UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA ELÉTRICA

JAQUELINE DA ROSA PERES

ANÁLISE E OTIMIZAÇÃO DE FERRAMENTA PARA O GERENCIAMENTO DE MACROS APLICADA EM PROJETOS ELÉTRICOS DE PAINEL DE AUTOMAÇÃO

São Leopoldo
2022

JAQUELINE DA ROSA PERES

ANÁLISE E OTIMIZAÇÃO DE FERRAMENTA PARA O GERENCIAMENTO DE MACROS APLICADA EM PROJETOS ELÉTRICOS DE PAINEL DE AUTOMAÇÃO

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica, pelo Curso de Engenharia Elétrica da Universidade do Vale do Rio dos Sinos (UNISINOS).

Orientadora: Prof.ª Dra. Ana Paula Mallmann

São Leopoldo

2022

Dedico este trabalho à minha mãe, Delurdes Gaeski da Rosa, que sempre me ensinou que o estudo mudaria minha vida.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois sua graça me alcança todos os dias e seu Amor se apresenta a mim de todas as formas.

Minha eterna gratidão à minha mãe, Lurdes, que é um exemplo de força e determinação. Obrigada pela paciência e cuidados no decorrer de todos estes anos. Sou assim, porque tu és! Te amo!

Sou grata, pois no decorrer dessa trajetória pude conhecer o amor. No campus da universidade encontrei um colega, que virou um grande amigo e hoje é meu esposo, obrigada Guilherme por andar junto comigo, pela paciência e conhecimentos compartilhados.

Sempre digo que sou uma construção, na qual cada pessoa que passou por mim contribuiu com um tijolinho. Obrigada aos diversos colegas que trilharam comigo essa longa estrada. Obrigada por ouvirem meus desabafos, enxugarem minhas lágrimas e por compartilharem comigo seus conhecimentos.

Obrigada à minha orientadora, Ana, pois te conhecer e assistir às tuas aulas foram um marco em minha vida acadêmica. Obrigada pela paciência e por não ter desistido de mim.

Agradeço por fazer parte da família Altus, pois me possibilita consolidar os conhecimentos adquiridos em sala de aula e estar em constante aprendizado.

Aos meus amigos, muito obrigada por entenderem minha ausência, por ouvirem meus desabafos, por se alegrarem comigo, vocês tornaram minha jornada mais leve.

“Até aqui nos ajudou o Senhor. 1° Samuel 7:12” (ALMEIDA, 2009).

RESUMO

Em um mundo acelerado, em que a busca constante se dá por processos ágeis e confiáveis, é necessário adequar ferramentas que atendam a estas demandas, a fim de maximizar o tempo e aumentar os lucros das companhias. Entendendo esta realidade, esse projeto buscou analisar e otimizar o uso da ferramenta *Cogineer Designer* no desenvolvimento do projeto elétrico de automação, para diminuir a quantidade de horas que o projetista emprega para a elaboração do projeto, mitigar possíveis erros de projeto e por consequência, retrabalho do projetista. O *Cogineer* é uma extensão do *software* EPLAN e foi através da biblioteca *PyAutoGUI que* o *desktop* foi automatizado, pois ela simula o uso do *mouse* e teclado do *desktop*. A ferramenta obteve um resultado satisfatório no que tange aos passos para o mapeamento da macro, porém os realizou em um tempo maior que os projetistas voluntários, foram apontadas, na análise de resultados, as possíveis situações que levaram a essa diferença no tempo de execução. Sugeriu-se, como trabalhos futuros, melhorias para que a ferramenta oferte um desempenho melhor e assim, possa ser implementada em outros processos.

Palavras-chave: EPLAN; *Cogineer*; *PyAutoGUI*; automação.

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE QUADROS

LISTA DE SIGLAS

CAGR	Taxa de Crescimento Anual Composta
CLP	Controlador Lógico Programável
COVID	Doença por Coronovírus-2019
HH	Homem-hora
RPA	Automação Robótica de Processos
TA	Tecnologia Assistiva
VPN	Rede Virtual Privada

SUMÁRIO

1 INTRODUÇÃO

Estamos vivendo um momento, cujo desenvolvimento tecnológico está acelerado e a busca é incessante para que as tarefas realizadas dentro das empresas demandem um tempo cada vez menor, de modo que a produtividade e a eficiência sejam mantidas. Entendendo este cenário, este trabalho propôs uma solução com os conceitos de termo em inglês *Robotic Process Automation* (RPA), a qual é uma tecnologia que visa automatizar tarefas frequentes nas organizações (QUALITAT GRUPO, 2018?).

Este projeto teve como objetivo analisar e otimizar o uso da ferramenta *Cogineer Designer* no setor de projeto elétrico de automação em uma empresa, no qual, primeiramente, implementou-se a automação do *desktop* e, em um segundo momento, a aplicabilidade da ferramenta foi analisada. Através da biblioteca *PyAutoGUI*, incorporada ao *Python*, foi realizada a automação do *desktop* para que ele realize o mapeamento das macros na etapa *Cogineer Designer.*

A execução de tarefas repetitivas com configurações monótonas demanda um tempo maior do projetista, podendo gerar erros que também ocasionam retrabalhos e todas estas variáveis são consideradas na proposta de venda dos projetos. Muitas vezes, o maior prazo de entrega faz com que o produto final não seja competitivo no mercado. Com a automatização destas tarefas, além de diminuir os custos de homem-hora (HH) e de mitigar os erros de projeto, o projetista terá maior disponibilidade para analisar outras demandas que envolvem o projeto elétrico, como a compra de equipamentos e a elaboração do *layout* do painel.

O *software* utilizado para o desenvolvimento dos projetos elétricos de automação é o EPLAN *Electric* P8 2022, que possui uma extensão chamada *Cogineer*. Esta ferramenta foi desenvolvida pela EPLAN para automatizar a elaboração dos projetos. No *Cogineer* há duas etapas para serem realizadas, uma é o *Designer*, onde ocorrem o mapeamento das macros de páginas ou de equipamentos e a outra é o *Project Builder*, onde o projeto é gerado.

2 FUNDAMENTAÇÃO TEÓRICA

Há alguns anos o termo RPA vem sendo muito discutido dentro das organizações devido ao fato de ser uma tecnologia criada para executar tarefas repetitivas e monótonas no ambiente empresarial, através de soluções de *software* (QUALITAT GRUPO, 2018?). Esta tecnologia possui aplicabilidade em diversos processos de negócio, tais como: folha de pagamento de funcionários, contas a receber e a pagar, processamento de faturas, gerenciamento de estoque, criação de relatórios, instalações de *software*, e também pode ser implementada em diversos setores, como: serviços financeiros, telecomunicações, energia e serviços públicos, imobiliário, entre outros (MADAKAM, 2019). Na Figura 1 pode-se verificar a diversidade de benefícios gerados pelo uso dessa tecnologia.

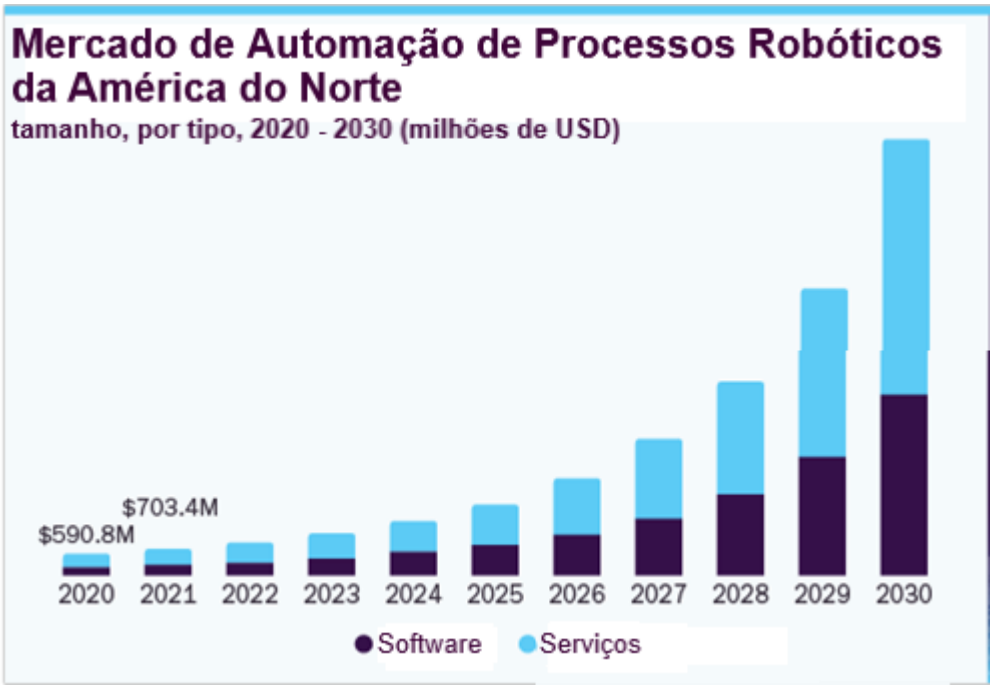
Figura 1 – Benefícios da tecnologia RPA

Fonte: BARUK (c2022?).

Devido à pandemia da Doença por Coronavírus (COVID-19) vem ocorrendo uma grande mudança nas operações das empresas e por isso, a tecnologia está evoluindo rapidamente. Consequentemente, prevê-se que a automação de processos robóticos tenha uma taxa de crescimento anual composta ou *Compound Annual*

Growth Rate (CAGR) de 38,2% até 2030. No Gráfico 1 é demonstrado a estimativa desse crescimento (GRAND VIEW RESEARCH, 2022).

Gráfico 1 – Crescimento do mercado de RPA



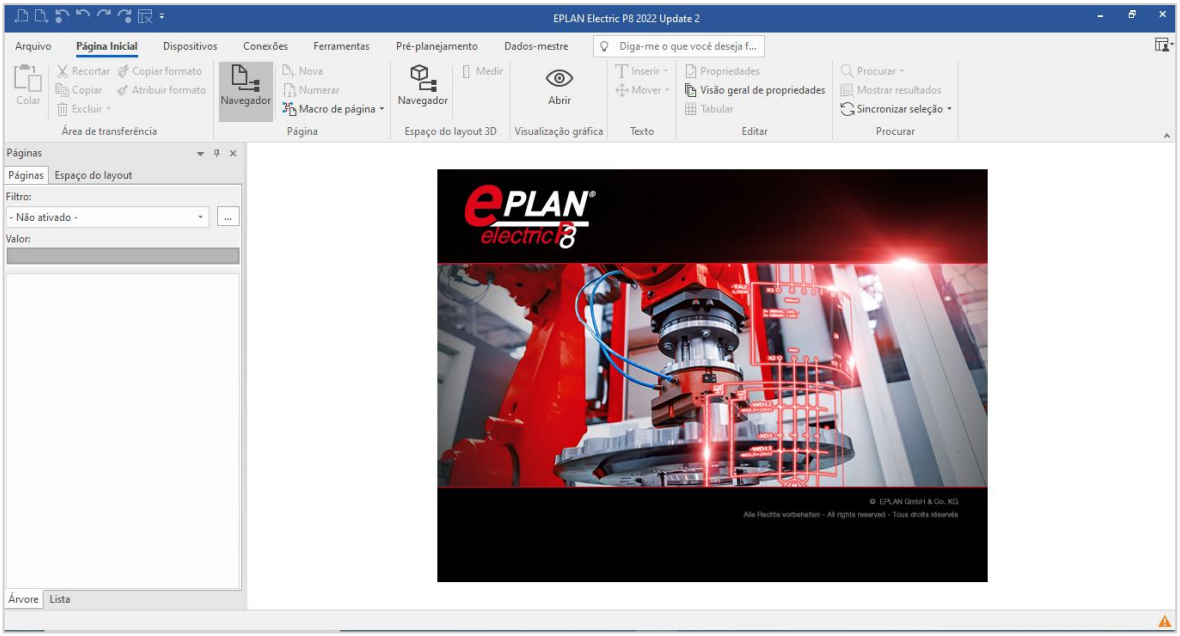
Fonte: Adaptado de Grand View Research (2022).

Existem dois modelos mais comuns de implementação da tecnologia de automação: a RPA assistida e a desassistida. Na assistida, o usuário trabalha juntamente com os robôs que automatizam as interações. Já na desassistida, o *software* robô executa as tarefas em máquinas virtuais ou servidores centrais, sem intervenção humana (QUALITAT GRUPO, 2018?).

2.1 *Software* EPLAN

Com o *software* EPLAN *Electric* P8 é possível realizar a elaboração de projetos nas áreas elétricas, de instrumentação, de controle e supervisão, entre outros. É uma ferramenta que também auxilia no planejamento e gerenciamento de projetos, pois incorporada com planilhas do Microsoft Excel, fornece relatórios que auxiliam na interpretação de dados do projetista (EPLAN, [2021d?]). A Figura 2 apresenta a área de trabalho do EPLAN *Electric* P8 2022.

