<i>Computer Aided Design</i>
CELESC	Centrais Elétricas de Santa Catarina
DLL	<i>Dynamic Link Library</i>
DPS	Dispositivo de proteção contra surtos
DR	Dispositivo diferencial residual
IDE	<i>Integrated Development Environment</i>
IDR	Interruptor diferencial residual
JSON	<i>JavaScript Object Notation</i>
NBR	Norma Brasileira
OOP	<i>Object Oriented Programming</i>
PI	Projeto integrador
TCC	Trabalho de Conclusão de Curso

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 Problema e justificativa	2
1.2 Objetivo geral	2
1.3 Objetivos específicos	2
1.4 Estrutura do documento	3
2 – REFERENCIAL TEÓRICO	4
2.1 Projeto de instalações elétricas	4
2.2 Etapas de elaboração do projeto de instalações elétricas	4
2.2.1 Informações preliminares	4
2.2.2 Quantificação do sistema	5
2.2.3 Determinação do padrão de atendimento	5
2.2.4 Quadros de distribuição e esquemas	5
2.2.5 Dimensionamento	6
2.2.6 Desenho das plantas	6
2.2.7 Elaboração dos detalhes construtivos	6
2.2.8 Memorial descritivo e cálculo	6
2.2.9 Elaboração da lista de material	7
2.3 Divisão das cargas da instalação (quadro de cargas)	7
2.4 Esquemas unifilar e multifilar	8
2.4.1 Disjuntor	8
2.4.2 IDR/DR - Interruptores de corrente diferencial-residual	10
2.4.3 DPS - Dispositivo de proteção contra surtos (de tensão)	10
2.5 Erros humanos	11
2.6 Conceitos computacionais	12
2.6.1 Ambiente de programação	12
2.6.2 .NET <i>Framework</i>	12
2.6.3 API (<i>Application Programming Interface</i>)	13
2.6.4 DLL (<i>Dynamic Link Library</i>)	13
2.6.5 <i>Microsoft Visual Studio</i>	14
2.6.6 <i>Windows Forms</i>	14
2.6.7 Repositório (Repo), <i>Git</i> e <i>Github</i>	14
2.6.8 CAD	14
2.6.9 AutoCAD	15
2.7 AutoLISP e Visual LISP	15
2.7.1 Blocos	15

2.7.2	Blocos Dinâmicos	16
2.8	Teste de Usabilidade	16
2.8.1	Heurística de Nielsen	17
2.9	Trabalhos Relacionados	17
2.9.1	PRO-Elétrica	17
2.9.2	<i>Plugin</i> de AutoCAD para automatização de etapas do desenvolvimento de projetos de instalações elétricas	18
2.9.3	Análise dos indicadores de desempenho da máquina <i>Reach Stacker</i> após a implementação de <i>software</i> testador de <i>joystick</i> no setor de manutenção portuária	19
3	– DESENVOLVIMENTO DO <i>PLUGIN</i>	20
3.1	Versão inicial - Projeto Integrador	21
3.1.1	Criação da tabela padrão do quadro de cargas	21
3.1.2	Programação do <i>plugin</i> para o CAD	21
3.1.2.1	Demonstração do <i>plugin</i>	24
3.2	Versão final - TCC	25
3.2.1	Interface - Alteração do <i>layout</i>	25
3.2.1.1	Inicialização	25
3.2.1.2	Menu geral	26
3.2.1.3	Configurações	27
3.2.1.4	Lista de material	27
3.2.1.5	Opções da lista de material	29
3.2.1.6	Cargas	29
3.2.1.7	Esquema de montagem	29
3.2.1.8	Sobre	31
3.2.1.9	Guia de ajuda - Manual de instruções	31
3.2.1.10	Quadro de cargas	32
3.2.2	Demonstração de projeto	32
4	– TESTE DE USABILIDADE	34
4.0.1	Objetivos do teste	34
4.0.2	Preparação do teste	34
4.0.2.1	Seleção dos participantes	34
4.0.2.2	Ambiente de teste	35
4.0.2.3	Condução dos testes	35
4.0.2.4	Questionário	36
4.0.3	Aplicação do teste	36
4.0.3.1	Roteiro dos testes	36
4.0.3.2	Arquivos de teste	36

4.0.4	Compilação dos resultados	36
4.0.4.1	Coleta, análise e apresentação dos dados	36
5	– RESULTADOS E DISCUSSÕES	42
5.1	<i>Plugin</i>	42
5.1.1	Indicadores de desempenho	43
5.2	Resultados do teste de usabilidade	45
5.2.1	Quadro de cargas	45
5.2.2	<i>Plugin</i>	46
5.2.3	<i>Feedback</i> geral	47
6	– CONCLUSÃO	49
6.1	Sugestões para trabalhos futuros	49
6.2	Considerações Finais	50
	Referências	51
	 Apêndices	 55
	APÊNDICE A –Manual do <i>plugin</i>	56
	APÊNDICE B –Roteiro do teste de usabilidade	69

1 INTRODUÇÃO

Uma das primeiras práticas aprendidas pelos seres humanos foi a contagem com os dedos ou também conhecida como: dactilonomia. Sendo cada dedo representado por uma unidade. Com a constante evolução humana, desenvolvemos o Ábaco, se não um dos primeiros, certamente um dos mais antigos instrumentos de cálculo desenvolvido. Essas simples ações criadas por nós, pode ser considerado como os primeiros passos que demos no desenvolvimento de computadores e criação de *softwares*.

“A programação de computadores segue uma técnica formal conhecida como ‘algoritmo’.” (WAZLAWICK, 2016) Os algarismos indianos (conhecidos como arábicos ou indo-arábicos), foram um ponta pé inicial para procedimentos (algoritmo) de adição e divisões com números longos, proporcionando a base para a construção das primeiras calculadoras mecânicas.

Com o passar do tempo evoluímos, os mecanismos de *hardware* e *software*, passando por computadores mecânicos, a criação do transistor, circuitos integrados, os primeiros computadores eletrônicos e na metade do século XX o surgimento da base das linguagens de programação. O objetivo de todo esse desenvolvimento tecnológico é apenas um: criar mecanismos que solucionem de forma ágil nossos problemas. A junção de *hardware* e *software* proporciona a criação de ferramentas que agilizam tarefas repetitivas e maçantes, especialmente nas áreas de engenharia.

Quando se tratando de projetos de instalações elétricas, uma parte fundamental do desenvolvimento é o quadro de cargas e esquema unifilar. Um esquema unifilar é uma representação gráfica simplificada de um sistema elétrico (FILHO, 1997), que mostra de forma clara e concisa a conexão entre os componentes principais, como disjuntores, cabos, dispositivos de proteção diferencial residual (DR), dispositivos de proteção contra surtos (DPS), cargas e etc. Faz-se importante por razões de documentação, segurança, manutenção, conformidade regulatórias e padronização.

Em um processo convencional, o projetista determina as cargas de um quadro, geralmente em uma planilha e, na sequência, esses dados são transcritos manualmente, tanto na forma de um desenho como em dados técnicos de especificação dos componentes, para um esquema unifilar/multifilar (CREDER, 2021).

Neste trabalho, será apresentada a criação de um *plugin* para o AutoCAD, destinado à automação da geração de desenhos de esquemas unificares. Este plugin visa simplificar e agilizar o processo, minimizando erros e aumentando a eficiência na elaboração de projetos elétricos.

1.1 Problema e justificativa

Um dos processos na concepção de projetos de instalações elétricas prediais é a representação gráfica de um esquema unifilar, um desenho simplificado de um quadro de distribuição contendo informações concisas sobre o sistema elétrico (FILHO, 1997). Grande parte dessa etapa consiste no trabalho do projetista em copiar dados de uma planilha e gerar o desenho correspondente às informações do quadro de cargas em um programa de desenho CAD. Esse processo, além de ser demorado, às vezes leva o projetista a cometer erros por descuido ou outros fatores, conforme abordado por (KANTOWITZ; SORKIN, 1983), resultando na reprodução de informações equivocadas no esquema unifilar.

Dada a complexidade e o tempo envolvidos no desenvolvimento de projetos elétricos, qualquer abordagem que possa otimizar esse processo é altamente benéfica. A automação não apenas reduz o risco de erros, mas também agiliza todo o ciclo de projeto, desde a concepção inicial até a implementação final. Essa economia de tempo e esforço não só melhora a eficiência do trabalho, mas também pode ter impactos positivos na segurança, custo e cronograma do projeto como um todo.

Portanto, visando agilizar e mitigar os erros do processo, foi desenvolvido, na disciplina de Projeto Integrador III do curso de Engenharia Elétrica do IFSC - Campus Itajaí, um plugin para software tipo CAD capaz de automatizar a criação de esquemas unifilares a partir do quadro de cargas. Com esse plugin, o projetista pode focar exclusivamente no desenvolvimento do quadro de cargas em uma planilha, deixando a geração do esquema unifilar para o final do processo, de forma automatizada.

Este trabalho explora as vantagens da automação em projetos elétricos, dando sequência ao desenvolvimento de um *plugin* para automação de esquemas unifilares em projetos elétricos, iniciado na disciplina de Projeto Integrador III do curso de Engenharia Elétrica. O trabalho adiciona novas funcionalidades e aprimora funcionalidades já existentes, além de avaliar, por meio de um teste de usabilidade, se sua aplicação foi bem-sucedida, demonstrando os resultados e vantagens dessa solução.

1.2 Objetivo geral

Estudar a aplicação e viabilidade do *plugin* desenvolvido para softwares de desenho CAD, voltado para a geração automática de esquemas unifilares com base em um quadro de cargas, a fim de aumentar a produtividade no processo de projetos elétricos industriais e prediais, e mitigar os erros humanos gerados ao longo da concepção do mesmo.

1.3 Objetivos específicos

- Desenvolver novas funcionalidades em relação ao *plugin* inicial, e aprimoramento de classes, métodos e parâmetros;

- Realizar o processo de validação do *plugin* por meio de testes de usabilidade, seguindo as etapas a seguir:
 - Definir o cronograma de testes;
 - Selecionar os participantes;
 - Determinar os exemplos para os testes;
 - Coletar os dados dos testes por meio de formulários preenchidos pelos usuários;
 - Avaliar o ganho de produtividade proporcionado pelo *plugin*;
 - Analisar o *feedback* fornecido pelos usuários;
 - Apresentar os resultados obtidos.
- Gerar a documentação do *plugin* (guia para usuário);
- Implementar novas funções, caso necessário, com base nas demandas do usuário.

1.4 Estrutura do documento

O documento está dividido em 6 capítulos. No capítulo 2 é apresentado o referencial teórico abordando os principais conceitos e estudos utilizado para a elaboração desse trabalho. São descritos os conceitos de instalações de projetos elétricos, uma abordagem que analisa as causas de erros humanos e por último, conceitos computacionais.

O capítulo 3 relata o desenvolvimento do *plugin* abordando a versão inicial, desenvolvimento das interfaces, ferramentas, criação da tabela padrão e apresentação da versão final.

O teste de usabilidade busca compreender se o programa (*plugin*) é utilizável e eficiente na sua proposta, os conceitos aplicados pelo teste são descritos no capítulo 4.

Na sequência, o capítulo 5 apresenta os resultados e discussões, apresentado os indicadores de desempenho e os resultados do teste de usabilidade.

Por fim, são apresentadas as conclusões finais deste trabalho, as propostas para trabalhos futuros e as conclusões finais apresentadas no capítulo 6.

2 REFERENCIAL TEÓRICO

Este capítulo tem como objetivo apresentar ao leitor os diferentes conceitos, tecnologias e termos técnicos utilizados neste trabalho.

2.1 Projeto de instalações elétricas

O projeto elétrico pode ser definido como, segundo (CREDER, 2021):

”O projeto elétrico pode ser definido como uma previsão escrita da instalação elétrica, contendo todos os seus detalhes, a localização dos pontos de utilização da energia elétrica, dos pontos de comandos, do trajeto dos eletrodutos e condutores, a divisão em circuitos, o dimensionamento da seção dos condutores, os dispositivos de manobra e proteção, a carga de cada circuito, a carga total etc.”

Para compor um projeto elétrico, o projetista deve estar atento às normativas empregadas, para proporcionar segurança e eficácia. Tratando-se do Brasil, as principais normativas são listadas abaixo, podendo ser dividido entre normas de âmbito nacional como as normas brasileira (ABNT NBR), e as normas da concessionária local:

- NBR-5410 (ABNT, 2004): Instalações elétricas de baixa tensão;
- N-321.0001 (CELESC, 2019): Fornecimento de energia elétrica em tensão secundária de distribuição;
- N-321.0003 (CELESC, 2023): Fornecimento de energia elétrica a edifícios de uso coletivo.

2.2 Etapas de elaboração do projeto de instalações elétricas

Um projeto consiste em várias etapas até que seu resultado final seja alcançado, é de suma importância seguir a linha de procedimento para que o projeto não resulte em retrabalho ou pior, uma incompatibilização entre disciplinas (projetos) ou normativa. As etapas serão descritas a seguir.

2.2.1 Informações preliminares

Nessa etapa o projetista busca obter o máximo de informações de todas as fontes, para assim formar uma concepção geral do projeto a ser desenvolvido, através de: (FILHO, 1997)

- Planta de situação:
Nessa planta o projetista terá noção dos acessos ao edifício, onde está localizada a rede de energia elétrica da concessionária, e os pontos de derivação para atendimento a edificação.
- Projeto arquitetônico:
Nessa planta indica os cortes, detalhes, pé direito, fachadas etc. Obtendo-se todas as dimensões da edificação.
- Projetos complementares:
Projeto estrutural, projeto sanitário, combate a incêndio, telecomunicações, sonorização, automação e entre outros. Esses projetos visam proporcionar a compatibilização entre disciplinas, para assim evitar encontros que podem vir a resultar em mudanças drásticas ou inviabilidade do projeto.
- Informações obtidas com o proprietário, arquiteto ou responsável:
Nessa etapa o projetista elétrico busca informações referentes a pontos específicos da instalação, seja um equipamento ou ponto de alimentação como de ar-condicionado; previsão de cargas para futuros acréscimos.

2.2.2 Quantificação do sistema

Com os dados preliminares obtidos, e baseando-se nas normativas vigentes, o projetista pode fazer o levantamento de previsão de cargas, sendo o posicionamento dos pontos e a quantificação da potência nominal. (FILHO, 1997)

2.2.3 Determinação do padrão de atendimento

Após a finalização da etapa anterior, o projetista determina a demanda e a categoria de fornecimento conforme a normativa vigente da concessionária local. Determinando, portanto, a provável demanda e o padrão de entrada de serviço. (FILHO, 1997)

2.2.4 Quadros de distribuição e esquemas

Com base nos pontos distribuídos e na potência nominal, nessa etapa o projetista elabora o quadro de cargas, que têm a função de representar o dimensionamento dos circuitos por meio de tabelas e esquemas. (FILHO, 1997)

2.2.5 Dimensionamento

Nesta etapa, o dimensionamento de todos os elementos do projeto, sendo que algumas etapas do dimensionamento podem ser ajustadas após, como por exemplo, a tubulação da instalação, visto que, o projetista ainda não tem noção da ocupação da tubulação somente após o lançamento de fiação. (FILHO, 1997)

2.2.6 Desenho das plantas

Esta etapa compreende basicamente:

1. Desenho dos pontos;
2. Localização dos quadros de distribuição;
3. Localização das caixas de passagem;
4. Localização do quadro de medidor(es);
5. Elaboração do quadro de cargas;
6. Caminhamento de tubulações;
7. Inserção da fiação;
8. Desenho do esquema unifilar;
9. Desenho do esquema vertical (prumada)(FILHO, 1997).

2.2.7 Elaboração dos detalhes construtivos

A finalidade dessa etapa é facilitar a interpretação do projeto, como por exemplo cortes verticais da instalação. Quanto mais detalhada o projeto melhor tende a ser a execução. (FILHO, 1997)

2.2.8 Memorial descritivo e cálculo

Nesta etapa, é gerado o documento com o objetivo de descrever de forma sucinta o projeto, justificando, as soluções adotadas e o que foi necessário utilizar. Composto por: (FILHO, 1997)

1. Dados básicos de identificação do projeto;
2. Dados quantitativos do projeto;
3. Descrição geral do projeto;
4. Documentação do projeto;
5. Cálculos das previsões de cargas;
6. Determinação da provável demanda;
7. Dimensionamento de condutores;
8. Dimensionamento de eletrodutos;
9. Dimensionamento dos dispositivos de proteção.

2.2.9 Elaboração da lista de material

Listagem de todos os materiais empregados no projeto, com especificações conforme fabricante e quantidade.

2.3 Divisão das cargas da instalação (quadro de cargas)

Para assegurar maior eficiência, segurança e facilitar a manutenção toda carga da instalação elétrica, a mesma deve ser dividida em vários circuitos, de modo que, previne o desligamento geral da instalação em caso de um elemento defeituoso na instalação; facilita a verificação e manutenção; possibilita a economia e eficiência ao utilizar em circuitos uma seção circular menor. (CREDER, 2021)

O termo *circuito* é empregado quando se tem um conjunto de equipamentos ligado no mesmo condutor e ao mesmo dispositivo de segurança (disjuntores termomagnéticos ou disjuntores residuais diferenciais (DR)).

Em sistemas polifásicos, os circuitos são distribuídos de forma a garantir melhor equilíbrio de cargas na instalação, e preferencialmente, seguindo a distribuição física de barramentos.

A carga que se considera para um equipamento ou circuito é a sua potência nominal absorvida, normalmente fornecida pelo fabricante ou calculada.

Por fim, os dados são preenchidos em uma tabela, com todas as informações de um circuito separada por linha e os dados de potência, seção do condutor, dispositivo de proteção, tensão de operação etc, separados por coluna. Ao final, obtêm-se a potência total do quadro, onde é então definida a proteção geral e a demanda da instalação desse quadro. A figura 1 mostra um modelo de quadro de cargas.

Figura 1 – Exemplo de um quadro de cargas.

QD - APARTAMENTO								
CIRCUITO	LOCAL	POTÊNCIA	FASE	FASE	FASE	CONDUTOR	DISJUNTOR DIN	IDR
		TOTAL(W)	A	B	C	(mm ²)	(A) - NBR NM60 898	(Fuga 30mA)
1	ILUMINAÇÃO COZINHA, SALAS E ÁREA DE SERVIÇO	600	600			3#1,5	Curva C – 1x10 A	
2	ILUMINAÇÃO QUARTOS E BWC	440		440		3#1,5	Curva C – 1x10 A	
3	TOMADAS COZINHA	1200			1200	3#4,0	Curva C – 1x25 A	
4	TOMADAS SALA DE ESTAR E SALA DE JANTAR	800	800			3#2,5	Curva C – 1x16 A	
5	TOMADAS FORNOS	2000		2000		3#4,0	Curva C – 1x20 A	DR1-4P-40A
6	TOMADAS QUARTOS e BWC	1300			1300	3#2,5	Curva C – 1x16 A	
7	TOMADAS ÁREA DE SERVIÇO	1200	1200			3#4,0	Curva C – 1x20 A	
8	CHUVEIRO SUÍTE 1	6600		6600		3#6,0	Curva B – 1x32 A	DR3-2P-40A
9	CHUVEIRO SUÍTE 2	6600			6600	3#6,0	Curva B – 1x32 A	DR3-2P-40A
10	CHUVEIRO SUÍTE 3	6600	6600			3#6,0	Curva B – 1x32 A	DR4-2P-40A
11	AR-CONDICIONADO SUÍTE 1	900			900	3#2,5	Curva C – 1x16 A	
12	AR-CONDICIONADO SUÍTE 2	900	900			3#2,5	Curva C – 1x16 A	
13	AR-CONDICIONADO SUÍTE 3	900		900		3#2,5	Curva C – 1x16 A	
14	AR-CONDICIONADO SALA DE ESTAR	900			900	3#2,5	Curva C – 1x16 A	
x10	ESPAÇO FÍSICO							
	TOTAL	30940	10100	9940	10900	5#10,0	Curva C – 3x50 A	
ESPECIFICAÇÃO: QUADRO ELÉTRICO - TIPO EMBUTIR - COM 3x16 (48) ESPAÇOS DISJUNTORES DIN – CONTENDO BARRAMENTO DE NEUTRO, TERRA E DEMAIS COMPONENTES MONTADOS.								

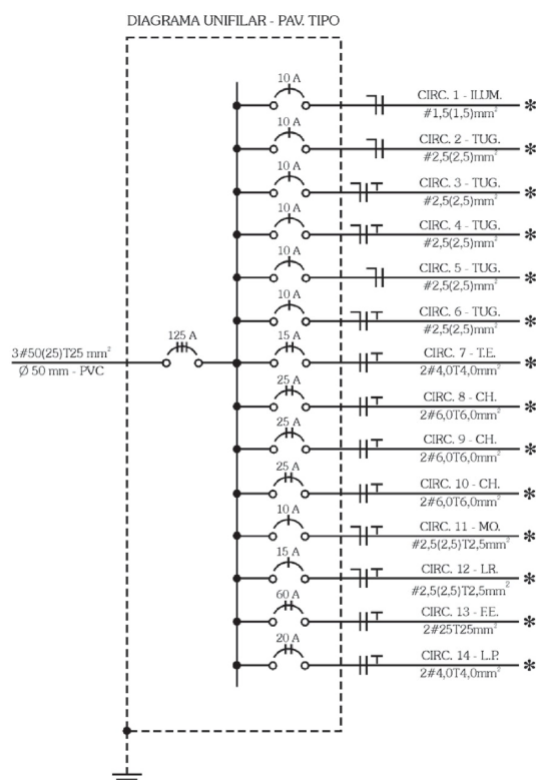
Fonte: Autor.

2.4 Esquemas unifilar e multifilar

Os esquemas unifilares são desenhos esquemáticos dos quadros de distribuição. Esses desenhos são executados a partir das informações contida nos quadros de distribuição de cargas e, de certa forma, constituem a representação desses quadros em forma de desenho. (FILHO, 1997)

Quando o esquema é dito unifilar trata-se da representação com apenas uma linha, sendo representada a fiação, circuito de saída, dispositivo de proteção etc. A figura 2 mostra o exemplo de um esquema unifilar.

Figura 2 – Exemplo de um esquema unifilar.



Fonte: Projeto de instalações elétricas prediais (FILHO, 1997).

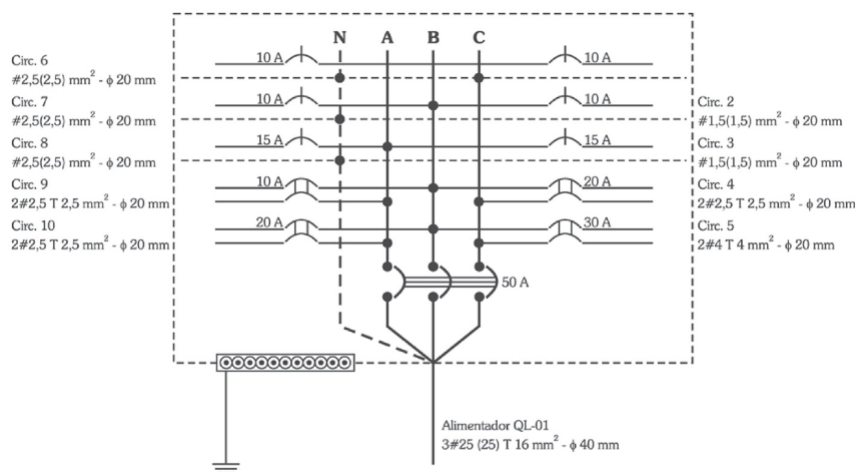
No caso do esquema multifilar, a representação tende a ser mais semelhante ao real, onde cada elemento do circuito tem uma linha exclusiva, representado de forma completa as conexões. A figura 3 mostra o exemplo de um esquema multifilar.

2.4.1 Disjuntor

Segundo a normativa ABNT NBR 5361:1998 (ABNT, 1998), define disjuntor como:

“Dispositivo de manobra (mecânico) e de proteção capaz de estabelecer, conduzir e interromper correntes em condições normais do circuito, assim como estabelecer, conduzir por tempo especificado e interromper correntes em condições anormais especificadas do circuito, tais como as de curto-circuito.”

Figura 3 – Exemplo de um esquema multifilar.



Fonte: Projeto de instalações elétricas prediais (FILHO, 1997).

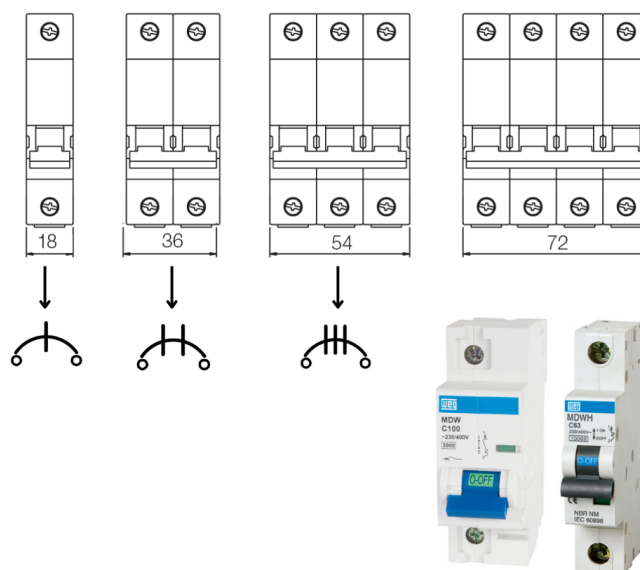
Os disjuntores podem ser classificados ainda como:

- **Disjuntor unipolar:** disjuntor com um único pólo (conexão);
- **disjuntor multipolar:** disjuntor com dois ou mais pólos (conexões).