Figura 2 – Área de trabalho do *software* EPLAN

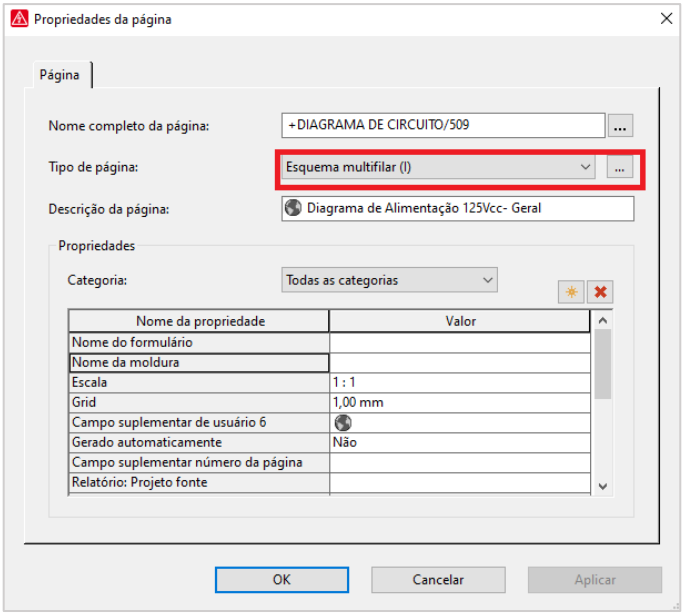


Fonte: EPLAN (2022).

O *software* possui muitas funcionalidades que facilitam a criação de desenhos esquemáticos e diagramas de circuito como a tecnologia macro, autoconexão dos equipamentos e circuitos, relatórios automáticos, entre outros (EPLAN, [2021d?]). Com a tecnologia macro pode-se criar um banco de dados com desenho de equipamentos, páginas com esquemas e diagramas elétricos que se repetem na criação do projeto (EPLAN, [2021d?]), resultando em menos esforços repetitivos.

Na elaboração do projeto, pode-se definir, na configuração da página, se o diagrama elétrico será unifilar ou multifilar, conforme é apresentado na Figura 3. No diagrama unifilar, há uma visão mais simplificada do projeto, o que facilita a verificação da quantidade e o modelo do condutor no trecho analisado. Já o diagrama multifilar representa detalhadamente toda a instalação elétrica (EPLAN, [2021c?]).

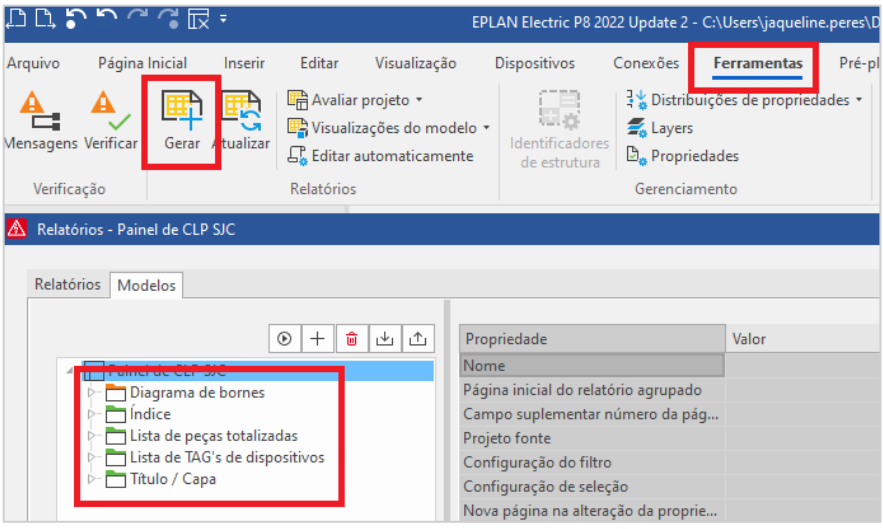
Figura 3 – Configuração das Propriedades de página



Fonte: EPLAN (2021).

Uma das diversas funcionalidades que o uso do *software* no desenvolvimento do projeto oferta é a geração automática de relatórios, como diagrama de bornes e plaquetas, que ajuda na hora da montagem do painel, assim como a lista de materiais, que facilita a aquisição dos equipamentos e insumos de montagem. Estes relatórios podem ser configurados de acordo com o critério do cliente. Na Figura 4, estão exibidos alguns modelos de relatórios que podem ser gerados no desenvolvimento do projeto elétrico.

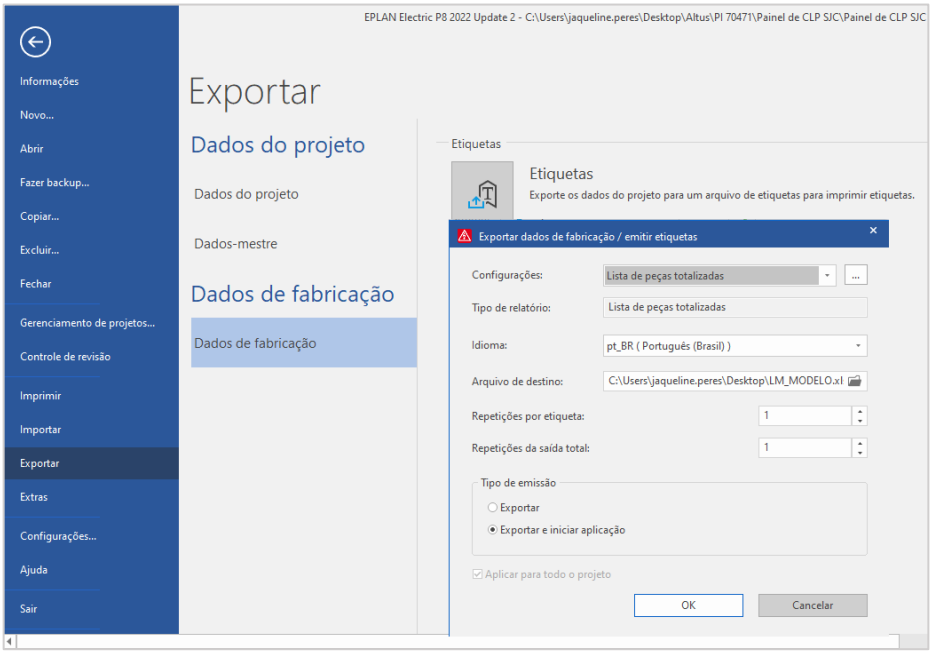
Figura 4 – Aba para gerar os relatórios



Fonte: EPLAN (2022).

Na Figura 5 está demonstrado como é configurado o software para exportação da lista de materiais do projeto.

Figura 5 – Aba para exportar lista de material

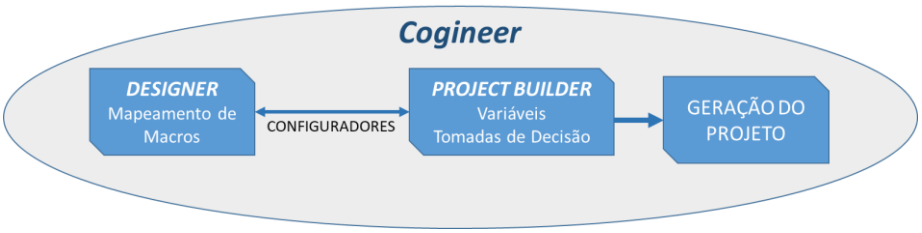


Fonte: EPLAN (2022).

2.1.1 *Cogineer*

O *Cogineer* é uma extensão do EPLAN *Electric* P8, através do qual é possível automatizar e padronizar a elaboração de projetos elétricos a partir de circuitos parciais predefinidos (macros) (EPLAN, [2021a?]). Há duas etapas de configurações principais, conforme demonstrada na Figura 6: *Designer,* cujo projetista mapeará as macros e definirá os configuradores para a próxima etapa e o *Project Builder*, que a partir das configurações definidas no *Designer* gerará a documentação do projeto (EPLAN, [2021e?]).

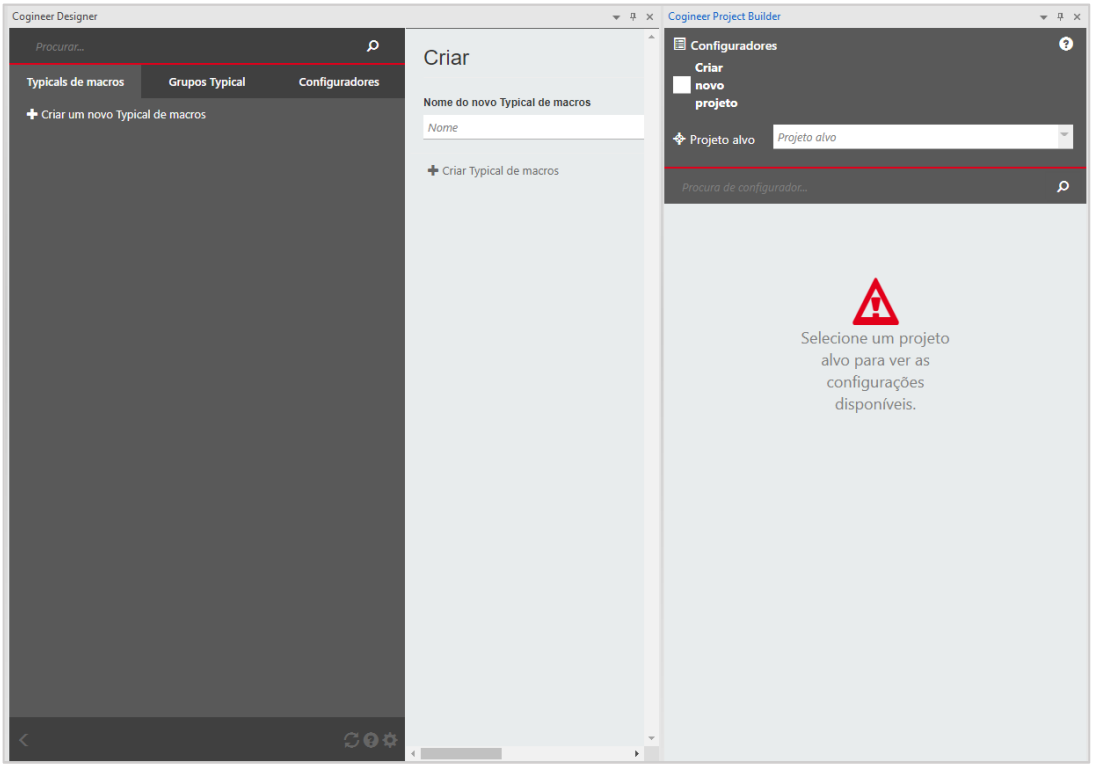
Figura 6 – Etapas das configurações do *Cogineer*



Fonte: Elaborada pela autora (2022).

A Figura 7 apresenta a interface do *Cogineer*, na qual é possível visualizar tanto o *Cogineer Designer* como o *Project Builder*.

Figura 7 – *Cogineer* na versão 2022 do EPLAN *Electric* P8



Fonte: EPLAN (2022).

2.1.1.1 *Cogineer Designer*

É no *Designer* que a estrutura de programação do projeto é realizada e nos configuradores estão todas as macros e grupos típicos que, se habilitados, são configurados no *Project Builder,* ou seja, faz a comunicação entre o *Designer* e o *Project Builder* (EPLAN, [2021e?]).

Na aba Macro-Típico cria-se condições para inserção de macros, que é uma estrutura organizacional necessária para a elaboração da documentação de projeto. Já o grupo Típico é um elemento organizacional que serve para agrupar as macros que que serão utilizadas; nele existem as Macros-Típicos ou Grupos Típicos e por fim, os Configuradores, aba onde serão configurados os acessos para a criação do projeto no *Project Builder* (EPLAN, [2021b?]. A Figura 8 aponta de modo genérico o funcionamento do *Designer*.

Figura 8 – Funcionamento no *Designer*

Fonte: Treinamento: Fundamentos de EPLAN *Cogineer* (2021).

No *Designer*, o projetista pode fazer uso de variáveis de configuração quando está no modo de edição de uma Macro-Típico ou de um Grupo Típico. Essas variáveis ficarão disponíveis no *Project Builder* e dão a possibilidade de alterar valores específicos nessa fase do projeto (EPLAN, [2021b?]. No Quadro 1, estão demonstrados os tipos de variáveis disponíveis.

Quadro 1 – Tipos de variáveis

Tipo de variável	Significado	Exemplo
Inteiro	Valores inteiros	1, 2, 3
Double	Número de ponto flutuante	3.456
String	Caractere	EPLAN
Booleano	Dois estados possíveis	Verdadeiro ou falso

Fonte: Adaptado de EPLAN ([2021a?]).

No Quadro 2, estão apontados os operadores disponíveis para a criação das configurações dentro do *Designer*.