Os disjuntores podem ser classificados, portanto, em unipolar/monopolar, bipolar, tripolar e tetrapolar. Há ainda classificação de caixa moldada, que são disjuntores maiores normalmente empregados para correntes elevadas, e os mesmos contam com uma alta capacidade de interrupção de curto circuito. A figura 4 mostra os modelos de disjuntores indicando os tamanhos em padrão DIN e abaixo a sua respectiva simbologia empregada para os disjuntores.

Figura 4 – Disjuntor modelo físico e simbologia.

MDW (2 A...63 A) SIW (40 A, 63 A)



Fonte: adaptado de (WEG, 2024).

2.4.2 IDR/DR - Interruptores de corrente diferencial-residual

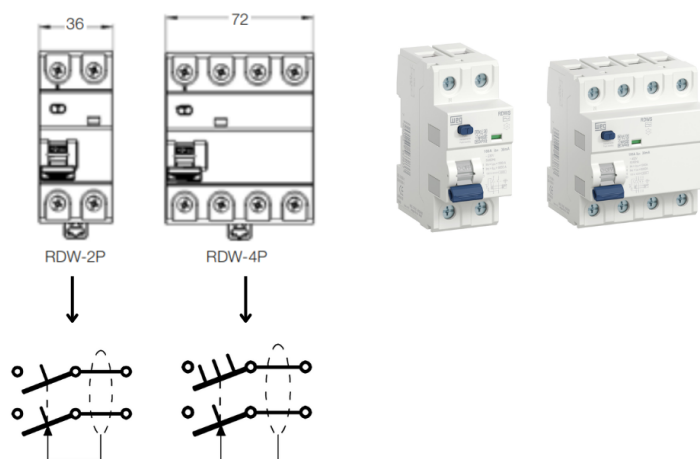
Assim como o disjuntor, o IDR e o DR são dispositivos mecânicos de interrupção. No entanto, esses dispositivos possuem a adição de um monitoramento interno da corrente de falta, ou seja, a corrente que flui em decorrência de uma falha, geralmente direcionada ao terra. A principal função tanto do IDR quanto do DR é proteger contra choques elétricos. Vale destacar que o IDR atua exclusivamente na presença de correntes de falta e não responde a situações de sobrecarga. Por outro lado, o DR, além de monitorar e atuar sobre correntes de falta, também é capaz de detectar e interromper correntes de sobrecarga, oferecendo uma proteção mais ampla.

Dispositivos chamados de IDR são dispositivos assim como os DR, com exceção de não contarem com dispositivo interno de proteção contra as sobrecorrentes. Os dispositivos podem ser classificados em bipolares e tetrapolares, basicamente. A figura 5 mostra o modelo físico com indicações de tamanho e o modelo de simbologia empregado.

Segundo a normativa NBR IEC 61008-1 (ABNT, 2022), define a utilização de dispositivos diferencial-residual como:

“...são destinados à proteção das pessoas contra os contatos indiretos às partes metálicas acessíveis da instalação...Eles podem ser utilizados para assegurar a proteção contra os perigos de incêndio resultantes de uma corrente de falta persistente à terra, sem intervenção do dispositivo de proteção contra as sobrecorrentes do circuito.”

Figura 5 – Disjuntor residual modelo físico e simbologia - adaptado pelo autor.



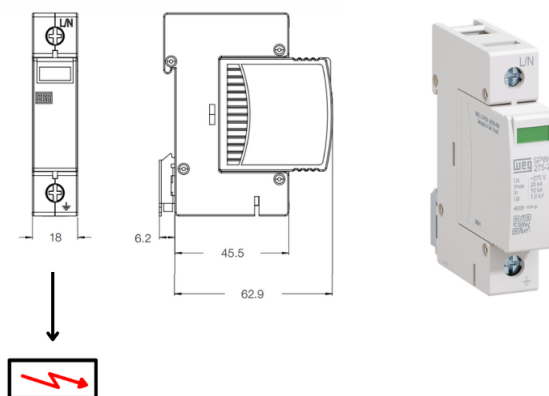
Fonte: adaptado de (WEG, 2023).

2.4.3 DPS - Dispositivo de proteção contra surtos (de tensão)

DPS (dispositivo de proteção contra surtos) são dispositivos que limitam os surtos de tensão ocasionados na rede, escoando a corrente de surto, assim evitam que aparelhos

eletro/eletrônicos venham a queimar ou apresentar falhas. Normalmente esses surtos são ocasionados por descargas atmosféricas (ABNT, 2021). A figura 6 mostra um modelo físico e abaixo a simbologia utilizada.

Figura 6 – DPS modelo físico e simbologia - adaptado pelo autor



Fonte: adaptado de (WEG, 2023)

2.5 Erros humanos

O livro *"Human Error"* (Erro Humano) de James Reason (REASON, 1990), aborda as causas e implicações dos erros cometidos pelos seres humanos em diversas situações, o autor destaca a importância de compreender os fatores que contribuem para os erros e propõe abordagens para promover mais ambientes seguros e eficientes. Uma das premissas básicas está na abordagem sistêmica, em que os seres humanos são suscetíveis a erros. Entendendo que não podemos mudar a condição humana, podemos mudar as condições sob as quais os humanos trabalham. Ou seja, criar um sistema redundante, ou criar barreiras que impeçam que um erro ocorra.

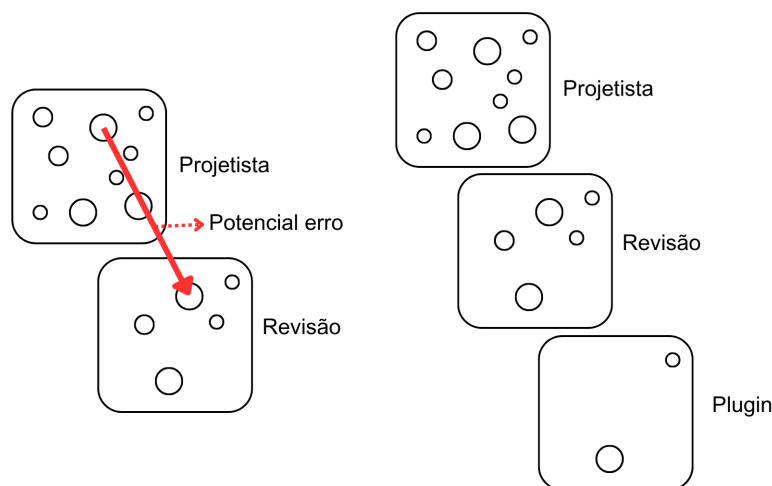
Uma das metáforas centrais apresentadas por Reason (REASON, 1990) é o modelo de "queijo suíço", que ilustra os mecanismos de proteção do sistema e as possíveis falhas. Essa analogia enfatiza que os erros só ocorrem quando os furos no queijo (representando as falhas) se alinham aos mecanismos de proteção, ressaltando a importância desses mecanismos na prevenção de erros.

Segundo (KANTOWITZ; SORKIN, 1983) as falhas podem ser diversas e classificadas em diferentes categorias, como por exemplo, falhas operacionais, execução de múltiplas tarefas, alta carga de trabalho, fadiga, capacitação, excesso de tarefas cognitivas e muitas outras. Sendo assim, os erros humanos são destacados por fatores individuais, tecnológicos e organizacionais (SILVA et al., 2017).

Utilizando-se da analogia proposta por Reason (REASON, 1990) do queijo suíço, tem-se que, um erro só é gerado quando há um alinhamento do furo do queijo entre os mecanismo que protegem da falha ocorrer, evidente que quanto menor a quantidade de mecanismos de proteção, mais propício que um erro ocorra. A figura 7 é uma adaptação

do queijo suíço aplicado para projetos elétricos, na busca de ilustrar os mecanismos de proteção, conforme o projeto avança os erros tendem a ser menores e com a adição de novos mecanismo os erros tendem a serem mitigados.

Figura 7 – Modelo do queijo suíço



Fonte: autor

2.6 Conceitos computacionais

2.6.1 Ambiente de programação

Um ambiente de programação, segundo (SEBESTA, 2018), também intitulado de *Integrated Development Environment* (IDE) ou ambiente de desenvolvimento integrado, engloba todas as ferramentas para o desenvolvimento de um *software*, podendo consistir em apenas um sistema de arquivos, um editor de textos, um compilador e um depurador.

2.6.2 .NET Framework

O *.NET Framework* nada mais é do que uma plataforma de desenvolvimento, que gerencia os aplicativos destinados para tal. Os dois principais componentes são o CLR (*Common Language Runtime*), o mecanismo de execução que manipula aplicativos em execução, e a biblioteca de classes (*class library*), que inclui APIs para leitura, escrita de arquivos, desenhos e muito mais (MICROSOFT, 2023b).

Aplicações do tipo *.NET* podem ser programadas em *C#* (lê-se *C Sharp*), *F#* (lê-se *F Sharp*), ou *Visual Basic*. O código é compilado pelo *CIL* (*Common Intermediate Language*) e então armazenado em *assemblies*, ou seja, arquivos com a terminação *.dll* ou *.exe*. Ao executar o arquivo o CLR (*Common Language Runtime*) converte em tempo de execução o código para a linguagem da máquina em que o mesmo está executando, seja ela *Linux*, *macOS*, *Windows*, *iOS*, *Android*, entre outras (MICROSOFT, 2023c). Há diversas vantagens na utilização da aplicação *.NET*, como por exemplo:

- Trabalhar com diversas linguagens e ambientes de programação, como por exemplo, C#, *Visual Basic*, *Python* e *JavaScript*;
- fácil integração entre aplicações do *Windows*, como por exemplo, *Microsoft Excel* e *Word*;
- permite o acesso a interfaces de programação avançadas com uma curva de aprendizado suave em comparação com linguagens de programação tradicional como o C++.

2.6.3 API (*Application Programming Interface*)

Em tradução Interface de Programação de Aplicativos (API) nada mais é do que um mecanismo que permite que dois *softwares* estabeleçam comunicação, funcionando como um contrato que define os protocolos de requisição e resposta (AWS, 2024).

A API .NET do AutoCAD permite que você manipule a aplicação e arquivos de desenho via programação utilizando as *assemblies* ou bibliotecas expostas (AUTODESK, 2024h).

2.6.4 DLL (*Dynamic Link Library*)

Biblioteca de vínculo dinâmico, em tradução, é uma biblioteca que contém código e dados que podem ser usados por mais de um programa ao mesmo tempo. O programa pode ser feito em módulos separados, facilitando em atualizações, de forma que não interfira globalmente no programa (MICROSOFT, 2024d).

A API do AutoCAD para a plataforma .NET foi desenvolvida utilizando vários arquivos DLL, que armazenam classes, estruturas, métodos e eventos para acessar objetos no arquivo de desenho da aplicação. Cada DLL estabelece *namespaces* (são conjuntos de identificadores que servem para agrupar funcionalidades comuns) distintos para organizar os componentes das bibliotecas de acordo com suas funcionalidades (AUTODESK, 2024e). As principais DLLs do AutoCAD .NET API são:

- **AcCoreMgd.dll:** utilizada ao trabalhar dentro do editor, publicar e plotar, e definir comandos e funções que podem ser chamados do AutoLISP.
- **AcDbMgd.dll:** utilizada ao trabalhar com objetos armazenados em um arquivo de desenho.
- **AcMgd.dll:** utilizada ao trabalhar com a aplicação e a interface do usuário.
- **AcCui.dll:** utilizada ao trabalhar com arquivos de personalização.

Vale ressaltar que antes de iniciar um projeto com API do AutoCAD, todas essas DLLs devem ser referenciadas no ambiente de programação, só assim faz-se possível utilizar os *namespaces* e os componentes da biblioteca (DLL).

2.6.5 Microsoft Visual Studio

O ambiente de programação (IDE) escolhido para construir esse projeto foi o *Visual Studio*, pelo fato de ser gratuito e ser a interface mais completa disponível, com ela faz-se possível escrever, editar, depurar, e compilar o código, muito além facilitando a integração com o *Windows Forms* e diversas outras extensões (MICROSOFT, 2023d).

2.6.6 Windows Forms

O *Windows Forms* é uma estrutura de interface do usuário (UI) para desenvolver aplicativos de desktop no Windows, oferecendo uma das maneiras mais eficientes de criar aplicativos com o auxílio do designer visual presente no Visual Studio. (MICROSOFT, 2023a)

2.6.7 Repositório (Repo), Git e Github

Repositórios de softwares, como Git, armazenam todos os diretórios e arquivos de um projeto, permitindo o controle de versão durante o desenvolvimento. Com eles, os desenvolvedores podem fazer modificações no código, mantendo um registro detalhado do progresso para recuperação de arquivos, se necessário. Essa centralização facilita o trabalho colaborativo, possibilitando a edição simultânea de arquivos e a colaboração eficiente (AWS, 2023).

Git é um sistema de controle de versão que efetivamente monitora as mudanças em arquivos, sendo especialmente útil quando você e um grupo de pessoas estão simultaneamente modificando os mesmos arquivos. O *GitHub* é uma plataforma em nuvem que funciona com o controle *Git*, facilitando o compartilhamento remoto do código (GITHUB, 2024).

A maneira para gerenciar esse projeto foi a utilização do *GitHub*, nele é possível fazer o versionamento do código, ou seja, cada alteração fica registrada podendo ser acessada a qualquer momento. Além disso, permite criar ramificações (*branches*), para que possa ser desenvolvidas partes separadas do programa. Oferece também um controle de *issues* (problemas) que podem ser cadastrados e trabalhados em conjunto para resolver problemas presentes no código, sendo possível registrar todas as etapas.

2.6.8 CAD

CAD - Desenho assistido por computador é o termo geralmente utilizado para designar toda a gama de software utilizado para a concepção de projetos de engenharia e/ou arquitetura etc, utilizando modelagem 2D e 3D (FILHO, 2010). .

Softwares CAD utilizam a modelagem por meio de vetores bidimensionais e tridimensionais, ou seja, todo elemento é desenhado no espaço tendo as coordenadas X, Y e Z. Isso implica em uma fácil visualização da modelagem no espaço virtual, além de ter uma

qualidade de processamento melhor quando comparado com desenhos do tipo *raster* (que são imagens geradas por *pixels*, comumente usando em modelagem BIM) (MOLDOVAN; MOLDOVAN, 2010).

2.6.9 AutoCAD

O modelo de *software* CAD mais difundido no mercado atualmente é distribuído pela empresa Autodesk, Inc. que teve sua primeira versão lançada em 1982 (AUTODESK, 2024f). Comumente empregado em projetos de engenharia, arquitetura, fabricação industrial, e diversas outras áreas. O AutoCAD trabalha com *prompts* de comando pelo teclado, que facilita a criação de entidades (linhas, círculos, polígonos e entre outros), além de diversas ferramentas que auxiliam o projetista, como a criação de blocos dinâmicos, tabelas, textos de referências (*fields*) e etc. Tudo isso somado, agiliza a concepção do projeto final (FILHO, 2010).

Além do vasto potencial disponibilizado pelo programa em si, o AutoCAD disponibiliza a API ObjectARX[®], nada mais é do que uma interface de programação de aplicativos (API) orientada a objetos em C++. Trata-se de uma DLL (*dynamic link library*), por ela faz-se possível chamar as funções diretas do AutoCAD. Portanto, essa API abre a arquitetura do AutoCAD, de forma que proporciona acesso direto às estruturas de bancos de dados do AutoCAD, ao sistema gráfico e à definição nativa de comandos (AUTODESK, 2024g). E a disponibilização da versatilidade de utilizar diversas linguagens de programação, como:

- AutoLISP e Visual Lisp (linguagem interna do programa)
- .NET Framework
- Active X[®] e COM

Para o desenvolvimento desse projeto, utilizou-se da API voltada para o .NET *Framework* utilizando a linguagem de programação C#.

2.7 AutoLISP e Visual LISP

AutoLISP é uma linguagem de programação designada para estender e customizar funcionalidades do AutoCAD (AUTODESK, 2024d). A abreviatura de LISP significa "*list processing*" (processamento de lista). Inicialmente a Autodesk escolheu a linguagem LISP como API de programação para a aplicação do AutoCAD. A linguagem LISP pode ser escrita utilizando editores de texto básicos, como por exemplo, o *Notepad* do *Windows* ou com extensões.

2.7.1 Blocos

Em programas CAD os blocos (*blocks*) são objetos que comumente representam símbolos, peças, detalhes e anotações. Um bloco nada mais é do que diversos objetos (linhas,

circulos, polilinhas) combinados em um único elemento (objeto ou entidade) (AUTODESK, 2024a). Um bloco garante produtividade e principalmente padronização nos projetos.

2.7.2 Blocos Dinâmicos

Um bloco dinâmico (*Dynamic Block*) é como um bloco comum, exceto pela possibilidade de criar regras, parâmetros e atributos que podem controlar sua aparência e comportamento quando inserido no ambiente CAD ou podendo até mesmo ser modificado posteriormente (AUTODESK, 2024b). As regras podem ser diversas, como por exemplo, regras de alinhamento, rotação, visibilidade, espelhamento, esticamento (*stretch*), multiplicação (*array*) entre muitas outras. Vale ressaltar que essas regras não podem ser aplicadas a um bloco comum, ou seja, um bloco comum não possui regras, porém um bloco comum pode ser transformado em um bloco dinâmico.

2.8 Teste de Usabilidade

O teste de usabilidade busca entender se o programa em si é usável pelos usuários e se consegue desenvolver tarefas propostas pelo programa. Por meio de combinações de métodos, que podem ser observações, entrevistas e questionário, é feita a coleta de dados e então avaliado a usabilidade do programa e o que precisa ser melhorado ou eliminado do programa (BARRETO et al., 2018).

O teste de usabilidade envolve quatro etapas, sendo elas:

Preparação para o teste de usabilidade: nesta etapa será feito todo o planejamento de condução do teste de usabilidade para que o mesmo seja bem sucedido.

Introdução ao teste de usabilidade: o participante (usuário) será situado sobre o que é necessário fazer e como vai ser conduzido todo o processo de teste, assim o participante terá uma boa noção e conforto para conduzir os testes.

Condução do teste de usabilidade: o avaliador deve acompanhar o participante para auxiliá-lo, caso necessário, e para conduzir o teste.

E por fim, o encerramento do teste: aqui então o participante preenche e coloca suas sugestões em um formulário, e então o teste pode ser reorganizado para uma nova seção, reiniciando o teste.

O teste, nesse caso, buscou aplicar tarefas representativas do dia a dia dos participantes e de utilização do programa, cobrindo todas as funcionalidades. Buscou-se também a aplicação em usuários típicos, ou seja, usuários que fazem parte do contexto de aplicação do programa (BARRETO et al., 2018).

Faz-se importante ressaltar também que, o teste de usabilidade não buscou a coleta de dados com o objetivo de auxiliar no processo de *design* do programa final, mas sim, avaliar a qualidade geral do produto final já pronto; isso se decorreu pela opção de desenvolver o programa por um método de inspeção heurística, ou seja, o programa foi

desenvolvido por princípios definidos de usabilidade.

2.8.1 Heurística de Nielsen

A avaliação heurística é um método de inspeção, desenvolvido por Nielsen, que busca desenvolver o *software* baseado em princípios de usabilidade (BARRETO et al., 2018).

Como dito na seção anterior, o programa foi desenvolvido partindo-se dos princípios de usabilidade, previamente definidos, sendo eles listados abaixo:

- Visibilidade do *status* do sistema: os usuários são informados do que está sendo executado pelo sistema. Então, buscou-se desenvolver mensagens de aviso quando uma tarefa foi concluída ou se houve algum erro durante o processo.
- Correspondência entre programa e mundo real: deve-se utilizar terminologias utilizadas pelos usuários.
- Controle do usuário e liberdade: o programa busca disponibilizar total controle ao usuário, mesmo em casos onde uma função possa ser escolhida por engano, oferecendo meios de saída do mesmo.
- Consistência e padrões: o sistema busca ser padronizado.
- Prevenção, recuperação e diagnóstico de erros: toda a aplicação foi provida de proteções contra erros que possam ser a vir ser geradas pelo usuário ou pelo programa, evitando erros inesperados e advertindo o usuário sobre.
- Flexibilidade, eficiência de uso e *design* minimalista: o sistema busca ser o mais simples e direto possível, de forma que mesmo usuários novos não tenham grandes dificuldades e que usuários recorrentes não precisam ficar relembrando a utilização do programa. Além, de possibilitar a personalização de alguns itens para agilizar processos, como por exemplo, pré-salvamento do formulário de inserção, que será aprofundado mais a frente em como se deu o desenvolvimento do *software*.
- Ajuda e documentação: Foi gerado documentação de utilização do programa, além de mensagens internas no programa que auxiliam na utilização do programa, mesmo para usuários sem conhecimento prévio do programa.

2.9 Trabalhos Relacionados

2.9.1 PRO-Elétrica

Conforme apresentação da empresa (MULTIPLUS, 2024): "O PRO-Elétrica é um *software* para projetos elétricos que detalha e dimensiona as instalações elétricas em baixa tensão, SPDA, instalação de placas fotovoltaicas On-Grid, cabeamento estruturado, loteamento com locação de postes e iluminação e automação residencial."

Embora o programa seja um *software*, é importante se atentar que é um *software* complementar, ou seja, ele complementa outro *software*, no caso faz-se necessário ter um

outro *software* tipo CAD compatível com o mesmo, por tanto, conclui-se que seja também um *plugin*, assim como o mesmo desenvolvido nesse trabalho.

Todavia, o programa desse trabalho busca uma especialidade ao focar nos padrões representativos de uma determinada empresa e não busca desenvolver as mesma ferramentas proposta pelo *plugin* do PRO-Elétrica, tratando-se de um programa completo e complexo voltado para a área.

2.9.2 *Plugin* de AutoCAD para automatização de etapas do desenvolvimento de projetos de instalações elétricas

O *plugin* em questão (D'AMARAL, 2021) trata-se de um trabalho de conclusão de curso do autor Matheus D'Amaral Gomes, feito também no Instituto Federal de Santa Catarina do Campus Itajaí. O programa também faz uma geração automatizada de esquemas unifilares. Os objetivos específicos do trabalho incluem a criação de funcionalidades que gerem automaticamente quadros de cargas, diagramas unifilares e listas de materiais, além de sugerir a quantidade mínima de tomadas e a carga destinada a luminárias em cada ambiente. A validação da funcionalidade do *plugin* foi realizada por meio de testes de usabilidade, garantindo que as novas ferramentas sejam práticas e eficientes para os usuários.

A metodologia utilizada envolveu uma pesquisa qualitativa de natureza aplicada, com objetivo exploratório-descritivo e método hipotético-dedutivo. Foram adotados procedimentos técnicos como pesquisas bibliográficas, documentais e de levantamento de dados, para proporcionar uma base teórica robusta e reunir informações essenciais para o desenvolvimento e validação do *plugin*.

As etapas de desenvolvimento e os resultados obtidos foram divididos entre a análise das funcionalidades implementadas e a análise dos dados gerados pelos testes, realizados com quatro profissionais experientes na área de projetos elétricos. Esses resultados e análises são apresentados no final do estudo, oferecendo uma visão detalhada sobre a eficácia do *plugin*.

Em conclusão, o estudo confirma a hipótese de que é viável desenvolver um *plugin* para AutoCAD que automatiza etapas na criação de projetos de instalações elétricas. Os resultados dos testes e a análise mostram que o *feedback* obtido foi positivo, indicando que o *plugin* melhora a eficiência e segurança dos projetos elétricos prediais.

Embora o programa desenvolvido nesse trabalho e o trabalho em questão sejam semelhantes, deve-se levar em consideração as observações levantadas no tópico anterior 2.9.1, além de algumas funcionalidades e padrões distintos que serão apresentados ao decorrer desse trabalho.

2.9.3 Análise dos indicadores de desempenho da máquina *Reach Stacker* após a implementação de *software* testador de *joystick* no setor de manutenção portuária

O *software* do trabalho de conclusão de curso da autora Isabelli Sasdelli Tavares (TAVARES, 2022), feito no Instituto Federal de Santa Catarina do Campus Itajaí. Desenvolveu-se um *software* para testar o *joystick* para máquina *Reach Stacker*. Um dos equipamentos mais utilizados em ambientes portuários é a empilhadeira tipo *reach stacker*, responsável pela movimentação de contêineres por curtas distâncias. Ela é composta por principais conjuntos, como a lança, o *spreader* e a cabine do operador. Dentro da cabine do operador, encontra-se o componente eletrônico *joystick*, que realiza todas as funções de movimentação da lança (esticar, recolher, subir e descer) e do *spreader* (abrir, fechar, deslocar, girar, inclinar e apalpar), além de funções de comunicação via rádio e buzina. Quando o *joystick* apresenta defeito, é necessário seguir uma série de passos para encontrar a causa raiz, desde a retirada do equipamento e instalação em outro para confirmação do problema até o envio para reparo externo. Esse procedimento é demorado e oneroso, tornando relevante o desenvolvimento de um testador de *joystick* para detectar rapidamente as falhas e permitir o conserto do componente de forma assertiva.

Este trabalho visou aprimorar o *software* e *hardware* inicialmente desenvolvidos para o testador de *joystick*, analisar o impacto da implementação no setor de manutenção através dos principais indicadores de desempenho (confiabilidade, disponibilidade, custo, MTTR, MTBF) e avaliar o uso do testador com os usuários finais por meio de um teste de usabilidade. A autora utilizou um método de teste de usabilidade que possui seis etapas: desenvolvimento do plano de teste, seleção e recrutamento de participantes, preparação do material do teste, condução do teste em si, conversa com o participante após o teste e análise dos resultados.

Em relação aos indicadores de desempenho, observou-se uma melhora significativa na confiabilidade, custo, MTTR e MTBF. A disponibilidade aumentou em relação às falhas de *joystick*, mas diminuiu de forma geral devido a outras falhas. Os testes de usabilidade com os usuários finais foram realizados através de um formulário digital, onde o *software* e funções como a geração de relatórios em PDF foram avaliados. Os resultados apresentados foram excelentes, com *feedbacks* interessantes para o desenvolvimento de trabalhos futuros.

3 DESENVOLVIMENTO DO *PLUGIN*

Este capítulo tem por objetivo relatar o desenvolvimento do *software*. O presente trabalho dá continuidade ao desenvolvimento do *plugin*, iniciado na disciplina de Projeto Integrador III do curso de Engenharia Elétrica do Instituto Federal de Santa Catarina (IFSC) Câmpus Itajaí. Portanto, a etapa final de desenvolvimento do *plugin* buscou desenvolver novas funcionalidades e aprimorar funcionalidades inicialmente desenvolvidas. Com o objetivo de melhor instruir o leitor, será apresentado tudo que foi desenvolvido inicialmente e às novas funcionalidades.

Assim, tem-se no quadro 1 as etapas resumidas do desenvolvimento do *software*, listados por funcionalidade.

Quadro 1 – Etapas do desenvolvimento do programa.

Versão do plugin	Etapa do desenvolvimento
Projeto integrador (Versão inicial)	Criação da tabela padrão do quadro de cargas
	Programação do plugin para o CAD
Versão final (TCC)	Geração automatizada da lista de material
	Exportação para planilhas excel e ambiente CAD da lista de material
	Criação de arquivo de salvamento para a interface
	Criação de ribbon tab do programa
	Melhorias de design do arquivo de quadro de cargas
	Melhorias e alterações de design da interface do ambiente CAD
	Importação de blocos dinâmicos automatizados para esquema de montagem
	Importação do quadro de cargas para o ambiente CAD
	Refatoração do código e outros

Fonte: autor.

O *plugin* em questão foi desenvolvido utilizando-se das APIs disponibilizada pela Autodesk, utilizando como o CAD AutoCAD, e a linguagem de programação C# com a *framework* .NET 4.7.2.

3.1 Versão inicial - Projeto Integrador

3.1.1 Criação da tabela padrão do quadro de cargas

Inicialmente, visto que programa como um todo usa como base informações vindas de uma planilha, teve-se que iniciar o desenvolvimento partindo da mesma. Logo, a partir de uma planilha do Excel criou-se um padrão de quadro de cargas, tendo por objetivo também criar como uma planilha de memorial de cálculos, ou seja, todos os cálculos necessários estão concentrados dentro da planilha; para que então o programa pudesse ser construído em torno desse padrão.

Buscou-se criar uma planilha com o máximo de informação possível, que pudesse a vir ser utilizadas ou não, e deixando cada tópico segregado para facilitar no processamento posterior dos dados em código. Cada item foi separado por coluna, sendo eles: nome do quadro, circuito, descrição (local), potência [W], fator de potência, potência [VA], fase - A, fase - B, fase - C, i [A] (corrente), quantidade de cabos, dividir cabos, reduzir terra, seção condutor [mm²], tipo de isolamento, tipo disjuntor, curva disjuntor, quantidade de pólos disjuntor, corrente do disjuntor, icu [kA], numeração do I/DR, quantidade de pólos I/DR, corrente I/DR, corrente de fuga do I/DR, DPS, proteção DPS, entrada de energia.

Após padronizado o modelo, procurou-se fazer a automação da planilha para leitura e classificação dos dados, para isso utilizou-se a API do *Microsoft Office (Microsoft.Office.Interop.Excel)* (MICROSOFT, 2024a); então o arquivo .xlsx ou .xlsm (extensão para arquivo do tipo Excel) é acessado e cada linha e coluna é acessada, correspondendo a cada célula, o valor é lido e armazenado dentro de uma estrutura (*struct*) do código, posteriormente ao processar as demais informações essa estrutura é usada como referências para as demais automações no código (MICROSOFT, 2024b).

3.1.2 Programação do *plugin* para o CAD

Como abordado no tópico 2.6, de conceitos computacionais, o AutoCAD disponibiliza uma API para acessar suas funcionalidades, esse projeto utiliza dessa funcionalidade em conjunto do *Visual Studio* e utilizando a linguagem de programação C#. Optou-se utilizar esse ambiente e esta linguagem por conveniência de ter um conhecimento prévio do autor, além de proporcionar múltiplas ferramentas na plataforma que auxiliaram na concepção final do trabalho, como por exemplo, na utilização do *Windows Forms*; e também por serem ferramentas já consolidadas no mercado e com uma boa base de referências *online*.

Portanto, nesta seção será apresentada a parte do *plugin* desenvolvido durante a disciplina de projeto integrador III, posteriormente as melhorias e novas funcionalidades implementadas durante a fase do TCC.

Inicialmente o programa precisava de duas informações básicas para iniciar o processo de automação em código, sendo elas, o caminho do arquivo e o ponto de inserção na tela (embora o ponto de inserção pudesse ser feito com uma técnica chamada *jigging*,

onde o objeto fica flutuando até escolher efetivamente o ponto, a princípio não foi escolhida essa técnica). Portanto criou-se um formulário (*forms*) com a entrada dos dados necessários para iniciar a automação, conforme a figura 8, mostra o respectivo *layout* que seria a apresentação final na disciplina de projeto integrador.

Figura 8 – Interface do *plugin* - Implementação inicial projeto integrador.



Fonte: autor.

A janela ou formulário apresentado na figura 8 é apresentada dentro do ambiente CAD. Para contextualização do leitor uma explicação breve sobre a inicialização do *plugin* dentro do ambiente CAD, para executar o *plugin* faz-se necessário que primeiro o usuário inicie o ambiente CAD e então por meio do comando "*netload*", digitado pelo *prompt* de comando do ambiente CAD, uma tela do *file explorer* (explorador de arquivos do *Windows*) é aberta e então o usuário carrega um aplicativo .NET (AUTODESK, 2022) pela extensão .dll, essa aplicação é onde está contido o programa, e então o programa será carregado dentro do ambiente CAD e pronto para ser utilizado.

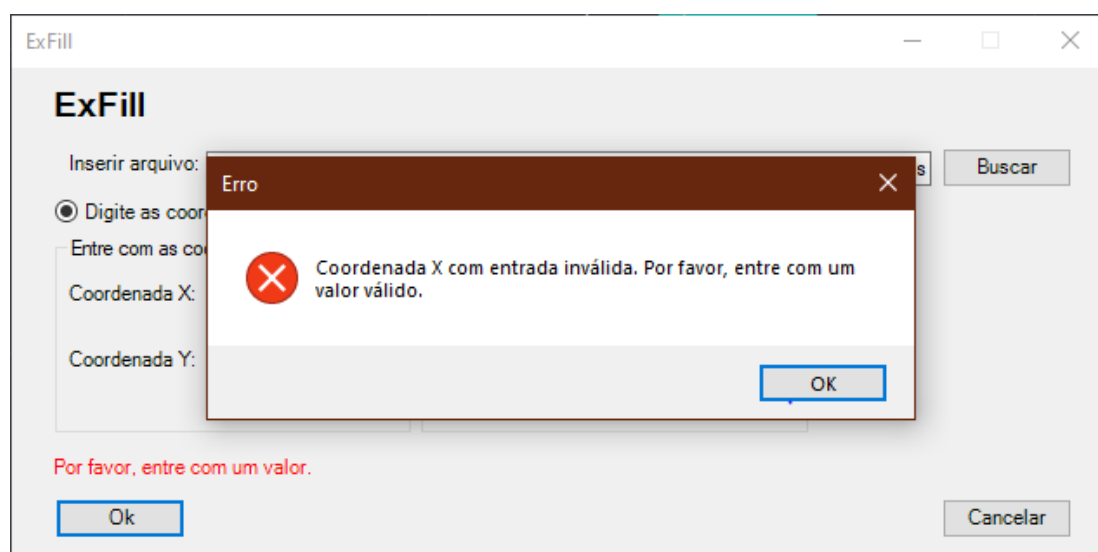
Conforme observa-se na figura 8 procura-se seguir a heurística de Nielsen abordada na seção 2.8.1, a interface busca ser limpa, direta e com mensagens de aviso que informam o usuário sobre o que está acontecendo no ambiente. O usuário pode escolher entre digitar as coordenadas ou inserir por ponto na tela, escolhendo diretamente na tela do ambiente de modelagem CAD onde será inserido o desenho.

Dentro da interface há sistemas que impedem que sejam cometidos erros por parte do usuário, por exemplo, se o mesmo tentar inserir o desenho sem pontos de coordenadas, o programa não irá inserir o desenho e indicará uma mensagem de aviso de que as coordenadas não foram corretamente inseridas, o mesmo se tentar inserir entradas inválidas, como caracteres não numéricos. A figura 9 demonstra um erro decorrido de uma entrada inválida como demonstração

O programa também conta com filtro de extensão de arquivos ao inserir o arquivo de referência dentro da aplicação, para que o usuário não selecione arquivos incompatíveis

com o requisitado e não gere erros dentro da aplicação, portanto o usuário apenas pode carregar arquivos .xlsx ou .xslm (extensão de planilhas do Excel). Caso venha a ter algum dado inserido equivocadamente dentro da planilha ou venha ser gerado algum erro interno, o programa irá inserir uma mensagem de aviso em vermelho na própria janela de interface, no canto inferior esquerdo, indicando que não foi possível inserir o desenho; caso a operação seja bem sucedida será exibido um mensagem em verde indicado que o desenho foi inserido com sucesso. Há também mensagens internas dentro do ambiente CAD inseridas no *prompt* de comando, caso venha a ter algum erro do CAD.

Figura 9 – Mensagem de erro de entrada inválida - Demonstração.

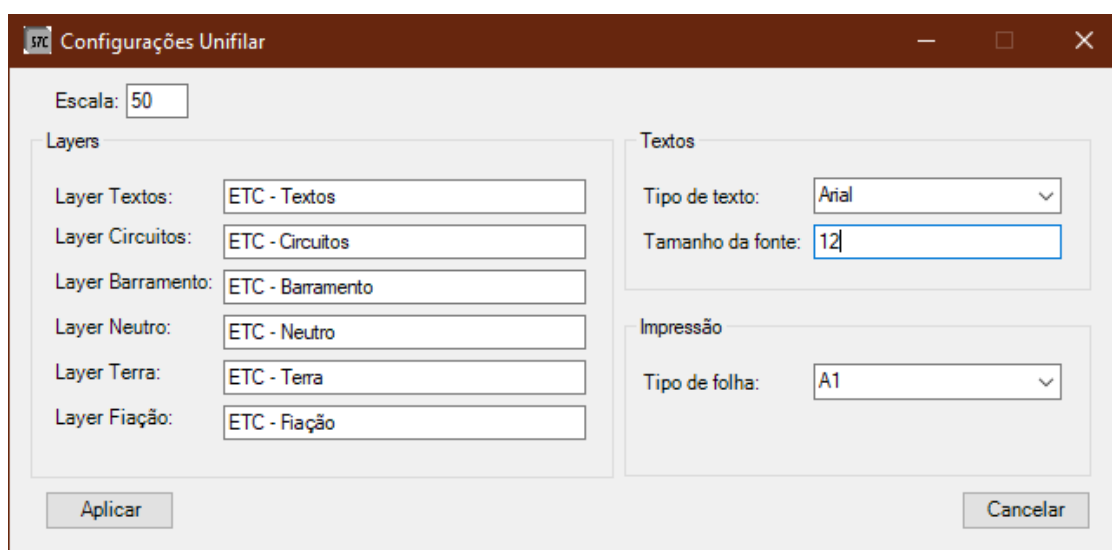


Fonte: autor.

A delimitação de grupo para configurações gerais, vista também na figura 8, foi implementada para complementar as informações necessárias na geração do esquema unifilar, composta portanto pela, tensão fase/neutro, indicações de onde vem a alimentação do quadro e o tamanho do eletroduto na alimentação. Essas informações foram complementadas no formulário, pois inicialmente não foram previstas implementá-las dentro do quadro de cargas. Vale ressaltar que, essas informações não são tratadas como limitação na geração do esquema unifilar, pois são complementares.

A versão inicial também conta com uma janela (formulário) para personalizar alguns característica do desenho (configurações), como por exemplo a alteração do nome das *layers* (são camadas/método para organizar os objetos por função ou propósito, sendo possível alterar a cor, tipo de linha, espessura da linha, etc, além de funcionalidades para filtragem e visibilidade (AUTODESK, 2024c)); também permitindo alterar o tipo de fonte, tamanho da letra e escala global do arquivo. A figura 10, demonstra a janela de configurações.

Figura 10 – Janela de configurações - Implementação inicial projeto integrador.

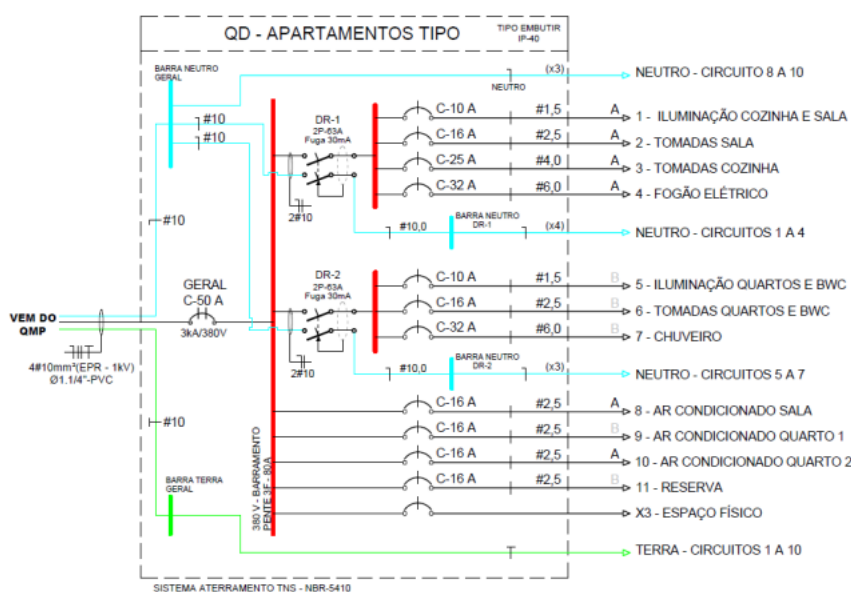


Fonte: autor.

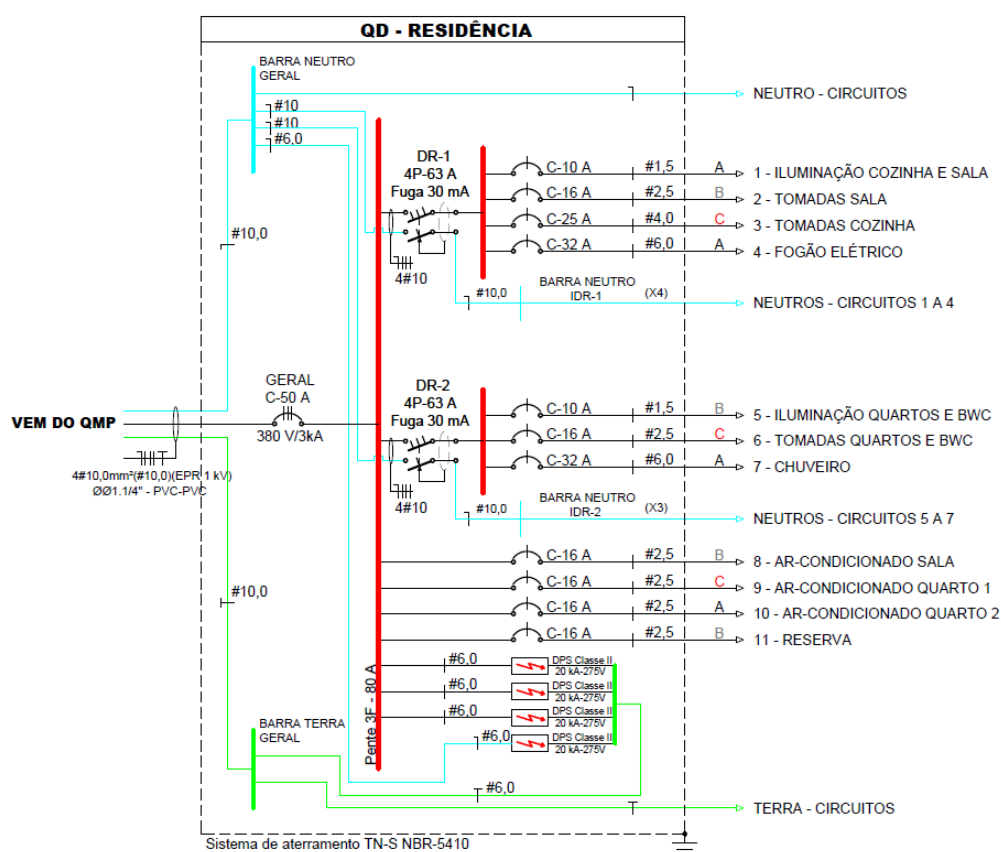
3.1.2.1 Demonstração do *plugin*

A figura 11, mostra o esquema unifilar referência (desenhado manualmente) utilizado para programar dentro do *plugin*, portanto, o *plugin* deve gerar um esquema semelhante ao desenho referência. E a figura 12, mostra a versão gerada pelo *plugin* dentro do ambiente CAD, ou seja, a aplicação gera o desenho de forma automatizada utilizando como referência o quadro de cargas, conforme pode-se observar, são idênticas.

Figura 11 – Desenho referência para o esquema unifilar (desenhado manualmente).



Fonte: autor.

Figura 12 – Desenho gerado pelo *plugin* (desenho gerado automaticamente).

Fonte: autor.

3.2 Versão final - TCC

3.2.1 Interface - Alteração do *layout*

Nesta seção serão apresentadas as alterações aplicadas ao *plugin* na versão final desenvolvida durante o Trabalho de Conclusão de Curso (TCC), diferenciando-se da versão inicial desenvolvida durante o a disciplina de Projeto Integrador III.

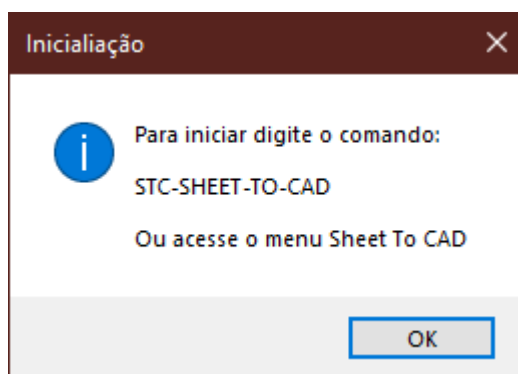
3.2.1.1 Inicialização

Houve alterações significativas no *layout*, agora ao iniciar o programa o usuário é apresentado com um mensagem de inicialização, indicando como pode ser iniciado o programa pelo *prompt* de comando e pela nova interface utilizando o *ribbon tab* do AutoCAD. A *ribbon tab* organiza as ferramentas em grupos de contexto, que permite que o usuário execute determinada ferramenta rapidamente por meio de uma interface de botões (AUTODESK, 2023), funcionando como um menu de guias/tabelas.

A figura 13, demonstra a mensagem de inicialização do programa, novamente aqui buscou-se seguir princípios da heurística de Nielsen abordada na seção 2.8.1, ao indicar para usuários sem conhecimento prévio como o *plugin* pode ser inicializado. Vale observar que na versão inicial era apenas possível inicializar o *plugin* por meio de *prompt*

de comando.

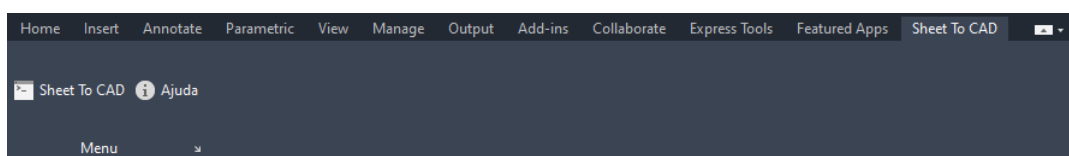
Figura 13 – Mensagem de inicialização do *plugin*.



Fonte: autor.

A figura 14, demonstra o *layout* adotado para a implementação do *plugin* em conjunto com a *ribbon tab* do AutoCAD. Como pode-se observar, há apenas duas opções disponível, uma para iniciar a janela do *plugin* e outra para acessar o arquivo de ajuda.

Figura 14 – *Ribbon tab* do *plugin*.



Fonte: autor.

3.2.1.2 Menu geral

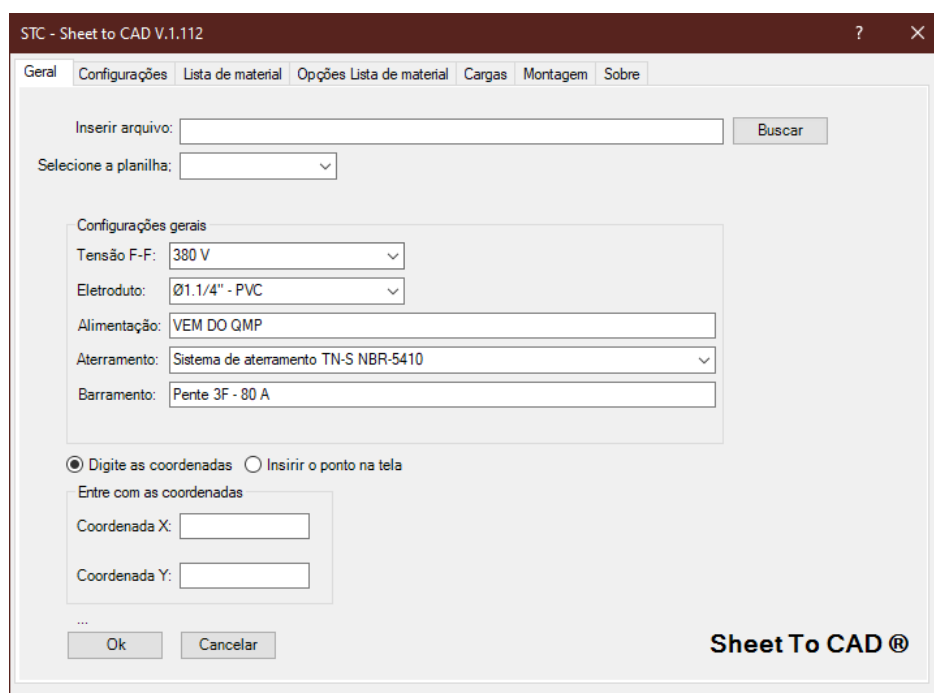
O menu geral, anteriormente visto na seção 3.1.2, sofreu mudanças significativas, atualmente a janela do *plugin* consta com uma guia de abas, assim todas as aplicações disponibilizadas pelo programa ficam concentradas em um único formulário, evitando a utilização de múltiplos *prompts* de comando dentro da aplicação. Foram criadas então as guias gerais, de configurações, geração de lista de material, configurações da lista de material, quadro de cargas, esquema de montagem e a guia sobre. Todas essas guias serão abordadas no decorrer das próximas seções. A figura 15, mostra o *layout* atual na aba de menu geral.

Dentro do menu geral alguns elementos foram realocados e novas funcionalidades adicionadas. Foi adicionada a possibilidade do usuário trabalhar com múltiplas planilhas dentro de um mesmo arquivo, então o mesmo pode selecionar em qual planilha está trabalhando pela opção de "Selecione a planilha". E o submenu de configurações gerais, foi complementado com opções de informações do tipo, de aterramento utilizado e o barramento geral do quadro.

O programa passou por uma reformulação de nome também, inicialmente com a nomeação de *ExFill* (em alusão ao nome Excel e *Fill* (preenchimento)), posteriormente ETC (*Excel To CAD*), e na última nomeação STC (*Sheet to CAD*) referenciando que o

mesmo trabalha com dados de uma planilha (*Sheet*) e tem interoperabilidade (*To*) para o CAD.

Figura 15 – Interface do *plugin* - Implementação final menu geral.



Fonte: autor.

3.2.1.3 Configurações

Na guia de configurações foram reformuladas também algumas configurações, sendo as principais, opção para salvar as configurações do formulário e configuração para tipo da representação de fase (esquema A-B-C ou R-S-T). A figura 16, demonstra o *layout* atual do menu de configurações.