Quadro 2 – Operadores

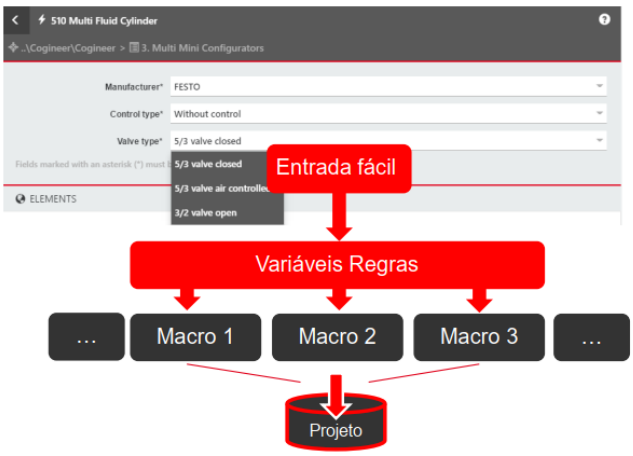
Operador	Significado	Válido para	Exemplo
+,-,*,/	Cálculo aritmético	Inteiro, double	= V1 * V2
'...'	Literal	String	= 'EPLAN'
+	Ligação de strings	String	= 'EPLAN'+' engenharia eficiente '
==, !=, <, <=, >,>=	Operadores de comparação	Inteiro, double	VERDADEIRO
not, and, or, xor	Operadores lógicos	Booleano	= (V1<=V2) e (V1>=V3)
&&, ||	Notação alternativa para operadores lógicos *e* e *ou*	Booleano	= (V1<=V2) && (V1>=V3)
if Condição **then** Valor1 **else** Valor2 **endif**	Valores condicionais	Tudo	= if V1==0 then V2 else V3 endif
ou			ou
Condição ? Valor1 : Valor2			= V1 == 0 ? V2:V3

Fonte: Adaptado de EPLAN ([2021a?]).

2.1.1.2 *Cogineer Project Builder*

Nesta etapa, o projetista gera a documentação do projeto, através dos configuradores que foram estabelecidos no *Designer* (EPLAN, [2021e?]). A Figura 9 demonstra o funcionamento do *Project Builder*.

Figura 9 – Funcionamento no *Project Builder*



Fonte: Treinamento: Fundamentos de EPLAN *Cogineer* (2021).

Nesta etapa, é necessário definir o projeto-alvo, que pode ser criado neste momento ou ser apontado o caminho em que ele está salvo. Dessa forma, como cada projeto possui seus formulários padrões é interessante ele já estar criado e nessa etapa, só ocorrerá a inserção das páginas ou equipamentos.

No próximo tópico serão apresentadas as características da linguagem de programação *Python*.

2.3 *Python*

Python é uma linguagem de programação de alto nível que oferta diversos recursos, como tipagem dinâmica e forte (onde o tipo de dado do valor deve ser do mesmo tipo da variável), orientação a objetos, multiparadigmas (programação funcional e imperativa), possui a sintaxe simples, logo o código será menor para criar, corrigir e manter. Utilizando *Python* há a possibilidade de programação *WEB*, *Desktop* e *Mobile* (MELO, [2021a?]).

Com *Python* é possível automatizar tarefas repetitivas, manipular e explorar dados através da biblioteca *PyAutoGUI* (Python DS, 2020), pois ela oferta métodos que controlam o *mouse* e o teclado e também possui funções de caixa de mensagem e de captura de tela (PyAutoGUI, 2019).

A seguir serão apresentados os trabalhos correlatos que foram utilizados como referência no desenvolvimento deste projeto.

2.4 Trabalhos correlatos

No Quadro 3, está o resumo de alguns trabalhos que foram utilizados como referência para o desenvolvimento desse projeto.

Quadro 3 – Trabalhos correlatos

Título	Ano	Autor	Relação
MARIANA - ASSISTENTE VIRTUAL PERSONALIZADO PARA LINUX *DESKTOP*	2021	SAUER, Luiz Felipe Corazza PILAN, José Rafael	Automação do teclado através da biblioteca *PyAutoGUI*
Automatização por Robô de *Software* para um Sistema Contábil	2020	CARVALHO, Maxmyller Ferreira de Freitas	Utiliza a biblioteca *PyAutoGUI* para automação do serviço contábil da Caixa Econômica Federal
Desenvolvimento de uma interface de mouse para pessoas com tetraplegia	2018	FRANÇA, Fábio F. *et al*	Fez uso da biblioteca *PyAutoGUI* para controlar o *mouse* e teclado.

Fonte: Elaborado pela autora (2022).

O trabalho de Sauer e Pilan (2021) apresentou o desenvolvimento de um Assistente Virtual, intitulado de Mariana, controlado por voz e implementado em sistemas operacionais Linux baseados em Ubuntu, uma das ferramentas utilizadas para a realização dessa automação, foi a biblioteca *PyautoGui,* assim como este trabalho apresenta. Mariana foi programada para realizar tarefas frequentes, como: realizar pesquisas, abrir *sites*, reconhecer e replicar falas em *softwares* de escritas, assim como reconhecer nomes de *sites* para abri-los, manipular janelas abertas e atuar no monitoramento de recursos do sistema.

Carvalho (2020) teve como objetivo criar um robô de *software* para ser instituído na empresa BWA Global, que realiza o serviço contábil chamado “SE-FIP-Conectividade Social” da Caixa Econômica Federal. Devido à crise econômica decorrente da pandemia da Covid-19, a empresa perdeu muitos colaboradores, o que motivou Carvalho a apresentar a ideia de automação de tarefas repetitivas. Entre algumas bibliotecas do *Python* utilizadas, Carvalho fez uso da *PyautoGui* para realizar a automação das interações com teclado e *mouse*, como utilizadas neste projeto.

O projeto do grupo formado por França et *al.* (2018) teve como objetivo principal a criação de uma Tecnologia Assistiva (TA), que busca possibilitar o uso de computadores por pessoas com tetraplegia, através de uma interface de interação. O grupo utilizou a biblioteca *PyautoGUI*, pois fornece métodos para controle do *mouse* e teclado, possibilitando assim, simular o *mouse* para criar a aplicação com o voluntário.

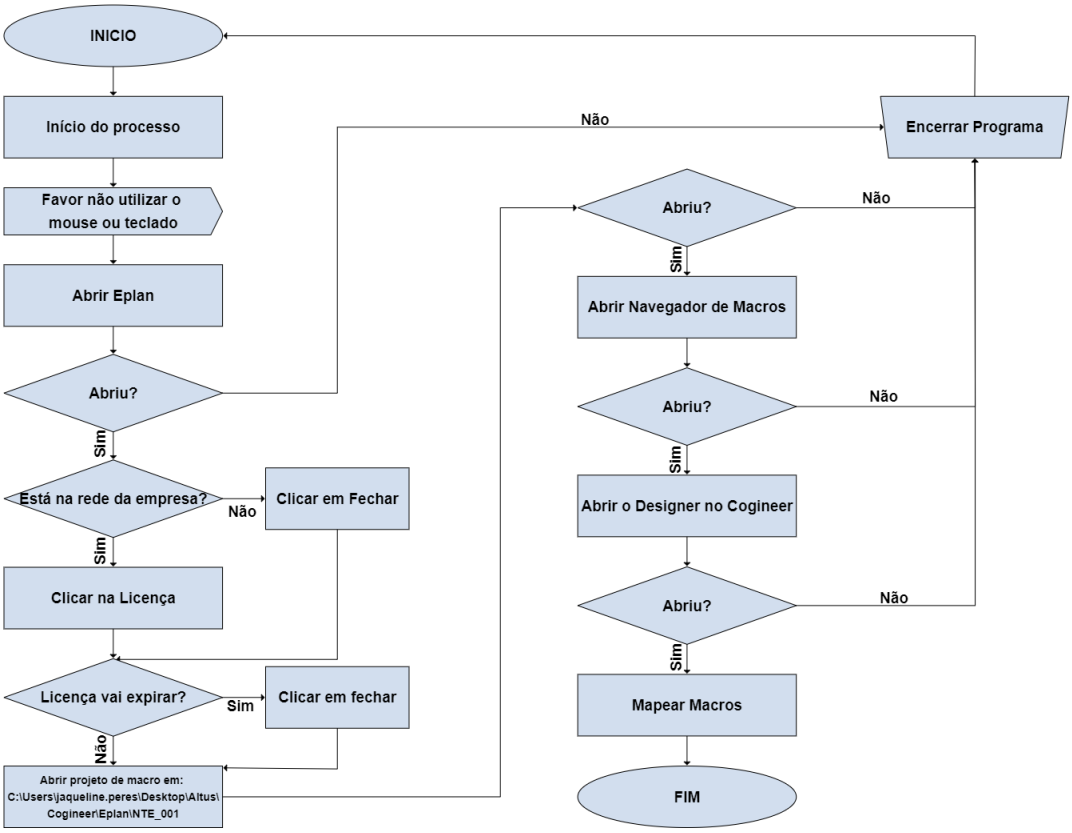
3 METODOLOGIA

Este trabalho foi desenvolvido para atender os objetivos propostos inicialmente, que são: a automação do *desktop* para gerar a redução dos custos de homem-hora (HH) no desenvolvimento do projeto elétrico, assim como mitigar possíveis erros de projetos.

Através da biblioteca *PyautoGUI*, o *desktop* foi automatizado, a fim de mapear as macros de página na etapa *Designer* dentro da extensão *Cogineer* do *software* EPLAN. Ao executar esse mapeamento, cria-se um banco de dados de configuradores disponíveis no *Project Builder* para uso em projetos futuros, que demandem o mesmo padrão de macro.

Primeiramente, foi gerado um fluxograma com as etapas macros que o programa executa, conforme indica a Figura 10.

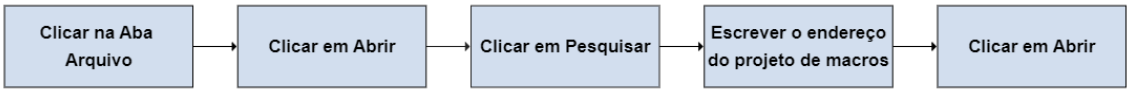
Figura 10 – Fluxograma do projeto proposto



Fonte: Elaborada pela autora (2022).

Visto que, para a correta execução do programa todos os passos que a máquina realiza precisam estar mapeados, criou-se um diagrama de blocos de cada etapa do fluxograma. Na Figura 11, estão apontados os passos para abrir o projeto de macros já criado anteriormente.

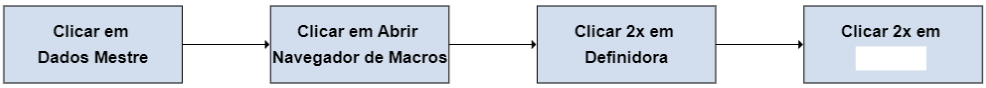
Figura 11 – Passos para abrir o projeto de macros



Fonte: Elaborada pela autora (2022).

Na Figura 12, estão demonstrados os passos para abrir o navegador de macros, que é o local onde ficam visíveis as macros de páginas para serem mapeadas no *Designer*.

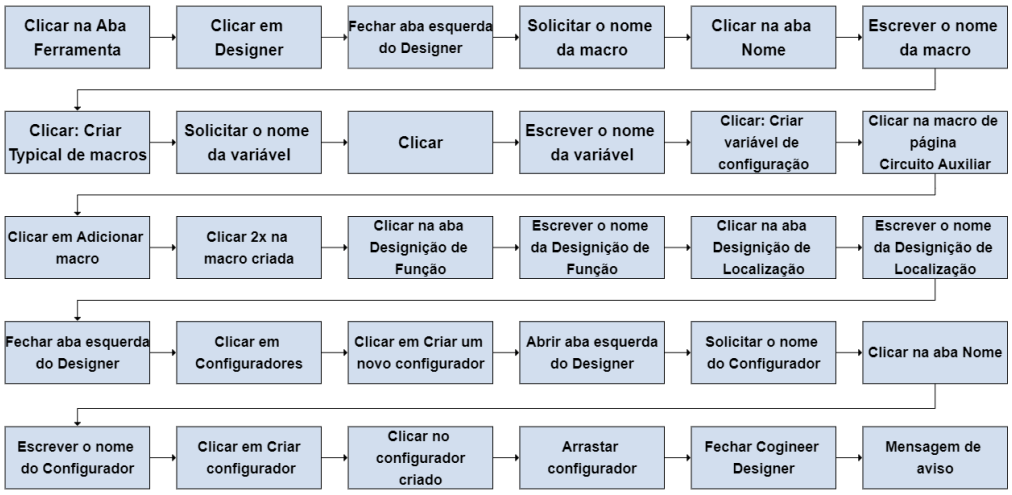
Figura 12 – Passos para abrir o navegador de macros



Fonte: Elaborada pela autora (2022).

Na Figura 13 estão exibidos os passos para abrir o *Cogineer Designer* e mapear as macros.

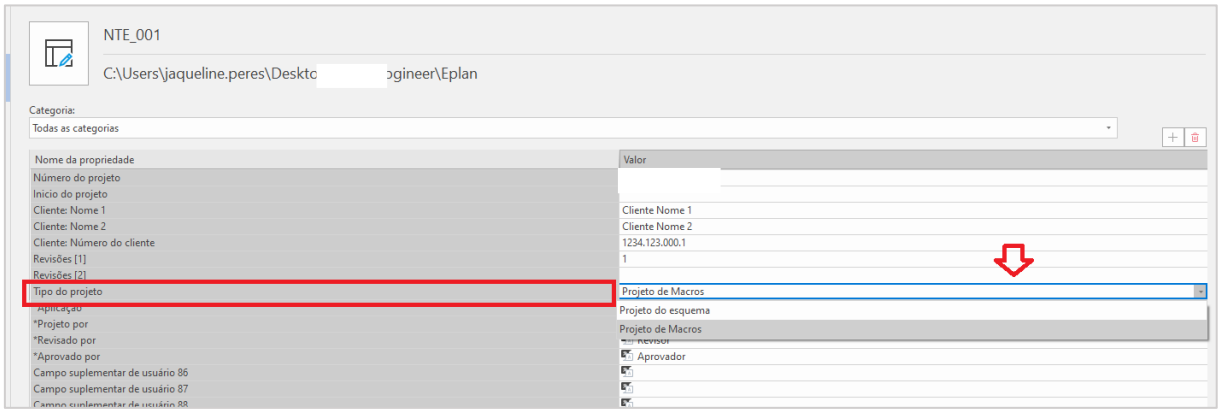
Figura 13 – Passos para abrir o *Cogineer Designer* e mapear as macros



Fonte: Elaborada pela autora (2022).

Existe um projeto padrão, chamado de NTE_001, no qual estão inseridas as páginas modelos para a elaboração do projeto elétrico, como: as páginas de alimentação do circuito auxiliar e equipamentos, assim como as páginas que estão representados os controladores lógicos programáveis (CLP's). Na sequência, este projeto padrão foi configurado para um projeto de macros, como a Figura 14 demonstra.

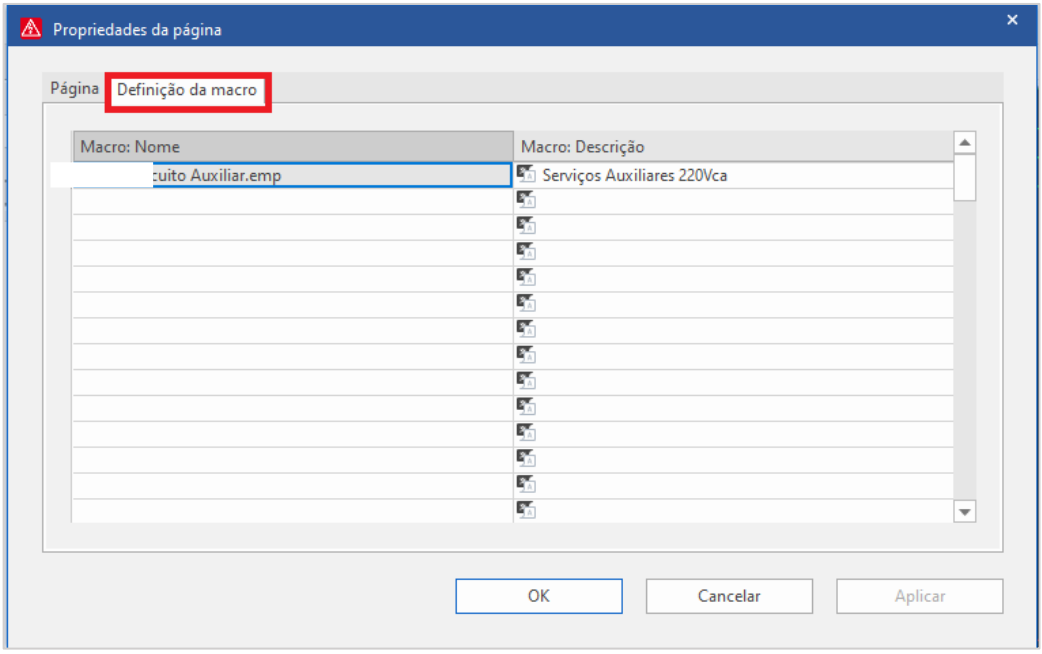
Figura 14 – Configuração para um projeto de macros



Fonte: EPLAN (2022).

A segunda etapa foi gerar as macros de páginas que foram utilizadas na etapa *Designer*, como é demonstrado na Figura 15.

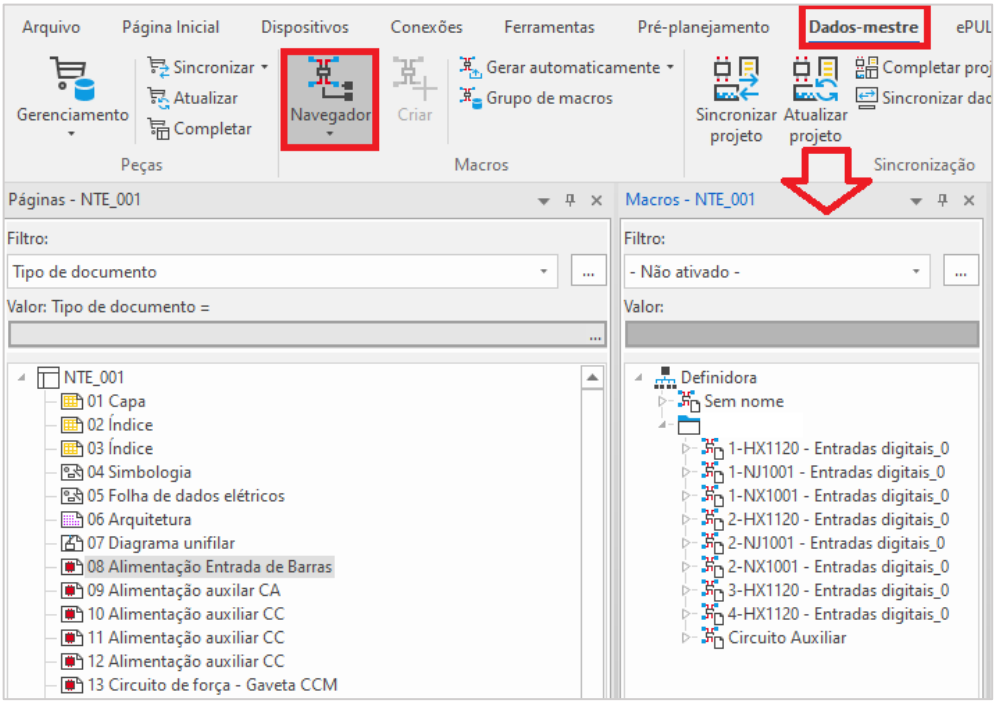
Figura 15 – Configuração da macro de página



Fonte: EPLAN (2022).

A descrição inserida na aba *Macro: Nome* é visualizada no navegador de macros, que é uma das etapas utilizadas para o mapeamento de macros no *Designer*, exibido na Figura 16.

Figura 16 – Navegador de macros



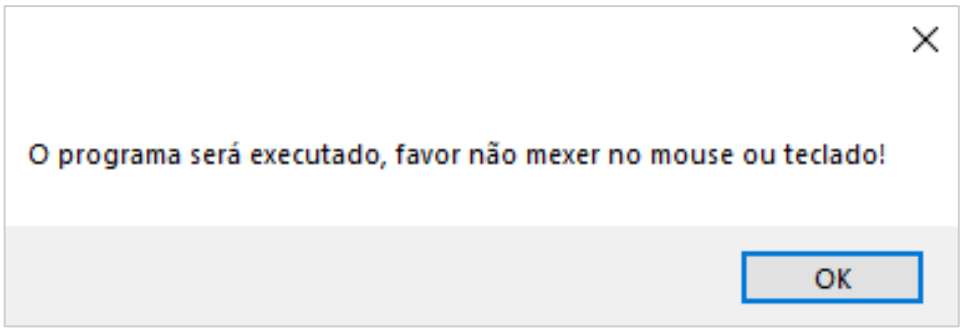
Fonte: EPLAN (2022).

Depois das configurações concluídas, passou-se à criação da ferramenta, cuja linguagem de programação escolhida para o desenvolvimento foi a *Python*, devido a sua sintaxe simples e de fácil aprendizado, assim como as suas inúmeras bibliotecas disponíveis, que auxiliam na elaboração do programa.

A principal biblioteca utilizada nesse projeto foi a *PyautoGUI*, porque ela é capaz de simular todas as ações que um usuário humano executa. Com o controle do *mouse* e do teclado, as tarefas repetitivas são automatizadas.

Quando o programa é iniciado, uma mensagem é exibida para o usuário humano, conforme Figura 17. Este cuidado evita a geração de conflitos e erros no interpretador.

Figura 17 – Mensagem para o usuário



Fonte: Elaborada pela autora (2022).

Através do módulo *os* foi utilizado o método *os.startfile*() para abrir o *software* EPLAN. Este método possibilita que um arquivo com seu programa associado inicie, ou seja, foi apontado o endereço do local do *software* na máquina para o *os.startfile,* que sinaliza para abri-lo.

Este projeto também fez uso do módulo *time* para uso da função *sleep*, pois ela suspende a execução do *thread* atual por um determinado tempo em segundos. *Threads* são linhas de execução concorrentes no programa, ou seja, a função *sleep* atrasa uma linha até que a anterior esteja concluída.

Como pode ser visto nos diagramas de blocos, demonstrados anteriormente, as principais funções de controle de *mouse* necessárias para o desenvolvimento do programa foram as funções *click*() e *doubleClick*(), onde a função *click*() reproduz um único clique com o botão esquerdo do mouse na sua posição atual e a *doubleClick*() reproduz dois cliques. Já a principal função de controle de teclado utilizada foi a *write*(), a qual digita os caracteres descritos no programa ou informados pelo usuário humano, fazendo uso das seguintes funções com *pyAutoGUI*: *pyautogui.click*(); *pyautogui.doubleClick*(); *pyautogui.write*().

Optou-se pelo uso de imagens das telas, pois ao estudar a sequência de telas que abrem, verificou-se que elas poderiam ficar levemente fora do local inicial. Logo, toda vez que o programa rodasse, seria necessário encontrar as coordenadas das imagens.

A biblioteca *PyAutoGUI* oferta uma função de localização de imagens, a *locateOnScreen*(), que foi muito utilizada neste projeto, devido à sua facilidade de uso. Ela localiza na tela do *desktop* a imagem passada como parâmetro quando for chamada.

A função localização *locateOnScreen*() realiza a verificação das imagens das telas, porque para a execução de determinado passo, é necessário o uso de uma tela específica, ou seja, a função é chamada com a imagem da tela que precisa verificar se está aberta. Se ao finalizar a verificação a tela foi encontrada, o programa executa a próxima linha do código, senão retorna à mensagem *ImageNotFoundException* e encerra o programa.

A função *locateOnScreen*() foi utilizada para adquirir as coordenadas da tela, pois ela retorna uma tupla de 4 inteiros (esquerda, superior, largura, altura). Em seguida, ela foi passada para a função *center*(), a fim de obter as coordenadas X e Y no centro do local em que há necessidade do clique (PyAutoGUI, 2019).

Tuplas são objetos no qual conseguimos armazenar informações e são comparadas às listas, porque possuem características similares, porém as listas são estruturas de dados que armazenam informações homogêneas e mutáveis, ou seja, elas possuem o mesmo tipo e significado e esses valores podem sofrer alterações. As tuplas armazenam informações heterogêneas e imutáveis, ou seja, elas podem apresentar tipo e significado diferentes e esses valores não podem ser alterados (ORESTES, 2018).

Criou-se a função denominada *abreImagemeClica*(), pois verificou-se que o mesmo trecho de código iria se repetir diversas vezes para atender os passos necessários na automação do *desktop*. Na Figura 18, está sendo apresentada a função gerada.

Figura 18 – Função *abreImagemeClica*

```
5  def abreImagemeClica(caminho):
6      locateImage = pyautogui.locateOnScreen(caminho)
7      locateImage2 = pyautogui.center(locateImage)
8      locateImageX, locateImageY = locateImage2
9      pyautogui.click(locateImageX, locateImageY)
```

Fonte: Elaborada pela autora (2022).

Houve casos em que foi necessário um duplo clique e para isso, gerou-se uma segunda função denominada *abreImagemeClica1*, cuja diferença está na solicitação dos dois cliques no local estipulado, como demonstrado na Figura 19.

Figura 19 – Função *abreImagemeClica1*

```
11 def abreImagemeClical(caminho):
12     locateImage = pyautogui.locateOnScreen(caminho)
13     locateImage2 = pyautogui.center(locateImage)
14     locateImageX, locateImageY = locateImage2
15     pyautogui.doubleClick(locateImageX, locateImageY)
```

Fonte: Elaborada pela autora (2022).