A maneira escolhida para salvar as informações foi por meio de arquivo JSON (*JavaScript Object Notation* - Notação de Objetos JavaScript). JSON é um formato de documento aberto que emprega texto estruturado de fácil compreensão para representar objetos de dados (LEE; ANJOS; SATTI, 2020). Uma das vantagens em utilizar JSON é a sua fácil implementação seja pela linguagem, podendo ser empregado por qualquer linguagem de programação, e pela fácil interpretação humana.

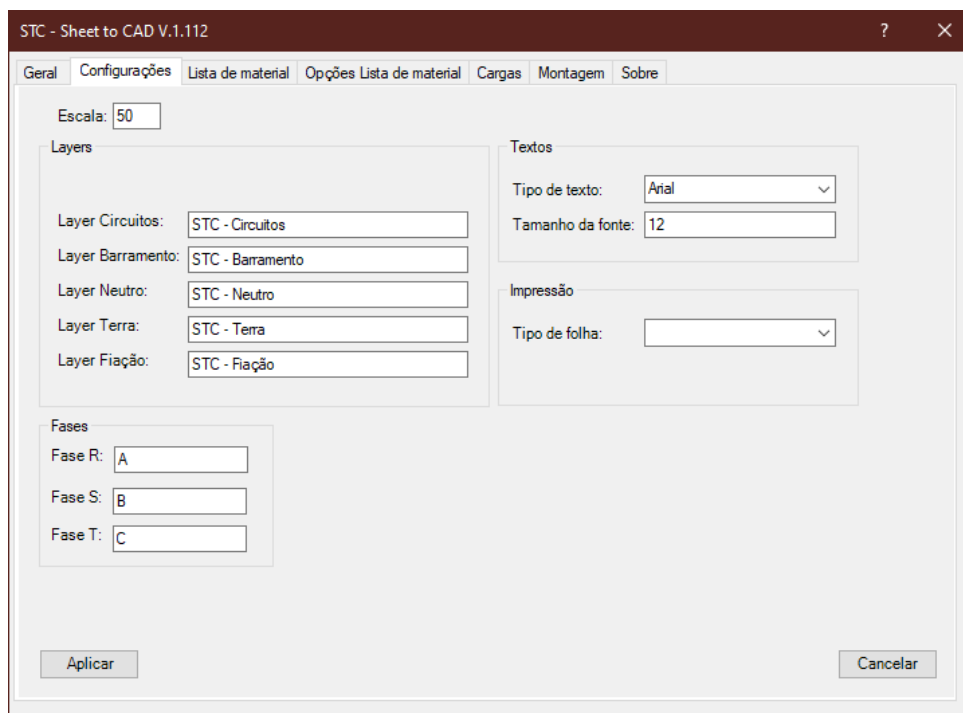
3.2.1.4 Lista de material

Este menu trata-se de uma nova funcionalidade implementada, geração de lista de material, atua como lista complementar a montagem do quadro, indicando as quantidades de disjuntores, terminais, demais dispositivos como o DR (diferencial-residual) e DPS (Dispositivo de proteção contra surtos) e também indica o tamanho do quadro mínimo baseando-se na quantidade de dispositivos no tamanho DIN (padronização da norma europeia para tamanho dos componentes) efetivos e o espaço reserva determinado pela

tabela 59 da norma NBR-5410 (ABNT, 2004). A figura 17, demonstra o *layout* com um exemplo aplicado.

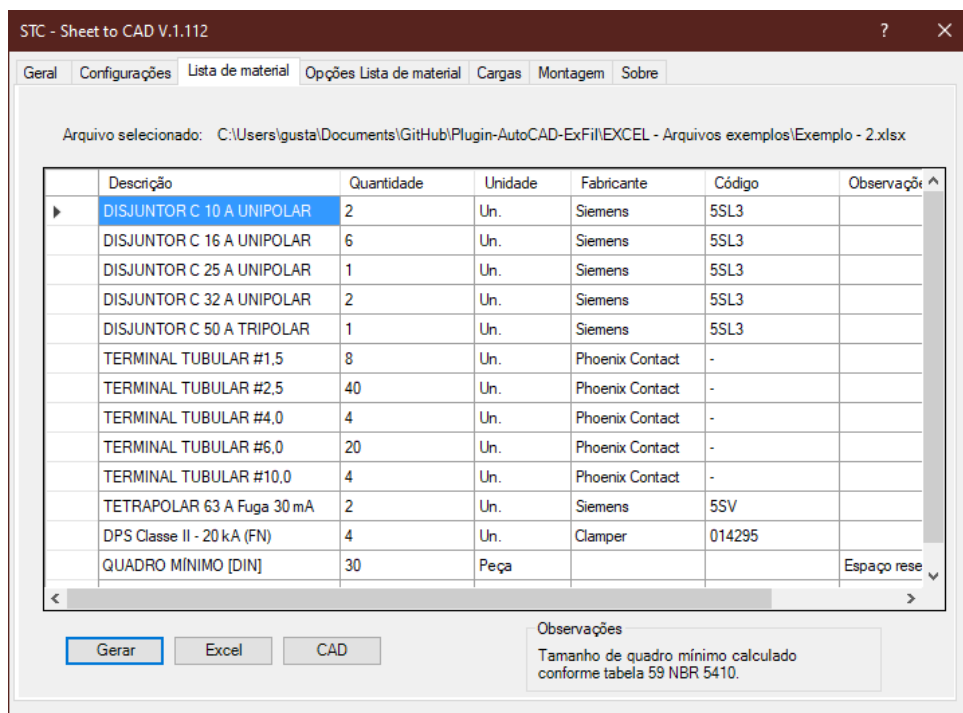
O programa também disponibiliza opções para exportar a lista para arquivos .xlsx e para dentro do ambiente CAD em forma de tabela.

Figura 16 – Interface do *plugin* - Implementação final menu de configurações.



Fonte: autor.

Figura 17 – Interface do *plugin* - Implementação lista de materiais.



Fonte: autor.

3.2.1.5 Opções da lista de material

Nesta guia é possível configurar algumas opções da geração da lista de material sendo elas, o fabricante e o código do fabricante, dessa forma a lista já é gerada conforme as necessidades do usuário. Trata-se de uma funcionalidade que nasceu em conjunto com a geração da lista de material. As informações presentes nesse menu podem ser salvas, juntamente com as demais informações, ou restauradas para configurações de fábrica. A figura 18 mostra o *layout* implementado.

Figura 18 – Interface do *plugin* - Implementação opções da lista de materiais.

STC - Sheet to CAD V.1.112

Gerar Configurações Lista de material Opções Lista de material Cargas Montagem Sobre

Fabricante

Disjuntor: Siemens

Disjuntor Residual: Siemens

Terminal: Phoenix Contact

DPS: Clamper

Código

Disjuntor: 5SL3

Disjuntor Residual: 5SV

Terminal: -

DPS: 014295

Salvar Restaurar

Fonte: autor.

3.2.1.6 Cargas

Com o intuito de simplificar o quadro de cargas, visto que o arquivo Excel contendo o memorial de cálculo é extenso, criou-se um menu para importação do quadro de cargas da planilha para dentro do CAD, então é gerado um quadro de cargas simplificado para que o mesmo possa a vir ser inserido dentro do ambiente CAD, constando como informação do projeto final. Portanto, trata-se de uma aplicação nova criada. A figura 19, demonstra o *layout* adotado.

3.2.1.7 Esquema de montagem

Um esquema de montagem é uma visualização da montagem física do quadro de distribuição, mostrando o posicionamento de todos os dispositivos de proteção, espaços

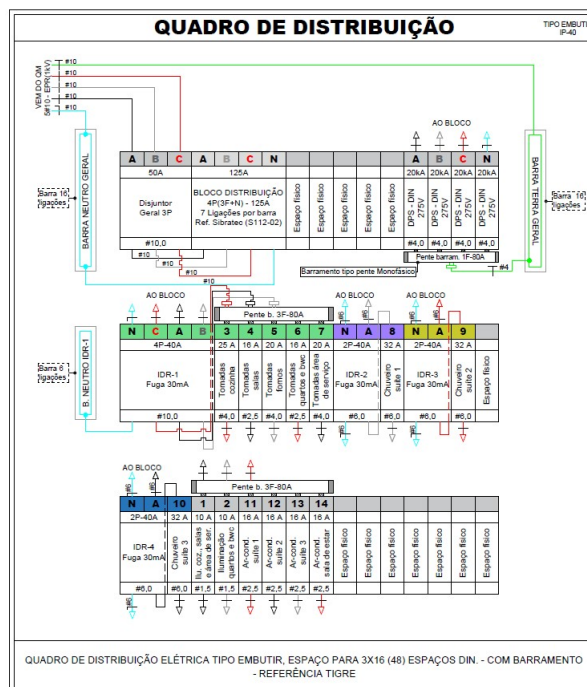
disponíveis no quadro, condutores, conexões, barramentos e tamanho do quadro. A figura 20 demonstra um modelo de esquema de montagem.

Figura 19 – Interface do *plugin* - Implementação quadro de cargas.

Circuito	Descrição	Potência	Fase A	Fase B	Fase C	Cor
1	ILUMINAÇÃO C...	200	200			#1.5
2	TOMADAS SALA	500		543		#2.5
3	TOMADAS COZI...	1500			1630	#4.0
4	FOGÃO ELÉTRI...	6500	6500			#6.0
5	ILUMINAÇÃO Q...	200		217		#1.5
6	TOMADAS QUA...	1100			1196	#2.5
7	CHUVEIRO	6600	7174			#6.0
8	AR-CONDICION...	1000		1087		#2.5
9	AR-CONDICION...	1000			1087	#2.5
10	AR-CONDICION...	1000	1000			#2.5
11	RESERVA	100		109		#2.5
(x3)	ESPAÇOS FISIC...	0				
*						

Fonte: autor.

Figura 20 – Modelo de esquema de montagem.

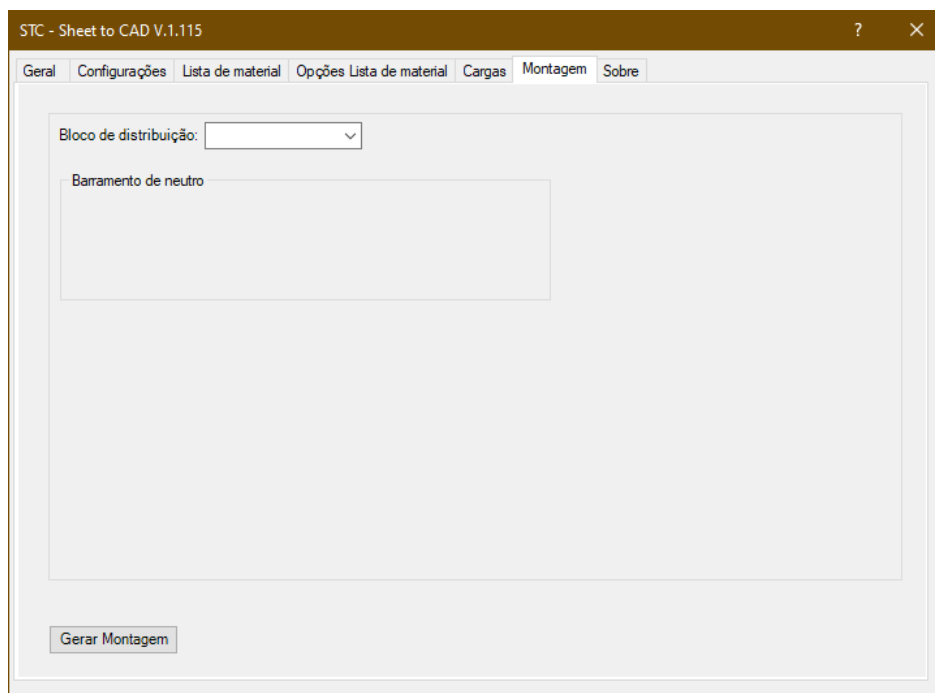


Fonte: autor.

Para agilizar o projeto de esquemas de montagem, foi criada uma funcionalidade que carrega automaticamente os blocos dos esquemas com as informações de cada elemento, conforme o arquivo lido pelo programa. Essa ferramenta, assim como a principal, visa

aumentar a velocidade do projeto e reduzir erros humanos ao pré-carregar os blocos com as informações dos circuitos. A figura 21 demonstra a guia de montagem da interface do *plugin*, algumas funcionalidades nessa guia ainda não foram completamente implementadas durante a documentação desse trabalho.

Figura 21 – Interface do *plugin* - Guia montagem.



Fonte: autor.

3.2.1.8 Sobre

O menu sobre consta informações sobre os desenvolvedores, meios de contato, versão do programa e acesso ao arquivo de ajuda. Conforme demonstrado na figura 22.

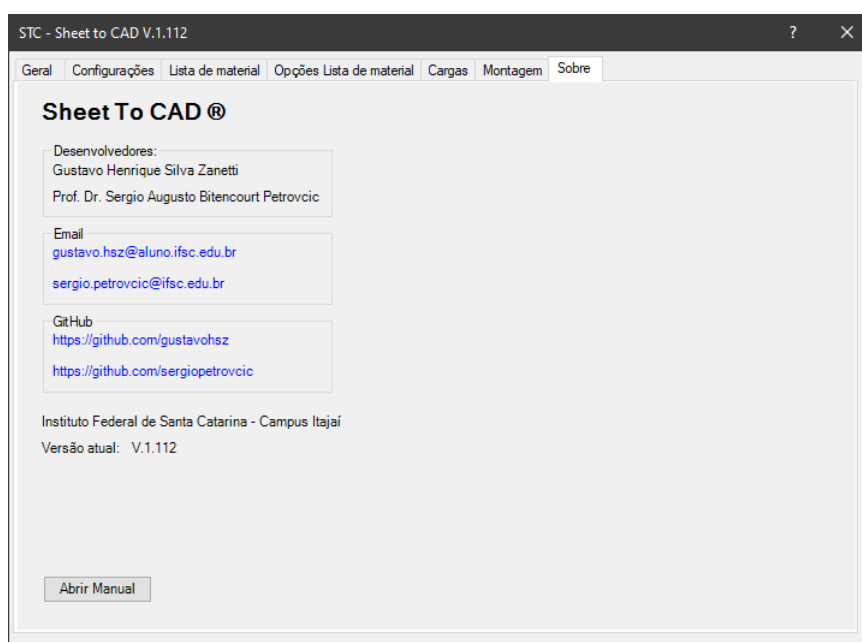
3.2.1.9 Guia de ajuda - Manual de instruções

O guia de ajuda ou manual de instruções é acessado pela aba "Sobre" na aplicação do formulário do *plugin* ou também pelo *ribbon tab* do CAD, no botão "Ajuda", visto na figura 14.

O manual busca orientar o usuário seja durante os testes ou em uma rotina cotidiana de uso, sanando dúvidas sobre a utilização do *plugin* e do quadro de cargas.

Nele constam informações de funcionalidades do *plugin*, mostrando a operacionalização das funcionalidades, com imagens detalhando os passos de utilização. Além de demonstrar o que não deve ser feito para evitar erros na execução, apesar de boa parte dos erros já serem tratados internamente na aplicação.

O manual está disponibilizado em arquivo .pdf dentro da pasta do aplicativo disponibilizado para o usuário. O mesmo foi adicionado ao apêndice A.

Figura 22 – Interface do *plugin* - Implementação menu sobre.

Fonte: autor.

3.2.1.10 Quadro de cargas

O quadro de cargas sofreu alterações, com relação à primeira versão, com a utilização formatações condicionais em busca de auxiliar o projetista. Foi criado um sistema de *highlights* (destaque) que destaca a célula selecionado, ou seja, conforme o projetista ou usuário muda de célula e o Excel automaticamente destaca a linha (em amarelo) e a coluna (em cinza claro), facilitando a identificação das mesmas, conforme demonstra a figura 23.

Utilizou-se uma formatação condicional que também indica se o circuito está em sobrecarga (destaque em vermelho). Conforme o item de coordenação entre condutores e dispositivo de proteção da NBR 5410 destaca, a corrente de projeto do circuito deve ser menor ou igual à corrente nominal do dispositivo de proteção (disjuntor, neste caso). Por exemplo, caso a corrente do circuito seja maior que a corrente do disjuntor (proteção), será indicado no quadro em vermelho que o disjuntor sofrerá sobrecarga.

O quadro também conta com listas suspensas para alguns valores padronizados como, seção do condutor, isolamento do condutor, corrente de disjuntor e etc. E caso o usuário venha a preencher um campo que não esteja no padrão do mesmo, a planilha emite uma mensagem de aviso para alertar o usuário sobre o campo não padronizado, conforme demonstra a figura 24.

3.2.2 Demonstração de projeto

A figura 25 demonstra a versão final do esquema automatizado gerado pelo *plugin*, bem como com as demais automações geradas pelas novas funcionalidades como, lista de material, quadro de cargas e os blocos para o esquema de montagem.

Figura 23 – Quadro de cargas utilizado em conjunto com o *plugin*.

A	B	G	H	I	J	K	L	M	N	O	P	Q	R
NOME DO QUADRO:	QD												
CIRCUITO	DESCRIÇÃO (LOCAL)	FASE - B	FASE - C	I [A]	QUANTIDADE DE CABOS	DIVIDIR CABOS?	REDUZIR TERRA?	SEÇÃO CONDUTOR	TIPO ISOLAÇÃO	TIPO DISJUNTOR	CURVA DISJUNTOR	QUANTIDADE PÓLOS DISJUNTOR	CORRENTE DISJUNTOR
1	ILUMINAÇÃO COZINHA E SALA			14,82	F+N+T	NÃO	NÃO	#1,5	PVC 750 V	DIN	C	UNIPOLAR	10 A
2	TOMADAS SALA	543		2,47	F+N+T	NÃO	NÃO	#0,5 #1,0	PVC 750 V	DIN	C	UNIPOLAR	16 A
3	TOMADAS COZINHA		1630	7,41	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	UNIPOLAR	25 A
4	FOGÃO ELÉTRICO			30,00	F+N+T	NÃO	NÃO	#1,5 #2,5	PVC 750 V	DIN	C	UNIPOLAR	32 A
5	ILUMINAÇÃO QUARTOS E BWC	217		0,99	F+N+T	NÃO	NÃO	#4,0 #6,0	PVC 750 V	DIN	C	UNIPOLAR	10 A
6	TOMADAS QUARTOS E BWC		1196	5,43	F+N+T	NÃO	NÃO	#10,0	PVC 750 V	DIN	C	UNIPOLAR	16 A
7	CHUVEIRO			30,00	F+N+T	NÃO	NÃO	#16,0 #25,0	PVC 750 V	DIN	C	UNIPOLAR	32 A
8	AR-CONDICIONADO SALA	1087		4,94	F+N+T	NÃO	NÃO	#35,0 #50,0	PVC 750 V	DIN	C	UNIPOLAR	16 A
9	AR-CONDICIONADO QUARTO 1		1087	4,94	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	UNIPOLAR	16 A
10	AR-CONDICIONADO QUARTO 2			4,94	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	UNIPOLAR	16 A
11	RESERVA	109		0,49	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	UNIPOLAR	16 A
(x3)	ESPAÇOS FÍSICOS			0,00									
DPS	DPS Classe II - 20 kA (FN)				3F+N+T	NÃO	NÃO	#6,0	PVC 750 V	DIN	C	TRIPOLAR	
PROTEÇÃO DPS													
ENTRADA DE ENERGIA	GERAL	1957	3913	35,58	3F+N+T	NÃO	NÃO	#10,0	EPR 1 kV	DIN	C	TRIPOLAR	50 A
FATOR DE DEMANDA	0,85												
TOTAL													

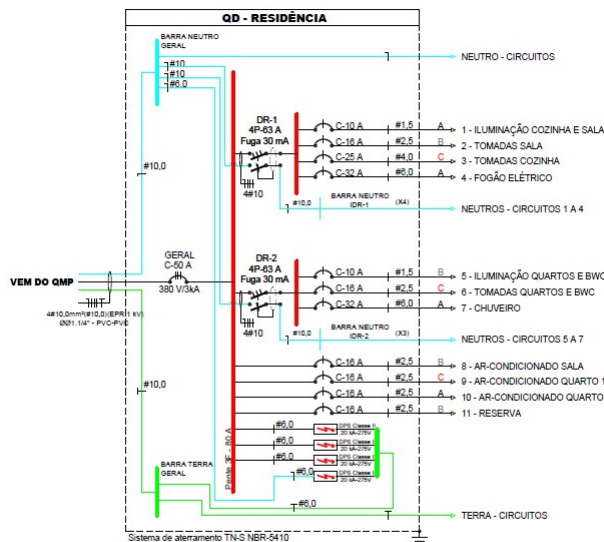
Fonte: autor.

Figura 24 – Quadro de cargas - Mensagem de aviso para informação não padronizada.

SE - B	FASE - C	I [A]	QUANTIDADE DE CABOS	DIVIDIR CABOS?	REDUZIR TERRA?	SEÇÃO CONDUTOR	TIPO ISOLAÇÃO	TIPO DISJUNTOR	CURVA DISJUNTOR	QUANTIDADE PÓLOS
		14,82	F+N+T	NÃO	NÃO	#1,5	PVC 750 V	DIN	1	
543		2,47	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	
	1630	7,41	F+N+T	NÃO	NÃO	#4,0	PVC 750 V	DIN	C	
		30,00	F+N+T	NÃO	NÃO	#6,0	PVC 750 V	DIN	C	
217		0,99	F+N+T	NÃO	NÃO	#4,0 #6,0	PVC 750 V	DIN	C	
	1196	5,43	F+N+T	NÃO	NÃO	#10,0	PVC 750 V	DIN	C	
		30,00	F+N+T	NÃO	NÃO	#16,0 #25,0	PVC 750 V	DIN	C	
087		4,94	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	
	1087	4,94	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	
		4,94	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	
109		0,49	F+N+T	NÃO	NÃO	#2,5	PVC 750 V	DIN	C	
		0,00								
			3F+N+T	NÃO	NÃO	#6,0	PVC 750 V	DIN	C	

Fonte: autor.

Figura 25 – Demonstração de projeto gerado pelo *plugin*, contendo - Esquema unifilar, lista de materiais, quadro de cargas simplificado e blocos para esquema de montagem.



Descrição	Quantidade	Unidade	Fabricante	Código	Observações
DISJUNTOR C 10 A UNIPOLAR	2	Un.	Siemens	SSL3	
DISJUNTOR C 16 A UNIPOLAR	6	Un.	Siemens	SSL3	
DISJUNTOR C 25 A UNIPOLAR	1	Un.	Siemens	SSL3	
DISJUNTOR C 32 A UNIPOLAR	2	Un.	Siemens	SSL3	
DISJUNTOR C 50 A TRIPOLAR	1	Un.	Siemens	SSL3	
TERMINAL TUBULAR #1,5	8	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #2,5	40	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #4,0	4	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #6,0	20	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #10,0	4	Un.	Phoenix Contact	-	
TETRAPOLAR 63 A Fuga 30 mA	2	Un.	Siemens	65V	
DPS Classe II - 20 kA (FN)	4	Un.	Clamper	014296	
QUADRO MÍNIMO (DIN)	30	Peça			Espaço reserva calculado: 4 [DIN]

Circuito	Descrição	Potência	Fase A	Fase B	Fase C	Condutor	Disjuntor	IDR
1	ILUMINAÇÃO COZINHA E SALA	200	200			#1,5	Curva C - 1x10 A	IDR1-4P-63 A
2	TOMADAS SALA	500		543		#2,5	Curva C - 1x16 A	IDR1-4P-63 A
3	TOMADAS COZINHA	1500			1630	#4,0	Curva C - 1x25 A	IDR1-4P-63 A
4	FOGÃO ELÉTRICO	6500	6500			#6,0	Curva C - 1x32 A	IDR1-4P-63 A
5	ILUMINAÇÃO QUARTOS E BWC	200		217		#1,5	Curva C - 1x10 A	IDR2-4P-63 A
6	TOMADAS QUARTOS E BWC	1100			1196	#2,5	Curva C - 1x16 A	IDR2-4P-63 A
7	CHUVEIRO	6600	7174			#6,0	Curva C - 1x32 A	IDR2-4P-63 A
8	AR-CONDICIONADO SALA	1000			1087	#2,5	Curva C - 1x16 A	
9	AR-CONDICIONADO QUARTO 1	1000			1087	#2,5	Curva C - 1x16 A	
10	AR-CONDICIONADO QUARTO 2	1000	1000			#2,5	Curva C - 1x16 A	
11	RESERVA	100		109		#2,5	Curva C - 1x16 A	
(x3)	ESPAÇOS FÍSICOS	0						

1	2	3	4	5	6	7	8	9	10	11
10 A	16 A	25 A	32 A	16 A	16 A	32 A	16 A	16 A	16 A	16 A
Iluminação cozinha e sala	Tomadas sala	Tomadas cozinha	Fogão elétrico	Iluminação quartos e BWC	Tomadas quartos e BWC	Chuveiro	Ar-condicionado sala	Ar-condicionado quarto 1	Ar-condicionado quarto 2	Reserva
#1,5	#2,5	#4,0	#6,0	#1,5	#2,5	#6,0	#2,5	#2,5	#2,5	#2,5

Fonte: autor.

4 TESTE DE USABILIDADE

O teste de usabilidade, conforme visto na seção 2.8, busca utilizar-se de métodos para avaliar um produto ou sistema (RUBIN; CHISNELL, 2011). As etapas do teste de usabilidade serão descritas a seguir.