A seguir, é descrito como foram realizados os passos de cada etapa da automação do *desktop*.

3.1. Abrir o *software* EPLAN

Essa seção aborda os passos que o *desktop* realiza para abrir o *software* EPLAN, no qual, após ele ser iniciado, podem ocorrer duas situações: uma delas é máquina não estar conectada na *Virtual Private Network* (VPN) da empresa e nesse caso, uma mensagem de aviso ao usuário é disparada. Para atender a essa possível situação, criou-se uma variável que, através da função *locateOnScreen*(), localiza na tela a imagem passada como parâmetro. Se a imagem for localizada, a função *abreImagemeClica*() é chamada para fechar a mensagem de aviso. Uma vez a máquina estando conectada à VPN da empresa, o programa passa a executar a próxima linha de instrução.

Para o projetista elaborar o projeto elétrico é necessário o uso de uma licença do *software*. Essas licenças podem ser utilizadas tanto na rede da empresa, como podem ser emprestadas por até 60 dias para cada máquina, possibilitando assim, o trabalho local.

Caso o projetista opte pelo empréstimo da licença para a máquina local, a segunda situação se consolida, na qual dez dias antes de expirar essa licença, o usuário passa a receber uma notificação quando abre o *software*. Para resolver essa situação, a função *locateOnScreen*() certificará se mensagem foi exibida para o usuário, ao ser confirmada, a função *abreImagemeClica*() é chamada para fechar a notificação. Caso a mensagem não tenha sido disparada, o programa passa a executar a próxima linha de instrução.

Na próxima seção, serão abordados os passos realizados para a abertura de um projeto de macros.

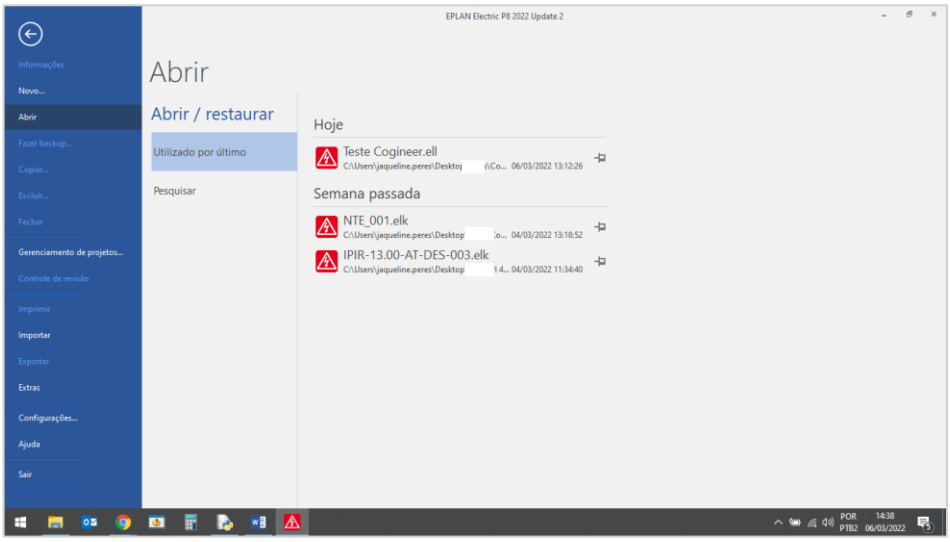
3.2. Abrir o projeto de macros

Depois do *software* EPLAN aberto, a próxima etapa foi abrir o projeto de macros, pois nele, estão vinculadas as macros de página.

Primeiramente, através da função *locateOnScreen*() é verificado se o EPLAN realmente foi aberto, cujo parâmetro da função é a imagem da área de trabalho do *software*. Sequencialmente, a função *abreImagemeClica*() é chamada e neste trecho do código, o parâmetro da função é a imagem da aba Arquivo.

A fim de garantir que a aba Arquivo tenha sido aberta, a função *locateOnScreen*() é chamada para localizar a imagem da Figura 20, possibilitando assim, que o programa passe a executar os próximos passos.

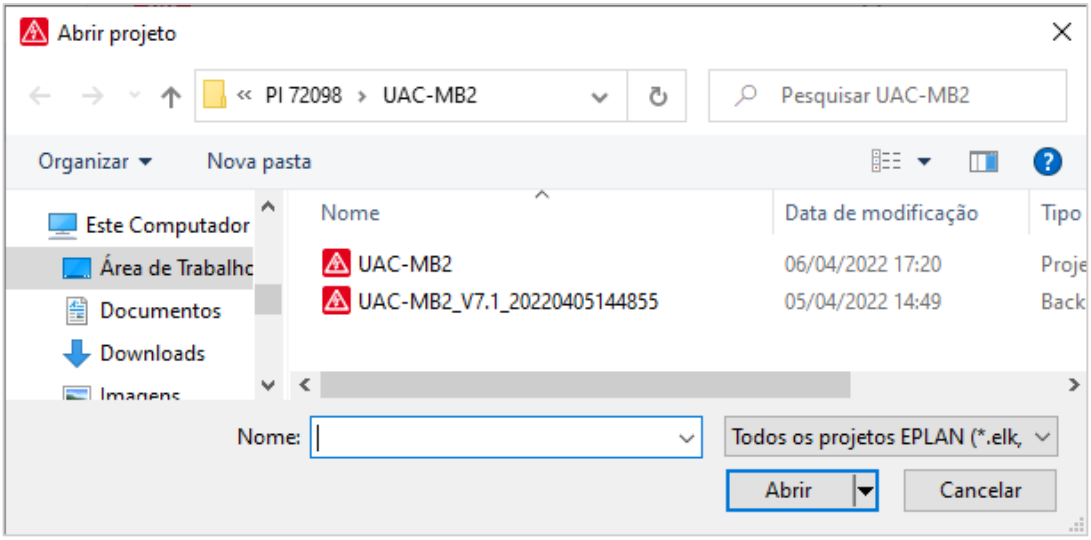
Figura 20 – Aba arquivo



Fonte: EPLAN (2022).

Posteriormente, a função *abreImagemeClica*() é chamada duas vezes, a fim de abrir a tela Pesquisar, conforme Figura 21, possibilitando assim, a abertura do projeto de macros.

Figura 21 – Tela para abertura do projeto de macros



Fonte: EPLAN (2022).

Com a tela Pesquisar aberta, a função *write*() é chamada e o parâmetro passado a ela é o endereço na máquina onde o projeto de macros foi salvo. Este endereço é constante, ou seja, ele é imutável. Após o endereço ser escrito na aba Nome, a função *abreImagemeClica*() é acionada para realizar o clique no botão Abrir, disponibilizando assim, o projeto no navegador de Páginas.

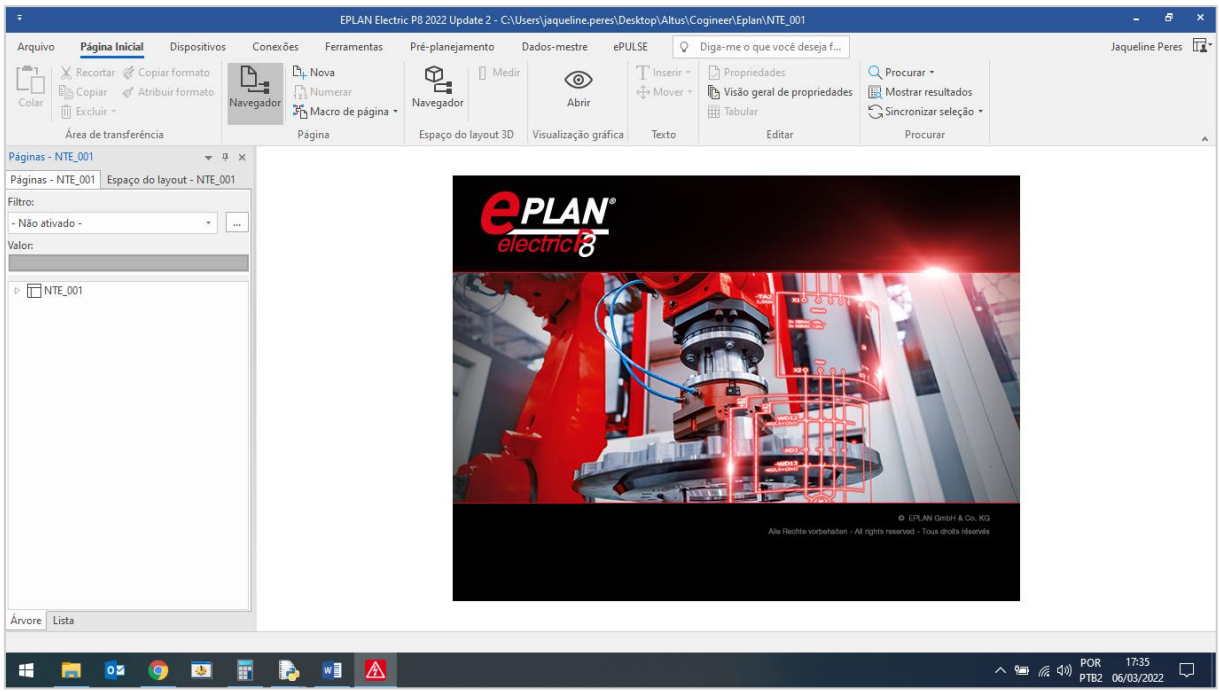
A seção a seguir, descreve os passos para abrir o navegador de macros no *software* EPLAN.

3.3. Abrir o navegador de macros

Posteriormente ao projeto de macros aberto, é necessário que o programa realize a abertura do navegador de macros, pois nele ficam exibidas as macros de páginas disponíveis para mapeamento.

Inicialmente, a função *locateOnScreen*() verificou se o projeto de macros realmente foi aberto. Esta verificação ocorreu através do parâmetro passado a ela, como é demonstrado na Figura 22.

Figura 22 – Tela de verificação para a abertura do navegador de macros

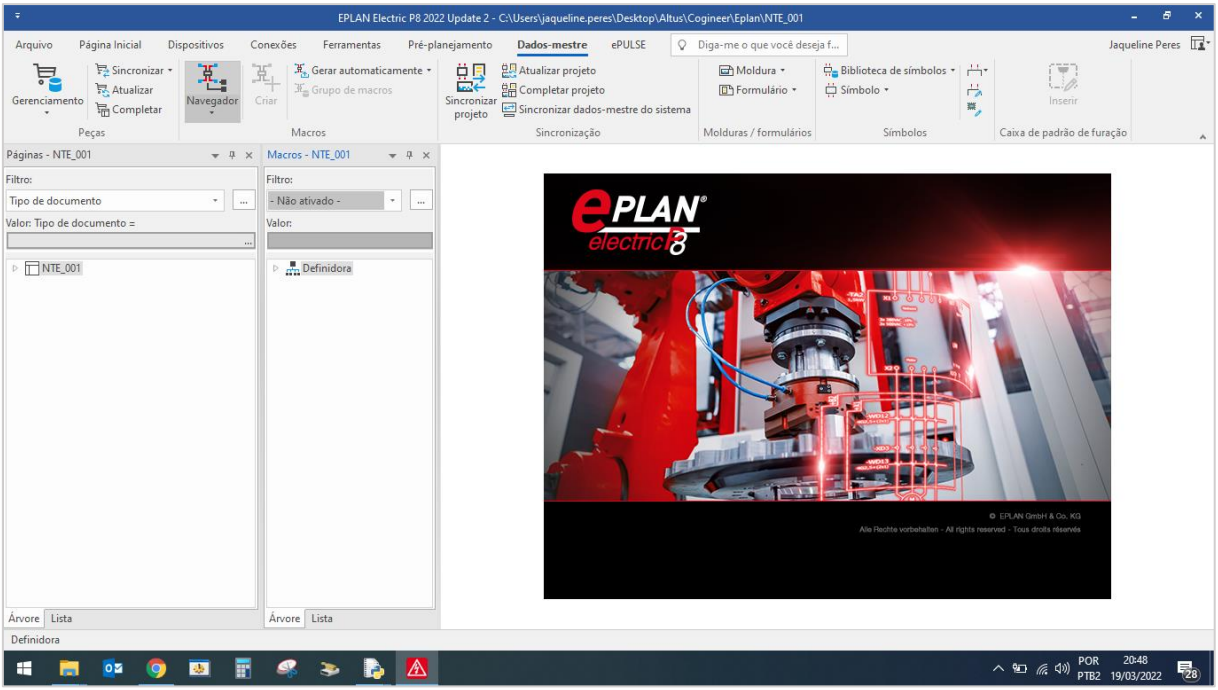


Fonte: EPLAN (2022).

Depois de confirmado que o projeto de macros foi aberto, a função *abreImagemeClica*() é chamada para abrir a aba Dados Mestre e sequencialmente, é verificado se a aba Dados Mestre foi aberta.

Com a aba Dados Mestre aberta a função *abreImagemeClica*() é solicitada para que realize a abertura do navegador de macros. Em seguida, a função *locateOnScreen*() verifica através da imagem passada como parâmetro se o navegador foi aberto, cuja Figura 23 exibe a imagem da tela passada como parâmetro.