4.0.1 Objetivos do teste

O teste de usabilidade adotado têm como objetivos:

- Analisar a usabilidade do *plugin*;
- Detectar potenciais falhas;
- Obter ideias para melhorias ou funcionalidades;
- Receber avaliações gerais sobre o *plugin*, analisando a eficiência e eficácia.

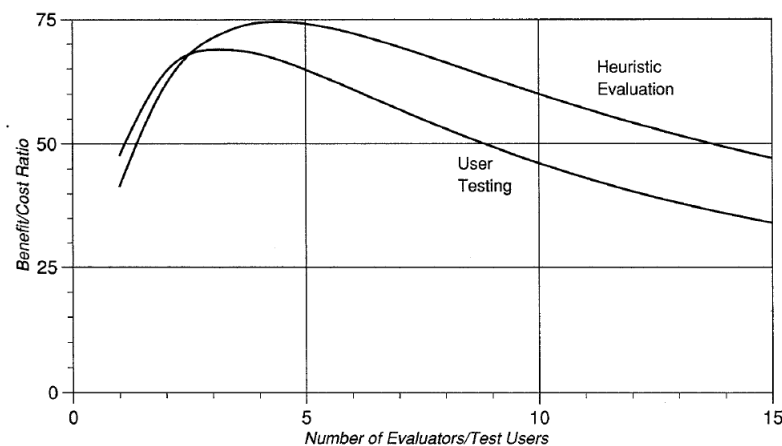
4.0.2 Preparação do teste

4.0.2.1 Seleção dos participantes

Determinar o perfil dos participantes dos testes é fundamental para garantir a coleta de dados válidos para o estudo. Considerando a questão abordada neste trabalho, o perfil dos participantes é descrito como profissionais envolvidos no desenvolvimento de projetos de instalações elétricas.

O número de participantes selecionados foi determinado conforme a pesquisa de Nielsen e Landauer (NIELSEN; LANDAUER, 1993), que estabelece a quantidade ideal com base na proporção de problemas identificados. De acordo com seus resultados, entre 4 e 5 participantes revelam mais de 75% dos problemas de usabilidade, uma proporção significativa, como indicado na Figura 26. Assim, optou-se por recrutar 4 participantes. Os dados dos participantes são apresentados no quadro 2.

Figura 26 – Relação entre benefício/custo e número de participantes.



Fonte: (NIELSEN; LANDAUER, 1993).

Quadro 2 – Dados sobre os participantes no teste de usabilidade.

Participante	Formação	Ocupação	Tempo de experiência
Participante 1	Superior Completo (engenharia elétrica)	Engenheiro(a) eletricista	>20 anos
Participante 2	Superior Completo (engenharia elétrica)	Engenheiro(a) eletricista	>10 anos
Participante 3	Superior Incompleto (engenharia elétrica)	Engenheiro(a) eletricista	>11 anos
Participante 4	Superior Completo (engenharia elétrica)	Engenheiro(a) eletricista	>3 anos

Fonte: autor.

4.0.2.2 Ambiente de teste

O local de testes foi realizado dentro do ambiente da empresa, pela simples facilidade de os participantes já se encontrarem no local. O local de testes foi deixado pré-configurado, com o computador ligado e o *software* na sua versão de *release* (versão estável do *software*) disponível para começar os testes. O quadro 3 destaca os itens utilizados no teste.

Quadro 3 – Compêndio itens do teste.

Itens	Descrição
Local	Escritório da empresa
Equipamento	Notebook com sistema operacional (SO) Windows 10
Programas necessários	AutoCAD versão 2019 - 2024 e plugin STC (objeto de estudo)
Documentos	Questionário pós-testes e manual de instruções
Moderador	Gustavo Henrique Silva Zanetti (Autor)

Fonte: autor.

4.0.2.3 Condução dos testes

Previamente, o usuário foi instruído com uma explicação de funcionamento do programa; portanto, foram repassadas todas as funcionalidades do *plugin* e do quadro de cargas, utilizando-se de um exemplo prático para demonstrar o funcionamento. Além

disso, foi explicado como o teste iria decorrer e o que era necessário ser feito, conforme demonstrado nas etapas do roteiro de teste montado para esse trabalho. Ao final do processo, o participante preencheu o formulário com questões qualitativas e quantitativas do programa como um todo, completando assim o teste de usabilidade.

Durante o teste, o moderador supervisionou e somente interveio ou engajou quando realmente necessário. O participante também contou com o manual do *plugin* e o roteiro de teste.

4.0.2.4 Questionário

Foi elaborado um questionário por meio do *Google Forms*, separado em 3 etapas conforme o alvo de estudo do teste, compondo perguntas qualitativas e quantitativas, o questionário foi separado em primeiro, perguntas sobre o quadro de cargas, sobre o *plugin* com o ambiente CAD e *feedbacks* gerais. As perguntas podem ser visualizadas na integra nas figuras 27 a 39 ou pelo [link](#) (ZANETTI, 2024b).

4.0.3 Aplicação do teste

4.0.3.1 Roteiro dos testes

O quadro 4 apresenta o roteiro de passos que o participante seguiu para proceder com os testes; o roteiro buscou seguir procedimentos de concepção de um projeto cotidiano, buscando abranger todas as funcionalidades. O arquivo do roteiro também foi adicionado ao apêndice B.

4.0.3.2 Arquivos de teste

Foi preparado previamente o quadro de cargas com algumas informações previamente lançadas para agilizar algumas etapas de projeto (não necessárias para fins de teste), assim o participante realiza apenas algumas alterações conforme requisição do procedimento de teste. Também foi preparado um roteiro contendo os procedimentos que o participante deveria realizar, disponível no apêndice B. Não foi necessário montar nenhum arquivo modelo(*template*) para o CAD, visto que o teste buscou apenas avaliar as ferramentas do *plugin*, sem a necessidade de utilização de blocos pré-configurados dentro do ambiente CAD. Logo, o teste utilizou-se do *template* padrão do AutoCAD.

4.0.4 Compilação dos resultados

4.0.4.1 Coleta, análise e apresentação dos dados

A coleta dos dados é realizada principalmente por meio de questionário, respondido no *Google forms* e por observações realizadas pelo moderador durante a condução do teste.

Quadro 4 – Roteiro do teste de usabilidade.

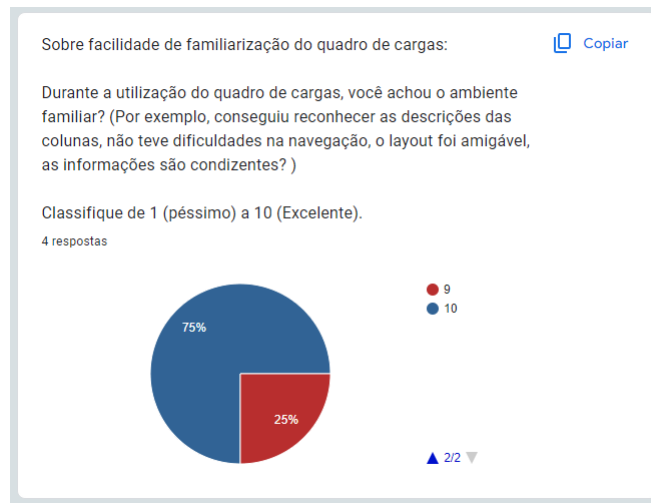
Procedimento	Instruções
1	Abra o arquivo de quadro de cargas (parcialmente preenchido) e adicione um circuito entre o circuito 1 e 2 existente no quadro, o novo circuito deverá compor o grupo de IDR 1 (previamente lançado), e a numeração dos circuitos deverá ser reordenada conforme a sequência de descrição, em ordem crescente.
2	Abra o autocad e com o comando "NETLOAD" carregue o arquivo que contém o plugin do programa.
3	Inicialize o plugin pelo comando ou pela interface do ribbon tab e carregue o quadro de cargas inicialmente modificado, na sequência selecione um ponto na tela para inserir o desenho ou digite uma coordenada.
4	Faça alterações de nome da nomenclatura de fases no menu de configurações e lance novamente o unifilar com o mesmo quadro de cargas, anteriormente carregado.
5	Gere a lista de material no menu e exporte para excel e no ambiente CAD.
6	Altere os itens de fabricante e código nas opções de lista de material e gere a lista novamente com as alterações.
7	Gere o quadro de cargas e os blocos do esquema de montagem.

Fonte: autor.

A análise dos dados, seguindo conforme objetivos tratados na seção 4.0.1, sendo divididos por dados qualitativos e quantitativos (RUBIN; CHISNELL, 2011). As perguntas juntamente com os resultados dos testes são apresentados nas figuras 27 a 39 abaixo.

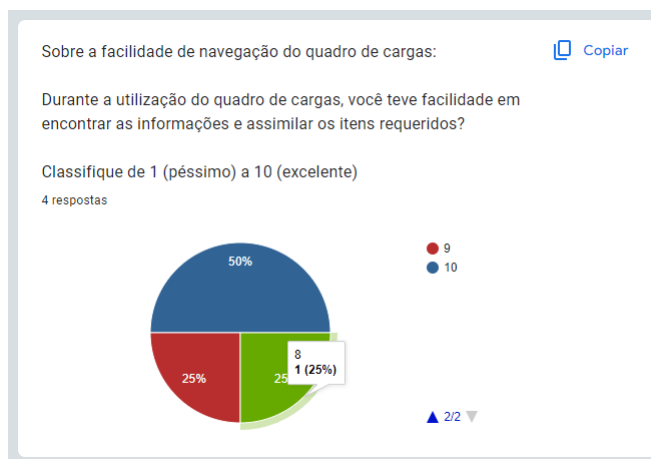
- Qualitativos: comentários livres separados por positivos e negativos;
- Quantitativos: respostas mensuradas por classificação numérica.

Figura 27 – Pergunta 1 - Quadro de cargas - Sobre a facilidade de familiarização.



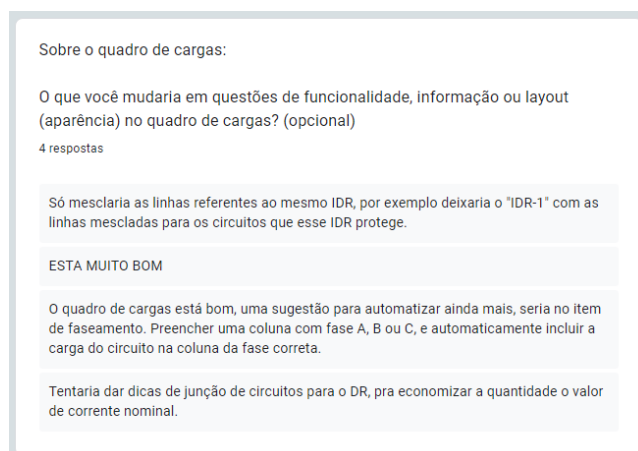
Fonte: Autor.

Figura 28 – Pergunta 2 - Quadro de cargas - Sobre a facilidade de navegação.



Fonte: Autor.

Figura 29 – Pergunta 3 - Quadro de cargas - Sugestões de melhorias.



Fonte: Autor.

Figura 30 – Pergunta 4 - Quadro de cargas - Erros ou dificuldades durante o teste.

Sobre o quadro de cargas:

Você se deparou com algum erro ou dificuldade durante o teste? Se sim, relate brevemente o que ocorreu.

4 respostas

Sem erro ou dificuldades

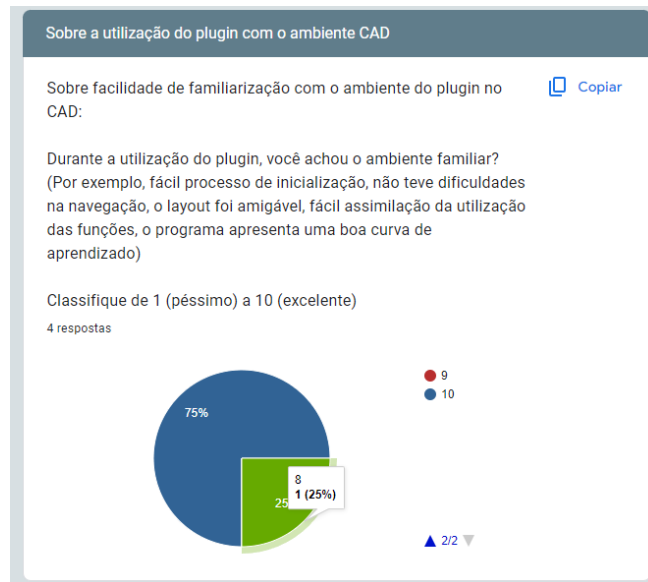
NAO

Não tive dificuldades.

Tudo certo.

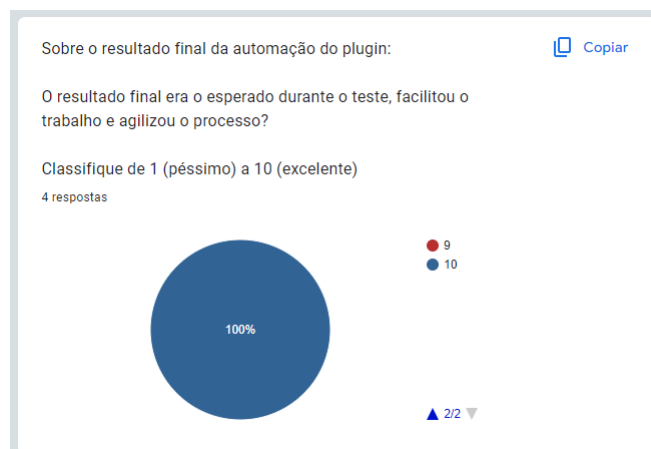
Fonte: Autor.

Figura 31 – Pergunta 1 - *Plugin* - Sobre a facilidade de familiarização.



Fonte: Autor.

Figura 32 – Pergunta 2 - *Plugin* - Sobre o resultado final da automação (esquema).



Fonte: Autor.

Figura 33 – Pergunta 3 - *Plugin* - Sugestões de melhorias.

Sobre o plugin no CAD:

O que você mudaria em questões de funcionalidade, informação ou layout (aparência) no plugin? (opcional)

4 respostas

Não mudaria nada de aparência, apenas explicaria melhor sobre a inserção das tabelas, unifilares e blocos de montagem no CAD.

ATALHO DE INICIALIZAÇÃO INCLUIR

Acredito que o default do ponto de inserção do diagrama no desenho poderia ser com clique na tela ao invés das coordenadas, que é menos usual, assim como o ponto de inserção dos outros itens (quadro de cargas, lista de materiais). Outro ponto que cabe melhoria é padronizar o ponto de inserção do objeto, pois alguns ficam alinhados no canto superior esquerdo, e outros no canto inferior esquerdo.

Nada a mudar.

Fonte: Autor.

Figura 34 – Pergunta 4 - *Plugin* - Erros ou dificuldades durante o teste.

Sobre o plugin no CAD:

Você se deparou com algum erro ou dificuldade durante o teste? Se sim, relate brevemente o que ocorreu.

4 respostas

Não.

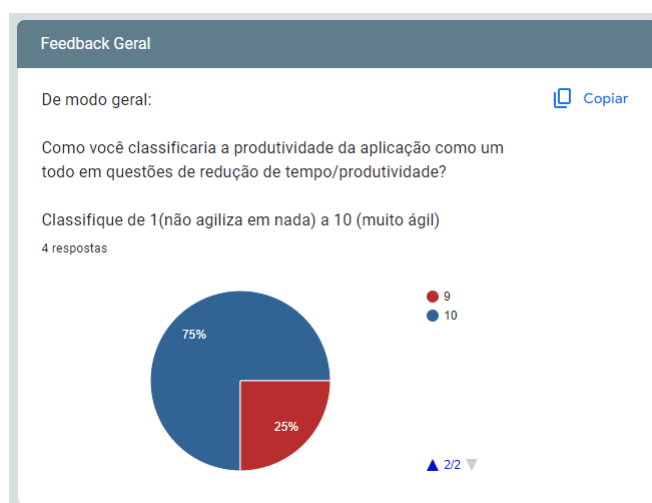
NAO

Não tive dificuldades.

Não

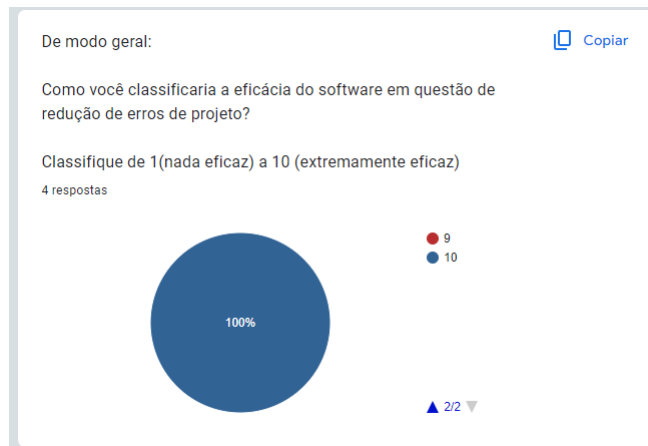
Fonte: Autor.

Figura 35 – Pergunta 1 - *Feedback* geral - Classificação de produtividade.



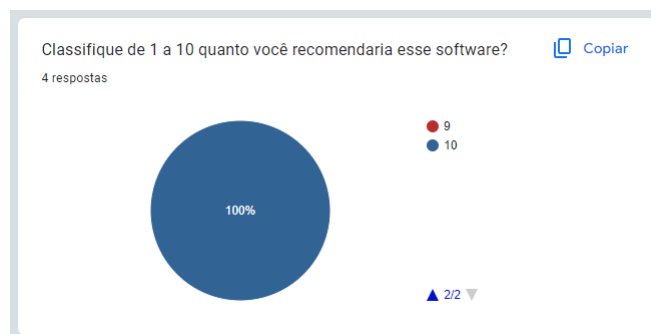
Fonte: Autor.

Figura 36 – Pergunta 2 - *Feedback* geral - Classificação de eficácia para redução de erros.



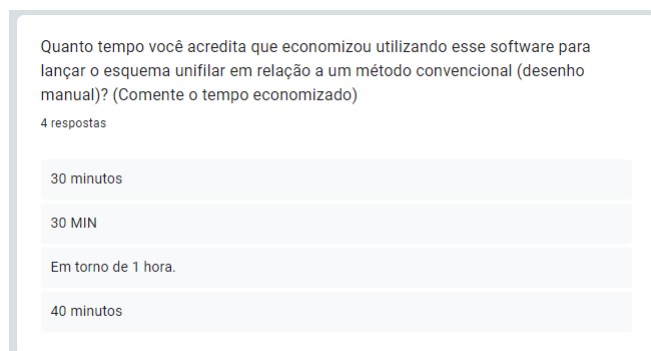
Fonte: Autor.

Figura 37 – Pergunta 3 - *Feedback* geral - Classificação de recomendação.



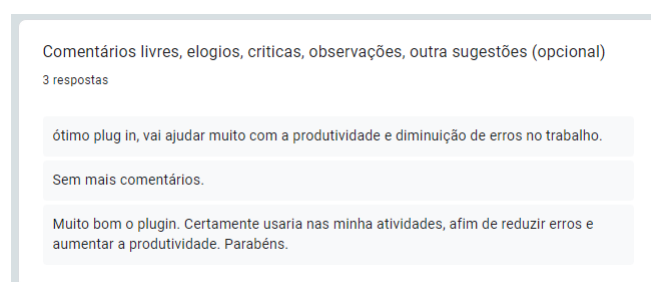
Fonte: Autor.

Figura 38 – Pergunta 4 - *Feedback* geral - Expectativa de tempo economizado.



Fonte: Autor.

Figura 39 – Pergunta 5 - *Feedback* geral - Comentários livres.



Fonte: Autor.

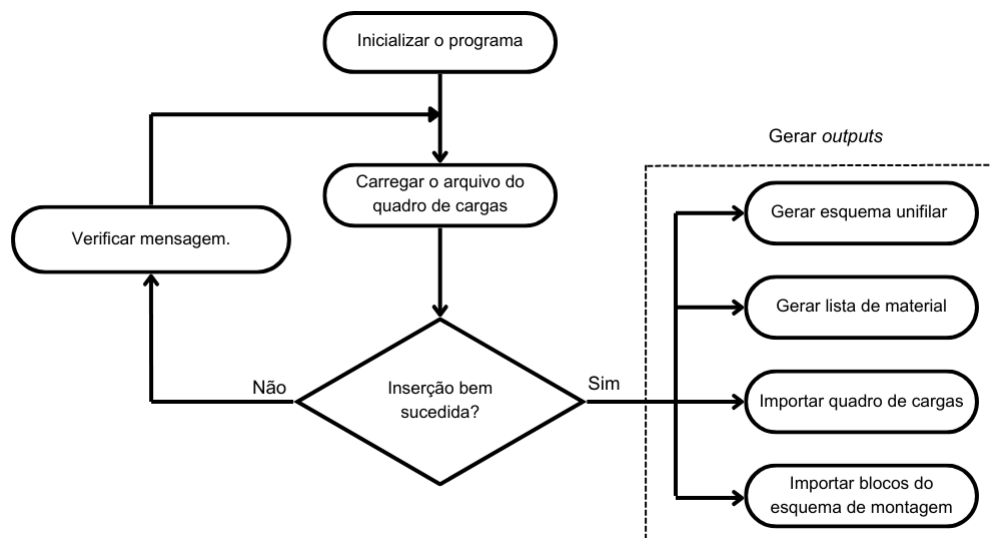
5 RESULTADOS E DISCUSSÕES

Neste capítulo, serão apresentados os resultados derivados da utilização do *plugin*, abrangendo as funcionalidades empregadas, a avaliação dos indicadores de desempenho e o teste de usabilidade realizado.

5.1 *Plugin*

O *plugin* é uma ferramenta que caracterizou a diminuição do tempo de projeto, pela sua ferramenta de automação de esquemas unifilares, além da redução de erros ocasionados em trabalhos manuais de desenho de esquema unifilar. A figura 40, apresenta de forma resumida o fluxograma operacional do *plugin*. O código desenvolvido pode ser acessado neste [link](#) (ZANETTI, 2024a).

Figura 40 – Fluxograma operacional do *plugin*.



Fonte: autor.

Na tabela 1, estão apresentadas às métricas geradas pela IDE do *Visual Studio* comparando entre as versões originais e a versão final do código, sendo a nova versão com diversas melhorias apresentadas anteriormente e refatoração do código.

Definições complementares às métricas do código podem ser encontradas no site da *Microsoft* (MICROSOFT, 2024c). Duas métricas são relevantes na análise no momento.

Sendo a primeira métrica o índice de facilidade de manutenção, definido entre a escala de 0 a 100, onde:

- 0 a 9: baixa facilidade de manutenção;
- 10 a 19: manutenção moderada;
- 20 a 100: boa capacidade de manutenção.

Portanto, observa-se que houve um aumento no índice de facilidade de manutenção

em relação a versão inicial do *plugin*, mesmo com um aumento de 72% de linhas de código fonte, em relação a versão inicial.

Tabela 1 – Métricas do código - Comparação entre a versão inicial e final.

Métrica	Versão Inicial (PI III)	Versão Final (TCC)
Índice de facilidade de manutenção	71	74
Complexidade Ciclomática	328	627
Profundidade de Herança	7	7
Acoplamento de classes	127	196
Linhas de código-fonte	3459	5954
Linhas de código executável	1605	2813

Fonte: autor.

Outro índice a ser analisado é a quantidade de linhas de código executável, nota-se que houve um aumento da quantidade de linhas executáveis, devido novas funcionalidades implementadas.

5.1.1 Indicadores de desempenho

Foram apurados dois indicadores para medir o desempenho do *plugin* sendo eles, tempo para projetar um esquema unifilar e redução de erros na geração do esquema unifilar. Portanto, foi feito o comparativo entre 5 projetos, compreendendo quadros de 15 a 25 disjuntores, sendo feito um pelo método manual, os resultados podem ser verificados pela tabela 2. E outro com a utilização do *plugin*, conforme demonstrado na tabela 3. Os resultados da média aritmética do tempo de execução são apresentados na tabela 4.

Faz-se importante salientar que o computador utilizado para rodar os testes possui um processador com 6 núcleos físicos, 16 GB de memória RAM e placa de vídeo Nvidia série RTX 3060. Idealmente, uma configuração mínima aceitável são as especificações reportadas pelo fabricante do *software* CAD (como AutoCAD), mas as configurações mínimas exigidas pelo fabricante têm impactos significativos no desempenho como um todo, ou seja, afetarão os tempos de execução do *plugin*.

Tabela 2 – Tempo de execução de projetos manualmente.

Quantidade de disjuntores	Tempo de projeto
20	24 min 32 s
24	25 min 50 s
25	31 min 55 s
22	23 min 00 s
15	22 min 36 s

Fonte: autor.

Tabela 3 – Tempo de execução de projetos pelo *plugin*.

Quantidade de disjuntores	Tempo de projeto
20	23 s
24	25 s
25	25 s
22	24 s
15	22 s

Fonte: autor.

Tabela 4 – Indicador de desempenho - Comparação da média aritmética do tempo de execução do projeto entre métodos.

Método	Tempo
Manual	25 min 35 s
<i>Plugin</i>	24 s

Fonte: autor.

Importante observar que o quadro de cargas já estava previamente lançado no teste de indicador de desempenho para o tempo, bem como para o indicador de desempenho para o erro na geração do esquema unifilar. Por tanto, é notável que o tempo de geração do esquema unifilar utilizado o *plugin* é drasticamente reduzido, correspondendo a 1,57% do tempo de execução do método manual.

A tabela 5 lista a quantidade de erros gerados em cada projeto executado de forma manual e a tabela 6, demonstra a média de erros entre os métodos gerados nos projetos, sendo considerado como erros, desalinhamentos, transcrição de dados, simbologia ou qualquer outro elemento que pudesse a vir ser caracterizado como um erro. Nota-se que o *plugin* não gerou erros em nenhum projeto, isso se deve principalmente pelo fato do *plugin* copiar exatamente o que está definido no quadro de cargas. O que implica em afirmar também que o quadro de cargas não é imune a erros; entretanto, nesta avaliação, considerou-se o que estava definido no quadro de cargas como correto.

Tabela 5 – Erros gerados pelo método manual

Nº projeto	Quantidade de erros
1	6
2	5
3	7
4	4
5	4

Fonte: autor.

Os erros gerados pelo ser humano na concepção do projeto podem variar em muitos aspectos, podendo ser gerados por uma simples falta de atenção ao copiar as informações, trabalho repetitivo de longa duração, cansaço na vista, etc. Portanto, é prudente afirmar que os erros podem variar frequentemente.

Tabela 6 – Indicador de desempenho - Comparação da média aritmética de erros gerados na execução do projeto entre métodos.

Método	Erros
Manual	5,2
<i>Plugin</i>	0

Fonte: autor.

5.2 Resultados do teste de usabilidade

Nesta seção são apresentados os resultados coletados no formulário aplicado para o teste de usabilidade.

5.2.1 Quadro de cargas

Foram aplicadas quatro perguntas aos participantes com relação ao quadro de cargas, sendo duas quantitativas e duas qualitativas. A tabela 7 e demonstra os resultados obtidos, sendo calculada a média aritmética e o desvio padrão.

A escala utilizada foi de 1 a 10, sendo que 1 representa o pior resultado (péssimo) e 10 o melhor resultado (excelente). De forma que todas as questões obtiveram ótimos resultados.

As duas perguntas qualitativas foram com relação a respostas livres sobre o que o participante mudaria em relação de funcionalidade, informação ou *layout* do quadro de cargas; e se o participante se deparou com algum erro ou dificuldade durante o teste.

De modo geral, nenhum usuário se deparou com erro ou dificuldade durante a utilização do quadro de cargas. Houve algumas sugestões de melhorias propostas que são pertinentes, mas que são justificadas, sendo elas:

Tabela 7 – Avaliações referentes ao quadro de cargas.

Pergunta	Média aritmética	Desvio Padrão
Sobre a facilidade de familiarização do quadro de cargas	9,75	0,50
Sobre a facilidade de navegação do quadro de cargas	9,25	0,96

Fonte: autor.

- Mesclar os grupos de IDR/DR para melhorar a visualização dos circuitos no quadro de cargas;
- Automatizar o preenchimento da distribuição de fases;
- Indicar corrente nominal utilizada no IDR/DR, levando em consideração a soma nominal da corrente dos dispositivos de proteção incorporados pelo grupo de IDR/DR.

O primeiro item é válido, entretanto, como o quadro de cargas foi concebido de forma que facilitou-se o algoritmo, a opção de utilizar índices para criar a distribuição dos IDR/DR foi a melhor opção encontrada.

Na segunda sugestão, buscou-se desenvolver o quadro de cargas de forma que o projetista decida a melhor maneira de fazer a distribuição de fases, visto que, em muitos casos, a distribuição de fases depende de como é a estrutura física do barramento do quadro e também como o quadro será montado, de forma que é inviável automatizar essa funcionalidade.

A última sugestão aborda o desenvolvimento de uma funcionalidade para indicar a utilização da corrente nominal do IDR/DR. Em situações onde o dispositivo abrange múltiplas fases e os circuitos estão distribuídos, sugeriu-se a necessidade de uma ferramenta que indique o nível de utilização de cada fase. Isso proporcionaria uma visualização mais clara e precisa da corrente nominal do dispositivo, permitindo uma gestão mais eficaz da energia elétrica.

5.2.2 *Plugin*

O questionário para o *plugin* seguiu a mesma linha abordada para o questionário do quadro de cargas. A tabela 8 demonstra os resultados obtidos, sendo todas as questões avaliadas com ótimos resultados.

De modo geral nessa etapa, nenhum usuário se deparou com erros na utilização do *plugin*. As sugestões requisitadas aos usuários são listadas a seguir:

- Melhorar a inserção das tabelas e blocos do esquema de montagem;
- Criar atalho para inicialização do *plugin*;
- Colocar como padrão a inserção de ponto na tela para os itens do *plugin*. Padronização do ponto de inserção dos objetos, pois alguns estão alinhados por arestas opostas.

A primeira sugestão foi corrigida posteriormente ao teste, em alguns casos o

Tabela 8 – Avaliações referentes ao *plugin*.

Pergunta	Média aritmética	Desvio Padrão
Sobre a facilidade de familiarização com o plugin	9,50	1,00
O resultado final foi o esperado, facilitou e agilizou o trabalho?	10,00	0,00

Fonte: autor.

formulário não estava ocultando automaticamente quando requisitado para o usuário escolher um ponto na tela, forçando-o a clicar na aplicação do CAD para então escolher o ponto, a funcionalidade foi ajustada de forma que, o formulário é ocultado automaticamente quando requisitado.

O atalho para o programa, embora o mesmo possa ser inicializado rapidamente em dois passos, requer a programação na linguagem LISP para configurar a inicialização conjunta com o CAD, podendo ser futuramente adicionada.

A inserção por ponto na tela padrão foi posteriormente ajustada, e a solução para padronizar o ponto de inserção pode ser implementada futuramente com a utilização de uma funcionalidade chamada *jigging*, permite que o usuário pré-visualize o objeto na tela e consiga compreender como o desenho está e em qual aresta o ponto de inserção se encontra (KOSHY, 2018), assim conseguindo definir o melhor lugar para o ponto de inserção (WALMSLEY, 2016).

5.2.3 *Feedback* geral

Nesta etapa foram feitas três perguntas quantitativas, utilizando o mesmo sistema de escala para notas, e duas qualitativas. Os resultados são apresentados na tabela 9 para as perguntas quantitativas, prevalecendo um ótimo resultado.

Tabela 9 – Avaliações referentes ao *feedback* geral.

Pergunta	Média aritmética	Desvio Padrão
Classifique a produtividade	9,75	0,50
Classifique a eficácia na redução de erros	10,00	0,00
Classifique quanto você recomendaria esse software	10,00	0,00

Fonte: autor.

As duas últimas perguntas são com relação, ao tempo economizado pelo usuário em relação a um método convencional (desenho manual), portanto o participante deveria informar o tempo que ele acha que economizou em um projeto em relação ao método

convencional. A tabela 10 apresenta o tempo que cada participante estimou economizar com a utilização do *plugin*.

A última pergunta destaca comentários livres como, elogios, críticas, observações e/ou outras sugestões. De modo geral, o *plugin* recebeu elogios e nenhum outro comentário adicional foi reportado.

O resultado do formulário pode ser acessado neste [link](#) (ZANETTI, 2024b).

Tabela 10 – Tempo estimado pelos participantes economizado com a utilização do *plugin*.

Participante	Tempo [minutos]
Participante 1	30
Participante 2	30
Participante 3	60
Participante 4	40

Fonte: autor.

6 CONCLUSÃO

O emprego deste *plugin* oferece claramente duas vantagens significativas. Em primeiro lugar, destaca-se a economia de tempo, uma vez que a criação manual de um esquema unifilar, da maneira convencional, pode demandar várias horas, a depender do tamanho do quadro de cargas e da quantidade de projetos, como pode-se observar pela tabela 4. A segunda vantagem está relacionada à mitigação de erros na transferência da informação do quadro de cargas para o esquema unifilar, comprovado pela tabela 6.

O *plugin* passou por melhorias e adição de novas funcionalidades, como visto no capítulo 3.2, destacando-se as novas funcionalidades dentre elas, a geração de lista de material, importação do quadro de cargas, geração de blocos para esquema de montagem, além de melhorias de *layout*, atalhos para acessar o programa pelo *ribbon tab*, desenvolvimento do manual e refatoração do código. Vale destacar também as melhorias desenvolvidas para o quadro de cargas, com o destaque de visualização para células.

Os indicadores de desempenho avaliados na seção 5.1.1 mostraram que o *plugin* tem um excelente desempenho, a tabela 4 mostrou que o tempo passou de aproximadamente 25 minutos para uma média de 24 segundos, ou seja, enquanto um único projeto pode ser executado pelo método convencional, isso corresponderia a aproximadamente 100 projetos executado no mesmo tempo de um projeto convencional. A tabela 6 também demonstrou que a mitigação de erros foi bem sucedida, em se tratando de lançamentos do desenho do esquema unifilar, não gerou nenhum erro.

O teste de usabilidade mostrou que o programa alcançou os objetivos inicialmente propostos, tanto em questões de produtividade quanto de eficácia na redução de erros, como demonstrado nas notas da tabela 8 e tabela 10. O teste também permitiu analisar melhorias a serem feitas futuramente, reportadas pelos participantes. Por fim, comprovando o êxito da implementação do *plugin*, ao mostrar que o desenvolvimento acelerou significativamente o trabalho e teve uma boa usabilidade reportada.

6.1 Sugestões para trabalhos futuros

Pretende-se dar continuidade no trabalho atualizando o *plugin*, com base nas sugestões requisitadas no teste de usabilidade e com implementações que inicialmente pretendia-se criar durante o TCC, mas que não foram implementadas pelo curto período de desenvolvimento disponível.

Das funcionalidades pretendidas a serem implementadas está, por exemplo, a criação de interfaces que permitam que o *plugin* trabalhe não apenas com o AutoCAD, mas também com outras aplicações CAD e outras fontes de dados, *Google Sheets*, *LibreOffice Calc*, etc. Além de melhorias das funcionalidades já implementadas no *plugin*, como por

exemplo, a aplicação de um algoritmo para a montagem do quadro (esquema de montagem), não apenas limitando-se a carregar os blocos para o esquema de montagem. O trabalho também dará melhoria contínua às aplicações já desenvolvidas.

Além disso, busca-se melhorar a automação do quadro de cargas, implementado funcionalidades que calculam a capacidade de condução de corrente do cabo do circuito, bem como a queda de tensão, cálculo de corrente corrigida, etc. Também indicado se o cabo e disjuntor são adequados para o circuito selecionado, assim, melhorando o processo de projeto.

6.2 Considerações Finais

Ao decorrer do desenvolvimento deste trabalho possibilitou-se o estudo aprofundado de aplicações de programação de *softwares* voltadas para a área de projetos de engenharia elétrica residencial e industrial. Provando-se um trabalho relevante ao demonstrar por meio de testes de usabilidade sua eficiência e eficácia na automação de projetos elétricos, melhorando a otimização de tempo e diminuindo possíveis erros de projeto.

É crucial continuar explorando e aprimorando soluções de automação para atender às crescentes demandas e complexidades dos projetos elétricos. Espera-se que este trabalho sirva como um ponto de partida para futuras pesquisas e desenvolvimentos nessa área, contribuindo para a evolução e inovação contínuas no campo da engenharia elétrica.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5361**: Disjuntores de baixa tensão. Rio de Janeiro, 1998. Citado na página 8.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5410**: Instalações elétricas de baixa tensão. Rio de Janeiro, 2004. Citado 2 vezes nas páginas 4 e 28.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR IEC 61643-11:2021**: Dispositivos de proteção contra surtos de baixa tensão parte 11: Dispositivos de proteção contra surtos conectados aos sistemas de baixa tensão - requisitos e métodos de ensaio. Rio de Janeiro, 2021. 106 p. Citado na página 11.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 61008-1**: Interruptores à corrente diferencial-residual para uso doméstico e similar sem dispositivo de proteção de sobrecorrentes (idr) parte 1: Regras gerais. Rio de Janeiro, 2022. 160 p. Citado na página 10.

AUTODESK. **NETLOAD (comando)**. 2022. Disponível em: <<https://help.autodesk.com/view/ACD/2022/PTB/?guid=GUID-D954BCC1-C4F0-488B-8F60-1D02D68940E0>>. Acesso em: 24 mar. 2024. Citado na página 22.

AUTODESK. **About the Ribbon**. 2023. Disponível em: <<https://help.autodesk.com/view/ACD/2023/ENU/?guid=GUID-D20EF1D7-4135-48A7-B68E-65BF3BFF3D70>>. Acesso em: 24 mar. 2024. Citado na página 25.

AUTODESK. **About Blocks**. 2024. Disponível em: <<https://help.autodesk.com/view/ACD/2024/ENU/?guid=GUID-7410E7FB-3E0D-4411-B8F6-DBD59C71E87D>>. Acesso em: 19 mar. 2024. Citado na página 16.

AUTODESK. **About Dynamic Blocks**. 2024. Disponível em: <<https://help.autodesk.com/view/ACD/2024/ENU/?guid=GUID-3C2FB982-3AF6-437B-987F-4EDF81EA0662>>. Acesso em: 19 mar. 2024. Citado na página 16.

AUTODESK. **About Layers**. 2024. Disponível em: <<https://help.autodesk.com/view/ACDLT/2024/ENU/?guid=GUID-6B3E3B5D-3AE2-4162-A5FE-CFE42AB0743B>>. Acesso em: 24 mar. 2024. Citado na página 23.

AUTODESK. **AutoLISP and Visual LISP (AutoLISP)**. 2024. Disponível em: <<https://help.autodesk.com/view/OARX/2023/ENU/?guid=GUID-49AAEA0E-C422-48C4-87F0-52FCA491BF2C>>. Acesso em: 18 abr. 2024. Citado na página 15.

AUTODESK. **Components of the AutoCAD .NET API (.NET)**. 2024. Disponível em: <<https://help.autodesk.com/view/OARX/2024/ENU/?guid=GUID-8657D153-0120-4881-A3C8-E00ED139E0D3>>. Acesso em: 17 mar. 2024. Citado na página 13.

AUTODESK. **CORPORATE INFO**. 2024. Disponível em: <<https://www.autodesk.com/company/newsroom/corporate-info>>. Acesso em: 19 mar. 2024. Citado na página 15.

AUTODESK. **The ObjectARX Programming Environment**. 2024. Disponível em: <<https://help.autodesk.com/view/OARX/2024/ENU/?guid=GUID-9A679CBB-F622-4A3F-BBE5-D4ED130EF303>>. Acesso em: 19 mar. 2024. Citado na página 15.

AUTODESK. **Overview of the AutoCAD .NET API (.NET)**. 2024. Disponível em: <<https://help.autodesk.com/view/OARX/2024/ENU/?guid=GUID-390A47DB-77AF-433A-994C-2AFBBE9996AE>>. Acesso em: 17 mar. 2024. Citado na página 13.

AWS. **What is Repo? (O que é Repo?)**. 2023. Disponível em: <<https://aws.amazon.com/pt/what-is/repo/>>. Acesso em: 17 mar. 2024. Citado na página 14.

AWS. **What is an API (Application Programming Interface)?** 2024. Disponível em: <<https://aws.amazon.com/what-is/api/>>. Acesso em: 17 mar. 2024. Citado na página 13.

BARRETO et al. **Interface humano-computador**. Editora Sagah, 2018. Disponível em: <<https://app.minhabiblioteca.com.br/#/books/9788595027374/>>. Acesso em: 21 mar. 2024. Citado 2 vezes nas páginas 16 e 17.

CELESC DISTRIBUIÇÃO S.A. **N-321.0001**: Fornecimento de energia elétrica em tensão secundária de distribuição. Florianópolis, 2019. Citado na página 4.

CELESC DISTRIBUIÇÃO S.A. **N-321.0003**: Fornecimento de energia elétrica a edificações de uso coletivo. Florianópolis, 2023. Citado na página 4.

CREDER, H. **Instalações elétricas**. Rio de Janeiro: Editora LTC, 2021. Citado 3 vezes nas páginas 1, 4 e 7.

D'AMARAL, G. M. **Plugin de AutoCAD para automatização de etapas do desenvolvimento de projetos de instalações elétricas**. 2021. Disponível em: <<https://repositorio.ifsc.edu.br/handle/123456789/2615>>. Acesso em: 22 mar. 2024. Citado na página 18.

FILHO, A. C. d. P. **A evolução do cad e sua aplicação em projetos de engenharia**. 2010. Citado 2 vezes nas páginas 14 e 15.

FILHO, D. L. L. F. **Projetos de instalações elétricas**. São Paulo: Editora Érica, 1997. Citado 7 vezes nas páginas 1, 2, 4, 5, 6, 8 e 9.

GITHUB. **About GitHub and Git**. 2024. Disponível em: <<https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>>. Acesso em: 17 mar. 2024. Citado na página 14.

KANTOWITZ, B. H.; SORKIN, R. D. **Human Factors: understanding people-system relationship**. [S.l.]: New York: Wiley, 1983. Citado 2 vezes nas páginas 2 e 11.

KOSHY, B. **Jig Tutorial - Jigs Explained - C# Code Walk-through (AutoCAD .net API)**. 2018. Disponível em: <<https://benkoshy.github.io/2018/01/24/what-is-a-jig.html>>. Acesso em: 13 abr. 2024. Citado na página 47.

LEE, J.; ANJOS, E.; SATTI, S. R. **SJSON: A succinct representation for JSON documents**. 2020. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0306437920301356>>. Acesso em: 24 mar. 2024. Citado na página 27.

MICROSOFT. **Desktop Guide (Windows Forms .NET)**. 2023. Disponível em: <<https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-8.0>>. Acesso em: 17 mar. 2024. Citado na página 14.

MICROSOFT. **Introdução ao .NET Framework**. 2023. Disponível em: <<https://learn.microsoft.com/pt-br/dotnet/framework/get-started/>>. Acesso em: 19 mar. 2024. Citado na página 12.

MICROSOFT. **What is .NET Framework?** 2023. Disponível em: <<https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet-framework>>. Acesso em: 19 mar. 2024. Citado na página 12.

MICROSOFT. **What is Visual Studio?** 2023. Disponível em: <<https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>>. Acesso em: 17 mar. 2024. Citado na página 14.

MICROSOFT. **Microsoft.Office.Interop.Excel Namespace**. 2024. Disponível em: <<https://learn.microsoft.com/pt-br/dotnet/api/microsoft.office.interop.excel?view=excel-pia>>. Acesso em: 24 mar. 2024. Citado na página 21.

MICROSOFT. **Passo a passo: programação do Office em C#**. 2024. Disponível em: <<https://learn.microsoft.com/pt-br/dotnet/csharp/advanced-topics/interop/walkthrough-office-programming>>. Acesso em: 24 mar. 2024. Citado na página 21.

MICROSOFT. **Valores de métrica de código**. 2024. Disponível em: <<https://learn.microsoft.com/pt-br/visualstudio/code-quality/code-metrics-values?view=vs-2022>>. Acesso em: 30 mar. 2024. Citado na página 42.

MICROSOFT. **What is a DLL**. 2024. Disponível em: <<https://learn.microsoft.com/en-us/troubleshoot/windows-client/setup-upgrade-and-drivers/dynamic-link-library>>. Acesso em: 17 mar. 2024. Citado na página 13.

MOLDOVAN, P. M. S.; MOLDOVAN, I. Cad software as a tool in design. 2010. Citado na página 15.

MULTIPLUS. **PRO-Elétrica Software para projetos elétricos**. 2024. Disponível em: <<https://multiplus.com/software/pro-eletrica/index.html>>. Acesso em: 22 mar. 2024. Citado na página 17.

NIELSEN, J.; LANDAUER, T. K. A mathematical model of the finding of usability problems. **Interchi'93**, 1993. Citado na página 34.

REASON, J. **Human Error**. [S.l.]: Cambridge: Cambridge University Press, 1990. Citado na página 11.

RUBIN, J.; CHISNELL, D. **Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests**. 2011. Disponível em: <https://books.google.com.br/books?id=L_e1MmVzMb0C>. Acesso em: 25 mar. 2024. Citado 2 vezes nas páginas 34 e 37.

SEBESTA, R. W. **Conceitos de Linguagens de Programação**. Porto Alegre: Editora Bookman, 2018. Citado na página 12.

SILVA, B. G. de J. et al. Confiabilidade humana: Uma abordagem atual do erro humano. **IX SIMPROD - Simpósio de engenharia de produção de Sergipe**, 2017. Citado na página 11.

TAVARES, I. S. **Análise dos indicadores de desempenho da máquina Reach Stacker após a implementação de software testador de joystick no setor de manutenção portuária**. 2022. Disponível em: <<https://repositorio.ifsc.edu.br/handle/123456789/2631>>. Acesso em: 13 abr. 2024. Citado na página 19.

WALMSLEY, K. **Jigging an AutoCAD leader with directional text using .NET**. 2016. Disponível em: <<https://www.keanw.com/2016/09/jigging-an-autocad-leader-with-directional-text-using-net.html>>. Acesso em: 13 abr. 2024. Citado na página 47.

WAZLAWICK, R. **História da computação**. [S.l.]: Editora Grupo GEN, 2016. Citado na página 1.

WEG. **Soluções integradas para instalações elétricas**. 2023. Disponível em: <<https://static.weg.net/medias/downloadcenter/h24/h69/WEG-MDW-MDWH-QDW-SPW-RDW-DWP-50023623-en.pdf>>. Acesso em: 18 abr. 2024. Citado 2 vezes nas páginas 10 e 11.

WEG. **Minidisjuntores MDW e MDWH**. 2024. Disponível em: <https://www.weg.net/catalog/weg/US/pt/c/Minidisjuntores-MDW-e-MDWH/p/MKT_WDC_GLOBAL_MINIATURE_CIRCUIT_BREAKERS_MDW_MDWH>. Acesso em: 18 abr. 2024. Citado na página 9.

ZANETTI, G. H. S. **Plugin CAD STC**. 2024. Disponível em: <<https://github.com/ifsc-itj/PluginCAD-STC>>. Acesso em: 17 mar. 2024. Citado na página 42.

ZANETTI, G. H. S. **Teste de usabilidade - Plugin STC - Sheet To CAD**. 2024. Disponível em: <https://docs.google.com/forms/d/1lKC_NrfomBmQPpe0K_5LkJkUx7gSgDL0_XJJVDQaW8mI/viewanalytics>. Acesso em: 13 abr. 2024. Citado 2 vezes nas páginas 36 e 48.

Apêndices

APÊNDICE A – Manual do *plugin*

MANUAL DE INSTRUÇÕES PLUGIN STC (SHEET TO CAD)

Elaboração: Gustavo Henrique Silva Zanetti

Orientador: Prof. Dr. Sergio Augusto Bitencourt Petrovic

IFSC – Instituto Federal de Santa Catarina – Campus Itajaí

SUMÁRIO

INICIALIZAR O <i>PLUGIN</i>	3
APRESENTAÇÃO DO PROGRAMA	4
COMO GERAR ESQUEMA UNIFILAR	5
SELECIONAR ARQUIVO	5
CONFIGURAÇÃO DO DESENHO	6
CONFIGURAÇÕES	7
LISTA DE MATERIAL	8
OPÇÕES DA LISTA DE MATERIAL	9
IMPORTAÇÃO DO QUADRO DE CARGAS	10
INSERIR BLOCOS DO ESQUEMA DE MONTAGEM	10
PREENCHIMENTO DO QUADRO DE CARGAS	11
PREENCHIMENTO INFORMAÇÕES DR/IDR.....	11
PREENCHIMENTO INFORMAÇÕES DPS, PROTEÇÃO DPS E ENTRADA DE ENERGIA	12
SOLUÇÃO DE ERROS	12

TABELA DE FIGURAS

FIGURA 1 - COMANDO NETLOAD	3
FIGURA 2 - SELEÇÃO DO ARQUIVO .DLL DO <i>PLUGIN</i>	3
FIGURA 3 - MENSAGEM DE INICIALIZAÇÃO.....	4
FIGURA 4 - MENU DO <i>RIBBON TAB</i> CRIADO PARA ACESSAR O PROGRAMA	4
FIGURA 5 - TELA INICIAL DO PROGRAMA	4
FIGURA 6 - SELECIONAR ARQUIVO	5
FIGURA 7 - SELEÇÃO DO ARQUIVO DO QUADRO DE CARGAS	5
FIGURA 8 - SELEÇÃO DA PLANILHA	6
FIGURA 9 - CONFIGURAÇÕES GERAIS DO ESQUEMA UNIFILAR.....	6
FIGURA 10 - DEFINIÇÃO DAS COORDENADAS DE INSERÇÃO DO ESQUEMA UNIFILAR	7
FIGURA 11 - RESULTADO FINAL DO ESQUEMA UNIFILAR GERADO	7
FIGURA 12 - MENU CONFIGURAÇÕES.....	8
FIGURA 13 - GERAR LISTA DE MATERIAL	8
FIGURA 14 - LISTA DE MATERIAIS GERADA.....	9
FIGURA 15 - MENU OPÇÕES LISTA DE MATERIAL	9
FIGURA 16 - MENU QUADRO DE CARGAS	10
FIGURA 17 - MENU ESQUEMA DE MONTAGEM	10
FIGURA 18 - INSERIR LINHAS NO QUADRO DE CARGAS.....	11
FIGURA 19 - DEMONSTRAÇÃO MENU SUSPENSO.....	11
FIGURA 20 - PREENCHIMENTO DR/IDR.....	12

INICIALIZAR O *PLUGIN*

Abra a aplicação do AutoCAD.