Figura 23 – Navegador de macros



Fonte: EPLAN (2022).

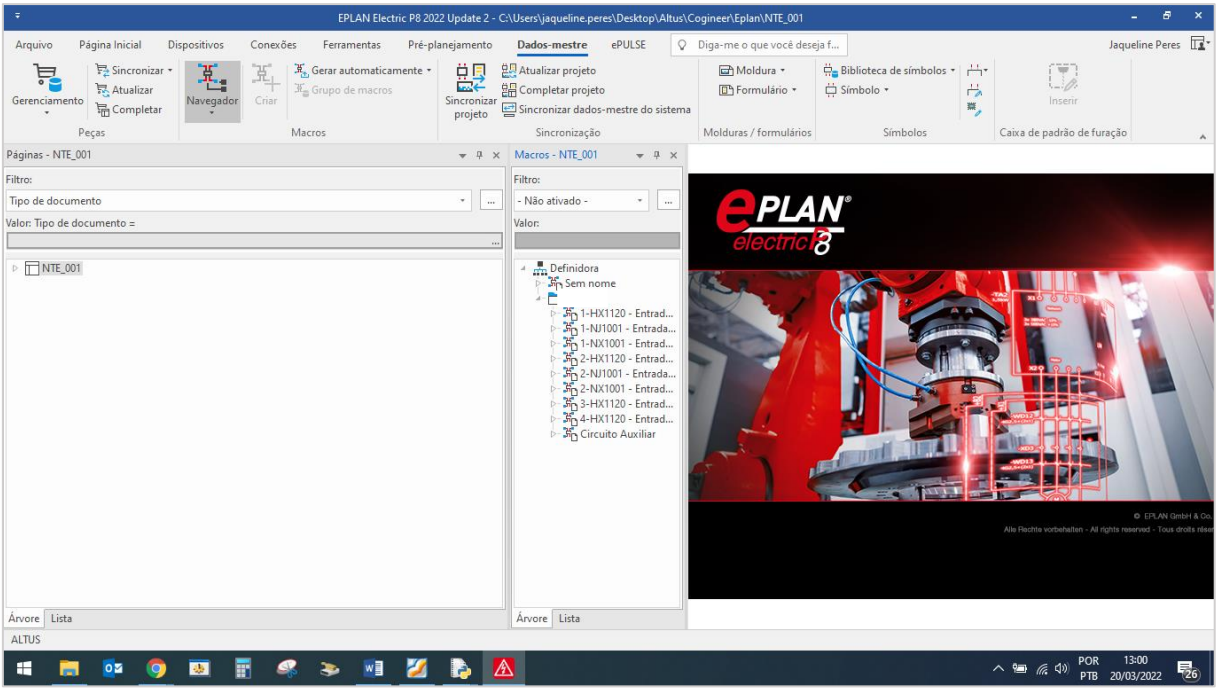
Depois de constatado que o navegador de macros foi aberto, a função *abreImagemeClica1*() é chamada duas vezes, sequencialmente, para deixar visível no navegador as macros de páginas que estão disponíveis no projeto de macros.

A próxima seção aborda como a extensão *Cogineer Designer* foi aberta.

3.4. Abrir a extensão *Cogineer Designer*

Essa seção demonstra os passos executados pelo *desktop* para abertura da extensão *Coginner Designer*, onde primeiramente, através da função *locateOnScreen*(), verificou-se que as macros de páginas estavam sendo exibidas no navegador de macros, conforme demonstra a Figura 24.

Figura 24 – Exibição das macros de páginas



Fonte: EPLAN (2022).

Posteriormente, a confirmação de que o navegador de macros foi aberto, a função *abreImagemeClica*() é chamada duas vezes para abrir o *Cogineer Designer*.

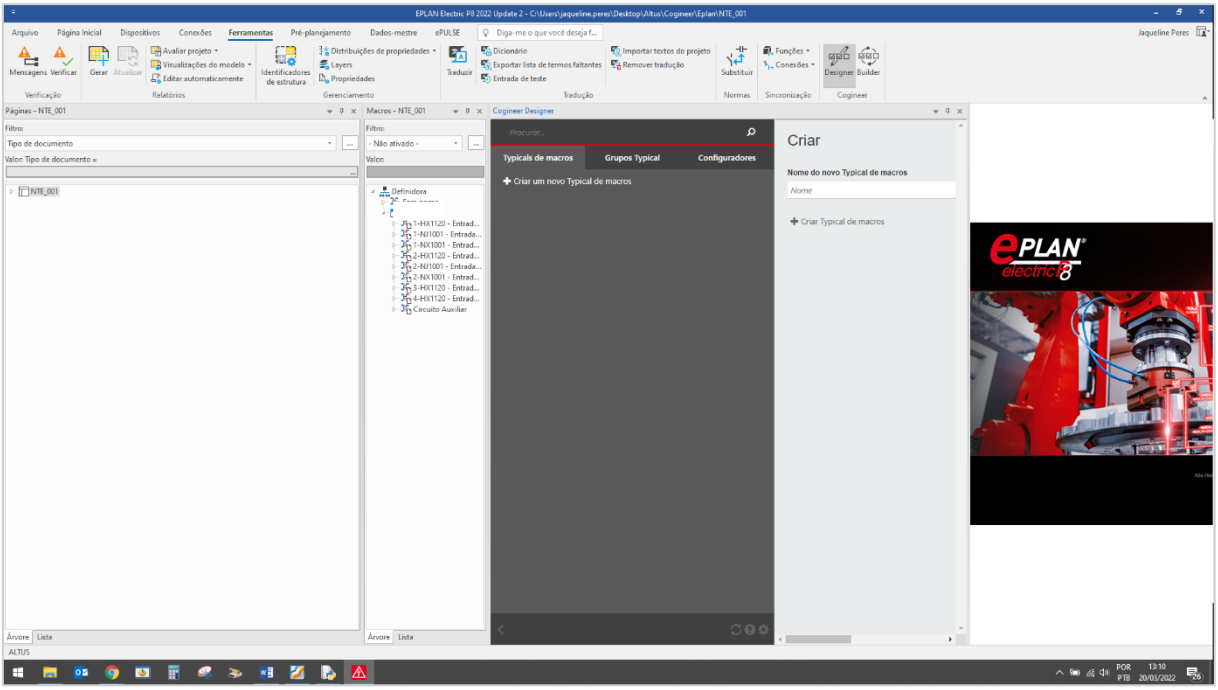
Na próxima seção, será abordado como ocorre o mapeamento das macros. Ressalta-se, sobretudo, que este projeto prevê o mapeamento de uma macro de páginas, intitulada de Circuito Auxiliar.

3.5. Mapear a macro

Essa seção relata os passos que a máquina executa para mapear a macro Circuito Auxiliar. Esse projeto estudou a inserção de apenas uma macro para, posteriormente, aplicar estes passos para macros de páginas dependentes, por exemplo, em um modelo de CLP pode possuir até quatro macros de páginas.

Para realizar os passos do mapeamento, primeiramente a função *locateOnScreen*() verificou se realmente a extensão *Cogineer Designer* foi aberta no EPLAN. A Figura 25 exibe a tela que a função recebeu como parâmetro.

Figura 25 – Tela do *Cogineer Designer* aberta no *software* EPLAN

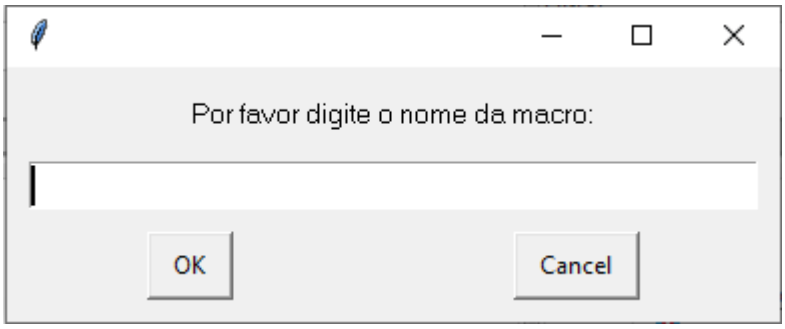


Fonte: EPLAN (2022).

Após o programa constatar que o *Cogineer Designer* foi aberto, a função *abreImagemeClica*() é chamada para fechar a tela lateral esquerda do *Designer*, disponibilizando assim, a tela para criar macros.

O próximo passo foi solicitar ao usuário humano para digitar o nome da macro, conforme exibido na Figura 26, que é armazenado na variável *nomemacro*.

Figura 26 – Escolha do nome da macro pelo usuário

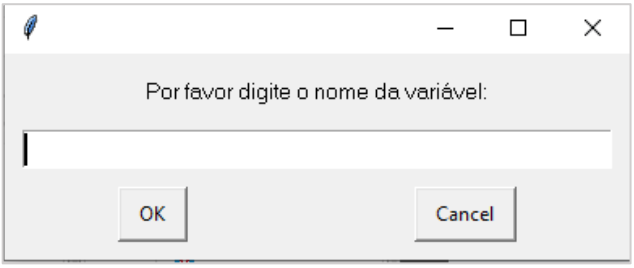


Fonte: Elaborada pela autora (2022).

Depois que a tela lateral foi minimizada, e verificada através da função *locateOnScreen*(), a função *abreImagemeClica*() é chamada para abrir a aba Nome

do novo Typical de macros e em seguida, a função *write*() é solicitada para escrever o nome da macro digitada pelo usuário anteriormente. Após a máquina digitar o nome da macro, a função *abreImagemeClica*() implementou o nome da macro que estará disponível no Configurador e em seguida, solicitou ao usuário humano para digitar o nome da variável, como é exibido na Figura 27.

Figura 27 – Escolha do nome da variável pelo usuário

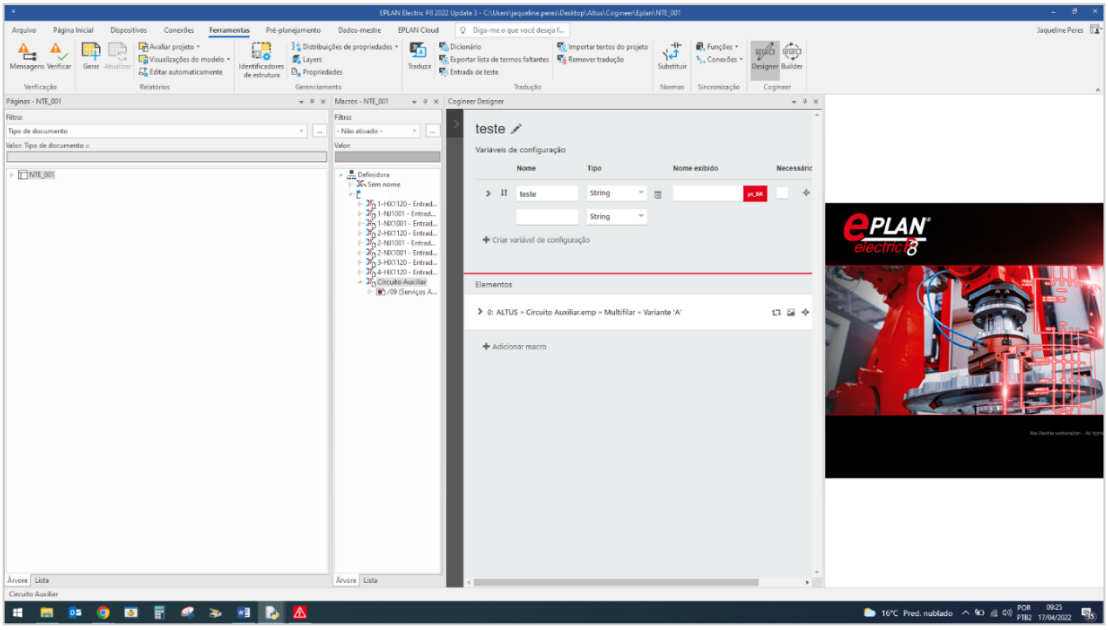


Fonte: Elaborada pela autora (2022).

Ao finalizar a interação com o usuário humano, a função *abreImagemeClica*() é chamada três vezes, sequencialmente, para realizar os passos, a fim de adicionar a macro de página Circuito Auxiliar.

Para verificar se a macro foi anexada como esperado, a função *locateOnScreen*() verifica se a tela passada como parâmetro foi aberta no *software* EPLAN, conforme demonstrado na Figura 28.

Figura 28 – Tela com a macro Circuito Auxiliar anexada



Fonte: EPLAN (2022).