Inicie um novo desenho.

Digite o comando “NETLOAD” e aperte “Enter”.

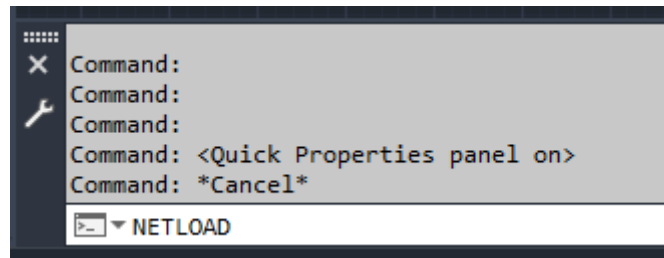


Figura 1 - Comando NETLOAD

Selecione o arquivo .dll do *plugin* e aperte “Abrir” (open), conforme demonstrado na Figura 2:

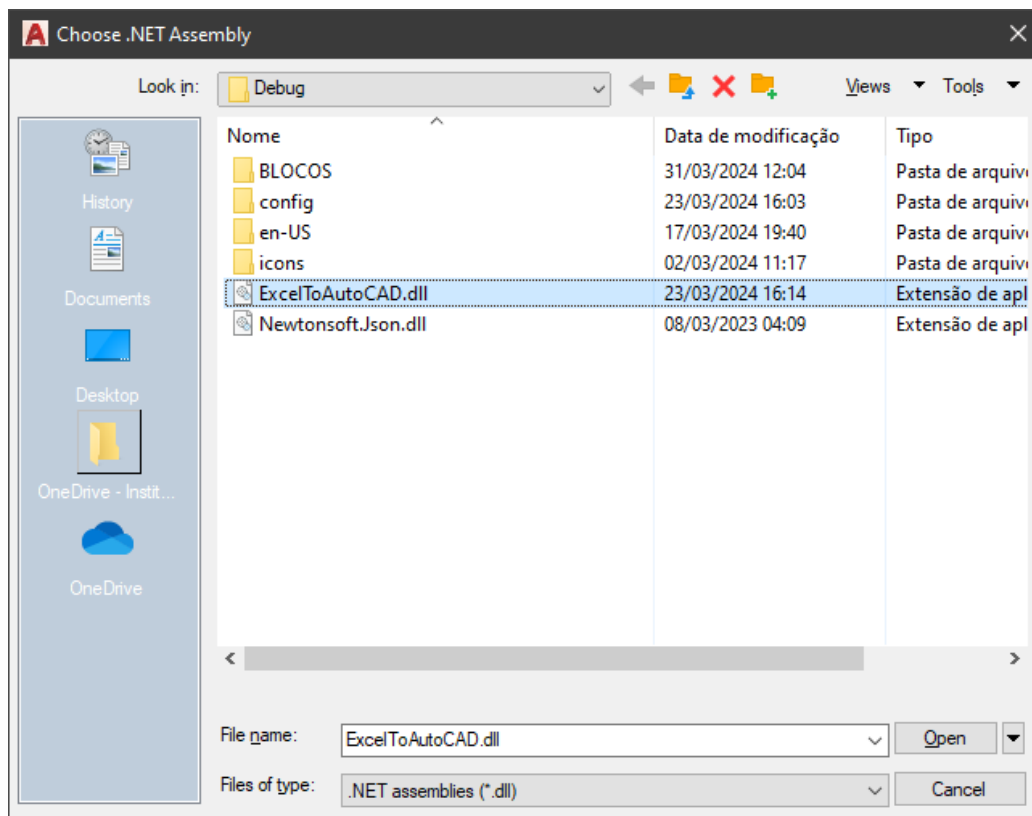


Figura 2 - Seleção do arquivo .dll do plugin

O plugin inicializa com uma mensagem de orientação, o usuário pode acessar o plugin pelo comando: “STC-SHEET-TO-CAD” ou acessar pela barra de tarefas (ribbon tab) superior na aba “Sheet to CAD”, visto na Figura 4.

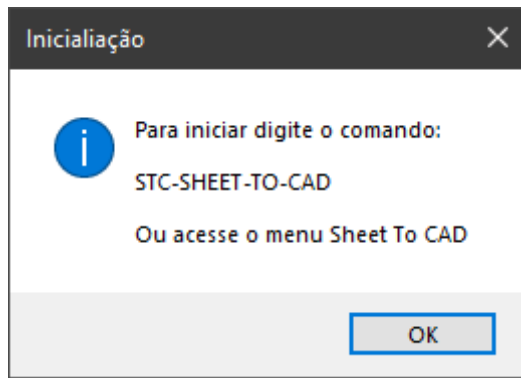


Figura 3 - Mensagem de inicialização

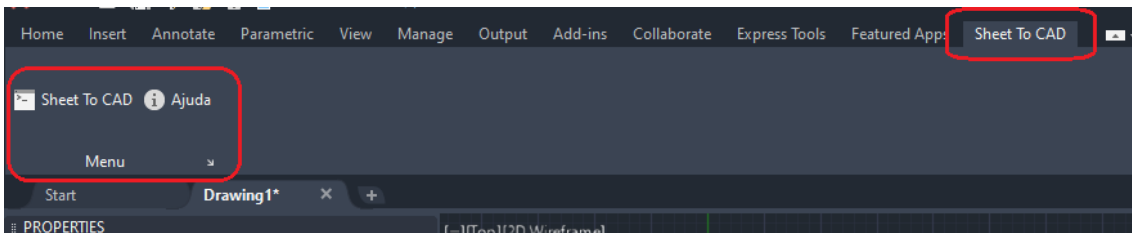


Figura 4 - Menu do *ribbon tab* criado para acessar o programa

Apresentação do programa

O programa trabalha apenas com uma janela (formulário), com as funcionalidades separadas por abas, conforme demonstra a Figura 5.

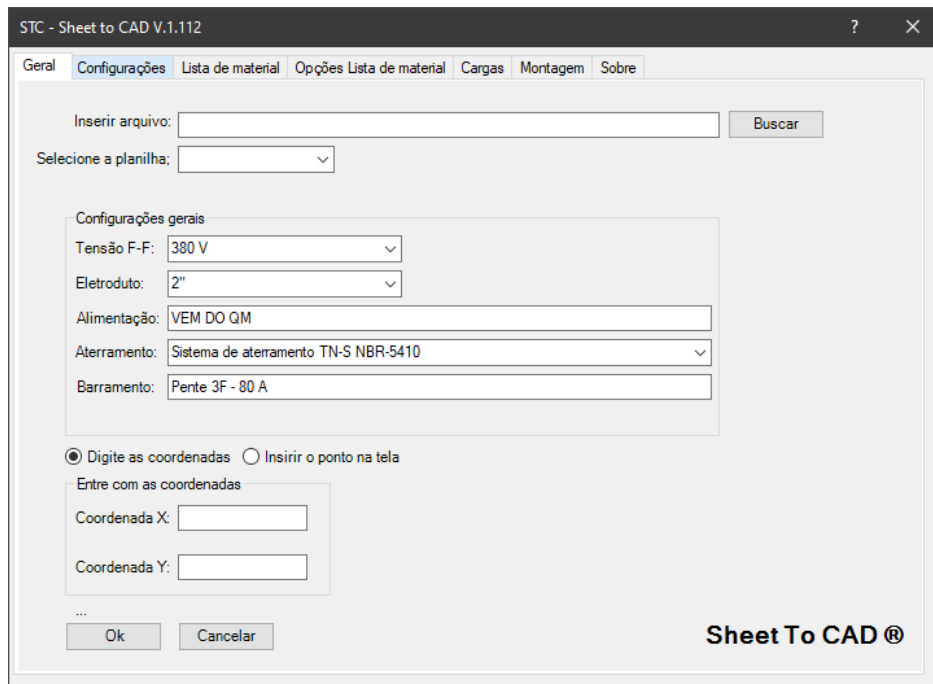


Figura 5 - Tela inicial do programa

Como gerar esquema unifilar

Selecionar arquivo

Para selecionar um arquivo, clique no botão buscar.

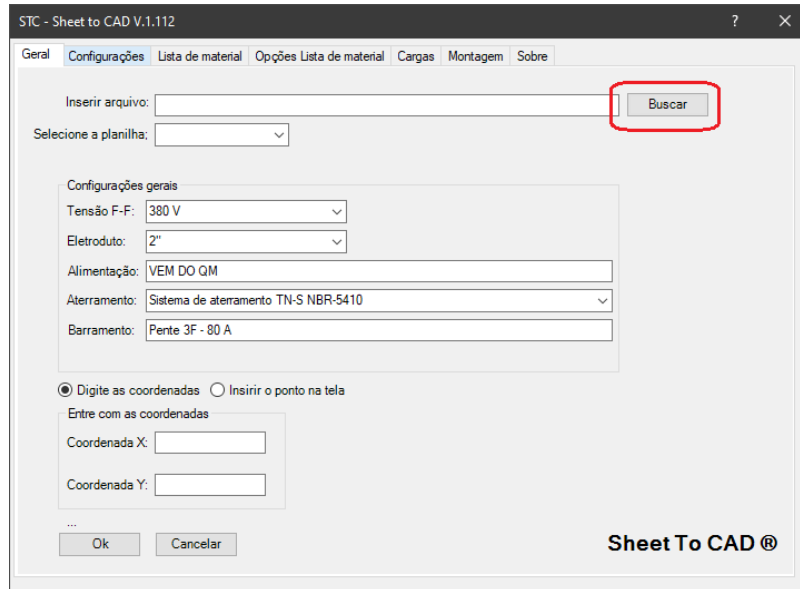


Figura 6 - Selecionar arquivo

O programa irá abrir o navegador de arquivo do *Windows (Windows explorer)*, então selecione o arquivo desejado e clique em abrir.

IMPORTANTE: Selecione apenas arquivos com a padronização do quadro de cargas, caso contrário o programa irá gerar um erro.

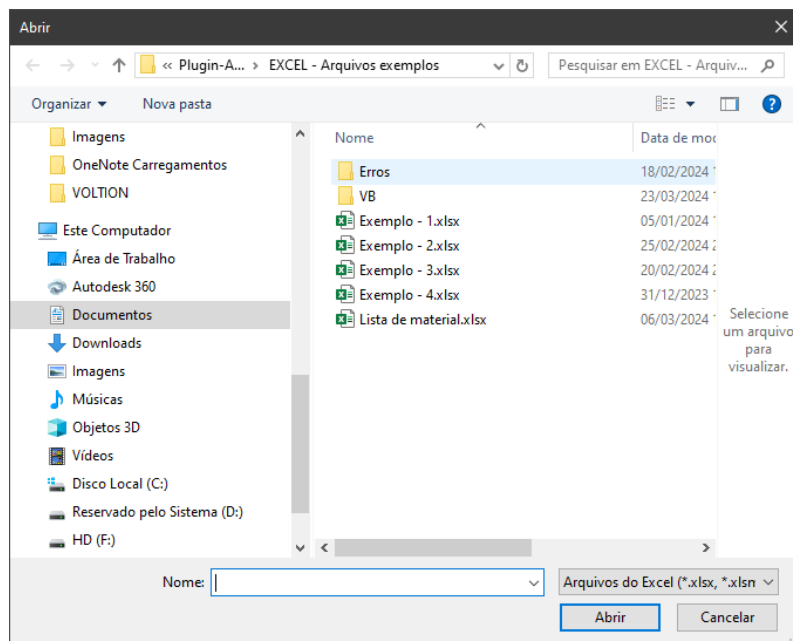


Figura 7 - Seleção do arquivo do quadro de cargas

Configuração do desenho

Após selecionado o arquivo, selecione a planilha desejada:

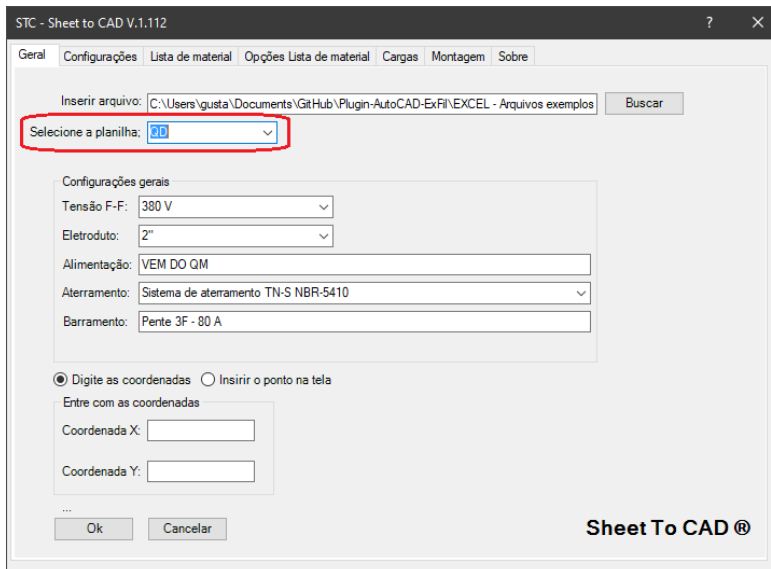


Figura 8 - Seleção da planilha

Caso deseje, você pode alterar as configurações gerais do esquema unifilar, conforme necessário no seu projeto:

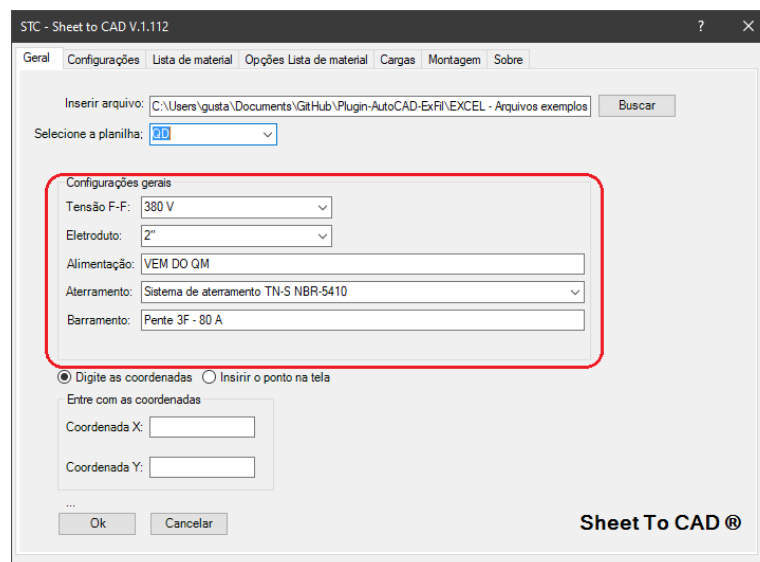


Figura 9 - Configurações gerais do esquema unifilar

Após, você pode escolher entre digitar as coordenadas, por exemplo, (0,0); ou escolher o ponto na tela, caso escolha por inserir o ponto na tela, você deve informar em qual ponto do espaço será inserido o desenho, o esquema é gerado iniciando pelo canto direito superior da tela.

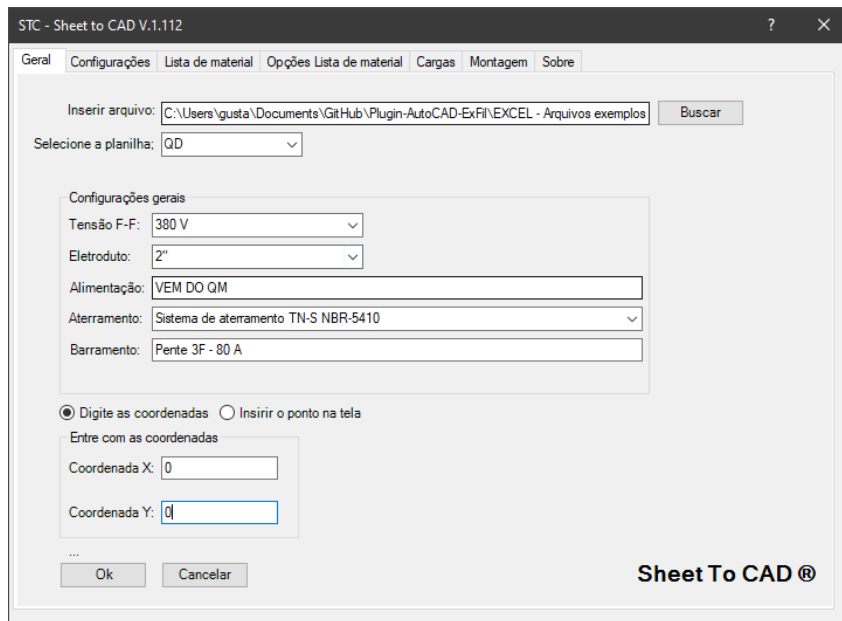


Figura 10 - Definição das coordenadas de inserção do esquema unifilar

Caso não ocorra nenhum erro o desenho será gerado no espaço de modelagem do CAD.

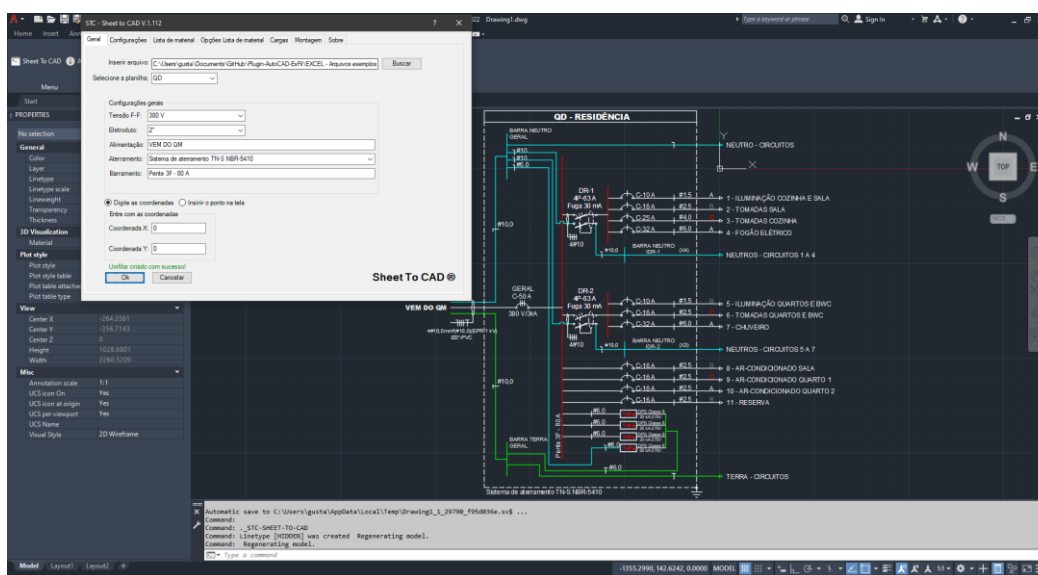


Figura 11 - Resultado final do esquema unifilar gerado

CONFIGURAÇÕES

A aba de configurações permite alterar a escala global do desenho (não recomendado fazer a alteração), mudar os nomes das Layers, tipo da fonte do texto, tamanho da fonte do texto e a nomenclatura das fases.

Após fazer as alterações clique em aplicar para aplicar as modificações desejadas, caso seja feita mudanças com relação ao esquema o mesmo deverá ser reinserido para que as mudanças sejam efetivadas.

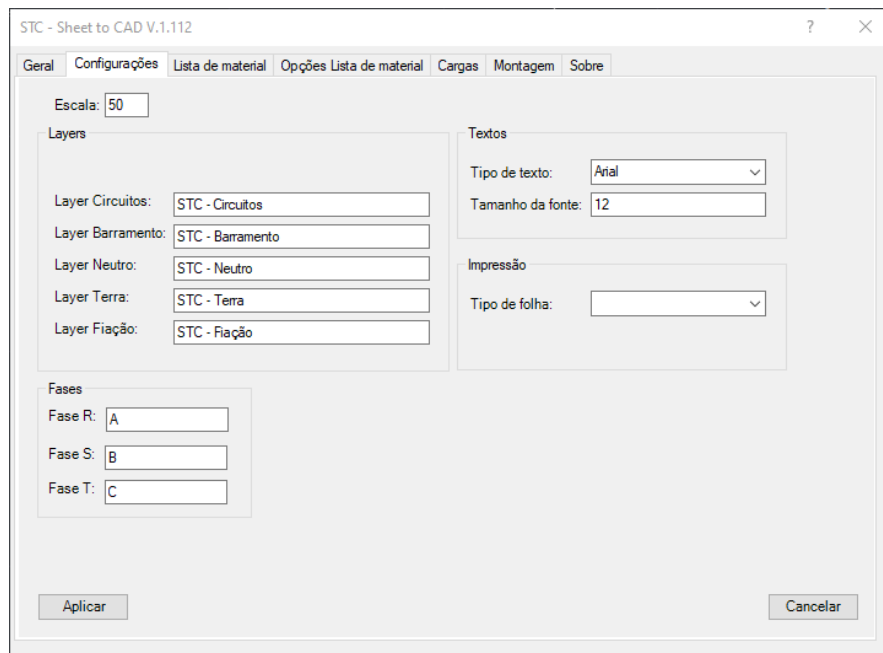


Figura 12 - Menu configurações

LISTA DE MATERIAL

A lista de material gera uma lista de material sugestiva para a montagem do quadro. Para gerar a lista clique no botão “Gerar”, conforme a Figura 13 demonstra:

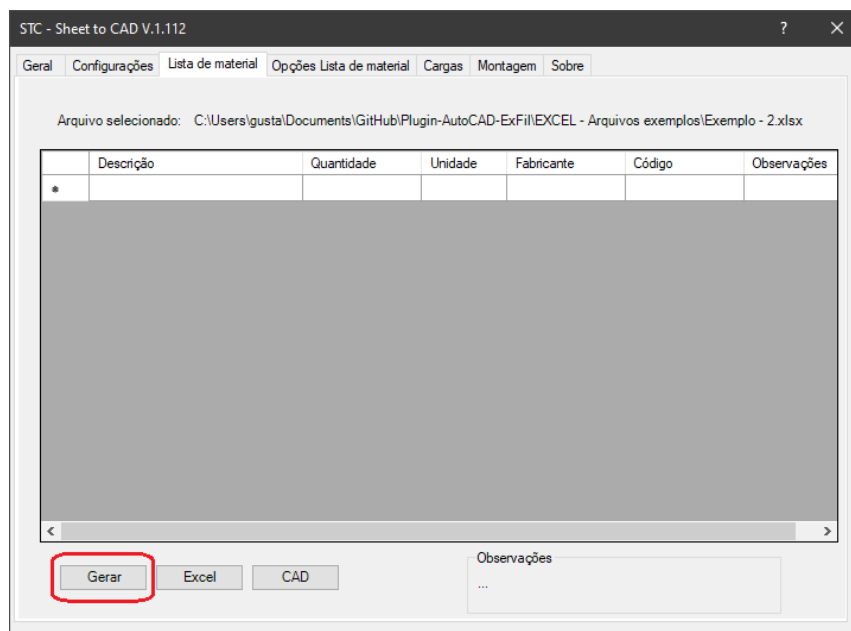


Figura 13 - Gerar lista de material

Após gerada a lista o usuário pode escolher entre inserir o desenho no CAD ou exportar para o Excel.

Descrição	Quantidade	Unidade	Fabricante	Código	Observações
DISJUNTOR C 10 A UNIPOLAR	2	Un.	Siemens	5SL3	
DISJUNTOR C 16 A UNIPOLAR	6	Un.	Siemens	5SL3	
DISJUNTOR C 25 A UNIPOLAR	1	Un.	Siemens	5SL3	
DISJUNTOR C 32 A UNIPOLAR	2	Un.	Siemens	5SL3	
DISJUNTOR C 50 A TRIPOLAR	1	Un.	Siemens	5SL3	
TERMINAL TUBULAR #1,5	8	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #2,5	40	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #4,0	4	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #6,0	20	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #10,0	4	Un.	Phoenix Contact	-	
TETRAPOLAR 63 A Fuga 30 mA	2	Un.	Siemens	5SV	
DPS Classe II - 20 kA (FN)	4	Un.	Clamper	014295	
QUADRO MINIMO [DIN]	30	Peça			Espaço rese

Figura 14 - Lista de materiais gerada

OPÇÕES DA LISTA DE MATERIAL

No menu de “Opções Lista de Material”, o usuário pode configurar o fabricante e o código do produto que deseje que seja contabilizado, lembrando que será aplicado para a planilha inteira gerada. Após alterar clique em salvar, caso deseje os dados podem ser restaurados no botão “Restaurar”, isso irá restaurar as configurações do programa para o padrão de fábrica.

Fabricante

Disjuntor: Siemens

Disjuntor Residual: Siemens

Terminal: Phoenix Contact

DPS: Clamper

Código

Disjuntor: 5SL3

Disjuntor Residual: 5SV

Terminal: -

DPS: 014295

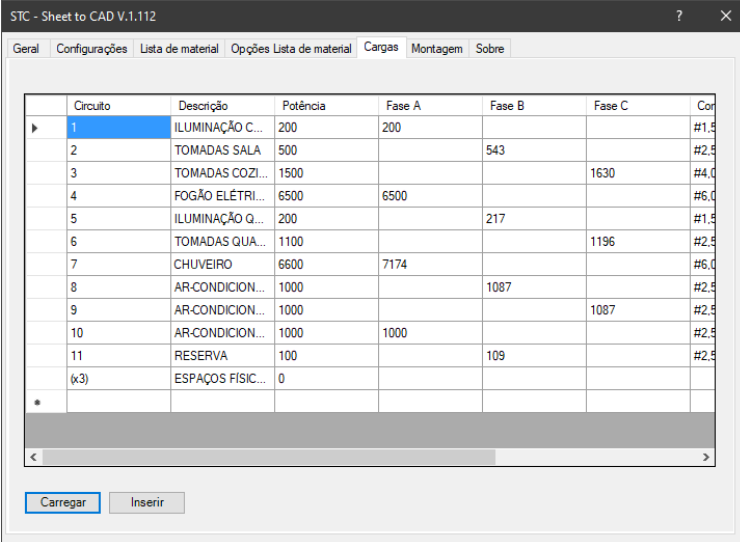
Salvar Restaurar

Figura 15 - Menu Opções Lista de Material

IMPORTAÇÃO DO QUADRO DE CARGAS

No menu “Cargas” o programa permite importar do arquivo Excel, inserido anteriormente, um quadro de cargas simplificado para que o mesmo seja inserido no espaço de desenho CAD.

Clique em “Carregar” e na sequência “Inserir” para inserir o quadro de cargas.



Circuito	Descrição	Potência	Fase A	Fase B	Fase C	Cor
1	ILUMINAÇÃO C...	200	200			#1.5
2	TOMADAS SALA	500		543		#2.5
3	TOMADAS COZI...	1500			1630	#4.0
4	FOGÃO ELÉTRI...	6500	6500			#6.0
5	ILUMINAÇÃO Q...	200		217		#1.5
6	TOMADAS QUA...	1100			1196	#2.5
7	CHUVEIRO	6600	7174			#6.0
8	AR-CONDICION...	1000		1087		#2.5
9	AR-CONDICION...	1000			1087	#2.5
10	AR-CONDICION...	1000	1000			#2.5
11	RESERVA	100		109		#2.5
(x3)	ESPAÇOS FISIC...	0				

Figura 16 - Menu quadro de cargas

INSERIR BLOCOS DO ESQUEMA DE MONTAGEM

No menu “Montagem” o programa permite importa, com base no arquivo do quadro de cargas, os blocos para fazer o esquema de montagem. Todos os blocos são importados com as informações de cada bloco previamente preenchidas.

Clique em “Gerar Montagem” e na sequência selecione um ponto na tela para gerar os blocos para o esquema de montagem.

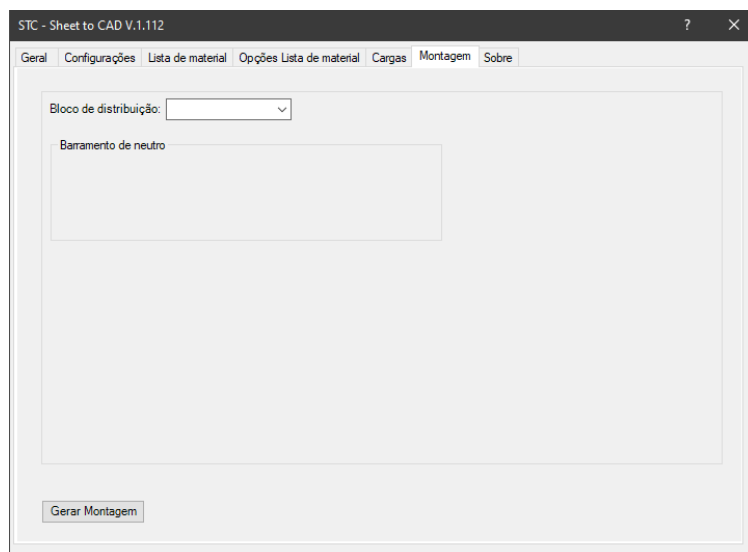


Figura 17 - Menu esquema de montagem

PREENCHIMENTO DO QUADRO DE CARGAS

IMPORTANTE: O usuário não deve jamais apagar uma coluna, caso seja feito, o programa irá gerar erros na geração do esquema unifilar.

O usuário pode inserir linhas sem problemas, conforme demonstra a Figura 18:

	A	B	C	D
1	NOME DO QUADRO:	QD-APTOS		
2				
3	CIRCUITO	DESCRIÇÃO (LOCAL)	POTÊNCIA [W]	FATOR DE POTÊNCIA
4	1	ILUMINAÇÃO	250	0,98
5	2	TOMADAS COZINHA	1200	0,98
6	3	CHUVEIRO 1	6600	0,98
7	4	TOMADAS QUARTOS, BWC E SALA	800	0,98
8	5	FOGÃO ELÉTRICO	6000	0,98
9				
10	6	AR-CONDICIONADO		
11	R1	RESERVA		
12	(5x)	ESPAÇO FÍSICO		
13				
14	DPS	DPS Classe II - 20 kA (FN)		
15	PROTEÇÃO DPS			
16	ENTRADA DE ENERGIA	GERAL		
17				
18	FATOR DE DEMANDA	0,85	13473	
19	TOTAL		15850	
20				

Figura 18 - Inserir linhas no quadro de cargas

As informações devem ser inseridas conforme descrição das colunas, em algumas células já é pré-preenchido um menu suspenso com valores padronizados.

	A	B	K	
1	NOME DO QUADRO:	QD-APTOS		
2				
3	CIRCUITO	DESCRIÇÃO (LOCAL)	CORRENTE DISJUNTOR	Icu
4	1	ILUMINAÇÃO	10 A	3
5	2	TOMADAS COZINHA	2 A	3
6	3	CHUVEIRO 1	4 A	3
7	4	TOMADAS QUARTOS, BWC E SALA	6 A	3
8	5	FOGÃO ELÉTRICO	10 A	3
9	6	AR-CONDICIONADO	16 A	3
10	R1	RESERVA	20 A	3
11	(5x)	ESPAÇO FÍSICO	25 A	3
12			32 A	3
13			40 A	3
14	DPS	DPS Classe II - 20 kA (FN)	50 A	3
15	PROTEÇÃO DPS		63 A	3
16	ENTRADA DE ENERGIA	GERAL	70 A	3
17				
18	FATOR DE DEMANDA	0,85		
19	TOTAL			
20				

Figura 19 - Demonstração menu suspenso

PREENCHIMENTO INFORMAÇÕES DR/IDR

O grupo de DR/IDR é organizado por indicações de índice, por exemplo, o grupo de DR/IDR 1 (um) deve ser indicado para cada circuito qual está incluso nesse grupo, e as informações devem ser preenchidas para cada linha de circuito, conforme demonstra a:

	A	B	R	S	T	U	V	W
1	NOME DO QUADRO:	QD-APTOS						
2								
3	CIRCUITO	DESCRIÇÃO (LOCAL)	CORRENTE DISJUNTOR	I _{cu} [kA]	NÚMERAÇÃO I/DR	QUANTIDADE DE PÓLOS I/DR	CORRENTE I/DR	FUGA I/DR
4	1	ILUMINAÇÃO	10 A	3kA	1	BIPOLAR	63 A	30 mA
5	2	TOMADAS COZINHA	25 A	3kA	1	BIPOLAR	63 A	30 mA
6	3	CHUVEIRO 1	32 A	3kA	1	BIPOLAR	63 A	30 mA
7	4	TOMADAS QUARTOS, BWC E SALA	16 A	3kA	2	BIPOLAR	63 A	30 mA
8	5	FOGÃO ELÉTRICO	32 A	3kA	2	BIPOLAR	63 A	30 mA
9	6	AR-CONDICIONADO	16 A	3kA	0			
10	R1	RESERVA	16 A	3kA	0			
11	(5x)	ESPAÇO FÍSICO		3kA	0			
12								
13	DPS	DPS Classe II - 20 kA (FN)						
14	PROTEÇÃO DPS							
15	ENTRADA DE ENERGIA	GERAL	40 A	3kA	0			
16								
17	FATOR DE DEMANDA	0,85						
18	TOTAL							

Figura 20 - Preenchimento DR/IDR

PREENCHIMENTO INFORMAÇÕES DPS, PROTEÇÃO DPS E ENTRADA DE ENERGIA

A sequência do “DPS, PROTEÇÃO DO DPS e ENTRADA DE ENERGIA” não deve ser alterada a ordem, e as informações devem ser preenchidos conforme requisição das informações presentes no cabeçalho da coluna.

IMPORTANTE: Deve ser deixado uma linha em branco entre a linha do DPS e os circuitos acima.

SOLUÇÃO DE ERROS

As etapas do plugin para inserir o esquema unifilar e demais aplicações contam com mensagens de erro quando uma operação é executada erroneamente pelo usuário, o usuário deve atentar-se ao que é dito na mensagem para corrigir o erro. Demais erros devem ser levados em consideração as operações redigidas nesse manual.

APÊNDICE B – Roteiro do teste de usabilidade

TESTE DE USABILIDADE - ETAPAS

ETAPA 1 – Alteração do quadro de cargas

Abra o arquivo de quadro de cargas (parcialmente preenchido) e adicione um circuito entre o circuito 1 e 2 existente no quadro.

O novo circuito deverá compor o grupo de IDR 1 (previamente lançado), e a numeração dos circuitos deverá ser reordenada conforme a sequência de descrição, em ordem crescente.

	A	B	C	D	E	F	G	H	I
1	NOME DO QUADRO:	QD - RESIDÊNCIA							
2									
3	CIRCUITO	DESCRIÇÃO (LOCAL)	POTÊNCIA [W]	FATOR DE POTÊNCIA	POTÊNCIA [VA]	FASE - A	FASE - B	FASE - C	I [A]
4	1	ILUMINAÇÃO COZINHA E SALA	200	0,92	217	200			0,99
5	2	TOMADAS COZINHA	1500	0,92	1630			1630	7,41
6	3	FOGÃO ELÉTRICO	6500	0,92	7065	6500			32,11
7	4	ILUMINAÇÃO QUARTOS E BWC	200	0,92	217		217		0,99
8	5	TOMADAS QUARTOS E BWC	1100	0,92	1196			1196	5,43
9	6	CHUVEIRO	6600	0,92	7174	7174			32,61
10	7	AR-CONDICIONADO SALA	1000	0,92	1087		1087		4,94
11	8	AR-CONDICIONADO QUARTO 1	1000	0,92	1087			1087	4,94
12	9	AR-CONDICIONADO QUARTO 2	1000	0,92	1087	1000			4,94
13	10	RESERVA	100	0,92	109		109		0,49
14	(x3)	ESPAÇOS FÍSICOS	0	0,92	0				0,00
15									
16	DPS	DPS Classe II - 20 kA (FN)							
17	PROTEÇÃO DPS								
18	ENTRADA DE ENERGIA	GERAL	19200			14874	1413	3913	
19									
20	FATOR DE DEMANDA	0,45	8640						
21	TOTAL		19200						

Dados do circuito a ser lançado:

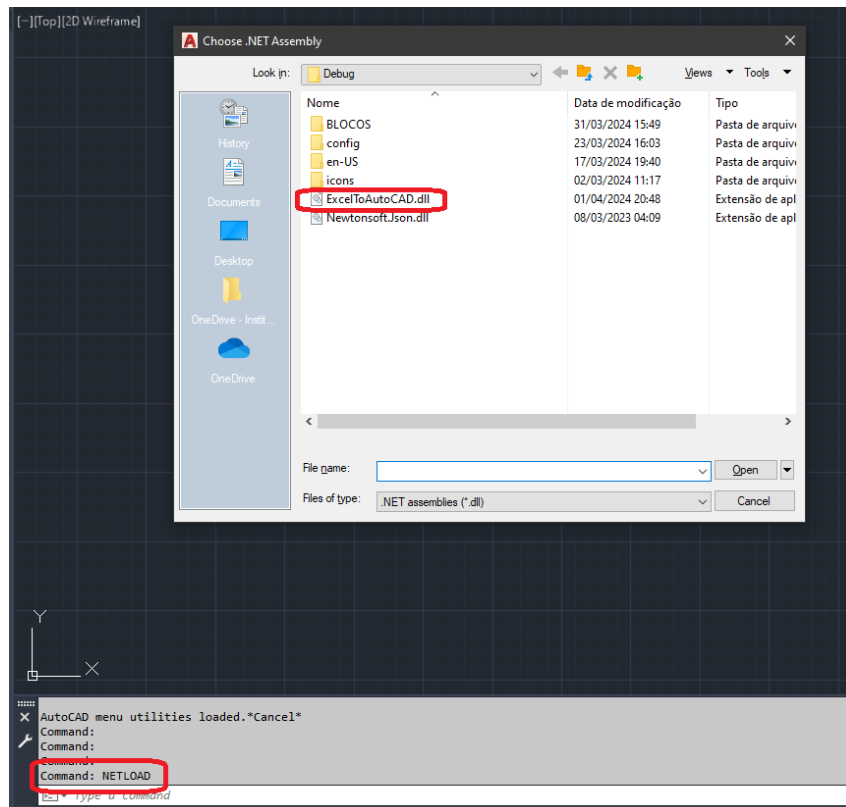
COLUNA	LINHAS
DESCRIÇÃO	TOMADAS SALA
POTÊNCIA	500 W
FASE	B
ESQUEMA	F+N+T
SEÇÃO DO CONDUTOR	#2,5
DISJUNTOR	C16 UNIPOLAR
GRUPO IDR	1

Dica: Para inserir uma nova linha clique em inserir e em seguida selecione a opção linha inteira.

CIRCUITO	DESCRIÇÃO (LO)
1	ILUMINAÇÃO COZINHA
2	TOMADAS COZINHA
3	FOGÃO ELÉTRICO
4	ILUMINAÇÃO QUARTO
5	TOMADAS QUARTO
6	CHUVEIRO
7	AR-CONDICIONADO
8	AR-CONDICIONADO
9	AR-CONDICIONADO
10	RESERVA
(x3)	ESPAÇOS FÍSICOS
DPS	DPS Classe II - 20 k
PROTEÇÃO DPS	
ENTRADA DE ENERGIA	GERAL
FATOR DE DEMANDA	0,45
TOTAL	

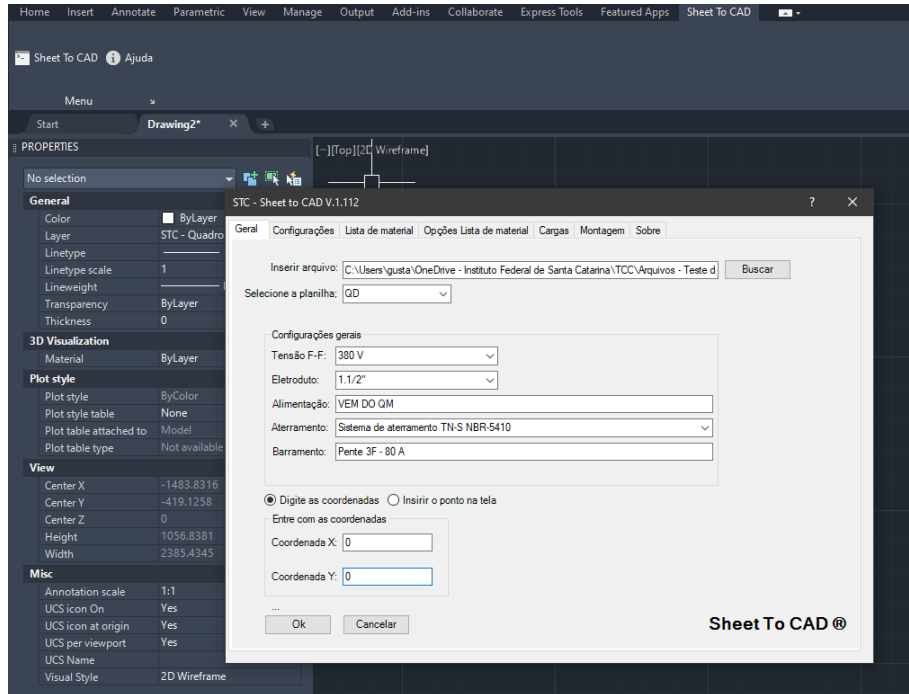
ETAPA 2 – Inicialização do plugin

Abra o AutoCAD e digite o comando “NETLOAD”, na sequência, carregue o arquivo que contém o plugin do programa.



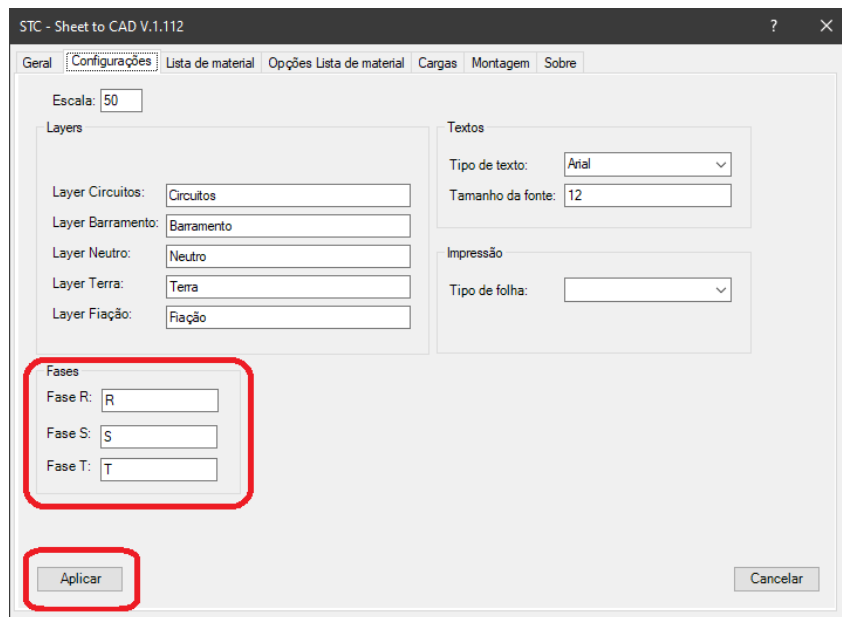
ETAPA 3 – Lançamento do esquema unifilar

Inicialize o plugin pelo comando “STC-SHEET-TO-CAD” ou pela interface do ribbon tab (menu superior) e carregue o quadro de cargas anteriormente modificado, na sequência selecione um ponto na tela para inserir o desenho ou digite uma coordenada.



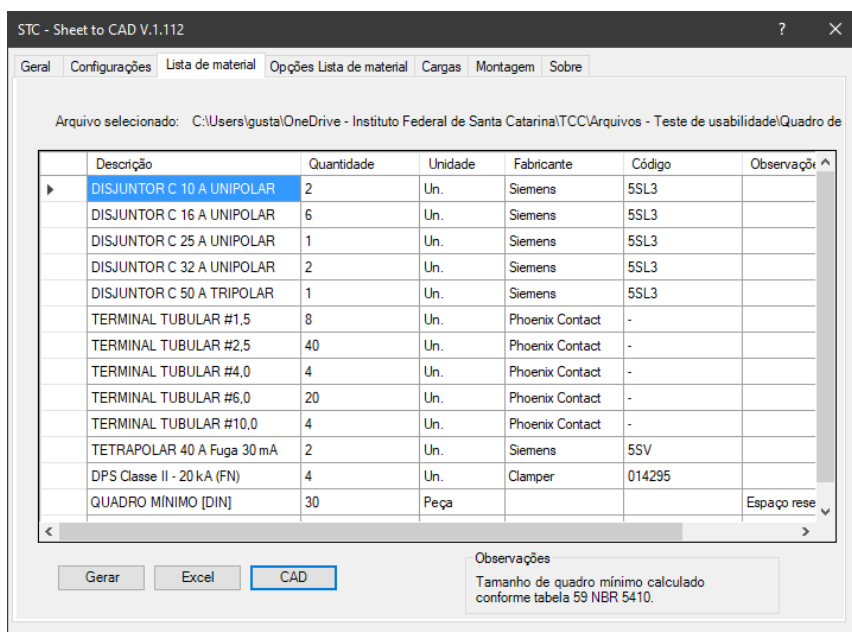
ETAPA 4 – Alterando as configurações

Acesse a aba de configurações e altere a nomenclatura das fases para R-S-T, na sequência clique em aplicar e gere novamente o esquema.



ETAPA 5 – Geração da lista de material

Na aba de Lista de material, clique em “Gerar”, após gerado a lista, clique no botão “Excel” para exportar a lista para o excel e salve o arquivo, na sequência clique em “CAD” para exportar a lista para dentro do ambiente CAD, selecione um ponto de inserção na tela.

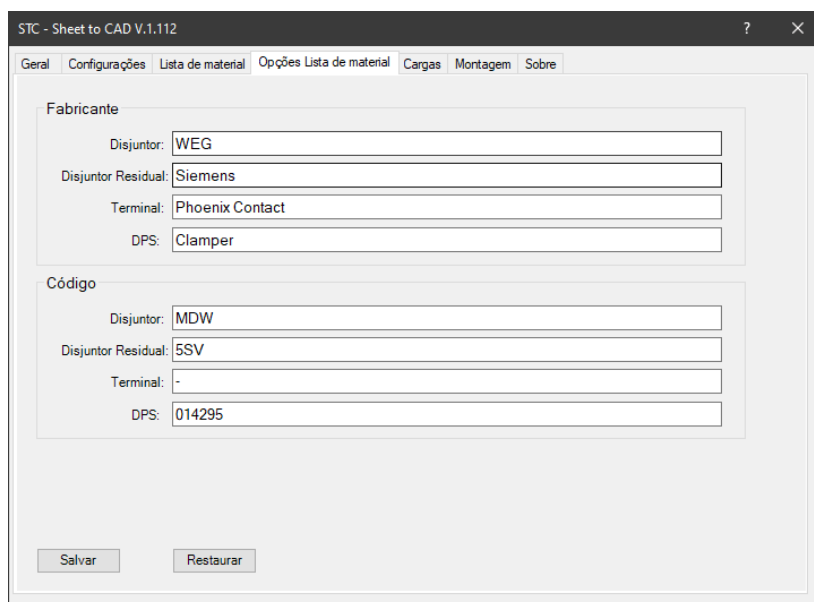


Descrição	Quantidade	Unidade	Fabricante	Código	Observações
DISJUNTOR C 10 A UNIPOLAR	2	Un.	Siemens	5SL3	
DISJUNTOR C 16 A UNIPOLAR	6	Un.	Siemens	5SL3	
DISJUNTOR C 25 A UNIPOLAR	1	Un.	Siemens	5SL3	
DISJUNTOR C 32 A UNIPOLAR	2	Un.	Siemens	5SL3	
DISJUNTOR C 50 A TRIPOLAR	1	Un.	Siemens	5SL3	
TERMINAL TUBULAR #1,5	8	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #2,5	40	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #4,0	4	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #6,0	20	Un.	Phoenix Contact	-	
TERMINAL TUBULAR #10,0	4	Un.	Phoenix Contact	-	
TETRAPOLAR 40 A Fuga 30 mA	2	Un.	Siemens	5SV	
DPS Classe II - 20 kA (FN)	4	Un.	Clamper	014295	
QUADRO MÍNIMO [DIN]	30	Peça			Espaço rese

Observações
Tamanho de quadro mínimo calculado conforme tabela 59 NBR 5410.

ETAPA 6 – Alterando as opções da lista de material

Na aba Opção Lista de Material, altere os campos de fabricante e código do componente, clique em salvar, e gere novamente a lista de material para que as aplicações sejam efetivadas.



Fabricante

Disjuntor: WEG

Disjuntor Residual: Siemens

Terminal: Phoenix Contact

DPS: Clamper

Código

Disjuntor: MDW

Disjuntor Residual: 5SV

Terminal: -

DPS: 014295

Salvar Restaurar

ETAPA 7 – Importando quadro de cargas e gerando blocos para o esquema de montagem.

Vá para a aba “Cargas” e clique em “Carregar” para importar o quadro de cargas da planilha Excel. Em seguida, clique em “Inserir” e escolha um ponto na tela para inserir o quadro de cargas.

Na sequência, vá para a aba “Montagem” e clique em “Gerar Montagem”, escolha um ponto na tela para importar os blocos de montagem.

Circuito	Descrição	Potência	Fase A	Fase B	Fase C	Cor
1	ILUMINAÇÃO C...	200	200			#1,5
2	TOMADAS SALA	500		543		#2,5
3	TOMADAS COZI...	1500			1630	#4,0
4	FOGÃO ELÉTRI...	6500	6500			#6,0
5	ILUMINAÇÃO Q...	200		217		#1,5
6	TOMADAS QUA...	1100			1196	#2,5
7	CHUVEIRO	6600	7174			#6,0
8	AR-CONDICION...	1000		1087		#2,5
9	AR-CONDICION...	1000			1087	#2,5
10	AR-CONDICION...	1000	1000			#2,5
11	RESERVA	100		109		#2,5
(x3)	ESPAÇOS FÍSIC...	0				
*						

Bloco de distribuição:

Barramento de neutro:

Gerar Montagem