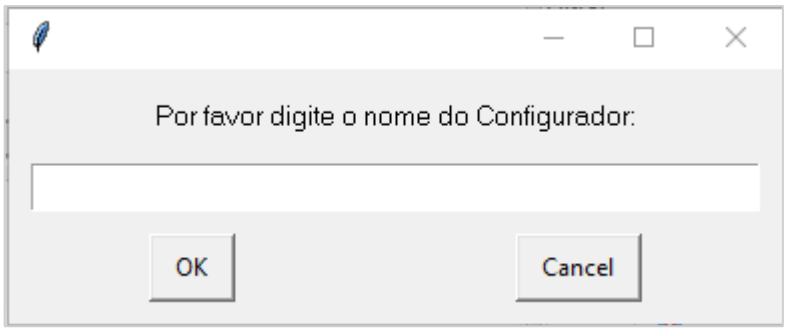
Posteriormente, a inserção da macro de página na Typical de macros, a função *abreImagemeClica1*() é chamada, a fim de maximizar sua estrutura para serem realizadas as configurações na Typical de macros.

O projeto utilizado para realização dos testes de aceitação da ferramenta é um projeto estruturado, no qual, ao ser criado, foram mantidas as nomenclaturas próprias que o *software* gera. Essas estruturas são chamadas de “Designação de função e localização” e precisam ser preenchidas corretamente para que, quando o projetista for gerar o projeto no *Cogineer Project Builder*, não haja erros e a geração do projeto possa ser concluída com sucesso.

A nomenclatura padrão do *software* da Designação de função é CA1 e da Designação de localização é EAA. Para realizar a inserção dos nomes das estruturas a função *abreImagemeClica*() é acionada por duas vezes, sequencialmente, assim como a função *write*().

Depois que o nome das estruturas é inserido na macro, a função *abreImagemeClica*() é chamada por quatro vezes para disponibilizar a aba onde foi realizado o Configurador. Nessa etapa, o usuário humano precisou definir o nome do configurador, que é solicitado, conforme evidencia-se na Figura 29.

Figura 29 – Escolha do nome do Configurador



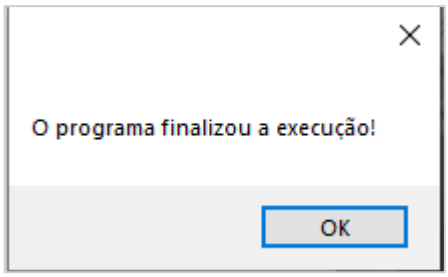
Fonte: Elaborada pela autora (2022).

Após o usuário ter digitado sua escolha para o nome do configurador, ela é armazenada em uma variável e através da função *write*() é escrita na aba Nome e em seguida, a função *abreImagemeClica*() é chamada para que o configurador possa ser gerado.

Esse configurador é disponibilizado para a etapa *Cogineer Project Builder* através da função *abreImagemeClica*().

Ao serem finalizados todos os passos necessários do mapeamento, o *Cogineer Designer* é encerrado através da função *abreImagemeClica*() e uma mensagem de aviso ao usuário é disparada, conforme demonstra a Figura 30.

Figura 30 – Mensagem de encerramento



Fonte: Elaborada pela autora (2022).

A seguir, é abordado como foi realizado a validação da ferramenta, ao qual consistiu primeiramente em verificar se os passos do mapeamento foram executados corretamente e posteriormente, avaliar o tempo que o *desktop* e projetistas voluntários levaram para mapear a macro.

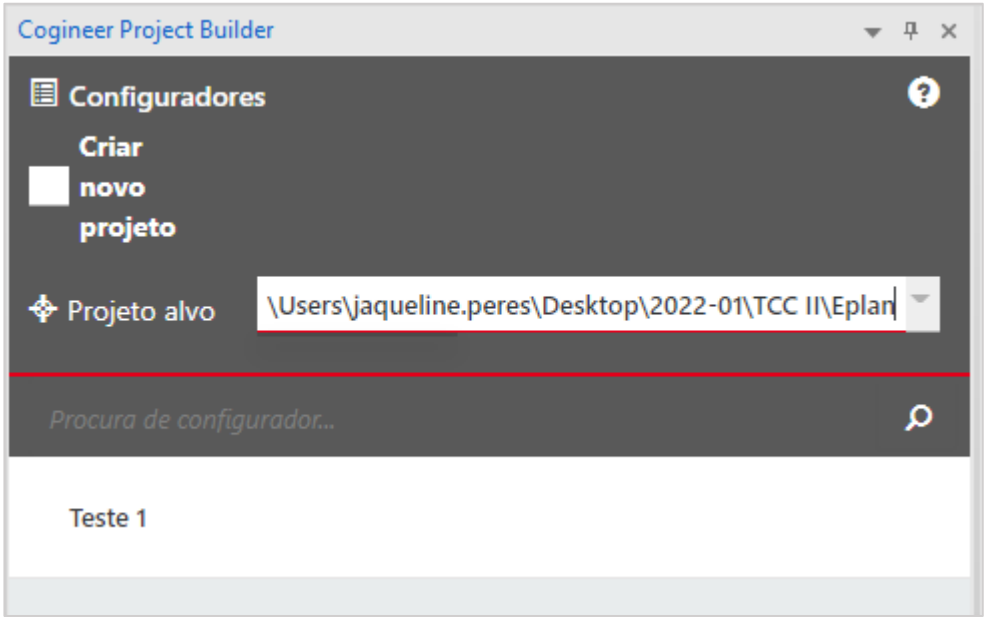
4 ANÁLISE DOS RESULTADOS

Neste capitulo, é apresentado como foi validado o correto funcionamento da ferramenta, assim como é analisado se o seu uso apresentou uma aplicação satisfatória para o setor de projeto elétrico de automação em uma empresa.

A primeira etapa foi verificar se o configurador criado ficou disponível na etapa *Cogineer Project Builder*, pois se a ferramenta realizou os passos de modo correto a macro mapeada ficaria disponibilizada através do configurador, constatando assim, o correto funcionamento da ferramenta.

Para fins de teste, o configurador gerado foi chamado de Teste 1 e ao ser inserido o endereço do projeto alvo, automaticamente o configurador disponível foi apresentado. A Figura 31 apresenta o configurador disponível na etapa *Cogineer Project Builder* para que o projetista pudesse gerar o projeto elétrico.

Figura 31 – Configurador disponível



Fonte: Elaborada pela autora (2022).

A próxima validação realizada foi a comparação do tempo que a máquina e projetistas voluntários do setor do projeto elétrico levam para mapear a macro e disponibilizar o configurador para a etapa *Cogineer Project Builder*. Hoje, no setor de projetos elétricos, da empresa em questão, há 3 projetistas.

Para essa validação, criou-se um formulário de avaliação, que pode ser conferido no Apêndice A, em que foram solicitados os dados dos voluntários, assim

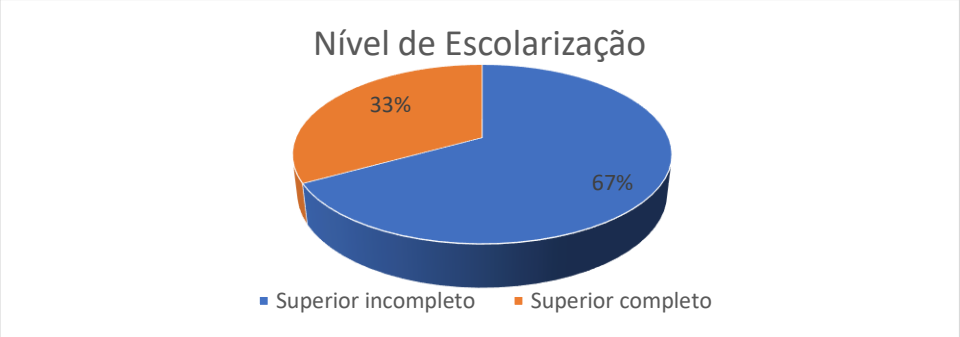
como o tempo que levaram para executar o mapeamento da macro e disponibilizar ao *Cogineer Project Builder* através do configurador criado.

Enviou-se aos voluntários o projeto de macros NTE_001, a macro de página Circuito Auxiliar e o formulário de avaliação. Após receberem os arquivos necessários para a realização dos testes, foi-lhes pedido que criassem um projeto novo para a execução dos passos do mapeamento.

A avaliação que se procurou realizar entre a ferramenta criada e o teste que os projetistas realizaram, levou em consideração a velocidade de execução, ou seja, o ganho de tempo na execução do mapeamento, podendo assim, verificar se o objetivo proposto de diminuir o homem-hora (HH) dos projetistas, foi atendido.

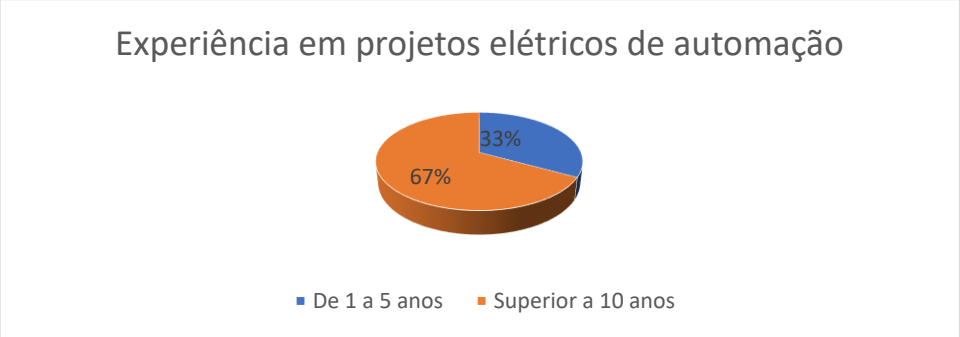
Com os dados dos voluntários obtidos na ficha de avaliação, agregou-se os dados concernentes aos níveis de escolarização e experiência tanto em projetos elétricos de automação, como no uso do *softwar*e EPLAN. Nos Gráficos 2, 3 e 4 podem ser conferidos os dados obtidos.

Gráfico 2 – Nível de escolarização dos projetistas



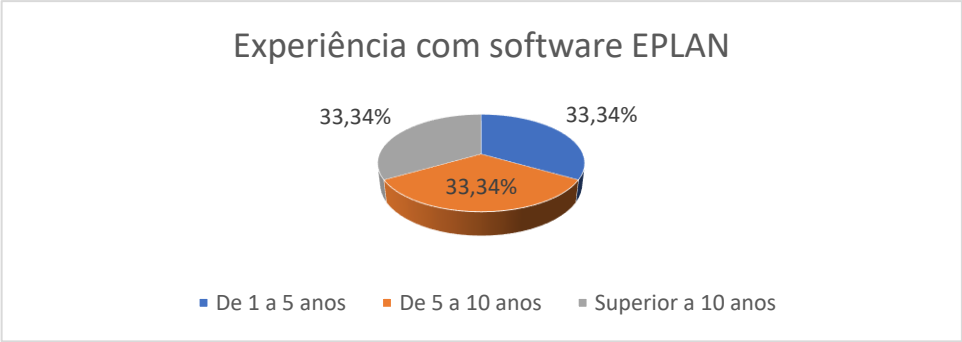
Fonte: Elaborado pela autora (2022).

Gráfico 3 – Nível de experiência em projetos elétricos de automação



Fonte: Elaborado pela autora (2022).

Gráfico 4 – Nível de experiência com o *software* EPLAN dos projetistas

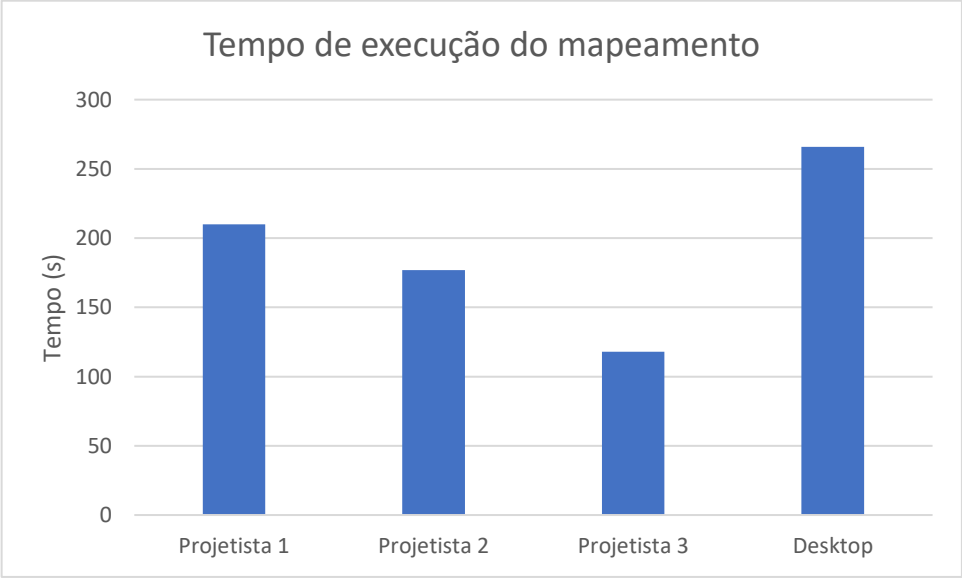


Fonte: Elaborado pela autora (2022).

Verificando os gráficos é possível perceber que os voluntários estavam habilitados para a realização do mapeamento, pois possuem vasta experiência em projetos elétricos de automação, assim como uma boa experiência com o *software* EPLAN. Todos os projetistas são eletrotécnicos, sendo um com o ensino superior concluído e dois em fase de conclusão.

O Gráfico 5 demonstra o tempo que os projetistas voluntários e o *desktop* levaram para a realização do mapeamento da macro Circuito Auxiliar.

Gráfico 5 – Tempo necessário para o mapeamento da macro



Fonte: Elaborado pela autora (2022).

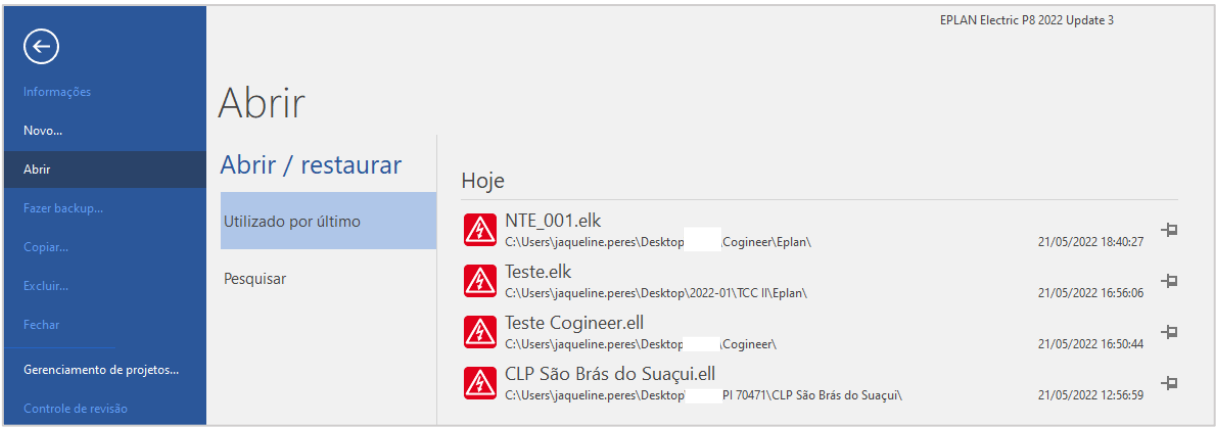
Analisando o gráfico 5 constatou-se que os projetistas executaram o mapeamento da macro em um tempo menor que o *desktop*, demonstrando dessa

forma, que o objetivo principal desse projeto, que visava diminuir o HH dos projetistas na realização do projeto elétrico, não foi alcançado.

Essa diferença de tempo se deve, também, devido algumas situações, tais como:

a) Na abertura do projeto de macros, se o projetista tivesse aberto recentemente o projeto, o endereço dele já estaria salvo na tela Abrir do *software* Eplan, como demonstra a Figura 32, diminuindo assim, o tempo em relação ao *desktop*, pois esse sempre executa os passos do programa.

Figura 32 – Tela Abrir do *software* EPLAN



Fonte: Elaborada pela autora (2022).

b) Outra situação é a interação com o usuário humano, que ocorrem em três momentos, escolha do nome da macro, da variável e do configurador. Esta interação demanda um tempo maior que o do projetista na execução, pois esse somente digita a escolha dos nomes, enquanto a máquina solicita, aguarda a escolha do nome ser digitada, clica no campo relacionado e escreve o nome escolhido.

Independente se a máquina tivesse executado em um tempo menor, constatou-se no decorrer do desenvolvimento da ferramenta, que o mapeamento de macros através *Cogineer Designer* neste formato, não atende ao setor de projetos elétrico, pois esse tipo de mapeamento é uma solução viável quando se gera um produto e não um projeto, onde há diversos tipos de cenários.

Vale ressaltar que no decorrer da elaboração da ferramenta foram encontradas algumas dificuldades, como por exemplo: quando o código foi interpretado para testes, algumas vezes, o processamento da máquina atrasou algumas ações, o que gerou alguns erros, encerrando assim o programa. Para solucionar esse problema atrasou-se alguns trechos do código através da biblioteca *time*.

Algumas melhorias seriam necessárias para tornar a ferramenta mais robusta, caso este projeto fosse validado para ser implementado no setor de projeto elétrico, como:

1) Implementar uma lógica condicional para quando o *desktop* estiver na VPN da empresa, pois a mensagem disparada é diferente, assim ela atenderia essa possível situação e programa continuaria sendo executado.
2) Implementar uma lógica condicional para atender quando o navegador de macros e/ou a Definidora do projeto de macros não forem fechados corretamente, na qual seria verificado se eles já estão abertos e se sim, passaria a executar o próximo trecho.
3) Estudar uma forma para deixar o programa mais confiável, pois para fins de entendimento, optou-se pela comparação através de imagens para executar os passos do mapeamento da macro. Porém, constatou-se que os *pixels* das imagens podem gerar erros, o que impede de o programa continuar executando.

5 CONCLUSÃO

O cenário atual com o intenso avanço da tecnologia tem provocado nas organizações uma busca incessante por ferramentas que otimizem os processos em seus diversos setores. Este trabalho propôs a análise e otimização do uso da ferramenta *Cogineer Designer* no setor de projeto elétrico de automação, cujo objetivo principal visava diminuir a quantidade de homem-hora (HH) na elaboração do projeto elétrico. Com o uso da biblioteca *PyAutoGUI*, incorporada ao *Python*, realizou-se a automação do *desktop* para o mapeamento de macros no *Cogineer Designer.*

Para o desenvolvimento da ferramenta averiguou-se, primeiramente, cada passo necessário para a máquina executar o mapeamento da macro, criando-se um fluxograma de todo o processo e posteriormente, diagramas de blocos de cada etapa, a fim de facilitar a criação do código e não gerar erros no interpretador.

Pode-se observar que a máquina executou com sucesso o mapeamento da macro, pois na etapa *Cogineer Project Builde*r, o configurador contendo a macro Circuito Auxiliar ficou disponível para a geração do projeto. Já em relação ao ganho de tempo esse projeto não alcançou o objetivo proposto, ficando comprovado, a partir de testes realizados juntamente com projetista voluntários, que o *desktop* executa o mapeamento em um tempo maior que eles.

De todo modo, ao ser estudada a ferramenta *Cogineer Designer* ao longo do desenvolvimento desse projeto, verificou-se junto ao setor de projeto elétrico que ela não atenderia a demanda do setor, como esperado. A partir dessa constatação e dos conhecimentos adquiridos ao longo desse processo, buscou-se junto a EPLAN outro modo de utilização da licença *Cogineer*.

Dessa forma, como sugestões de trabalhos futuros, almeja-se: tornar a ferramenta mais robusta, como descrito na Análise de Resultados; ensinar o *desktop* inserir várias páginas dos cartões de CLP; ensinar à máquina os passos para atender a etapa do *Cogineer Project Builder*; implementar o mesmo método em outras áreas que também realizam tarefas repetitivas.

REFERÊNCIAS

ALMEIDA. **A Bíblia da mulher**. 2009. 2. Ed. Barueri, SP. Sociedade Bíblica do Brasil.

BARUK. **Automação robótica de processos**. Rio de Janeiro: baruk, [c2022?]. Disponível em: <https://baruk.me/automacao-processos-roboticos-rpa-para-empresa/>. Acesso em: 25 mai. 2022.

CARVALHO, Maxmyller Ferreira de Freitas. **Automatização por robô de software para um sistema contábil**. 2020. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Universidade Federal do Rio Grande do Norte, Natal, 2020. Disponível em: <https://repositorio.ufrn.br/bitstream/123456789/43643/3/AutomatizacaoPorRobo_Carvalho_2020.pdf>. Acesso em: 19 fev. 2022.

EPLAN. **Cogineer**. [*S. l.*]: EPLAN, [2021a?]. Disponível em: <https://www.EPLAN.help/help/cogineercloud/en-us/Content/htm/cogineer_intro.htm>. Acesso em: 04 set. 2021.

EPLAN. **Designer**. [*S. l.*]: EPLAN, [2021b?]. Disponível em: <https://www.EPLAN.help/help/cogineercloud/enus/Content/htm/cogineer_d_basics.htm>.Acesso em: 04 set. 2021.

EPLAN. **EPLAN.help**. [*S. l.*]: EPLAN, [2021c?]. Disponível em: <https://www.EPLAN.help/pt-br/Infoportal/Content/Plattform/2.9/EPLAN_Help.htm>. Acesso em: 23 out. 2021.

EPLAN. **Introduction**. [*S. l.*]: EPLAN, [2021d?]. Disponível em: <https://www.EPLAN-software.com/solutions/EPLAN-platform/EPLAN-electric-p8/>. Acesso em: 20 ago. 2021.

EPLAN. **Project Builder**. [*S. l.*]: EPLAN, [2021e?]. Disponível em: <https://www.EPLAN.help/help/cogineercloud/enus/Content/htm/cogineer_pb_basics.htm>. Acesso em: 04 set. 2021.

FRANÇA, Fábio F. *et al*. **Desenvolvimento de uma interface de mouse para pessoas com tetraplegia**. Campus Natal: 5° Senid, mai. 2018. Disponível em: <https://www.upf.br/_uploads/Conteudo/senid/2018-artigos-resumidos/179263.pdf>. Acesso em: 19 fev. 2022.

MADAKAM, Somayya. *et al*. **The Future Digital Work Force:** Robotic Process Automation (RPA). São Paulo: SciELO Brazil, 2019. Disponível em: <https://www.scielo.br/j/jistm/a/m7cqFWJPsWSk8ZnWRN6fR5m/?lang=en>. Acesso em: 22 mai. 2022.

MELO, Diego. **O que é Python? [Guia para iniciantes]**. [*S. l.*]: Tecnoblog, [2021a?]. Disponível em: <https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/#:~:text=Python%20%C3%A9%20uma%20linguagem%20de,vantagens%20neste%20guia%20para%20iniciantes>. Acesso em: 11 out. 2021.

ORESTES, Yan. **Conhecendo as tuplas no Python**. São Paulo: Alura, 20 set. 2018. Disponível em: <https://www.alura.com.br/artigos/conhecendo-as-tuplas-no-python>. Acesso em: 09 abr. 2022.

PYAUTOGUI. **Welcome to PyAutoGUI's documentation!**[*S. l.*]: PyautoGUI, c2019. Disponível em: <https://pyautogui.readthedocs.io/en/latest/>. Acesso em: 30 nov. 2021.

PYTHON DS. **RPA para automação de interfaces gráfica com PyAutoGUI**. [*S. l.*]: PythonDS, 31 jan 2020. Disponível em: <http://www.pythonds.com.br/posts/pyagui.html>. Acesso em: 30 nov. 2021.

PYTHON. **os – Miscellaneous operating system interfaces**. [*S. l.*]: Python, c2001. Disponível em: <https://docs.python.org/3/library/os.html?highlight=os%20startfile#os.startfile>. Acesso em: 06 mar. 2022.

QUALITAT GRUPO. **Automação Robótica de Processos**. São Paulo: QUALITAT GRUPO, [2018?]. Disponível em: <https://www.channel360.com.br/wp-content/uploads/2019/02/ebook-automacao-robotica-processos.pdf>. Acesso em: 20 de fev. 2022.

GRAND VIEW RESEARCH. **Robotic process automation market size, share & trends analysis report by type, by service, by application, by deployment, by organization, by region, and segment forecasts, 2021 – 2028**. San Francisco: Grand View Research, Apr. 2021. Disponível em: https://www.grandviewresearch.com/industry-analysis/robotic-process-automation-rpa-market. Acesso em: 20 fev. 2022.

SAUER, Luiz Felipe Corazza; PILAN, José Rafael. **MARIANA - ASSISTENTE VIRTUAL PERSONALIZADO PARA LINUX DESKTOP**. São Paulo: Revista Tekhne e Logos, 3 dez. 2021 . Disponível em: <http://revista.fatecbt.edu.br/index.php/tl/article/view/668>. Acesso em: 19 fev. 2022.

APÊNDICE A – FORMULÁRIO PARA ANÁLISE DO MAPEAMENTO DA MACRO

FORMULÁRIO PARA ANÁLISE DO MAPEAMENTO DA MACRO

Nome Completo:
Data:

Nível de Escolarização	
	Ensino Médio
	Ensino Técnico
	Ensino Superior Incompleto
	Ensino Superior

Experiência em Projetos Elétrico de Automação	
	Menos de 1 ano
	De 1 a 5 anos
	De 5 a 10 anos
	Superior a 10 anos

Experiência com *software* EPLAN	
	Menos de 1 ano
	De 1 a 5 anos
	De 5 a 10 anos
	Superior a 10 anos

Os passos para realizar o mapeamento da macro estão claros?
Sim ( ) Não ( )

Sente-se apto para realizar essa atividade ?
Sim ( ) Não ( )

Quanto tempo foi necessário para o mapeamento: _____

Observações/Sugestões:
