

Universidade Federal de Minas Gerais
Escola de Engenharia
Curso de Graduação em Engenharia de Controle e Automação

Projeto Final de Curso
Automatização do Teste de Endereçamento em CLP
Rockwell para empresa do Segmento Siderúrgico

Watson Salerno Parente Santana

Orientador: Prof. Vítor Costa da Silva Campos
Supervisor: Eng. Ronaldo Schittini Gardoni Júnior

Belo Horizonte, Junho de 2023

Monografia

Automatização do Teste de Endereçamento em CLP Rockwell para empresa do Segmento Siderúrgico

Monografia a ser submetida à banca examinadora designada pelo Colegiado Didático do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Minas Gerais, como parte dos requisitos para futura aprovação na disciplina Projeto Final de Curso II.

Belo Horizonte, 07 de Julho de 2023

Resumo

A demanda por projetos de engenharia de automação voltados para a indústria siderúrgica vem crescendo de forma significativa nos últimos anos, principalmente em função das inovações tecnológicas trazidas pela chamada Indústria 4.0. Empresas fornecedoras de soluções para esse mercado, como a Primetals Technologies, parceira da execução deste trabalho, estão inseridas em um contexto de alta competitividade em que se sobressai aquela com capacidade de entregar bons resultados com um menor gasto de tempo. Dessa maneira, é grande a necessidade de se encontrar ferramentas e estratégias de trabalho que possibilitem agilizar ou até mesmo eliminar processos repetitivos e de baixo valor agregado, para que assim os esforços da equipe de engenharia possam se concentrar em atividades de maior complexidade, como a programação de lógicas para Controladores Lógicos Programáveis (CLPs). Diante disso, o presente trabalho apresenta os procedimentos utilizados para se obter uma aplicação desenvolvida em *Visual Basic for Application* (VBA), capaz de realizar a leitura, em um planilha de *Excel* padronizada, dos dados que compõem o endereçamento de *hardware* de um projeto de automação, como os nomes, tipos e endereços de variáveis. Com isso, a aplicação possibilita o teste destes endereços de maneira automática, tendo como resultando uma planilha parcialmente preenchida no que compõe a etapa de Testes Internos da Primetals Technologies, para CLPs Rockwell.

Agradecimentos

Agradeço primeiramente à minha família que me acompanhou, apoiou e me reconfortou nos momentos mais difíceis desta trajetória. Eterna gratidão aos meus pais, Tatiane e Weliton, por todos os esforços que foram realizados para que o meu sonho em me formar em uma das melhores Universidades do país, fosse possível. Um abraço ao meu irmão, que mesmo não estando neste plano, tenho certeza que sempre me acompanhou nesta trajetória.

Cito aqui também familiares e amigos especiais que fazem parte desta minha vitória: minha avó Rosilda, meus primos e primas Paulo César, Pedro Victor, Ana Luiza e Maria Eduarda, minhas tias Lyramar e Graciela e meus melhores amigos: Yan, Luciano, Eduardo, Ítalo e Gabriel. Não menos importante, todos os meus amigos de faculdade com quem compartilhei muitas risadas, dificuldades e bons momentos ao longo destes anos.

Agradeço também a duas pessoas muito especiais que moldaram meu amor pelo conhecimento: Pedrina e Antônia. Vocês foram parte essencial desta trajetória e carregaram todo carinho, atenção e ensinamentos recebidos. Um agradecimento especial para meu orientador Vítor Costa, que mesmo diante de diversas dificuldades pela qual passei durante a elaboração deste projeto e diante de uma pandemia, me apoiou e me aconselhou sabiamente.

Finalmente, gratidão à empresa Primetals Technologies e aos meus colegas de profissão, que em nenhum momento negaram ajuda, recursos ou demonstraram falta de vontade para colaborar comigo no desenvolvimento do projeto.

Conteúdo

Resumo	i
Agradecimentos	ii
Lista de Figuras	vi
1 Introdução	1
1.1 Motivação e Justificativa	1
1.2 Objetivos do Projeto	3
1.3 Local de Realização	3
1.4 Estrutura da Monografia	3
2 Revisão Bibliográfica	5
2.1 A Quarta Revolução Industrial	5
2.2 Impactos da Indústria 4.0 em Empresas de Projetos de Engenharia	6
2.3 Conceitos para Projetos de Automação na Indústria 4.0	7
2.4 <i>Visual Basic for Application</i>	7
2.4.1 <i>Dynamic Data Exchange - DDE</i>	8
2.4.2 <i>Component Object Model/Distributed Component Object Model</i>	9
2.4.3 <i>Object Linking and Embedding</i>	10
2.4.4 Diferenças entre DDE x OLE x COM/DCOM	10
2.5 Conceitos Principais sobre RPA	11
2.5.1 Objetivos e benefícios do sistema RPA	12
2.6 Aplicações de RPA na indústria/nas empresas	13
2.7 Estrutura de Hardware de um Controlador Lógico Programável	15
2.7.1 Conceitos Chave: Lista de Entrada/Saída	16
2.7.2 Conceitos Chave: Formato de Endereçamento Físico de dispositivos	16
3 Metodologia	19
3.1 Recursos Empregados	19
3.2 Concepção do projeto baseado em um estudo de caso	19
3.3 Comunicação entre as aplicações	20
3.3.1 Configuração <i>Studio 5000®Logix Emulator</i>	21
3.3.2 Configuração <i>Logix</i>	24
3.3.3 Configuração <i>RSLinx Classic</i>	26
3.4 Possíveis configurações da Árvore de <i>Hardware</i> e uso de <i>Tags</i> Locais	27
3.5 Planilha de Testes Internos	30

4	Discussão	33
4.1	Premissas de funcionamento adequado da Macro	33
4.1.1	Preenchimento dos Campos	33
4.1.2	<i>Parameters and Local Tags</i> - Como tratar	37
4.2	Vantagens do uso da Planilha	41
5	Conclusões	47
5.1	Considerações Finais	47
5.2	Propostas de Continuidade	48
6	Anexos	49
6.1	<i>Main</i>	49
6.2	<i>GerarValorRandomico</i>	50
	Referências Bibliográficas	51

Lista de Figuras

1.1	<i>Rack</i> com o Controlador Lógico Rockwell e os cartões (módulos). Fonte: Autor.	2
2.1	Gráfico de casos de automação de processos. Fonte: [34]	14
2.2	Estrutura de <i>Hardware</i> de um CLP. Fonte: [36]	15
3.1	Visão geral para comunicação e funcionamento do projeto. Fonte: Elaborado pelo Autor.	20
3.2	Interface inicial <i>Rockwell Automation Studio 5000®Logix Emulator</i> . Fonte: Autor.	21
3.3	Processo de declaração da CPU Virtual. Fonte: Autor.	21
3.4	Processo de declaração da CPU Virtual. Fonte: Autor.	22
3.5	Processo de declaração da CPU Virtual. Fonte: Autor.	22
3.6	Processo de declaração do cartão de E/S. Fonte: Autor.	22
3.7	Processo de declaração do cartão de E/S. Fonte: Autor.	23
3.8	Processo de declaração do cartão de E/S. Fonte: Autor.	23
3.9	Processo de declaração do cartão de E/S. Fonte: Autor.	23
3.10	Estrutura final da configuração do ambiente <i>Logix Emulator</i> utilizado para o Projeto. Fonte: Autor.	24
3.11	<i>Logix Designer 1</i> - Passo a passo para entrar online com a aplicação. Fonte: Autor.	24
3.12	<i>Logix Designer 2</i> - Passo a passo para entrar online com a aplicação. Fonte: Autor.	25
3.13	<i>Logix Designer 3</i> - Passo a passo para entrar online com a aplicação. Fonte: Autor.	25
3.14	<i>Logix Designer 4</i> - Passo a passo para entrar online com a aplicação. Fonte: Autor.	25
3.15	<i>Logix Designer 5</i> - Passo a passo para entrar online com a aplicação. Fonte: Autor.	26
3.16	Interface inicial <i>RSLinx Classic</i> . Fonte: Autor.	26
3.17	DDE <i>Topic Configuration</i> para o <i>RSLinx Classic</i> . Fonte: Autor.	27
3.18	Criado um novo <i>topic</i> com o caminho apontado para o CLP emulado, no <i>Virtual Backplane</i> . Fonte: Autor.	27
3.19	Modelo de Árvore de <i>Hardware</i> . Fonte: Elaborado pelo Autor.	28
3.20	Visão da árvore de <i>Hardware</i> visto dentro do ambiente <i>RSLogix Designer</i> . Fonte: Elaborado pelo Autor.	29

3.21	Visão do endereçamento equivalente à configurações 3.19 e 3.20. Fonte: Elaborado pelo Autor.	29
3.22	Arquitetura tradicional de <i>Hardware</i> - Cartões de Entrada/Saída no mesmo <i>rack</i> do CLP. Fonte: Autor.	29
3.23	Visão da <i>Árvore de Hardware</i> visto dentro do ambiente <i>RSLogix Designer</i> . Fonte: Elaborado pelo Autor.	30
3.24	Visão do endereçamento equivalente à configurações 3.22 e 3.23. Fonte: Elaborado pelo Autor.	30
3.25	Esquema de aceitação do uso de ferramentas Automatizadas. Fonte: [49].	31
3.26	Modelo da Planilha de Testes de Equipamentos e Endereçamento. Fonte: Elaborado pelo Autor.	32
4.1	Erro ao criar o <i>Topic</i> , nomenclatura incorreta. Elaborado pelo Autor.	34
4.2	Campo preenchido <i>versus Topic</i> criado. Elaborado pelo Autor.	34
4.3	Cenário de teste: Ausência do endereço físico. Elaborado pelo Autor.	35
4.4	Cenário de teste: Ausência da <i>Tag</i> . Elaborado pelo Autor.	35
4.5	Cenário de teste: Ausência do tipo de Sinal. Elaborado pelo Autor.	35
4.6	Cenário de teste: Ausência do preenchimento dos Campos. Teste não executado. Elaborado pelo Autor.	35
4.7	Cenário de teste: Preenchimento incorreto do campo “Tipo”. Teste executado com erro de Endereçamento. Elaborado pelo Autor.	37
4.8	Cenário de teste: Preenchimento incorreto do campo “Endereço”. Teste executado com erro de Endereçamento. Elaborado pelo Autor.	37
4.9	Alteração de escopo de <i>Tag</i> Local. Elaborado pelo Autor.	37
4.10	Endereçamento de <i>Tag</i> Local. Elaborado pelo Autor.	38
4.11	<i>Tag</i> declarada em um escopo local, mas não registrada de maneira correta na planilha. Elaborado pelo Autor.	39
4.12	Interrupção abrupta do funcionamento do <i>Excel</i> . Elaborado pelo Autor.	39
4.13	Sugestão de organização e uso da planilha - Organização a nível de CLP. Elaborado pelo Autor.	39
4.14	Sugestão de organização e uso da planilha - Organização a nível de teste. Elaborado pelo Autor.	40
4.15	Alteração de uma <i>Task</i> para o formato acíclico. Elaborado pelo Autor.	41
4.16	Resultado do Teste. Elaborado pelo Autor.	41
4.17	<i>Tags</i> Globais pré-teste. Elaborado pelo Autor.	42
4.18	Planilha de <i>Tag</i> Globais pré-teste. Elaborado pelo Autor.	43
4.19	<i>Tags</i> Locais pré-teste. Elaborado pelo Autor.	43
4.20	Planilha de <i>Tag</i> Locais pré-teste. Elaborado pelo Autor.	44
4.21	<i>Tags</i> Globais pós-teste. Elaborado pelo Autor.	44
4.22	Planilha de <i>Tag</i> Globais pós-teste. Elaborado pelo Autor.	45
4.23	<i>Tags</i> Locais pós-teste. Elaborado pelo Autor.	45
4.24	Planilha de <i>Tag</i> Locais pós-teste. Elaborado pelo Autor.	46

Capítulo 1

Introdução

1.1 Motivação e Justificativa

Ao longo de sua história, o homem busca constantemente desenvolver métodos ou ferramentas que facilitem a execução de determinada atividade [1]. Foi assim com a invenção de moinhos de vento e carros movidos à tração animal, por exemplo. Mais recentemente, por volta dos anos de 1950, o termo automação começou a se popularizar tendo como significado a movimentação automática de materiais. Nos dias atuais, este termo possui uma representação muito maior e mostra-se presente no cotidiano de praticamente todas as pessoas, desde em seus despertadores, programados previamente para tocar em determinado horário, até em robôs aspiradores de pó. No setor secundário, ou seja, na indústria, a realidade não é diferente. Segundo pesquisa de 2020 da Associação Brasileira de Automação [2], em média, 68% das linhas de produção presentes em fábricas são automatizadas por meio de sensores, equipamentos de precisão e sistemas compostos por Controladores Lógicos Programáveis (CLPs), entre outros.

A automação industrial tem por definição o objetivo de, a partir da aplicação de tecnologias em processos produtivos, aumentar a eficiência e a produtividade, minimizar falhas, tornar estes processos mais seguros e precisos e, por fim, reduzir qualquer esforço humano desnecessário. Visando estes benefícios, a Quarta Revolução Industrial tem impulsionado fortemente a automação de fábricas e indústrias, criando oportunidades de crescimento e modernização do setor e corroborando para vantagens competitivas no mercado [3]. Conceitos como integração de sistemas, gerenciamento de dados, digitalização, Internet das Coisas, robótica, entre outros estão em evidência e são alvo de pesquisa e desenvolvimento de grandes multinacionais fornecedoras de *softwares* industriais, como a Rockwell Automation, que possuem em seu portfólio soluções como o PlantPAX [4]. A ferramenta Rockwell essencialmente é uma plataforma que unifica as ferramentas da empresa em um ambiente digitalizado, que permite a análise e compartilhamento de diversos dados de uma planta, sempre visando o aumento de produtividade.

Uma vez que o setor industrial é responsável por uma parcela de, aproximadamente, 21% do Produto Interno Bruto (PIB) brasileiro [5], pode-se afirmar que é grande a demanda por indústrias cada vez mais automatizadas e eficientes. Além disso, segundo diagnóstico levantado pela organização GSIBrasil em 2020 [2], cerca de 43% das indústrias nacionais aguardavam uma melhora do cenário global no contexto da pandemia do Covid-19 para investir em automação visando a modernização das suas fábricas, o que tende a elevar ainda

mais esta demanda. Por isso, existem no Brasil diversas empresas de engenharia focadas em automação industrial, que desenvolvem soluções para atender às necessidades de um mercado caracterizado por estar em constante desenvolvimento, principalmente com o avanço da Quarta Revolução Industrial. Dentre estas empresas, destaca-se a Primetals Technologies que atua diretamente em projetos de engenharia voltados para a automação de plantas siderúrgicas e vem, progressivamente, recebendo por parte de seus clientes requisições por uma execução de projetos rápida, assertiva e com custos reduzidos, visto que as fábricas buscam minimizar o intervalo entre a imobilização do capital investido e o início da produção na nova planta ou na planta atualizada.

Dado que se observa uma tendência da indústria de manufatura em buscar soluções que amplifiquem a sua eficiência de produção e assertividade, é possível compreender o porquê dos controladores lógicos programáveis (CLPs) serem amplamente utilizados para controlar os processos tecnológicos em diversas áreas da indústria [6] e estarem presentes na maior parte de projetos de engenharia. No entanto, o desenvolvimento do *software* de um CLP, principalmente em suas etapas iniciais, pode acarretar um gargalo na produção de uma empresa de projetos de automação industrial, principalmente com a ocorrência de projetos com uma quantidade elevada de pontos de entrada e saída. A definição da lista de entradas e saídas (E/S) possui um papel crucial no desenvolvimento do projeto, onde erros de endereçamento e validação do mesmo podem implicar em um mau desempenho da aplicação para o período de testes e de comissionamento. Por apresentar uma alta intervenção humana na redigitação e digitação de dados fundamentais, a etapa de testes está sujeita a erros, conforme indicam estudos de confiabilidade humana [7].

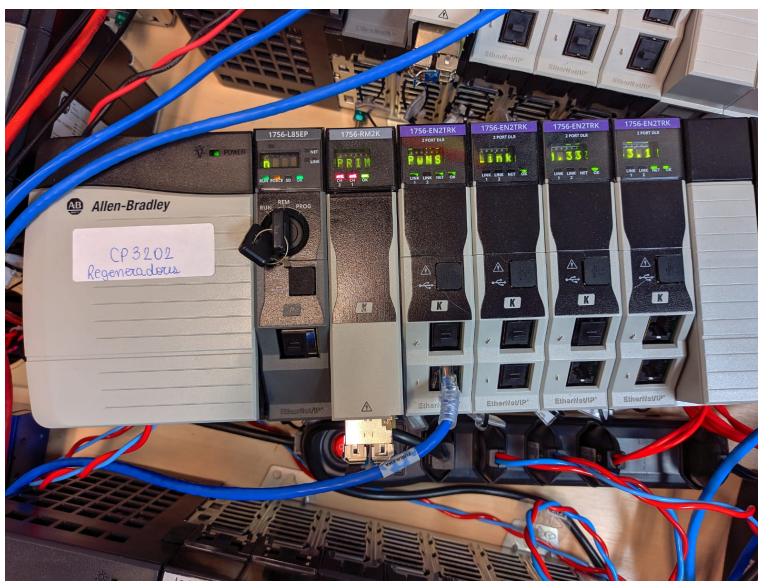


Figura 1.1: Rack com o Controlador Lógico Rockwell e os cartões (módulos). Fonte: Autor.

Diante do cenário da necessidade de uma alta confiabilidade no teste de endereçamento de entradas e saídas de um CLP, o Projeto Final de Curso objetiva desenvolver uma solução onde a pessoa técnica responsável pelo teste (Engenheiro ou Projetista) da Primetals Technologies, possa focar seus esforços em eventuais correções pontuais de endereçamento ou no teste lógico dos dispositivos associados aos mesmo, reduzindo assim os esforços de validação dos mesmo. Para realização do teste, utiliza-se uma planilha padrozinada de testes.

Visando reduzir o gargalo de teste de diversos pontos de E/S, algumas alternativas de automação surgem e para isso, ferramentas como as citadas abaixo entram em cogitação:

- *Visual Basic for Application (VBA)*;
- *Robotic Process Automation (RPA)*.

A planilha padronizada é utilizada na maior parte dos projetos executados pela empresa e contém informações essenciais para o teste do programa elaborado para o CLP, tais como: documento referência, endereço físico dos cartões de entrada e saída a serem utilizados, *tag* proposto para o equipamento, função e tipo de equipamento.

1.2 Objetivos do Projeto

Seguindo o contexto apresentado na introdução deste trabalho, utilizando os conhecimentos adquiridos ao longo do curso e o conhecimento adquirido ao longo da carreira profissional, o trabalho tem como objetivo:

- a. Redução do tempo de teste de endereçamento;
- b. Elaborar uma planilha padronizada para uso do corpo técnico;
- c. Validação dos resultados e aceitação da ferramenta, por parte do corpo técnico da empresa.

1.3 Local de Realização

A empresa parceira do projeto em questão é a Primetals Technologies Brazil Ltda., multinacional da área de engenharia e projetos voltados para a siderurgia. Seu escritório de Belo Horizonte está localizado na Rua Matias Cardoso, 169, 6º andar, bairro Santo Agostinho. O projeto será desenvolvido fora das dependências da empresa, na residência do aluno. No entanto, as dependências da Primetals poderão ser utilizadas caso haja necessidade.

1.4 Estrutura da Monografia

O trabalho está dividido em cinco capítulos. O Capítulo 1 apresenta uma breve introdução sobre o cenário de crescimento na automação de processos, com uma visão geral sobre indústria. O tema do Projeto Final de Curso é contextualizado, o respectivo problema apresentado e os objetivos almejados com a solução.

No Capítulo 2 apresentam-se as definições e conceitos associados às inovações presentes na automação e das respectivas ferramentas apresentadas na introdução, sendo estes embasados em documentos técnicos obtidos por meio de uma pesquisa bibliográfica. São apresentados conceitos abordados no trabalho, assim como possíveis soluções disponíveis no mercado.

O Capítulo 3, tem como foco apresentar a arquitetura do sistema elaborado, detalhando em um nível técnico os critérios de organização da aplicação.

O Capítulo 4 consiste dos resultados obtidos e mensurados com a aplicação. Discute-se as premissas de funcionamento, as dificuldades encontradas pelo mecanismo escolhido para teste, assim como os resultados obtidos.

Por fim, um capítulo para a conclusão sumariza todos os resultados mais relevantes, possibilidades futuras para otimização da ferramenta e os próximos passos visados para o projeto.

Capítulo 2

Revisão Bibliográfica

Este capítulo aborda conceitos e estudos relacionados à indústria 4.0, assim como um visão mais detalhadas das ferramentas citadas no Capítulo 1 e conceitos pertinentes ao entendimento da estrutura de um Controlador Lógico Programável. Ressalta-se que o mercado está em franca expansão e passa por um processo de constantes atualizações.

2.1 A Quarta Revolução Industrial

Schwab [8] define que "revolução" é quando uma mudança acontece de maneira radical. O mundo foi e é moldado por revoluções tecnológicas que afetaram e afetam fortemente o contexto social e econômico da sociedade. A máquina a vapor, a construção de ferrovias, a eletricidade, as linhas de montagem, os semicondutores e a computação foram os principais agentes que transformaram o modo de se enxergar o mundo e a estrutura social em suas respectivas épocas. Nos tempos atuais, segundo Blanchet [9], o momento é o de tecnologias como a inteligência artificial, a rede móvel de alta velocidade, a *internet* das coisas, a integração de plantas industriais etc, que caracterizam a Quarta Revolução industrial, também chamada de Indústria 4.0.

De acordo com um artigo da Siemens [10], o conceito de Indústria 4.0 surgiu na Feira de Hannover de 2011, onde empresas de tecnologia, universidades e centros de pesquisa alemães criaram a proposta de um novo modelo para o funcionamento de fábricas. Essa ideia consistia em um controle descentralizado do processo e integração de dispositivos inteligentes, visando a otimização da produção através das informações trocadas por eles. Os princípios fundamentais desta proposta, são [11]:

- **Descentralização:** O aumento da complexidade da produção traz como dificuldade manter a base do sistema de controle centralizado. Por isso, a descentralização da tomada de decisão se mostra essencial à aplicação de soluções para a Indústria 4.0. Este conceito engloba computadores inteligentes distribuídos pelos componentes de um sistema de produção que são capazes de trocar entre si informações sobre o estado atual e com isso tomar decisões que corroborem para alcançar um objetivo;
- **Operação em Tempo Real:** Um sistema deve ser capaz de reagir à mudanças de demandas ou problemas de operação. Para isso, é essencial a aquisição e análise constante dos dados gerados por equipamentos inteligentes visando uma resposta em tempo real;

- **Orientação a Serviços:** É uma estrutura de desenvolvimento de *software* que visa a criação de aplicações que aperfeiçoem um serviço;
- **Virtualização:** Baseando-se no monitoramento e em dados oriundos de sensores, cria-se uma versão digital dos processos, que permite um espelhamento daquilo que é físico por meio de modelos matemáticos. Assim é possível realizar simulações, monitorar, prever problemas e tomar decisões para obter um melhor funcionamento do sistema;
- **Modularidade:** Sistemas modulares podem se ajustar de acordo com mudanças na demanda de um sistema ou uma customização de um produto apenas estendendo ou substituindo um módulo individualmente; e
- **Interoperabilidade:** Sistemas que são capazes de estabelecerem conexão com outros sistemas. Mais sistemas conectados significa maior quantidade de dados coletados e maior tomadas de decisão em tempo real.

De modo que esses princípios sejam aplicados na indústria, tecnologias inovadoras, principalmente de *software* e *hardware*, estão se mostrando cada vez mais necessárias neste ambiente.

2.2 Impactos da Indústria 4.0 em Empresas de Projetos de Engenharia

Como sugere Prasad [12], o atual contexto de mercados globalizados, que exige produtos de maior qualidade e com preços competitivos, está obrigando as empresas a aumentarem o nível de automação de seus processos de fabricação para atender a essas demandas. Por outro lado, os processos estão cada vez mais complexos e devem ser suficientemente flexíveis para se adaptarem às necessidades do mercado. A tudo isso se soma a necessidade de minimizar o tempo necessário para modificar a linha produção, a fim de reduzir o tempo de inatividade da linha e diminuir custos de produção.

Em suma, segundo Eneko [13], os processos de fabricação devem ser flexíveis e adaptáveis à fabricação de lotes de produtos personalizados, que atendem às necessidades e expectativas dos clientes fim. Todas essas adaptações e mudanças nos processos de fabricação devem ser realizados de forma ágil, rápida e confiável, para que não afetem negativamente o eficiência do processo.

As empresas de projetos de engenharia de automação exercem um papel fundamental para a indústria de manufatura, já que através delas há a possibilidade de que as inovações e atualizações advindas da quarta revolução industrial sejam levadas para as fábricas. Quando uma grande indústria contrata uma empresa de projetos para realizar alguma melhoria em alguma de suas plantas, espera-se que o resultado traga alguns dos pilares da indústria 4.0. Ao mesmo tempo, os projetos devem ser realizados de com a maior eficiência possível, dado que, como foi apresentado, as fábricas que não se atualizam sofrem com a perda de competitividade dentro do mercado.

2.3 Conceitos Fundamentais para Projetos de Automação no âmbito da Indústria 4.0

Essa crescente demanda por processos de produção flexíveis e eficientes resulta em uma necessidade cada vez maior de sistemas elétricos e soluções de automação integradas durante todo o ciclo de vida de plantas metalúrgicas [10]. Além disso, as abordagens de reutilização, padronização e modularidade vêm ganhando cada vez mais importância nos últimos anos. Segundo Magar et al. [14], a engenharia de automação mudou o foco dos sistemas únicos para linhas de produtos. Embora o paradigma da engenharia tenha mudado, os sistemas de automação industrial dentro de uma linha de produtos são criados com as mesmas ferramentas de engenharia dos sistemas individuais. Assim, torna-se relevante analisar quais requisitos as ferramentas de engenharia devem atender para aumentar o reaproveitamento em uma linha de produtos de sistemas de automação industrial. Os requisitos identificados são interessantes pelos seguintes motivos: primeiro, eles são relevantes para desenvolvedores de ferramentas de engenharia, uma vez que visam fornecer produtos competitivos. Em segundo lugar, eles são relevantes para desenvolvedores de sistemas de automação industrial, uma vez que reutilizam artefatos pré-existentes no projeto de novos sistemas de automação industrial com a ajuda de ferramentas de engenharia.

Como complementam Magar et al. [14], em projetos industriais a engenharia raramente é executada do zero. Na maioria dos casos, os desenvolvedores de sistemas de soluções industriais estão interessados em reutilizar os resultados do trabalho de projetos anteriores semelhantes, a fim de aumentar a eficiência na engenharia. Os resultados do trabalho reutilizáveis são geralmente chamados de “artefatos” e podem ser múltiplos. Por exemplo, requisitos, documentações, desenhos CAD, diagramas de fiação, componentes de *software*, arquiteturas de referência, modelos de engenharia ou casos de teste são artefatos adequados para reutilização em projetos semelhantes.

Além da questão de reutilização, Ulrich [15] demonstra a importância do conceito de modularidade, um dos pilares da Indústria 4.0, segundo Hermann et al [16]. Em projetos de sistemas de engenharia, este termo se refere à utilização de unidades independentes, que quando colocados para interagir entre si, formam uma variedade de uma solução. Atualmente, em qualquer empresa é importante que seja dado foco à modularidade, seja para facilitar a personalização de produtos, seja para aumentar a eficiência de suas soluções.

Este trabalho tem o objetivo de desenvolver uma aplicação independente que desempenhe bem a função de deixar a execução de projetos mais eficaz. Além disso, esta aplicação tem a premissa de padronizar o início do desenvolvimento de projetos de *software* de CLP. Por isso, nota-se que os pilares da Indústria 4.0 estão diretamente conectados com os objetivos deste trabalho.

2.4 *Visual Basic for Application*

O *Visual Basic for Application* (VBA) [17] é uma linguagem de programação desenvolvida pela *Microsoft*, que foi introduzida pela primeira vez em 1993 como parte do *Microsoft Office*. O VBA é uma extensão do *Visual Basic* (VB), uma linguagem de programação amplamente utilizada para desenvolvimento de aplicativos Windows. O VBA permite a automação e personalização de aplicativos do *Microsoft Office*, como o Excel, Word, PowerPoint e Ac-

cess, através da escrita de macros[18].

Uma das principais aplicações do VBA é a automação de tarefas repetitivas em aplicativos do *Microsoft Office*. Com o VBA, os usuários podem criar macros para executar ações complexas com apenas um clique, como realizar cálculos automatizados no Excel, gerar relatórios personalizados no Word ou criar apresentações dinâmicas no PowerPoint. Além disso, o VBA permite a interação entre diferentes aplicativos do Office, permitindo a transferência de dados e o controle de um aplicativo a partir de outro.

Um exemplo de uso do VBA é a criação de uma macro no Excel para automatizar a geração de relatórios. Através do VBA, é possível escrever código para importar dados de uma fonte externa, realizar cálculos, aplicar formatação personalizada e gerar o relatório final de forma automática[19]. Isso economiza tempo e reduz erros, tornando o processo mais eficiente.

2.4.1 *Dynamic Data Exchange - DDE*

O *Dynamic Data Exchange (DDE)* é um mecanismo de comunicação entre aplicativos no ambiente *Windows*. Originalmente, o protocolo DDE foi arquitetado para permitir troca de qualquer tipo de dados, como é explicado por Rodent (1992) [20]. Ele permite que os aplicativos compartilhem dados e atualizações em tempo real, facilitando a troca de informações entre diferentes programas. O DDE foi introduzido pela primeira vez pela *Microsoft* na década de 1980 e foi amplamente utilizado em várias aplicações, especialmente em ambientes corporativos[21].

No DDE, os aplicativos são divididos em dois papéis[22]: o aplicativo de servidor e o aplicativo de cliente. O aplicativo de servidor é responsável por fornecer os dados que serão compartilhados, enquanto o aplicativo de cliente é responsável por receber e exibir os dados do servidor. A comunicação entre os aplicativos ocorre por meio de "conversações" que são estabelecidas entre eles.

O DDE possibilita a transferência de informações em tempo real entre aplicativos, permitindo que o VBA envie e receba dados de outras aplicações, como planilhas do Excel, sistemas de gerenciamento de dados, sistemas de controle industrial, entre outros. Isso torna o VBA uma ferramenta poderosa para automatizar tarefas e integrar diferentes sistemas.

Para utilizar a comunicação DDE no VBA, é necessário estabelecer uma conexão com o aplicativo desejado e definir os tópicos e itens de dados relevantes[22]. O VBA oferece funções e métodos específicos para estabelecer a conexão DDE, enviar comandos e receber dados de volta. Essa comunicação pode ser usada para acessar e manipular dados externos, executar comandos em outros aplicativos e atualizar informações em tempo real.

Um exemplo prático de uso do VBA com DDE é a criação de um painel de controle que receba informações de sensores em tempo real e atualize automaticamente os dados em uma planilha do Excel. O VBA pode estabelecer a conexão DDE com o sistema de aquisição de dados, receber os valores dos sensores e atualizar os valores nas células da planilha. Dessa forma, é possível monitorar e analisar os dados de forma contínua e automatizada.

Um dos principais benefícios do DDE é a capacidade de atualizar automaticamente os dados exibidos nos aplicativos clientes à medida que os dados no aplicativo de servidor são alterados. Isso permite que os usuários visualizem informações em tempo real e ajuda na sincronização de dados entre diferentes aplicativos.

No entanto, com o avanço da tecnologia e o surgimento de outros mecanismos de comu-

nicação, como OLE (*Object Linking and Embedding*) e COM (*Component Object Model*), o uso do DDE tem diminuído significativamente[23]. Isso se deve, em parte, às limitações do DDE, como a falta de suporte a recursos avançados e a incompatibilidade parcial com alguns ambientes operacionais 64 bits.

2.4.2 *Component Object Model/Distributed Component Object Model*

Component Object Model (COM)

O *Component Object Model* (COM)¹ recebeu esse nome quando ficou claro que o modelo seria projetado para promover a interoperabilidade dos componentes. Essa abordagem aos componentes foi adotada para encorajar os desenvolvedores a escrever de forma a permitir que os componentes possam interoperar entre si[25]. A adoção dessa arquitetura permite que os componentes possam ser integrados ao modelo imediatamente.

Além da reengenharia de toda a arquitetura, os desenvolvedores do COM também aproveitaram a oportunidade para corrigir alguns dos problemas que haviam sido identificados durante o uso do OLE1 (*Object Linking and Embedding - version 1.0*) em relação ao gerenciamento e criação de componentes.

A infra-estrutura COM foi projetada para ser mais poderosa e fazer uso da tecnologia de objeto de uso geral, em vez de adotar tecnologia específica para apenas um domínio. Um exemplo disso foi o OLE1, que se concentrou nos aspectos do documento. Como resultado da adoção dessa abordagem mais geral, o modelo é capaz de fornecer um ambiente mais poderoso para os desenvolvedores.

Uma das principais vantagens do COM é sua capacidade de oferecer serviços de objeto, permitindo que um objeto ofereça funcionalidades e recursos para outros objetos por meio de interfaces. Isso facilita a construção de aplicativos escaláveis e modulares, onde diferentes componentes podem interagir e cooperar de forma eficiente[26].

Um conceito comum que permeia a tecnologia COM é o uso de "interfaces", que permite que algumas linguagens de programação, como *Java*, *C++*, *Python* e entre outras tirem proveitos deste ambiente, assim como para o ambiente DCOM [27].

Para fazer uso da arquitetura COM, um ambiente de programação totalmente novo foi desenvolvido chamado OLE. A nova versão do OLE é uma reescrita total do OLE1. Ambos os modelos OLE compartilham muito pouco em comum. Com a nova versão do OLE fornecendo mais suporte do que apenas objetos sendo vinculados e incorporados, decidiu-se abandonar o acrônimo de *Object Linking and Embedding* que anteriormente era associado ao OLE e será abordado em 2.4.3.

Distributed Component Object Model (DCOM)

O modelo DCOM é um conjunto de extensões para o ambiente COM e foi desenvolvido e distribuído depois que o sistema COM foi implantado no mercado. O DCOM fornece comunicações de componentes distribuídos. Um dos principais objetivos da arquitetura DCOM é interceptar todas as mensagens de comunicação entre processos locais e substituí-las por um protocolo de rede baseado no *Distributed Computing Environment* (DCE). Além disso,

¹O acrônimo COM por um curto período, em 1994, significou *Common Object Model*, como resultado de uma contribuição da *Digital* e da *Microsoft*[24]

o protocolo também pode ser responsável por rotear a solicitação por meio de um ambiente de rede.

As principais vantagens do DCOM conforme descrito em *Microsoft and Digital Equipment Corporation* (1996a)[23][28] é a capacidade de:

- Fornecer um esquema de gerenciamento de memória compartilhada que pode ser usado entre os componentes;
- A forma como interage com a rede no que diz respeito à interoperabilidade da rede e à transparência das comunicações;
- A capacidade de fornecer carregamento e descarregamento dinâmico de componentes conforme necessário;
- E fornecer um rico ambiente de relatórios de erros e status para desenvolvedores.

Além disso, o DCOM fornece recursos de segurança para garantir que somente os clientes autorizados possam acessar os componentes distribuídos. Ele utiliza autenticação e criptografia para proteger as comunicações entre os clientes e servidores[29].

2.4.3 *Object Linking and Embedding*

OLE (*Object Linking and Embedding*) é uma tecnologia de integração de objetos que permite a comunicação e compartilhamento de informações entre aplicativos em um ambiente *Microsoft*. Ela foi desenvolvida pela *Microsoft* como uma estratégia de ultrapassar as limitações do DDE e é amplamente utilizada para facilitar a interoperabilidade e o trabalho em conjunto de diferentes aplicativos[30].

No modelo OLE, objetos podem ser vinculados ou incorporados em documentos ou aplicativos. A vinculação permite que um objeto seja atualizado automaticamente quando o objeto original é modificado, enquanto a incorporação torna o objeto parte integrante do documento ou aplicativo, permitindo sua manipulação sem depender do objeto original.

O OLE permite que os usuários criem documentos compostos, nos quais podem combinar informações de diferentes fontes. Por exemplo, é possível incorporar uma planilha do Excel em um documento do Word e atualizar automaticamente os dados da planilha sem sair do aplicativo do Word.

Essa tecnologia é especialmente útil em cenários onde diferentes aplicativos precisam trabalhar juntos de forma integrada. Por exemplo, em um sistema de gerenciamento de banco de dados, pode-se incorporar um gráfico criado em um programa de visualização de dados em um relatório do sistema de gerenciamento de banco de dados, permitindo a apresentação visual dos dados de forma mais eficiente, como em um ambiente Excel ou Power Point.

2.4.4 **Diferenças entre DDE x OLE x COM/DCOM**

Com os itens anteriores apresentados, é possível sumarizar a diferença entre DDE (*Dynamic Data Exchange*), OLE (*Object Linking and Embedding*), e COM/DCOM (*Component Object Model/Distributed Component Object Model*). Todas categorizam-se como tecnologias que desempenham um papel fundamental na comunicação e integração de componentes

de software em ambientes distribuídos. Embora compartilhem semelhanças, existem diferenças significativas entre elas.

DDE é uma tecnologia mais antiga desenvolvida pela *Microsoft* para permitir a troca de dados entre aplicativos em execução no sistema operacional *Windows*. Ele permite que os aplicativos compartilhem informações por meio de uma conexão de cliente/servidor. Por exemplo, um aplicativo de planilha pode enviar dados atualizados para um aplicativo de gráficos usando DDE. No entanto, o DDE tem suas limitações em termos de complexidade e segurança[22].

OLE, por outro lado, é uma tecnologia mais avançada e expansível que evoluiu a partir do DDE. OLE permite que os objetos de um aplicativo sejam vinculados (*linking*) e incorporados (*embedding*) em outros aplicativos. Por exemplo, um documento do *Microsoft Word* pode incorporar uma planilha do Excel. OLE fornece uma maneira mais flexível e rica de compartilhar dados e funcionalidades entre aplicativos. Além disso, OLE permite a manipulação direta dos objetos incorporados[31].

COM/DCOM é uma tecnologia mais abrangente que vai além da vinculação e incorporação de objetos. É uma arquitetura de componentes que define um modelo de programação para criar, distribuir e usar componentes de software. COM é usado para criar componentes reutilizáveis e definir interfaces padronizadas para a comunicação entre esses componentes. DCOM, por sua vez, estende o COM para permitir a comunicação e interação entre componentes distribuídos em redes[28][25].

Um exemplo prático do uso dessas tecnologias é a integração de um aplicativo de processamento de texto com um aplicativo de gráficos. Usando DDE, o aplicativo de processamento de texto pode enviar dados atualizados para o aplicativo de gráficos para criar gráficos dinâmicos. Com OLE, o aplicativo de processamento de texto pode incorporar diretamente o gráfico em seu documento. Já com COM/DCOM, os componentes de processamento de texto e gráficos podem se comunicar e compartilhar dados de maneira mais eficiente em um ambiente distribuído

2.5 Conceitos Principais sobre RPA

Em síntese, o *Robotic Process Automation* (RPA) é um software, um ambiente de desenvolvimento, que oferece a seus usuários a capacidade de automatizar tarefas digitais de maneira simples [32]. O RPA também pode ser entendido como uma tecnologia responsável por emular o comportamento de execução de processos digitais que envolvam um ou mais sistemas em sua camada de apresentação, mediante a criação de robôs virtuais ou *Bots*. É importante salientar que a intenção principal da ferramenta é dar suporte à criação de uma força de trabalho digital que seja capaz de interagir com qualquer aplicação ou sistema da mesma forma que seres humanos o fazem. Ressaltando a diferença, visto que um agente digital é responsável por executar diversos *Bots* (podendo ser de áreas diferentes) e, além disso, ele não descansa e possui precisão em sua execução evitando erros humanos.

Com intuito de facilitar o entendimento sobre o processo de automação de processos realizado por meio da ferramenta RPA, a seguir, apresenta-se as definições dos principais conceitos relacionados:

- **Bot:** ou robô virtual, é um *script* virtual desenvolvido por um usuário na ferramenta de RPA, com o objetivo de executar alguma tarefa ou processo digital.

- **Arquitetura de RPA:** configuração do ambiente integrado que roda todos os processos, cada companhia cliente da ferramenta de RPA é responsável por definir e manter uma arquitetura da solução do sistema.
- **Sala de controle virtual:** servidor de controle virtual, além disso seu nome tem contato com o conceito de uma sala de controle fabril, porque é o servidor responsável por orquestrar, auditar e garantir o funcionamento da operação integrada entre as máquinas de RPA. Ou seja, necessita de apresentar ao administrador do ambiente de RPA a saúde dos agentes digitais (computadores virtuais) que rodam os *Bots* (processos entre sistemas).
- **Agente digital:** computadores virtuais (execução 24x7) responsáveis pela força de trabalho digital, é o único capaz de executar robôs virtuais (*Bots*) entre sistemas. Ou seja, neste documento será definido que um agente digital pode possuir diversos *Bots*, por exemplo: um agente digital tem a capacidade de executar processos de Recursos Humanos e em outro momento, executar processos tributários.
- **Ambiente de desenvolvimento:** dispositivo (físico ou virtual) em que o software RPA contratado para se fazer o desenvolvimento da automação de processos está instalado. Existem diversas fornecedoras desta tecnologia no mercado que ao serem contratadas, disponibilizam o ambiente de desenvolvimento integrado para esta automação.
- **Mapeamento de processo:** precursor para toda automação de processo, trata-se da tarefa inicial para toda criação de um novo robô virtual é o mapeamento de processo.

2.5.1 Objetivos e benefícios do sistema RPA

Os sistemas virtuais de RPA possuem objetivos claros para otimizar e gerar a automação de processos digitais advindos de uma ferramenta digital. Já o RPA tem como objetivo final a simplificação, agilidade, escalabilidade e segurança para o desenvolvimento desta automação de processos.

Sendo assim, as organizações responsáveis pela aquisição da ferramenta têm benefícios como:

- a. Aumento da produtividade;
- b. Aumento da eficiência;
- c. Redução de custos;
- d. Maior atribuição de tarefas analíticas ao empregado.

O primeiro benefício citado advém da aceleração de fluxos de trabalho e da permissão da tecnologia para a execução de processos simultâneos. Ou seja, mais processos poderão ser executados com a mesma equipe, sintetizando a máxima de “fazer mais com menos”.

O segundo benefício pode ser justificado, uma vez que os *bots* são *scripts* que não estão sujeitos a alterações momentâneas e que, portanto, executam repetidamente o que foram programados para fazer. Sendo assim, garantem boa performance e principalmente um aumento da eficiência.

A redução de custos ocorre devido a especificidade dos bots, visto que permitem seu reuso e sua execução em paralelo de processos iguais ou não. Portanto, devido à alta escalabilidade da ferramenta, é possível que haja uma redução de custos da companhia para execução de processos existentes.

O último benefício listado ocorre por conta da realidade que diversas funções exercidas por empregados ao manusear diferentes sistemas, requerem muita ação manual e repetitiva, sem que haja necessidade de análise e um posicionamento cognitivo. Portanto, ao atribuir essas tarefas aos *bots*, a organização possui a possibilidade de atribuir tarefas analíticas ao empregado, deixando-os com a parte cognitiva e inteligente da execução de tarefas.

2.6 Aplicações de RPA na indústria/nas empresas

Sendo uma tecnologia recente, o *Robotic Process Automation* se baseia na criação de robôs virtuais (*Bots*) para integração de diversos sistemas, com a prerrogativa de emular um comportamento humano composto por tarefas repetitivas, de alto volume, e sob regras bem definidas. Portanto, este tópico apresenta referências na área de RPA que envolvem soluções trazidas por esta nova tecnologia, bem como aplicações no mercado atual.

Em [33], o autor analisa os benefícios econômicos e sociais trazidos pela implementação de um RPA na companhia Energias de Portugal, S.A. (EDP). No referido trabalho, um estudo de caso da ferramenta na área de Recursos Humanos para cadastro de informações de novos empregados e atualizações, daqueles já existentes, em um sistema de gestão corporativo da EDP. Deve-se ressaltar que este tipo de tarefa é totalmente manual e, anteriormente à implantação do RPA, a execução sempre ficou sob responsabilidade de um analista. Sendo assim, percebe-se que, com o avanço da tecnologia, esta ação rotineira torna-se algo desnecessário e tem-se a possibilidade de automatizar a tarefa, ou seja, o cadastro manual, gerando economias em horas gastas com esse processo e permitindo ao mesmo analista aumentar sua produtividade como ser pensante, utilizando de sua criatividade, análise crítica e pensamento cognitivo, evitando a execução repetitiva de ler dados de uma fonte e copiar para o sistema de gestão qualquer.

O uso de RPA tem se tornado realidade em diversos segmentos de negócios, tais como finanças e contabilidade, recursos humanos, cadeia de suprimentos e tecnologia da informação. Sempre considerando, conforme dito anteriormente, a automação de tarefas repetitivas, com regras bem definidas, manuais e extensas. A Figura 2.1, a seguir, apresenta um gráfico relacionando os possíveis casos de automação de funções e tarefas, passíveis de receberem a tecnologia do RPA.

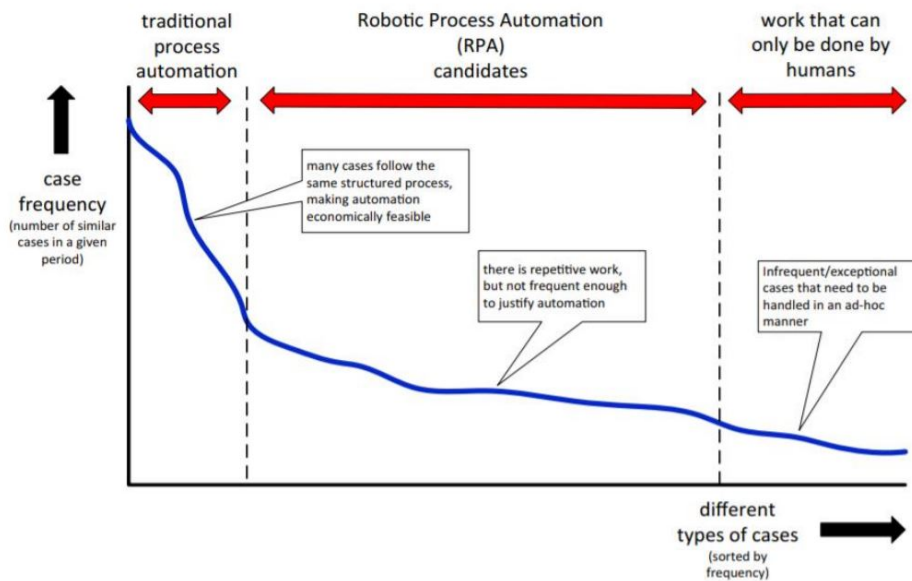


Figura 2.1: Gráfico de casos de automação de processos. Fonte: [34]

A Figura 2.1, apresenta um gráfico dos casos passíveis de automação de processo. No eixo das ordenadas a variável é a frequência de execução dos processos (números similares de casos num período estudado), já no eixo das abscissas a variável representa os diferentes tipos de casos (ordenados pela frequência). Na primeira divisão da figura, apresenta os processos tradicionais de automação, em que são justificados pelo comentário de que a maioria dos casos nesta divisão são processos estruturados, permitindo a viabilidade econômica da automação. Na segunda divisão, os candidatos à tecnologia RPA são trazidos à tona, com a justificativa de que há trabalho repetitivo, mas não possuem uma frequência alta, justificando a automação. Por fim, na última divisão os trabalhos que apenas humanos são capazes de executá-los, infrequentes e com muitas exceções às regras que fazem com que as tarefas precisam ser tratadas manualmente em cada situação.

Outro aspecto relevante sobre o RPA é a expressiva dificuldade para mensurar o real impacto que pode trazer para a organização que visa sua implantação. Entretanto, em [35], tem-se um estudo aplicado em que foram analisados e mensurados os resultados pós-implantação de um RPA, levando-se em consideração esta dificuldade de se medir tal impacto. O processo candidato à automação pelo RPA envolve a emissão de notas fiscais em uma empresa do ramo metalúrgico e a conclusão do trabalho aponta para uma implementação do RPA ainda precária, devido a diversos fatores, com destaque para a dificuldade de mapeamento de processos, em que o desenho e a concepção da demanda pré-projeto se mostra uma possível barreira para implantação. Portanto, a tecnologia ainda encontra dificuldades, contudo, se a concepção da demanda, proveniente do mapeamento de possíveis processos candidatos, for bem realizada e um planejamento bem-feito, o RPA pode ser bem-sucedido.

2.7 Estrutura de Hardware de um Controlador Lógico Programável

Para entender os conceitos associados à estrutura de *Hardware* de um Controlador Lógico Programável (CLP), é necessário entender o que é um CLP em primeiro lugar. De maneira sucinta, um CLP pode ser definido como um computador com diversas peças de *hardware* associadas entre si, ou já integrado no próprio.

Dentre os componentes de hardware que podem integrar um CLP, estão [36]:

- **Backplane**[37]: O *Backplane* pode ser entendido como a “espinha dorsal” de um CLP. É onde diferentes tipos de componentes (Processador, Cartões de entrada e saída e cartões de comunicação) são interligados para troca de dados entre si.
- **Fonte de Alimentação**: Como o nome sugere, é quem fornece alimentação para o Processador. É responsável por fazer a conversão de alimentação AC para DC.
- **Módulos de entrada e saída**: São responsáveis de interligar dispositivos digitais ou analógicos, de campo, com o Processador. Como exemplos de módulos de entradas, tem-se: *Switches*, *Encoders* e Transmissores. Como dispositivos de saída, tem-se: Relés, lâmpadas e válvulas.
- **Processador**: É composto pela unidade de processamento central (CPU) e a memória (responsável por armazenar os valores de variáveis). Em termos gerais, é a unidade responsável por processar as lógicas de controle de dispositivos de campo.
- **Dispositivo de Programação**: Ambiente onde os projetos de CLP's são desenvolvidos. Para a uma aplicação Rockwell, tem-se o *Studio 5000*. Em um ambiente Siemens, a equivalência seria o TIA-Portal ou SIMATIC PCS 7.

Os conceitos acima podem ser visualizados conforme a figura abaixo:

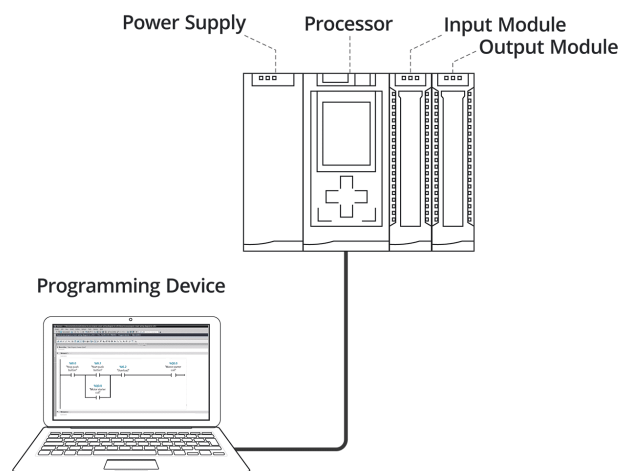


Figura 2.2: Estrutura de *Hardware* de um CLP. Fonte: [36]

2.7.1 Conceitos Chave: Lista de Entrada/Saída

Um conceito pertinente a ser entendido é a lista de instrumentação de um projeto, conhecido como Lista de Entrada e Saídas, ou “Lista de I/O”. A lista de I/O referencia todas as informações enviadas ou recebidas pelo CLP. Ela possui um papel fundamental na elaboração de um projeto de automação, pois é a fundação para dimensionamento do controlador a ser utilizado, além de fornecer uma referência em relação ao número de dispositivos a serem controlados e monitorados[38].

2.7.2 Conceitos Chave: Formato de Endereçamento Físico de dispositivos

Outro ponto pertinente que se relaciona à Estrutura de *Hardware* de um CLP é a estrutura de endereçamento de dispositivos no ambiente Rockwell. Para sinais de entrada e saída digitais, o formato do endereçamento assemelha-se à:

- Local:6:I.Data.X
- Local:6:O.Data.X

Já para as entradas e saídas analógicas assemelham-se à:

- Local:4:I.ChXData
- Local:4:O.ChXData

Em cada um dos formatos, o “X” representa a posição no cartão de entrada/saída. Em um cartão de 8 entradas ou saídas, esse valor varia de 0-7.

Em um ambiente Siemens[39], o formato de endereçamento é diferente do mostrado acima. Sinais binários podem ser representados conforme:

- Input: %I1.0
- Output :%Q1.0
- Memory: %M10.8
- DBs: %DB1.DBX11.3

Palavras (inteiros de 16-bits) são representadas como:

- Input: %IW10
- Output: %QW10
- Memory: %MW14
- DBs: %DB1.DBW11

Enquanto palavras duplas (Inteiros de 32-bits ou Real) são representados:

2.7. ESTRUTURA DE HARDWARE DE UM CONTROLADOR LÓGICO PROGRAMÁVEL17

- Input: %ID10
- Output: %QD100
- Memory: %MD200
- DBs: %DB1.DBD288

Capítulo 3

Metodologia

Nesta seção, são abordados primeiramente os recursos de *software* empregados no trabalho, seguidos pela descrição do modelo de projeto de automação a ser utilizado como base para a construção da solução e pela estratégia empregada na idealização da aplicação que cumpre o objetivo deste trabalho.

3.1 Recursos Empregados

Durante o desenvolvimento do projeto, foram empregadas algumas ferramentas de *software*. São elas:

- Rockwell Automation RSLinx Classic[40];
- Rockwell Automation Studio 5000@Logix Emulator[41];
- Rockwell Automation RSLogix 5000 v20.05[42];
- *Microsoft Excel* 2016;
- *Microsoft Visual Basic for Applications* 2016;
- VMWare Workstation Player [43];
- *Notepad++* [44];

Visando concentrar os recursos de *software* descritos anteriormente em um único ambiente de desenvolvimento, foi configurada através do programa VMWare uma máquina virtual, ou seja, uma simulação de um ambiente computacional com capacidade de executar as mesmas tarefas que uma máquina física. Esta máquina possui Windows 7 de 64 bits e tem instalados todos os programas mencionados.

3.2 Concepção do projeto baseado em um estudo de caso

A finalidade do projeto elaborado é ser utilizado no dia-a-dia da empresa, assim é necessário que este fizesse-se presente no mesmo. Baseado em um projeto ambientado completamente em uma plataforma Rockwell, a necessidade de validação do endereçamento na etapa

de testes internos é de extrema importância. Atividades de alto grau de intervenção humana ficam sujeitas a erros[7] e pode gerar uma carga excessiva de retrabalho.

A demanda e o impacto de eventuais erros crescem proporcionalmente a lista de E/S. Projetos que contemplam mais de uma área ou um diversas sistemas interligados, costumam ter uma vasta lista de dispositivos de sensoriamento, que incluem:

- Entrada/Saída Digitais;
- Entrada/Saída Analógicos;
- RTD;
- HART.

Entendido estes conceitos e partindo de uma abordagem generalista, o projeto visa a capacidade de interoperabilidade com qualquer tipo de lógica operacional, desde que seja em um ambiente com uso de CLPs Rockwell.

3.3 Comunicação entre as aplicações

A comunicação entre os dispositivos pode ser visto de maneira geral conforme a Figura 3.1.

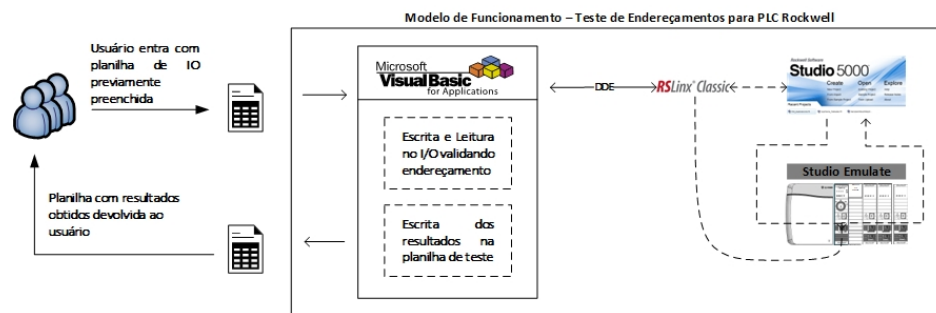


Figura 3.1: Visão geral para comunicação e funcionamento do projeto. Fonte: Elaborado pelo Autor.

O projeto foi fundamentado, diante de todas as possibilidades apresentadas no Capítulo 2, no protocolo DDE. A opção pela utilização do DDE vem da capacidade de integração não apenas com o ambiente *Visual Basic for Application*(VBA), mas também pela capacidade de funcionar nativamente - dentro das devidas configurações - com o *RSLinx Classic*.

Embora a versão mais atual - *FactoryTalk Linx Gateway* [45] - incorpore bem as funcionalidades já existentes no *RSLinx Classic* além de trazer novos recursos, o *RSLinx Classic* possui uma boa otimização e trabalha bem em ambientes operacionais com baixos recursos computacionais alocados. Isso permite que, independentemente da versão do controladores, ou do ambiente operacional, a aplicação desenvolvida tenha altas chances de compatibilidade e sucesso.

Na Figura 3.17, é indicado uma aba para comunicação DDE/OPC. É nela onde fazemos as configurações para que a comunicação com a aplicação DDE no ambiente VBA seja possível. No entanto, para validar a comunicação, foi necessário configurar um programa teste, com *Tags* de Entrada/Saída. Os tópicos 3.3.1 e 3.3.2 mostram estas etapas.

3.3.1 Configuração *Studio 5000®Logix Emulator*

A configuração para o *Rockwell Automation Studio 5000®Logix Emulator* consiste em instanciar um *hardware* virtual que mimetiza um CLP físico. A Figura 3.2 mostra a interface de inicialização da aplicação. Replicando um *rack*, com seus respectivos *slots* e *backplane* - meio físico por onde cartões são energizados e comunicam-se entre si[46]-, permite simular a comunicação com a aplicação que contém as respectivas *tags* e endereços físicos.

As Figuras 3.3 até 3.6 mostram o processo de instanciação da *CPU* virtual. As Figuras 3.6 até a figura 3.10 mostram o processo de declaração de um cartão virtual de 32 pontos de entrada e saída, além da configuração final utilizada no ambiente *Logix Emulator* para o Projeto Final de Curso.

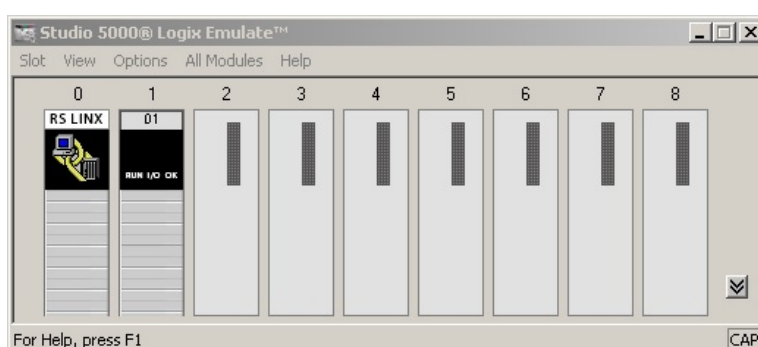


Figura 3.2: Interface inicial *Rockwell Automation Studio 5000®Logix Emulator*. Fonte: Autor.

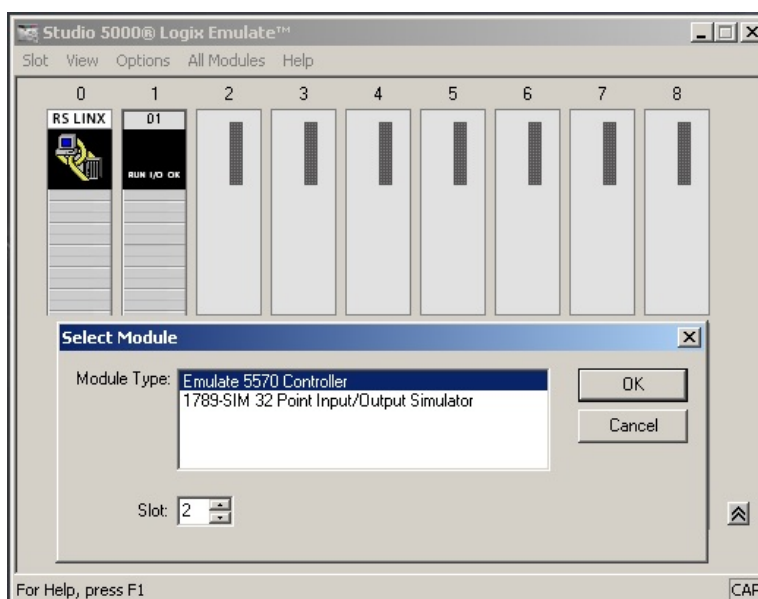


Figura 3.3: Processo de declaração da *CPU* Virtual. Fonte: Autor.

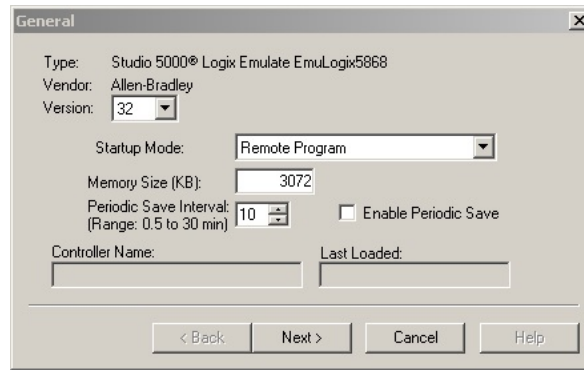


Figura 3.4: Processo de declaração da CPU Virtual. Fonte: Autor.

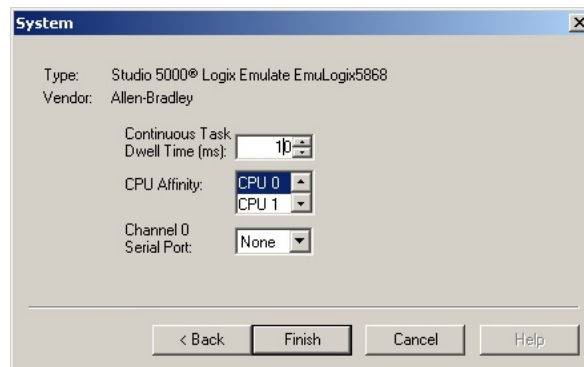


Figura 3.5: Processo de declaração da CPU Virtual. Fonte: Autor.

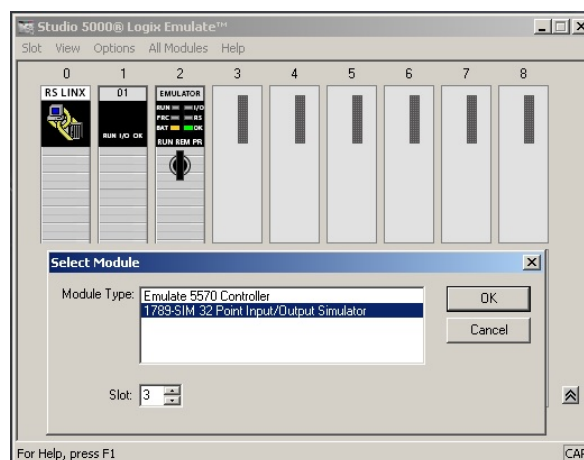


Figura 3.6: Processo de declaração do cartão de E/S. Fonte: Autor.

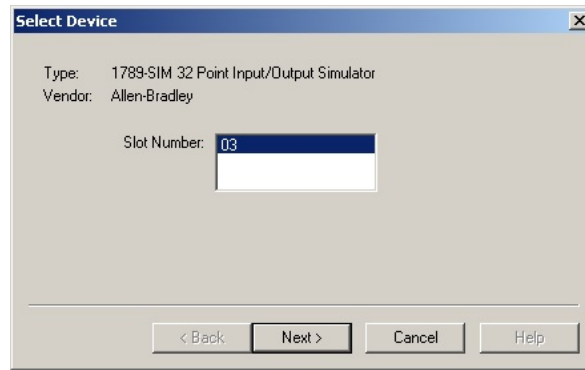


Figura 3.7: Processo de declaração do cartão de E/S. Fonte: Autor.

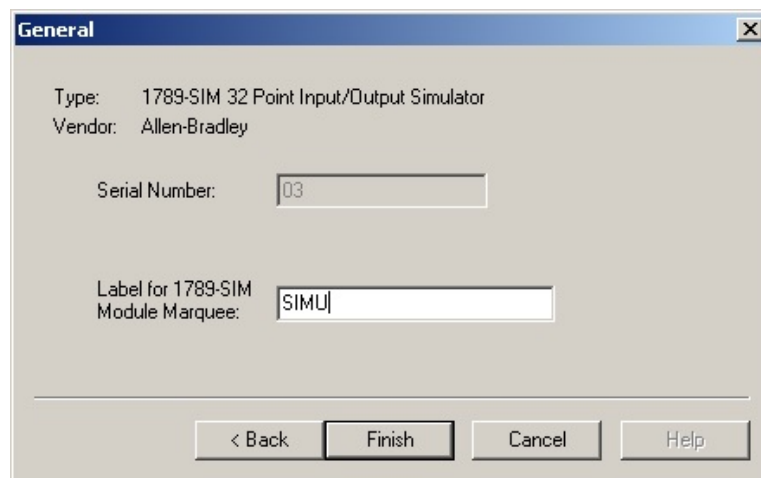


Figura 3.8: Processo de declaração do cartão de E/S. Fonte: Autor.

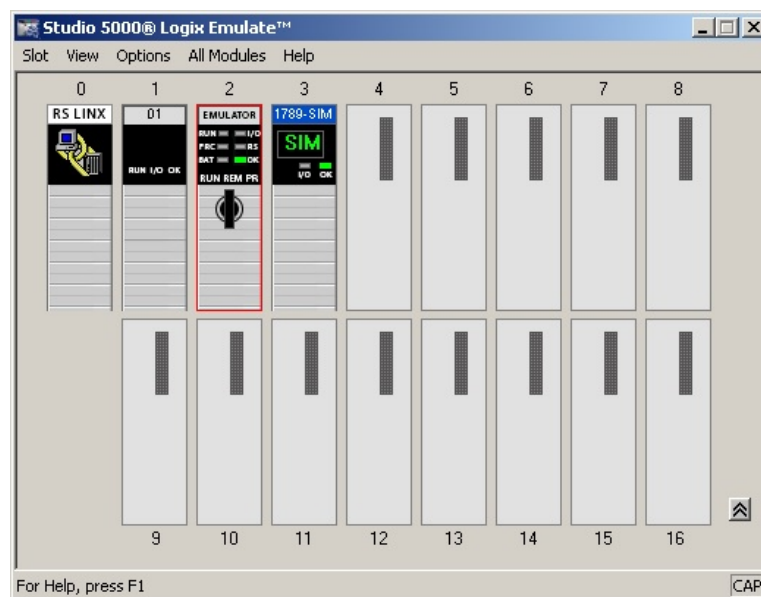


Figura 3.9: Processo de declaração do cartão de E/S. Fonte: Autor.

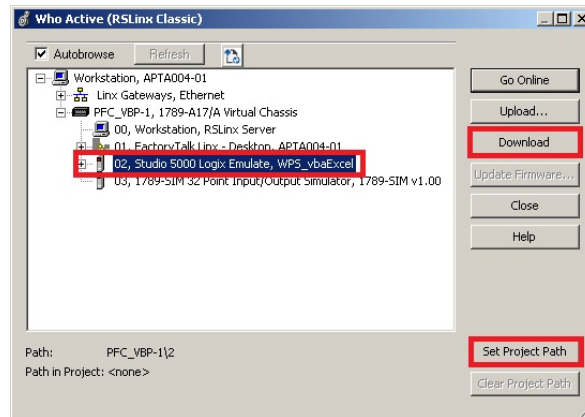


Figura 3.12: *Logix Designer 2* - Passo a passo para entrar online com a aplicação. Fonte: Autor.

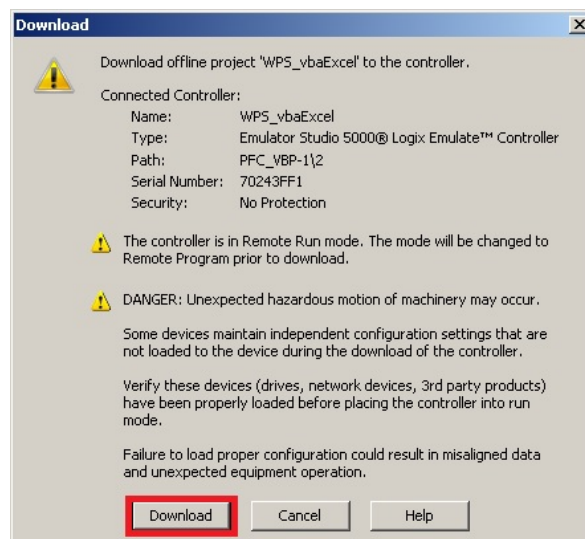


Figura 3.13: *Logix Designer 3* - Passo a passo para entrar online com a aplicação. Fonte: Autor.

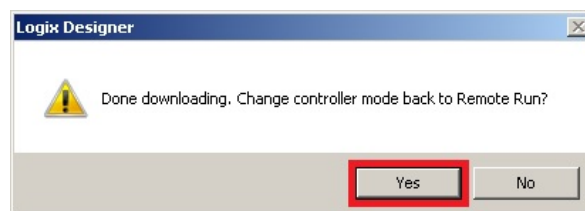


Figura 3.14: *Logix Designer 4* - Passo a passo para entrar online com a aplicação. Fonte: Autor.

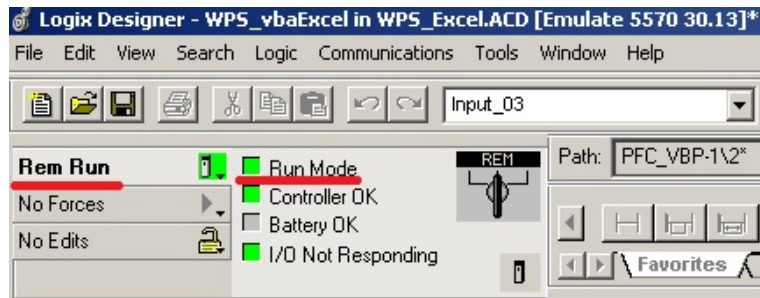


Figura 3.15: *Logix Designer 5* - Passo a passo para entrar online com a aplicação. Fonte: Autor.

3.3.3 Configuração *RSLinx Classic*

Com as configurações de CLP feitas, para que a comunicação DDE seja estabelecida, são necessários três argumentos, como descrito em [47]. São eles:

1. *Application*: *String* identificando a aplicação que pode participar de uma comunicação DDE. Usualmente é definido como a própria instância *.exe*. Para o projeto, a instância declarada é *RSLINX*.
2. *Topic*¹: *String* que representa o nome do tópico reconhecido pela *Application*. O processo para criação do *topic* é composto pelas Figuras 3.16, 3.17 e 3.18.
3. *Item*: *String* que representa o nome do dado reconhecido pela *Application*. Para o projeto, ela é representada tanto por *tags* quanto os endereços físicos do controlador.

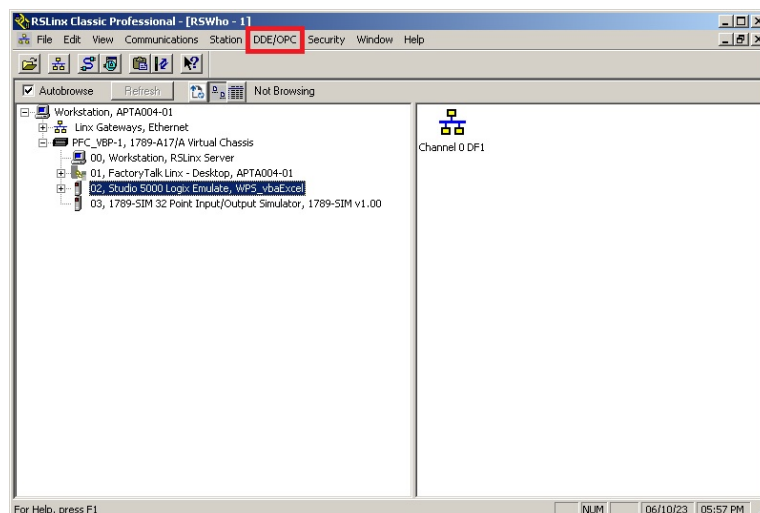


Figura 3.16: Interface inicial *RSLinx Classic*. Fonte: Autor.

¹Conforme será abordado em 4.1.1, é sugerido ao corpo técnico que a criação do *Topic* dentro *RSLinx Classic* seja feito com o nome do CLP a ser testado.

3.4. POSSÍVEIS CONFIGURAÇÕES DA ÁRVORE DE HARDWARE E USO DE TAGS LOCAIS 27

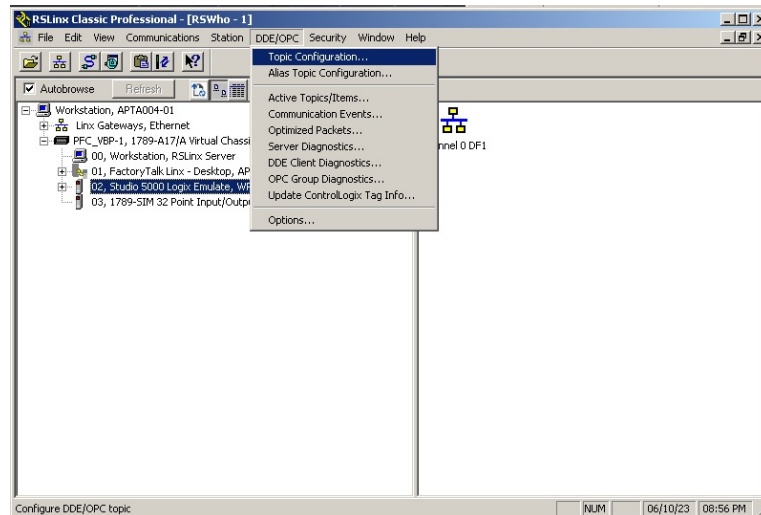


Figura 3.17: DDE *Topic Configuration* para o *RSLinx Classic*. Fonte: Autor.

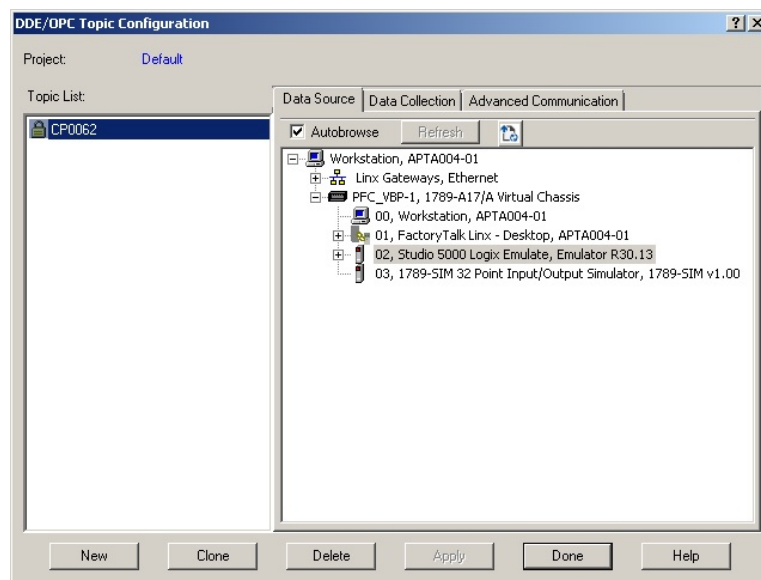


Figura 3.18: Criado um novo *topic* com o caminho apontado para o CLP emulado, no *Virtual Backplane*. Fonte: Autor.

3.4 Possíveis configurações da *Árvore de Hardware* e uso de *Tags Locais*

Um conceito a se entender antes de iniciar o teste de endereçamento, é perceber como a estrutura de *hardware* utilizada influencia sobre a nomenclatura do endereço físico de um cartão de Entrada/Saída.

Em plantas/projetos em que um CLP encontra-se distante do conjunto de cartões de entradas/saídas, a comunicação costuma ocorrer da seguinte maneira:

1. Um cartão Ethernet/IP comunica-se com um conversor Ethernet/IP - Fibra Óptica, como um 1783-ETAP[48];
2. O sinal via fibra é convertido novamente, utilizando o mesmo modelo de cartão, para Ethernet/IP e enviado para um cartão Ethernet;
3. O cartão Ethernet/IP, como o um cartão 1756-EN2T [48], comunica-se com os cartões de Entrada/Saída por meio do *backplane* do rack.

Neste caso, a estrutura de hardware assemelharia-se a imagem mostrada em 3.19.

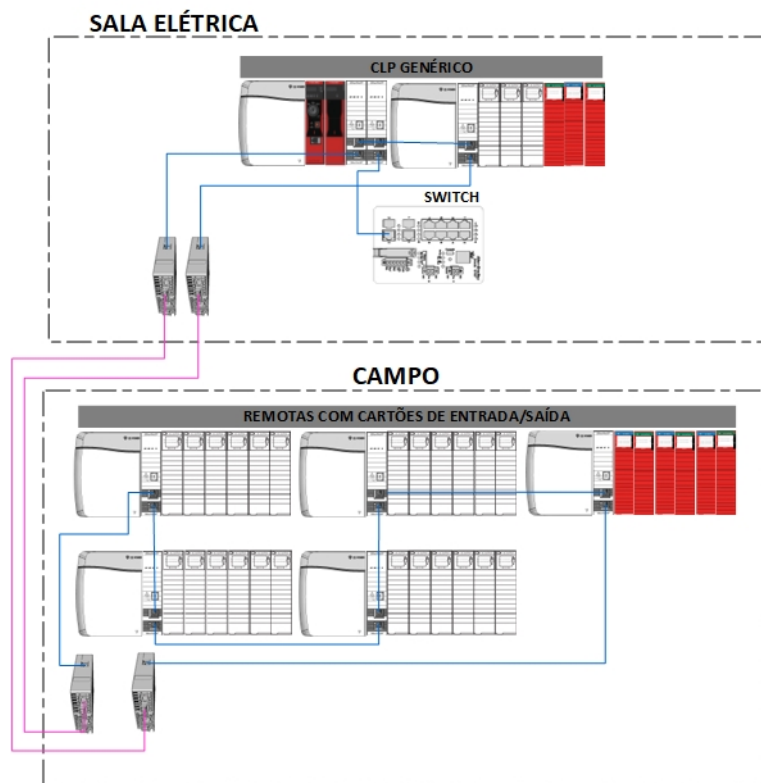


Figura 3.19: Modelo de *Árvore de Hardware*. Fonte: Elaborado pelo Autor.

A estrutura de *hardware*, dentro do ambiente *RSLogix 5000* assemelha-se à 3.20 e o endereçamento físico assemelha-se à 3.21.

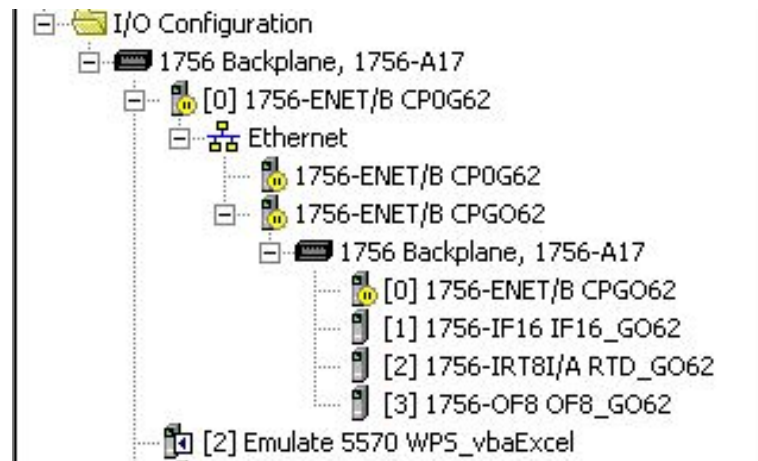


Figura 3.20: Visão da árvore de *Hardware* visto dentro do ambiente *RSLogix Designer*.
Fonte: Elaborado pelo Autor.

CP60621		AB:1756_ENET_17SL0T1:0
CP60620		AB:1756_ENET_17SL0T0:0
CP60621C		AB:1756_IF8_Float:C:0
CP60621:1		AB:1756_IF8_Float:I:0
CP60622C		AB:1756_IRT8I:C:0
CP60622:1		AB:1756_IRT8I:I:0
CP60622:0		AB:1756_IRT8I:O:0
CP60623C		AB:1756_OF8_Float:C:0
CP60623:1		AB:1756_OF8_Float:I:0
CP60623:0		AB:1756_OF8_Float:O:0

Figura 3.21: Visão do endereçamento equivalente à configurações 3.19 e 3.20. Fonte: Elaborado pelo Autor.

Em uma configuração mais tradicional, em que os cartões de Entrada/Saída encontram-se no mesmo *rack* do CLP, como mostrado na Figura 3.22, a árvore de *hardware* seria representada, genericamente, como mostrado pela Figura 3.23 e a estrutura de endereçamento aproxima-se de 3.24.



Figura 3.22: Arquitetura tradicional de *Hardware* - Cartões de Entrada/Saída no mesmo *rack* do CLP. Fonte: Autor.



Figura 3.23: Visão da *Árvore de Hardware* visto dentro do ambiente *RSLogix Designer*.
Fonte: Elaborado pelo Autor.

Local:3:C			AB:1756_MODULE:C:0
Local:3:I			AB:1756_MODULE_DINT_8Bytes:I:0
Local:3:O			AB:1756_MODULE_DINT_4Bytes:O:0
Local:4:C			AB:1756_IF8H:C:0
Local:4:I			AB:1756_IF8H_Analog:I:0
Local:5:C			AB:1756_OF8H:C:0
Local:5:I			AB:1756_OF8H_Analog:I:0
Local:5:O			AB:1756_OF8H:O:0
Local:6:C			AB:1756_DI:C:1
Local:6:I			AB:1756_DI:I:0

Figura 3.24: Visão do endereçamento equivalente às configurações 3.22 e 3.23. Fonte: Elaborado pelo Autor.

3.5 Planilha de Testes Internos

Uma das maneiras de assegurar um controle de qualidade dos testes realizados e aprovados nos *softwares* de CLPs desenvolvido pela Primetals, está na utilização de uma planilha para controle de testes internos. A planilha possui um formato padrão que é compatível com qualquer possível área de programação ou interface de equipamento sendo utilizado, seja *Siemens*, *Rockwell*, *Emerson* ou *ABB*.

Conforme relato por [49], o processo de aceitação da automatização de processos é composto por itens que contribuem para o sucesso da aplicação. Eles são descritos no esquema mostrado na Figura abaixo.



Figura 3.25: Esquema de aceitação do uso de ferramentas Automatizadas. Fonte: [49].

As *Expectativas Realistas* traduzem-se na forma que o corpo técnico entende sobre a capacidade de teste da aplicação. Para isso, discussões prévias foram realizadas com alguns membros da equipe para entender a percepção sobre a aplicação.

A *Influência do Usuário* enquadra-se nas sugestões de funcionalidades que o usuário faz durante o período de desenvolvimento da aplicação. Usuários que participam do desenvolvimento, tendem a ser menos resistentes quanto a uso da aplicação.

O *Treinamento* consiste na capacitação de uso da ferramenta. A depender do grau de complexidade associada ao uso ou etapas de uso da aplicação, o usuário pode ser desmotivado a fazê-lo.

O *Comprometimento da Gerência* trata do incentivo que a nível gerencial da equipe faz sobre o uso da ferramenta. A percepção de aceite e de uso da ferramenta pelo grupo destinado a ela é maior com um incentivo dos níveis mais altos de gerência da empresa ([50] e[51]).

Considerando os conceitos apresentados, o modelo de planilha de testes elaborada, utiliza um modelo já conhecido pelo corpo técnico e utilizado no dia-a-dia. Pequenas alterações foram realizadas, sobre influência do usuário ². O novo modelo, com alterações para atender a Automatização de Testes de Endereçamentos, é mostrado pela Figura 3.26.

²Durante a etapa de testes para validação da aplicação, foram aceitas opiniões e sugestões para atender possíveis expectativas e criar uma familiaridade com o desenvolvimento, de forma a facilitar o uso da mesma.

Especificação Funcional do Equipamento										Testar Endereçamento		
Nome:				PLC:						CP8002		
Ativo:												
Tipo:												
Resolução:												
Endereço:												
Endereço (Hexadecimal):												
Execução dos Funcionamentos no Equipamento												
Equipamento	Doc. Referência	Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK	OK/NOK	X	Observação	
ES_000002		Local#1 Data 0	LSL_000002_Inc_PV	Alarmes	DI	NÍVEL DO TANQUE 0		OK				
ESL_000002		Local#1 Data 1	LSL_000002_Inc_PV	Alarmes	DI	NÍVEL DO TANQUE 1		OK				
PSI_000002		Local#1 CO#Data	PSI_000002_Inc_PV	Extensão analógica	AI	SENSOR DE PRESSÃO (PSI#0002)		OK				
PSI_000002		Local#1 CO#Data	PSI_000002_Inc_PV	Extensão analógica	AI	SENSOR DE PRESSÃO (PSI#0002)		OK				
TE_000002		Local#1 CO#Data	TE_000002_Inc_PV	Extensão analógica	AI	TEMPERATURA DO ENFRIAMENTO S		OK				
TE_000002		Local#1 CO#Data	TE_000002_Inc_PV	Extensão analógica	AI	TEMPERATURA DO ENFRIAMENTO S		OK				
TE_000002		Local#1 CO#Data	TE_000002_Inc_PV	Extensão analógica	AI	TEMPERATURA DO ENFRIAMENTO T		OK				
VE_000002		Local#1 CO#Data	VE_000002_Inc_PV	Extensão analógica	AI	VELOCIDADE		OK				
VE_000002		Local#1 CO#Data	VE_000002_Inc_PV	Extensão analógica	AI	VELOCIDADE		OK				
RI_000002		RACK00FC_11 CO#Data	Programa#RI_000002_Inc_PV	Extensão analógica	AI	MEDIDOR DE VAZÃO DE RESFRIAMENTO		OK				
RI_000002		RACK00FC_11 CO#Data	Programa#RI_000002_Inc_PV	Extensão analógica	AI	MEDIDOR DE VAZÃO DE RESFRIAMENTO		OK				
RI_000002		RACK00FC_21 CO#Data 0	Programa#RI_000002_Inc_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO		OK				
RI_000002		RACK00FC_21 CO#Data 1	Programa#RI_000002_Inc_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO		OK				
RI_000002		RACK00FC_21 CO#Data 2	Programa#RI_000002_Inc_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO		OK				
RI_000002		RACK00FC_11 CO#Data	Programa#RI_000002_Inc_PV	Extensão analógica	AI	MEDIDOR DE TEMPERATURA PRÉ RESFRIAMENTO		OK				
RI_000002		RACK00FC_11 CO#Data	Programa#RI_000002_Inc_PV	Extensão analógica	AI	MEDIDOR DE TEMPERATURA PRÉ RESFRIAMENTO		OK				
RI_000002		RACK00FC_11 CO#Data	Programa#RI_000002_Inc_PV	Extensão analógica	AI	MEDIDOR DE TEMPERATURA PRÉ RESFRIAMENTO		OK				

Figura 3.26: Modelo da Planilha de Testes de Equipamentos e Endereçamento. Fonte: Elaborado pelo Autor.

Dentre os campos apresentados na Figura 3.26, os de maior importância para o funcionamento da aplicação, por depender diretamente do seu preenchimento, são os campos:

- Endereço(*Tag*);
- *Tag* Proposto;
- Tipo.

Os itens citados fazem-se de tal importância dado a forma de funcionamento da aplicação.

- O campo *Tipo* irá definir o tipo de valor que será enviado para o teste, podendo ser um valor real, inteiro ou binário (1 ou 0);
- Os campos *Endereço(Tag)* e *Tag Proposto* funcionam como o mecanismo de verificação. Se um endereço físico de uma *tag* está incorreto, o campo Endereçamento (OK/NOK) será marcado. Se a *tag* do endereço físico está incorreta, o mesmo ocorre.

As premissas de funcionamento serão apresentadas a fundo no próximo capítulo(4.1), onde serão apresentados resultados tanto do preenchimento adequado quanto do preenchimento “inadequado” da planilha 3.26.

Capítulo 4

Discussão

Para a execução do projeto, algumas premissas foram assumidas. Este capítulo busca explorá-las, apresentar os resultados e dificuldades encontradas ao longo dos testes realizados para validação da aplicação.

4.1 Premissas de funcionamento adequado da Macro

Conforme citado em 3.5, o preenchimento adequado da planilha é o primeiro passo para um bom funcionamento da aplicação. No entanto, por ainda depender de uma interferência humana para o preenchimento (explorado em 5.2), os campos estão suscetíveis à eventuais erros de preenchimento, como é abordado por [7].

Assim sendo e prevendo possíveis cenários como o descrito acima, a aplicação possui alguns mecanismos que asseguram que o teste seja realizado em sua totalidade, mesmo com o preenchimento incorreto da planilha.

4.1.1 Preenchimento dos Campos

O primeiro cenário trata do preenchimento do campo “PLC”. A aplicação foi preparada para definir o *Topic* 3.3.3, como o nome do próprio CLP em teste. Em caso de não preenchimento ou preenchimento incorreto, uma mensagem será mostrada ao usuário, conforme Figuras 4.1 e 4.2

```
Public Function OpenRSLinx()  
  
Dim topic As String  
  
topic = Range("G5").Text  
  
On Error Resume Next  
'Abre conexao com o RSLinx  
  
OpenRSLinx = DDEInitiate("RSLinx", topic)  
'verifica se comunicacao foi estabelecida  
  
If Err.Number <> 0 Then
```

```

MsgBox "Error ao conectar-se ao Topic RSLinx",
    vbaExclamation, "Error"
OpenRSLinx = 0 'Retorna falso em caso de erro
End If

End Function

```

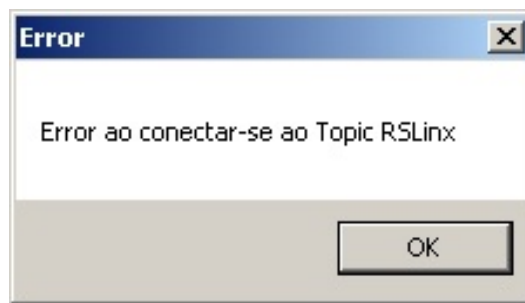


Figura 4.1: Erro ao criar o *Topic*, nomenclatura incorreta. Elaborado pelo Autor.

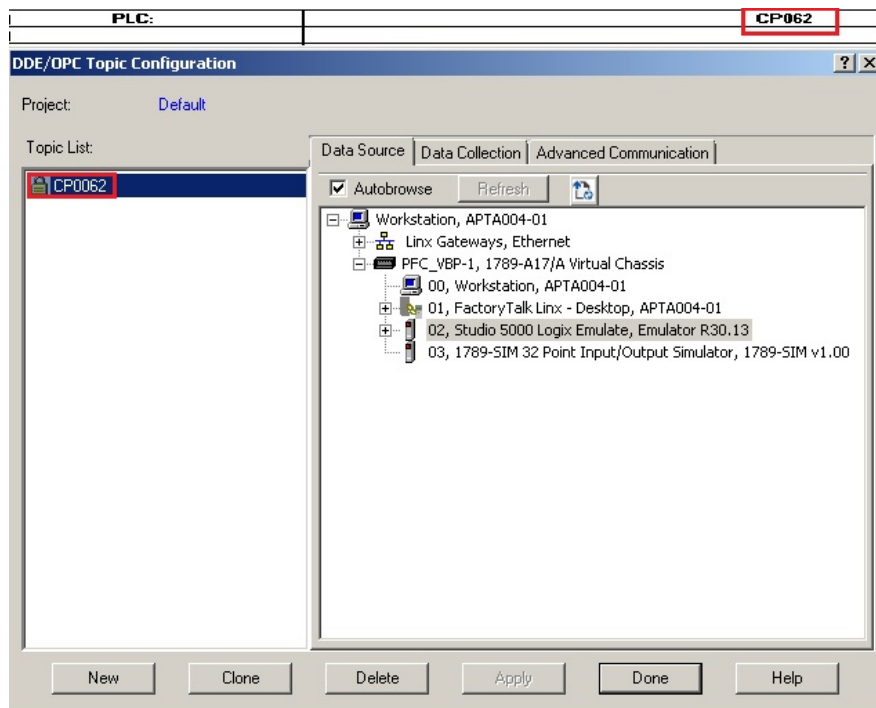


Figura 4.2: Campo preenchido *versus* *Topic* criado. Elaborado pelo Autor.

O segundo cenário trata de um eventual esquecimento por parte do corpo técnico em preencher um dos campos. Para isso, o código verifica se todos os campos foram preenchidos antes de iniciar o teste, conforme mostrado abaixo.

```

If IsEmpty(AddressRange.Item(i, 1)) And
    IsEmpty(TagRange.Item(i, 1)) And IsEmpty(TypeRange.Item(i,
1)) Then

```

```

Exit For
Else
  If IsEmpty(TagRange.Item(i, 1)) Then
    AddressingOkCell.Value = "NOK"
    i = i + 1
  Else
    AddressingOkRange.Item(i, 1).Value =
      Treats_Address_DDE(TagSearchValue,
        AddressSearchValue, TypeSearchValue)
    If AddressingOkRange.Item(i, 1).Value = "OK" Then
      AddressingOkCell.Value = "OK"
    Else
      AddressingOkCell.Value = "NOK"
    End If
    i = i + 1
  End If
End If

```

A aplicação continuará sua execução, no entanto, o endereçamento será marcado como **"NOK"**. As Figuras 4.3, 4.4 e 4.5 mostram alguns cenários de teste.

Teste dos Equipamentos						
Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK Endereçamento
Local 61 Data 0	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE BAIXO		NOK
Local 61 Data 1	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE MUITO BAIXO		OK
Local 41 OnOffData	RT_3M0562_3_Ins_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO		OK

Figura 4.3: Cenário de teste: Ausência do endereço físico. Elaborado pelo Autor.

Teste dos Equipamentos						
Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK Endereçamento
Local 61 Data 0	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE BAIXO		NOK
Local 61 Data 1	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE MUITO BAIXO		OK
Local 41 OnOffData	RT_3M0562_3_Ins_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO		OK

Figura 4.4: Cenário de teste: Ausência da Tag. Elaborado pelo Autor.

Teste dos Equipamentos						
Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK Endereçamento
Local 61 Data 0	LSL_3M0562_Ins_PV	Alarmes		NIVEL DO TANQUE BAIXO		NOK
Local 61 Data 1	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE MUITO BAIXO		OK
Local 41 OnOffData	RT_3M0562_3_Ins_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO		OK

Figura 4.5: Cenário de teste: Ausência do tipo de Sinal. Elaborado pelo Autor.

Para o caso em que o nenhum dos três campos principais são preenchidos, o teste será interrompido. Se a primeira linha de teste contém todos os campos vazios, o teste não será realizado, conforme mostrado pela Figura 4.6.

Teste dos Equipamentos						
Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK Endereçamento
Local 61 Data 0	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE BAIXO		OK
Local 61 Data 1	LSL_3M0562_Ins_PV	Alarmes	DI	NIVEL DO TANQUE MUITO BAIXO		OK
Local 41 OnOffData	RT_3M0562_3_Ins_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO		OK

Figura 4.6: Cenário de teste: Ausência do preenchimento dos Campos. Teste não executado. Elaborado pelo Autor.

Isso ocorre pela forma que aplicação trata o fim do intervalo de leitura e teste de endereçamento. Quando todos os campos (Endereço, Tag e Tipo), deixam de serem preenchidos, conforme mostrado abaixo, o *loop* de teste será interrompido.

```

If IsEmpty(AddressRange.Item(i, 1)) And
  IsEmpty(TagRange.Item(i, 1)) And IsEmpty(TypeRange.Item(i,
  1)) Then
  Exit For
Else

```

O terceiro cenário trata o preenchimento inadequado no tipo de endereço, i.e., uma *tag* declarada como Digital, isto é, *booleana*, recebe o endereço físico de um sinal analógico, ou vice-versa, o código confirma que o sinal não está correto mas ainda permite o fim da execução, marcando o endereçamento como “NOK”. O resultado dos testes são mostrados nas Figuras 4.7 e 4.8.

```

Function VerificaTipoEndereco(address As String, iotype As String)
  As Integer

  Dim pos As Integer
  Dim substring_digital As String
  Dim substring_digitalS As String 'sinal digital safety
  Dim substring_analog As String

  substring_digital = ".Data."
  substring_digital2 = ".Pt"
  substring_analog = ".Ch"

  If iotype = "DI" Or iotype = "DO" Then
    pos = InStr(address, substring_digital)
    If pos > 0 Then
      VerificaTipoEndereco = 1
    Else
      pos = InStr(address, substring_digital2)
      If pos > 0 Then
        VerificaTipoEndereco = 1
      Else
        VerificaTipoEndereco = 0
      End If
    End If
  End If

  Else
    pos = InStr(address, substring_analog)
    If pos > 0 Then
      VerificaTipoEndereco = 2
    Else
      VerificaTipoEndereco = 0
    End If
  End If

End Function

```

Testes Internos

Especificação Funcional do Equipamento

Testar Endereçamento

Cliente: _____ P.I.C.: _____ CP062

Arqs: _____

Tipo: _____

Responsável: Rudson Santana

Data (dd/mm/aaaa): 14/03/2017

Descrição das Funcionalidades do Equipamento

Equipamento	Doc. Referência	Endereço(Tag)	Tag Proposta	Função	Tipo	Descrição	Resultado	OK/NOK	Endereçamento
UL_30002		Local 4.1 Data 2	UL_30002_01ng_PV	Nível	DI	NÍVEL DO TANQUE BASTO	OK		OK
UL_30002		Local 4.1 Data 2	UL_30002_02ng_PV	Nível	DI	NÍVEL DO TANQUE BASTO	OK		NOK
RT_30002_1		Local 4.1 Data 2	RT_30002_1_01ng_PV	Entrada analógica	AI	SENHOR DE PRESSÃO PRIMÁRIO	OK		OK
RT_30002_2		Local 4.1 Data 2	RT_30002_2_01ng_PV	Entrada analógica	AI	SENHOR DE PRESSÃO SECUNDÁRIO	OK		OK
TE_30002		Local 4.1 Data 2	TE_30002_01ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE B	OK		OK
TE_30002		Local 4.1 Data 2	TE_30002_02ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE C	OK		OK
TE_30002		Local 4.1 Data 2	TE_30002_03ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE D	OK		OK
VE_30002		Local 4.1 Data 2	VE_30002_01ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE E	OK		OK
VE_30002		Local 4.1 Data 2	VE_30002_02ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE F	OK		OK
FL_3000_01		RA005F001.1 Data 2	Programa:FL_3000_01ng_PV	Entrada analógica	AI	MEDIDOR DE VAZÃO PRÉ REFRIGERANTE	OK		NOK
FL_3000_02		RA005F001.1 Data 2	Programa:FL_3000_02ng_PV	Entrada analógica	AI	MEDIDOR DE VAZÃO PRÉ REFRIGERANTE	OK		OK
GL_3000_01		RA005F001.1 Data 2	Programa:GL_3000_01ng_PV	Entrada analógica	AI	MEDIDOR DE VACUAMENTO DE GÁS PRÉ REFRIGERANTE	OK		OK
GL_3000_02		RA005F001.1 Data 2	Programa:GL_3000_02ng_PV	Entrada analógica	AI	MEDIDOR DE VACUAMENTO DE GÁS PRÉ REFRIGERANTE	OK		OK
TL_3000_01		RA005F001.1 Data 2	Programa:TL_3000_01ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		OK
TL_3000_02		RA005F001.1 Data 2	Programa:TL_3000_02ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		OK
TL_3000_03		RA005F001.1 Data 2	Programa:TL_3000_03ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		OK
TL_3000_04		RA005F001.1 Data 2	Programa:TL_3000_04ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		OK

Figura 4.7: Cenário de teste: Preenchimento incorreto do campo “Tipo”. Teste executado com erro de Endereçamento. Elaborado pelo Autor.

Testes Internos

Especificação Funcional do Equipamento

Testar Endereçamento

Cliente: _____ P.I.C.: _____ CP062

Arqs: _____

Tipo: _____

Responsável: Rudson Santana

Data (dd/mm/aaaa): 14/03/2017

Descrição das Funcionalidades do Equipamento

Equipamento	Doc. Referência	Endereço(Tag)	Tag Proposta	Função	Tipo	Descrição	Resultado	OK/NOK	Endereçamento
UL_30002		Local 4.1 Data 2	UL_30002_01ng_PV	Nível	DI	NÍVEL DO TANQUE BASTO	OK		OK
UL_30002		Local 4.1 Data 2	UL_30002_02ng_PV	Nível	DI	NÍVEL DO TANQUE BASTO	OK		NOK
RT_30002_1		Local 4.1 Data 2	RT_30002_1_01ng_PV	Entrada analógica	AI	SENHOR DE PRESSÃO PRIMÁRIO	OK		OK
RT_30002_2		Local 4.1 Data 2	RT_30002_2_01ng_PV	Entrada analógica	AI	SENHOR DE PRESSÃO SECUNDÁRIO	OK		OK
TE_30002		Local 4.1 Data 2	TE_30002_01ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE B	OK		OK
TE_30002		Local 4.1 Data 2	TE_30002_02ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE C	OK		OK
TE_30002		Local 4.1 Data 2	TE_30002_03ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE D	OK		OK
VE_30002		Local 4.1 Data 2	VE_30002_01ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE E	OK		OK
VE_30002		Local 4.1 Data 2	VE_30002_02ng_PV	Entrada analógica	AI	TEMPERATURA DO REFRIGERANTE F	OK		OK
FL_3000_01		RA005F001.1 Data 2	Programa:FL_3000_01ng_PV	Entrada analógica	AI	MEDIDOR DE VAZÃO PRÉ REFRIGERANTE	OK		OK
FL_3000_02		RA005F001.1 Data 2	Programa:FL_3000_02ng_PV	Entrada analógica	AI	MEDIDOR DE VAZÃO PRÉ REFRIGERANTE	OK		OK
GL_3000_01		RA005F001.1 Data 2	Programa:GL_3000_01ng_PV	Entrada analógica	AI	MEDIDOR DE VACUAMENTO DE GÁS PRÉ REFRIGERANTE	OK		OK
GL_3000_02		RA005F001.1 Data 2	Programa:GL_3000_02ng_PV	Entrada analógica	AI	MEDIDOR DE VACUAMENTO DE GÁS PRÉ REFRIGERANTE	OK		OK
TL_3000_01		Local 4.1 Data 2	Programa:TL_3000_01ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		NOK
TL_3000_02		RA005F001.1 Data 2	Programa:TL_3000_02ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		OK
TL_3000_03		RA005F001.1 Data 2	Programa:TL_3000_03ng_PV	Entrada analógica	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERANTE	OK		OK

Figura 4.8: Cenário de teste: Preenchimento incorreto do campo “Endereço”. Teste executado com erro de Endereçamento. Elaborado pelo Autor.

4.1.2 Parameters and Local Tags - Como tratar

Tratando-se do uso de *Tags* locais, um programador pode optar por declarar *tags* dentro de um programa em uma determinada *Task*, periódica ou contínua. Quando declarada como *Tag* local, o escopo de leitura desta fica limitada apenas ao programa em que ela foi declarada em uma situação comum.

No entanto, é possível que por questões organizacionais, o programador opte pelo o uso de *tags* locais recebendo o endereço físico. Para isso, o Logix permite que uma tag tenha seu escopo alterado. Conforme mostrado pelas Figuras 4.9 e por 4.10, definindo uma tag como *Input Parameter*, *Output Parameter*, *InOut Parameter* ou *Public Parameter*, podemos mudar o escopo de acesso e com isso, permitir que uma *tag* local seja endereçada.

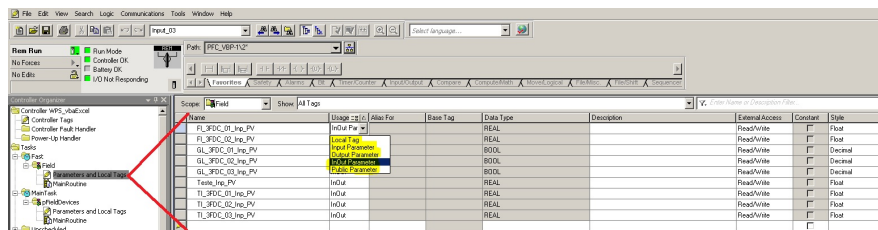


Figura 4.9: Alteração de escopo de Tag Local. Elaborado pelo Autor.

Name	Usage	Value	Force Mark	Style	Data Type	Description	Constant	Connections
FI_3FDC_01_Inp_PV	InOut	874.63		Float	REAL		FF	BACKFDC_11.ChkData
FI_3FDC_02_Inp_PV	InOut	836.88		Float	REAL		FF	BACKFDC_11.ChkData
GI_3FDC_01_Inp_PV	InOut	1		Decimal	BOOL		FF	BACKFDC_21.Data1
GI_3FDC_02_Inp_PV	InOut	1		Decimal	BOOL		FF	BACKFDC_21.Data2
TI_3FDC_01_Inp_PV	InOut	3.60		Float	REAL		FF	BACKFDC_11.ChkData
TI_3FDC_02_Inp_PV	InOut	163.68		Float	REAL		FF	BACKFDC_11.ChkData
TL_3FDC_01_Inp_PV	InOut	73.87		Float	REAL		FF	BACKFDC_11.ChkData
TL_3FDC_02_Inp_PV	InOut	107.42		Float	REAL		FF	BACKFDC_11.ChkData

Figura 4.10: Endereçamento de *Tag* Local. Elaborado pelo Autor.

Declarar uma *Tag* como local, mesmo com o escopo alterado de local para *Input Parameter*, *Output Parameter*, *InOut Parameter* ou *Public Parameter*, implica em uma mudança na forma de ler a mesma, para qualquer aplicação que tente fazer leitura desta, seja um supervisor, seja uma ferramenta de automatização.

Para o exemplo das Figuras 4.9 e 4.10, para que o acesso seja possível, é necessário que o seguinte formato seja utilizado:

$$Program : [1].[2] \quad (4.1)$$

Onde em 4.1, temos:

- O item [1] = *Nome do Programa*;
- O item [2] = *Nome da Tag*.

Tomando o exemplo de 4.10, se quiséssemos referenciar a *tag FI_3FDC_01_Inp_PV* fora do Programa em que foi criado, faríamos conforme 4.2.

$$Program : Field.FI_3FDC_01_Inp_PV \quad (4.2)$$

Dentro do cenário de teste com o uso do DDE, um erro na declaração correta do formato apresentado em 4.1, ou até em um cenário de declaração incorreta da *tag* (como um erro de digitação), revela uma limitação da comunicação pelo protocolo DDE.

A validação do teste de endereçamento é feito por meio da função *DDERequest*, que por padrão não retorna nenhum *Error Code*, apenas ocorrerá um erro no tempo de execução[52]. Isso dificulta não só o tratamento de eventuais erros no preenchimento do campo *Tag*, como o tratamento do código como um todo.

Se uma *tag* for alterada na planilha sem a devida correção no CLP, ou sem apontar o escopo da mesmo - para *tags* locais - e durante o *request* não for encontrada, o aplicação irá travar e ocorrerá um “*crash*” do *Excel*, conforme apresentado pelas Figuras 4.11 e 4.12.

Teste dos Equipamentos						
Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK
Local:SI.DatA.0	LS_3M052_inp_PV	Alarmes	DI	NIVEL DO TANQUE BAIXO		OK
Local:SI.DatA.1	LSL_3M052_inp_PV	Alarmes	DI	NIVEL DO TANQUE BAIXO		OK
Local:AI.CHIDatA	PII_3M052_1_inp_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO		OK
Local:AI.CHIDatA	PII_3M052_2_inp_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO SECUNDÁRIO		OK
Local:AI.CHIDatA	TE_3M052_inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO S		OK
Local:AI.CHIDatA	TE_3M052_inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO S		OK
Local:AI.CHIDatA	TE_3M052_inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO T		OK
Local:AI.CHIDatA	VE_3M052_inp_PV	Entradas analógicas	AI	VIBRAÇÃO RADIAL		OK
Local:AI.CHIDatA	VI_3M052_inp_PV	Entradas analógicas	AI	VIBRAÇÃO RADIAL		OK
RACK03FDC11.CHIDatA	II_3FDC_01_inp_PV	Entradas analógicas	AI	MEDIDOR DE VAZÃO PRÉ RESFRIAMENTO		OK
RACK03FDC11.CHIDatA	II_3FDC_02_inp_PV	Entradas analógicas	AI	MEDIDOR DE VAZÃO PÓS RESFRIAMENTO		OK
RACK03FDC31.DatA.0	Program FAN_01_3FDC_01_inp_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO		OK
RACK03FDC31.DatA.1	Program FAN_02_3FDC_02_inp_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS SERPENTINAS		OK

Figura 4.11: Tag declarada em um escopo local, mas não registrada de maneira correta na planilha. Elaborado pelo Autor.

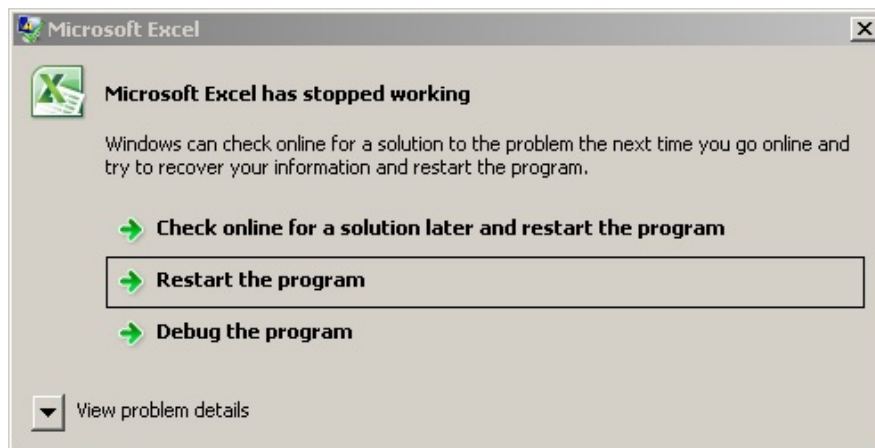


Figura 4.12: Interrupção abrupta do funcionamento do Excel. Elaborado pelo Autor.

Neste cenário, uma recomendação de uso ao corpo técnico da empresa é de o realizar teste de tags locais de um programa, utilize uma aba exclusiva para o programa, conforme mostrado em 4.13 e 4.14.



Figura 4.13: Sugestão de organização e uso da planilha - Organização a nível de CLP. Elaborado pelo Autor.

Teste de Endereçamento							
PLC			CPM2				
Watan Saitama 1480-703							
Descrição das Funcionalidades do Equipamento							
Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK Endereçamento	OK/NOK Aprovação
8A010FDC11.010040	Program.Fun.F1_SFDC_D1_1P	Estados analógico	AI	MEDIDOR DE VAZÃO PRÉ REFRIGERAMENTO			
8A010FDC11.010040	Program.Fun.F1_SFDC_Q2_1P	Estados analógico	AI	MEDIDOR DE VAZÃO PÓS REFRIGERAMENTO			
8A010FDC11.010040	Program.Fun.F1_SFDC_D1_1P	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ REFRIGERAMENTO			
8A010FDC11.010040	Program.Fun.F1_SFDC_Q2_1P	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PÓS REFRIGERAMENTO			
8A010FDC11.010040	Program.Fun.F1_SFDC_D1_1P	Estados analógico	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERAMENTO			
8A010FDC11.010040	Program.Fun.F1_SFDC_Q2_1P	Estados analógico	AI	MEDIDOR DE TEMPERATURA PÓS REFRIGERAMENTO			
8A010FDC11.010040	Program.Fun.F1_SFDC_D1_1P	Estados analógico	AI	MEDIDOR DE TEMPERATURA SERPENTINAS			
8A010FDC11.010040	Program.Fun.F1_SFDC_Q2_1P	Estados analógico	AI	MEDIDOR DE TEMPERATURA PÓS REFRIGERAMENTO			

Figura 4.14: Sugestão de organização e uso da planilha - Organização a nível de teste. Elaborado pelo Autor.

Outra limitação encontrada durante o teste utilizando a planilha é para o teste de sinais declaradas em escopo local, mas programas definidos em *tasks* acíclicas. Conforme definido por Erickson[53], *tasks* acíclicas, ou assíncronas, não estão vinculadas a um ciclo de varredura(*scan*) predefinido. Elas são ativadas em momentos específicos e são comumente utilizadas em rotinas *Safety*[54].

Nos primeiros testes realizados, a aplicação apresentava um comportamento completamente anormal, onde um sinal marcado como “OK” poderia ser alterado para “NOK”. A principal hipótese para o problema era uma possível discrepância entre o intervalo de envio/leitura do *DDE* e a alteração do valor dentro do programa.

Realizando alguns teste e implementando uma verificação de erro no *DDERequest*, foi possível normalizar a situação, permitindo que rotinas *Safety* também sejam testadas. Os resultados são apresentados nas Figuras 4.15 e 4.16.

```
Sub TrataErroDDE ()
' Exibe a mensagem de erro e as possiveis causas
MsgBox "Erro ao estabelecer conexao com o Servidor DDE.
Possiveis causas:" & vbCrLf & _
" - " & IIf(Err.Number = 1, "Erro ao iniciar comunicacao", "
- Erro ao encontrar a tag"), _
vbExclamation, "Erro de conexao DDE"
End Sub
```

```
data_value = DDERequest(rslinx, read_tag)
On Error Resume Next
If IsEmpty(data_value) Then
TrataErroDDE
Range("A90").ClearContents
DDETerminate rslinx
Treats_Address_DDE = "NOK"
Exit Function
ElseIf
```

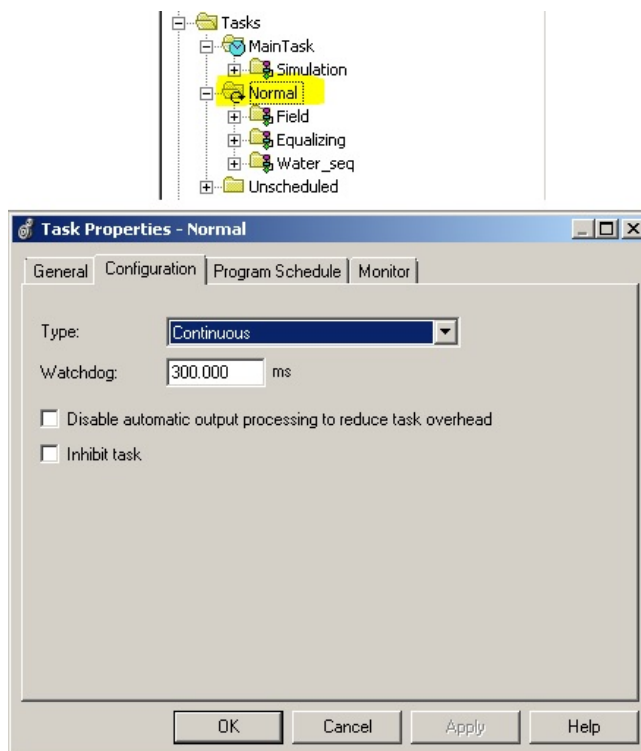


Figura 4.15: Alteração de uma *Task* para o formato acíclico. Elaborado pelo Autor.

Teste dos Equipamentos								
Especificação Funcional de Equipamento								
Nome:		R/C		CP882				
Área:								
Título:		Vantagem Variável						
Responsável:		Vantagem Variável						
Data (dd/mm/aaaa):		14/03/2012						
Descrição das Funcionalidades do Equipamento								
Especificar todos os Funcionamentos do Equipamento (Modo de operação, funcionamento, admissional, comando, alarme)								
Equipamento	Doc. Referência	Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK
TL_PFC_01		RA025F0C3.1.040Data	Programa TL_PFC_01_PFC.P	Entradas analógicas	AI	MEDIDOR DE VAZÃO PRÉ REFRIGERAMENTO		OK
TL_PFC_02		RA025F0C3.1.040Data	Programa TL_PFC_02_PFC.P	Entradas analógicas	AI	MEDIDOR DE VAZÃO PÓS REFRIGERAMENTO		OK
OL_PFC_01		RA025F0C2.1.040Data	Programa TL_PFC_01_PFC.P	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ REFRIGERAMENTO		OK
OL_PFC_02		RA025F0C2.1.040Data	Programa TL_PFC_02_PFC.P	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PÓS REFRIGERAMENTO		OK
OL_PFC_03		RA025F0C2.1.040Data	Programa TL_PFC_03_PFC.P	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PÓS REFRIGERAMENTO		OK
TL_PFC_07		RA025F0C3.1.040Data	Programa TL_PFC_07_PFC.P	Entradas analógicas	AI	MEDIDOR DE TEMPERATURA PRÉ REFRIGERAMENTO		OK
TL_PFC_08		RA025F0C3.1.040Data	Programa TL_PFC_08_PFC.P	Entradas analógicas	AI	MEDIDOR DE TEMPERATURA PÓS REFRIGERAMENTO		OK
TL_PFC_09		RA025F0C3.1.040Data	Programa TL_PFC_09_PFC.P	Entradas analógicas	AI	MEDIDOR DE TEMPERATURA PÓS REFRIGERAMENTO		OK

Figura 4.16: Resultado do Teste. Elaborado pelo Autor.

4.2 Vantagens do uso da Planilha

Originalmente, o teste de endereçamento era feito forma manual, com o programador escrevendo um valor manualmente no endereço físico e verificando a alteração na *Tag* associada. Considerando que num cenário normal cada *Tag*/ponto leva em média 3 minutos ou 180 segundos para ser testada¹, uma lista de Entrada/Saída com cerca de 1200 pontos, seria gasto uma média de 216.000 segundos, ou cerca de 60 horas para fazer um teste completo de todo o endereçamento.

Considerando que o preenchimento da planilha de testes com os endereços já faz parte da rotina de teste, isto é, não podendo ser contabilizado como um tempo gasto de “preenchimento”, e que se seguidas todas as premissas de teste, gasta-se em média menos de 1 segundo de teste por ponto na pior das hipóteses, teríamos para o mesmo cenário de 1200 pontos de

¹O tempo médio por ponto foi calculado empiricamente e com respaldo do corpo técnico da empresa

entrada e saída um tempo de teste de aproximadamente 1200 segundos, ou aproximadamente 20 minutos ou 1/3 de hora.

Isso representa um ganho percentual de aproximadamente 99,45% no tempo de execução dos testes. Uma melhora extremamente significativa no tempo de teste que pode ser redirecionado para correções de lógicas de processos ou afins.

A próxima sequência de imagens demonstra o funcionamento da aplicação, antes do teste 4.17, 4.18, 4.19 e 4.20, e após o teste 4.21, 4.22, 4.23 e 4.24.

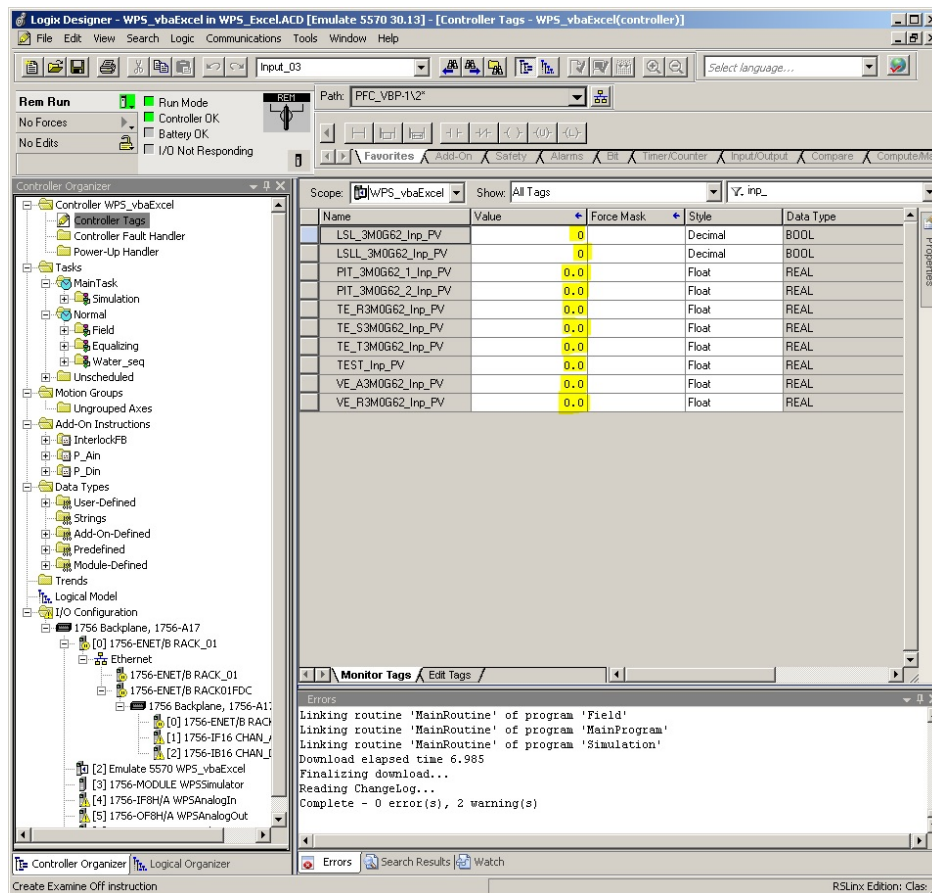


Figura 4.17: Tags Globais pré-teste. Elaborado pelo Autor.

Endereço(Tag)	Tag Propósito	Função	Tipo	Descrição	Resultado	OK/NOK	Endereçamento	OK/NOK	Aprovaçã
Local:SI:Data:1	LVL_3M0502_Inp_PV	Alarmes	DI	NÍVEL DO TANQUE BAIXO					
Local:SI:Data:1	LVL_3M0502_Inp_PV	Alarmes	DI	NÍVEL DO TANQUE MUITO BAIXO					
Local:SI:CH:Data:1	PIT_3M0502_2_Inp_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO					
Local:SI:CH:Data:1	PIT_3M0502_2_Inp_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO SECUNDÁRIO					
Local:SI:CH:Data:1	TE_3M0502_Inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO D					
Local:SI:CH:Data:1	TE_3M0502_Inp_PV	Entradas analógicas	AITD	TEMPERATURA DO ENROLAMENTO D					
Local:SI:CH:Data:1	TE_3M0502_Inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO T					
Local:SI:CH:Data:1	VE_3M0502_Inp_PV	Entradas analógicas	AI	VIBRAÇÃO LOCAL					
Local:SI:CH:Data:1	VE_3M0502_Inp_PV	Entradas analógicas	AITD	VIBRAÇÃO LOCAL					

Figura 4.18: Planilha de Tag Globais pré-teste. Elaborado pelo Autor.

Name	Usage	Value	Force Mask	Style	Data Type
FL_3FDC_01_Inp_PV	InOut	0.0		Float	REAL
FL_3FDC_02_Inp_PV	InOut	0.0		Float	REAL
GL_3FDC_01_Inp_PV	InOut	0		Decimal	BOOL
GL_3FDC_02_Inp_PV	InOut	0		Decimal	BOOL
GL_3FDC_03_Inp_PV	InOut	0		Decimal	BOOL
Teste_Inp_PV	InOut	0.0		Float	REAL
TL_3FDC_01_Inp_PV	InOut	0.0		Float	REAL
TL_3FDC_02_Inp_PV	InOut	0.0		Float	REAL
TL_3FDC_03_Inp_PV	InOut	0.0		Float	REAL

Figura 4.19: Tags Locais pré-teste. Elaborado pelo Autor.

Endereço(Tag)	Tag Proposto	Função	Tipo	Descrição	Resultado	OK/NOK	OK/NOK Aproveçã
RACK0FDC1.1.ChData	Program Field_T_3FDC_03_Inp_PV	Etradas analógicas	AI	MEDIDOR DE VAZÃO PRÉ RESFRIAMENTO			
RACK0FDC1.1.ChData	Program Field_T_3FDC_03_Inp_PV	Etradas analógicas	AI	MEDIDOR DE VAZÃO PÓS RESFRIAMENTO			
RACK0FDC2.1.Data.0	Program Field_Q_3FDC_02_Inp_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO			
RACK0FDC2.1.Data.1	Program Field_Q_3FDC_02_Inp_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PÓS RESFRIAMENTO			
RACK0FDC2.1.Data.2	Program Field_Q_3FDC_02_Inp_PV	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PÓS RESFRIAMENTO			
RACK0FDC1.1.ChData	Program Field_T_3FDC_03_Inp_PV	Etradas analógicas	AI	MEDIDOR DE TEMPERATURA PRÉ RESFRIAMENTO			
RACK0FDC1.1.ChData	Program Field_T_3FDC_03_Inp_PV	Etradas analógicas	AI	MEDIDOR DE TEMPERATURA PÓS RESFRIAMENTO			
RACK0FDC1.1.ChData	Program Field_T_3FDC_03_Inp_PV	Etradas analógicas	AI	MEDIDOR DE TEMPERATURA SERPENTINAS			
RACK0FDC1.1.ChData	Program Field_T_3FDC_03_Inp_PV	Etradas analógicas	AI	MEDIDOR DE TEMPERATURA PÓS RESFRIAMENTO			

Figura 4.20: Planilha de Tag Locais pré-teste. Elaborado pelo Autor.

Name	Value	Force Mask	Style	Data Type
LSL_3M0G62_Inp_PV	1		Decimal	BOOL
LSL_3M0G62_Inp_PV	1		Decimal	BOOL
PIT_3M0G62_1_Inp_PV	568.2		Float	REAL
PIT_3M0G62_2_Inp_PV	777.64		Float	REAL
TE_R3M0G62_Inp_PV	494.8		Float	REAL
TE_S3M0G62_Inp_PV	730.61		Float	REAL
TE_T3M0G62_Inp_PV	554.34		Float	REAL
TEST_Inp_PV	0.0		Float	REAL
VE_A3M0G62_Inp_PV	319.03		Float	REAL
VE_R3M0G62_Inp_PV	726.75		Float	REAL

Figura 4.21: Tags Globais pós-teste. Elaborado pelo Autor.

Endereço(Tag)	Tag Propósito	Função	Tipo	Descrição	Resultado	OK/NOK	Endereçamento	Aprovação
Local:SI:Data.1	LSL_RMS052_Inp_PV	Alarmes	DI	NÍVEL DO TANQUE BAIXO		OK		
Local:SI:Data.1	LSL_RMS052_Inp_PV	Alarmes	DI	NÍVEL DO TANQUE MUITO BAIXO		OK		
Local:AI:CH:Data	PIT_RMS052_2_Inp_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO PRIMÁRIO		OK		
Local:AI:CH:Data	PIT_RMS052_3_Inp_PV	Entradas analógicas	AI	SENSOR DE PRESSÃO SECUNDÁRIO		OK		
Local:AI:CH:Data	TE_RMS052_Inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO D		OK		
Local:AI:CH:Data	TS_RMS052_Inp_PV	Entradas analógicas	RTD	TEMPERATURA DO ENROLAMENTO S		OK		
Local:AI:CH:Data	TE_RMS052_Inp_PV	Entradas analógicas	AI	TEMPERATURA DO ENROLAMENTO T		OK		
Local:AI:CH:Data	VE_RMS052_Inp_PV	Entradas analógicas	AI	VIBRAÇÃO LOCAL		OK		
Local:AI:CH:Data	VR_RMS052_Inp_PV	Entradas analógicas	RTD	VIBRAÇÃO REMOTA		OK		

Figura 4.22: Planilha de Tag Globais pós-teste. Elaborado pelo Autor.

Name	Usage	Value	Force Mask	Style	Data Type
FL_3FDC_01_Inp_PV	InOut	939.16		Float	REAL
FL_3FDC_02_Inp_PV	InOut	132.24		Float	REAL
GL_3FDC_01_Inp_PV	InOut	1		Decimal	BOOL
GL_3FDC_02_Inp_PV	InOut	1		Decimal	BOOL
GL_3FDC_03_Inp_PV	InOut	1		Decimal	BOOL
Teste_Inp_PV	InOut	0.0		Float	REAL
TL_3FDC_01_Inp_PV	InOut	785.95		Float	REAL
TL_3FDC_02_Inp_PV	InOut	680.96		Float	REAL
TL_3FDC_03_Inp_PV	InOut	196.25		Float	REAL

Errors

```

Linking routine 'MainRoutine' of program 'Field'
Linking routine 'MainRoutine' of program 'MainProgram'
Linking routine 'MainRoutine' of program 'Simulation'
Download elapsed time 6.985
Finalizing download...
Reading ChangeLog...
Complete - 0 error(s), 2 warning(s)
    
```

Figura 4.23: Tags Locais pós-teste. Elaborado pelo Autor.

The screenshot shows an Excel spreadsheet with the following data table:

Descrição das Funcionalidades do Equipamento							
Teste dos Equipamentos							
Endereço(Tag)	Tag Proposta	Função	Tipo	Descrição	Resultado	OK/NOK Endereçamento	OK/NOK Aproveitamento
BACKUPDC11.CHData	Program Field 01_3FDC_01_IP	Eteradas analógicas	AI	MEDIDOR DE VAZÃO PRÉ RESFRIAMENTO	OK		
BACKUPDC11.CHData	Program Field 01_3FDC_02_IP	Eteradas analógicas	AI	MEDIDOR DE VAZÃO PÓS RESFRIAMENTO	OK		
BACKUPDC11.CHData	Program Field 01_3FDC_01_IP	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO	OK		
BACKUPDC11.Data.1	Program Field 01_3FDC_02_IP	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PRÉ RESFRIAMENTO	OK		
BACKUPDC11.Data.2	Program Field 01_3FDC_01_IP	Alarmes	DI	SENSOR DE VAZAMENTO DE GÁS PÓS RESFRIAMENTO	OK		
BACKUPDC11.CHData	Program Field 01_3FDC_01_IP	Eteradas analógicas	AI	MEDIDOR DE TEMPERATURA PRÉ RESFRIAMENTO	OK		
BACKUPDC11.CHData	Program Field 01_3FDC_02_IP	Eteradas analógicas	AI	MEDIDOR DE TEMPERATURA PRÉ RESFRIAMENTO	OK		
BACKUPDC11.CHData	Program Field 01_3FDC_01_IP	Eteradas analógicas	AI	MEDIDOR DE TEMPERATURA HERPENTINAS	OK		
BACKUPDC11.CHData	Program Field 01_3FDC_02_IP	Eteradas analógicas	AI	MEDIDOR DE TEMPERATURA PÓS RESFRIAMENTO	OK		

Figura 4.24: Planilha de Tag Locais pós-teste. Elaborado pelo Autor.

Capítulo 5

Conclusões

5.1 Considerações Finais

Conforme descrito em 4.2, o uso da planilha proporciona ganhos extraordinários de produtividade e tempo para um programador que esteja testando sua aplicação. Em um mundo cada vez mais digitalizado e que diversas ferramentas surgem como meios para melhorar e otimizar nossas atividades, é necessário abrir-se para tais e reconhecer o potencial que elas trazem.

O mercado de aplicativos de produtividade refletem a demanda, com um crescimento previsto para 9,1% entre 2021 e 2028, sendo avaliado em aproximadamente U\$D8,14 milhões[55][56]. Isso ocorre porque “tempo” tornou-se um dos principais ativos do mercado, não apenas em uma esfera profissional, mas também na esfera social.

Em uma empresa de projetos como a Primetals Technologies, onde custos de projetos são mensurados em horas de trabalho/execução, não poderíamos ter um cenário diferente. Enxergar padrões de atividades repetitivas, que possuem um alto grau de intervenção humana e que estão sujeitas a erros por diversos fatores, seja exaustão, desatenção e afins, permite que soluções sejam desenvolvidas buscando mitigar, facilitar e assegurar uma boa confiabilidade da atividade.

Assim sendo, a ideia de teste de endereçamento foi concebida, orquestrada e testada. As decisões de escolha de método de teste, ambiente, ferramentas foram pautadas também em custos, tanto em termos financeiros quanto em tempo. Caso optasse por utilizar uma abordagem com RPA, seria necessário o uso de recursos computacionais adicionais ou até mesmo licenças, a depender da aplicação utilizada. Utilizando COM/DCOM, seria necessário por ventura criar uma interface que fizesse a comunicação com a aplicação, diferentemente do suporte nativo do DDE para com o *RSLinx Classic* e o *Excel*, o que do ponto de vista de manutenção, pode representar uma limitação para membros externos da bancada de desenvolvimento, visto que o VBA está mais presente no dia-a-dia do corpo técnico da empresa.

Como já abordado, é importante entender que o fator humano é o principal limitador na implementação de automatizações, seja pela capacidade de não conseguir reproduzir atividades complexas que dependem de intervenção humana, seja por não existir uma aceitação do uso de ferramentas de automatização[57][58]. Assim, obter uma aceitação pelo grupo de usuários que fará uso da ferramenta torna-se tão ou mais importante, que a ferramenta em si.

Mesmo diante das premissas abordadas no capítulo anterior, a familiaridade com o ambiente de teste e respaldado pela discussão no tópico 3.5, ainda é possível que o corpo técnico

sinta-se capacitado e estimulado a utilizar a aplicação, diante da economia de tempo adquirida e apresentada.

Vistos os resultados apresentados em 4.2 atendem o objetivo almejado em 1.2, assim como a aceitação discutida em parágrafos anteriores, é possível concluir que o projeto atendeu todas suas expectativas, mesmo diante das limitações existentes.

Mesmo em desuso, o DDE é um protocolo conhecido e relativamente bem documentado, e em um ambiente onde às vezes os recursos de *hardware* são ou podem ser limitados, optar por uma escolha que seja capaz de rodar mesmo em ambientes mais antigos parece o caminho mais sensato.

No entanto, isso não limita a aplicação que mesmo utilizando um protocolo relativamente defasado, ainda é robusto para sistemas atuais. Na próxima seção, serão abordadas perspectivas futuras e melhorias.

5.2 Propostas de Continuidade

De acordo com o [59], a Siemens é a principal fabricante de CLPs no mercado e responde naturalmente, pelo maior *market share* do mesmo. Na sequência, temos grande nomes como *Mitsubishi Electric*, *Rockwell Automation*, *Emerson*, *ABB* entre outras. Assim, a etapa natural seria a adaptação do teste de endereçamento para um ambiente *Siemens*.

Outras melhorias mapeadas para a aplicação consistem em:

- Eliminar a necessidade de intervenção humana na inserção dos dados, automatizando e cruzando as informações da Lista de E/S com a lista de *Tags* criadas dentro do CLP;
- Fazer uso da versão mais atual do *RSLinx Classic*, denominada *FactoryTalk Linx Gateway*, explorado em 3.3.

A aplicação *FactoryTalk Linx Gateway* possui um suporte nativo também para DDE e o *Real-Time Data*, que utiliza-se do protocolo OLE e não esbarra nas limitações já conhecidas do DDE, como[45]:

- DDE comumente para de responder quando precisa gerenciar múltiplas requisições;
- DDE possui dificuldade em recalcular ou apresentar novos dados em tempo-real à medida que são transferidos para a planilha;
- O DDE tende a ter problemas de desempenho, incluindo queda atualizações, tempo de resposta ineficiente e desempenho não confiável.

Isso permitiria que outras aplicações fora do ambiente VBA sejam utilizados, a exemplo de uma aplicação Python.

Capítulo 6

Anexos

Aqui são apresentados algumas outras funções utilizadas durante o desenvolvimento da aplicação.

6.1 *Main*

```
Sub Main()  
    Dim AddressSearchValue As String  
    Dim TagSearchValue As String  
    Dim TypeSearchValue As String  
    Dim AddressRange As Range  
    Dim TagRange As Range  
    Dim TypeRange As Range  
    Dim AddressingOkRange As Range  
    Dim AddressingOkCell As Range  
    Dim i As Integer  
  
    Set AddressRange = Range("C15:C53")  
    Set TagRange = Range("D15:D53")  
    Set TypeRange = Range("F15:F53")  
    Set AddressingOkRange = Range("I15:I53")  
    i = 1  
  
    For Each AddressingOkCell In AddressingOkRange  
        AddressSearchValue = AddressRange.item(i, 1).Value  
        TagSearchValue = TagRange.item(i, 1).Value  
        TypeSearchValue = TypeRange.item(i, 1).Value  
  
        If IsEmpty(AddressRange.item(i, 1)) And  
           IsEmpty(TagRange.item(i, 1)) And  
           IsEmpty(TypeRange.item(i, 1)) Then  
            Exit For  
        Else  
            If IsEmpty(TagRange.item(i, 1)) Then  
                AddressingOkCell.Value = "NOK"  
            End If  
        End If  
    Next  
End Sub
```

```
        i = i + 1
    Else
        AddressingOkRange.item(i, 1).Value =
            Treats_Address_DDE(TagSearchValue,
                AddressSearchValue, TypeSearchValue)
        If AddressingOkRange.item(i, 1).Value = "OK" Then
            AddressingOkCell.Value = "OK"
        Else
            AddressingOkCell.Value = "NOK"
        End If
        i = i + 1
    End If
End If
Next AddressingOkCell
i = 0

End Sub
```

6.2 GerarValorRandomico

```
Function GerarValorRandomico() As Double
    'Definir a semente do gerador de numeros aleatorios
    Randomize
    'Gerar um valor real aleatorio entre 0 e 1000
    GerarValorRandomico = Rnd() * 1000
End Function
```

Referências Bibliográficas

- [1] Guilherme Santos. *O que é Automação Industrial?* 2021. URL: <https://www.automacaoindustrial.info/o-que-e-automacao-industrial/>. (Acesso em: 02/06/2023).
- [2] GS1Brasil. *Índice de Automação do Mercado Brasileiro e de Consumidores*. URL: <https://www.gs1br.org/indices-e-pesquisas/estudos-e-pesquisas/diagnostico-de-automacao>. (Acesso em: 02/06/2023).
- [3] ABDI. *Agenda brasileira para a Indústria 4.0*. URL: <http://www.industria40.gov.br>. (Acesso em: 20/05/2023).
- [4] Rockwell Automation. *Rockwell Automation - PlantPAx Distributed Control System™*. URL: <https://www.rockwellautomation.com/en-us/capabilities/process-solutions/process-systems/plantpax-distributed-control-system.html>. (Acesso em: 22/05/2023).
- [5] Eduardo Cucolo e Simon Ducroquet. *Como é composto o PIB*. URL: <https://arte.folha.uol.com.br/mercado/2020/entenda-como-e-feito-o-pib/>. (Acesso em: 02/06/2023).
- [6] Savage. *What Are PLCs and Why Are They Important?* URL: <https://www.finelinemarineelectric.com/blog/what-are-plcs-and-why-are-they-important/>. (Acesso em: 02/06/2023).
- [7] Manuel Pereira Do Nascimento Neto. «Proposição de uma Sistemática para Avaliação de Confiabilidade Humana em Mina a Céu Aberto». Tese de doutoramento. Universidade Federal do Rio Grande do Sul, 2014.
- [8] Klaus Schwab. *A Quarta Revolução Industrial*. (Acesso em: 30/05/2023). Edipro, 2016. ISBN: 9788572839785.
- [9] Max Blanchet. «Industrie 4.0: the new industrial revolution. How Europe will succeed». Em: *Industrie 4.0: the new industrial revolution. How Europe will succeed*. 2014.
- [10] Siemens. *Um guia prático sobre a Indústria 4.0*. (Acesso em: 01/06/2023). URL: <https://new.siemens.com/br/pt/empresa/stories/industria/industria-4-0.html>.
- [11] INSTITUTO DE ESTUDOS PARA O DESENVOLVIMENTO INDUSTRIAL IEDI. «Indústria 4.0: A Quarta Revolução Industrial e os Desafios para a Indústria e para o Desenvolvimento Brasileiro». Em: . (2017), p. 54.

- [12] Brian Prasad. «Analysis of pricing strategies for new product introduction». Em: *Pricing Strategy and Practice* 5 (dez. de 1997), pp. 132–141. DOI: 10.1108/09684909710184626.
- [13] Fernández Alonso Eneko. «Generación Automática del Proyecto de Control para Máquinas Modulares, Utilizando XML, para Plataforma TIA Portal». Em: . (set. de 2017), pp. 1–89.
- [14] C. R. Magar, N. Jazdi e P. Göhner. «Requirements on engineering tools for increasing reuse in industrial automation». Em: *ETFA2011*. 2011, pp. 1–7. DOI: 10.1109/ETFA.2011.6058976.
- [15] Karl Ulrich. «Fundamentals of Product Modularity». Em: *Management of Design: Engineering and Management Perspectives*. Dordrecht: Springer Netherlands, 1994, pp. 219–231. ISBN: 978-94-011-1390-8. DOI: 10.1007/978-94-011-1390-8_12. URL: https://doi.org/10.1007/978-94-011-1390-8_12.
- [16] M. Hermann, T. Pentek e B. Otto. «Design Principles for Industrie 4.0 Scenarios». Em: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. 2016, pp. 3928–3937. DOI: 10.1109/HICSS.2016.488.
- [17] Microsoft. *Introdução ao VBA no Office*. URL: <https://docs.microsoft.com/pt-br/office/vba/library-reference/concepts/getting-started-with-vba-in-office>. (Acesso em: 04/06/2023).
- [18] Michael Alexander e John Walkenbach. *Excel VBA Programming for Dummies*. (Acesso em: 08/03/2023). John Wiley & Sons, 2018.
- [19] Rob Bovey. *Professional Excel development: The definitive guide to developing applications using Microsoft Excel, VBA, and .Net*. (Acesso em: 09/03/2023). Addison-Wesley Professional, 2009.
- [20] Herman Rodent. «Supporting the DDE System Topic». Em: *Microsoft Development Network Library* (1992).
- [21] Wikipedia. *Dynamic Data Exchange*. (Acesso em: 02/06/2023). URL: https://en.wikipedia.org/wiki/Dynamic_Data_Exchange.
- [22] Microsoft. *Dynamic Data Exchange - Client, Server, and Conversation*. (Acesso em: 16/04/2023). URL: <https://learn.microsoft.com/en-us/windows/win32/dataxchg/about-dynamic-data-exchange#client-server-and-conversation>.
- [23] Dean Thompson et al. «Distributed component object model (DCOM)». Em: *Monash University, Department of Software Development, Melbourne, Australien* (1997).
- [24] David Kruglinski. *Inside Visual C++ (Microsoft Programming Series)*. (Acesso em: 08/06/2023). USA: Microsoft Press; 3rd edition (January 1, 1996), 1996. ISBN: 9781556158919.
- [25] Microsoft. *The Component Object Model*. URL: <https://learn.microsoft.com/en-us/windows/win32/com/the-component-object-model>. (Acesso em: 09/06/2023).
- [26] Dale Rogerson. *Inside Com (Microsoft Programming Series)*. (Acesso em: 09/03/2023). USA: Microsoft Press (January 27, 1997), 1996. ISBN: 1572313498.

- [27] David N. Gray et al. «Modern Languages and Microsoft's Component Object Model». Em: *Commun. ACM* 41.5 (1998), pp. 55–65. ISSN: 0001-0782. DOI: 10.1145/274946.274957. URL: <https://doi.org/10.1145/274946.274957>.
- [28] Microsoft. *[MS-DCOM]: Distributed Component Object Model (DCOM) Remote Protocol*. URL: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-dcom/4a893f3d-bd29-48cd-9f43-d9777a4415b0. (Acesso em: 09/06/2023).
- [29] Microsoft. *[MS-RPCE]: Remote Procedure Call Protocol Extensions*. URL: https://learn.microsoft.com/pt-br/openspecs/windows_protocols/ms-rpce/290c38b1-92fe-4229-91e6-4fc376610c15. (Acesso em: 09/06/2023).
- [30] Kraig Brockschmidt. *Inside OLE (2nd Ed.)* (Acesso em: 07/06/2023). USA: Microsoft Press, 1995. ISBN: 1556158432.
- [31] Microsoft. *OLE Background*. URL: <https://learn.microsoft.com/en-us/cpp/mfc/ole-background?view=msvc-170>. (Acesso em: 09/06/2023).
- [32] Inc. Automation Anywhere. *ROBOTIC PROCESS AUTOMATION (RPA)*. URL: <https://www.automationanywhere.com/rpa/robotic-process-automation>. (Acesso em: 23/05/2022).
- [33] Ricardo José Salvado Martins. «Benefícios da implementação de um centro de excelência de Robotic Process Automation». Tese de doutoramento. Instituto Superior de Economia e Gestão, 2018.
- [34] WMP Van der Aalst, M Bichler e A Heinzl. *Robotic process automation. Bus Inf Syst Eng* 60 (4): p.2. 2018.
- [35] A. d. S. Scherman. *A influência da implantação de RPA (Robotic Process Automation) nos processos relacionados a emissão de notas em uma empresa do ramo metalúrgico*. URL: <https://www.lume.ufrgs.br/handle/10183/204580>. (Acesso em: 11/04/2022).
- [36] Ted Mortenson. *PLC Hardware Explained. 2020*. URL: <https://realpars.com/plc-hardware/#:~:text=%E2%80%93The%20Hardware%20components%20of%20a,digital%20or%20analog%20field%20devices..> (Acesso em: 04/07/2023).
- [37] Library Automation Direct. *The Base-ics: PLC Backplane & More*. URL: <https://library.automationdirect.com/plc-hardware/>. (Acesso em: 04/07/2023).
- [38] Automation Forum. *I/O List*. URL: <https://automationforum.co/i-o-list/>. (Acesso em: 04/07/2023).
- [39] Stream Controls. *Siemens S7 Addressing*. URL: <https://streamcontrols.com/help/streamscada/SiemensS7Addressing.html>. (Acesso em: 04/07/2023).
- [40] Rockwell Automation. *RSLinx Classic - Obtendo Resultados com o Guia*. Português. Versão 2.54. (Acesso em: 09/06/2023). 2008. 58 pp.

- [41] Rockwell Automation. *Studio 5000 Logix Emulate Getting Results Guide*. English. Versão LGEM5K-GR016M-EN-E. (Acesso em: 09/06/2023). Nov. de 2022. 945 pp.
- [42] Rockwell Automation. *Manual de referência de instruções gerais dos Controladores Logix 5000*. Português. Versão 1756-RM003T-PT-P. (Acesso em: 09/06/2023). Nov. de 2018. 71 pp.
- [43] VMware. *VMware Workstation Player*. (Acesso em: 06/06/2023).
- [44] Don Ho. *notepad++*. (Acesso em: 05/06/2023). URL: <https://notepad-plus-plus.org/>.
- [45] Rockwell Automation. *FactoryTalk Linx Gateway Getting Results Guide*. English. Versão FTLG-GR001F-EN-E. (Acesso em: 28/05/2023). Jan. de 2023. 135 pp.
- [46] C. A Collins. *Programmable Controllers: An Engineer's Guide*. (Acesso em: 10/06/2023). Elsevier: Microsoft Press; 1st edition (January 1, 1996), 2088. ISBN: 075065757X.
- [47] Microsoft. *DDE Function*. (Acesso em: 11/02/2023). URL: <https://support.microsoft.com/en-us/office/dde-function-79e8b21c-2054-4b48-9ceb-d2cf38dc17f9>.
- [48] Rockwell Automation. *EtherNet/IP Network Devices*. English. Versão ENET-UM006B-EN-P. (Acesso em: 10/06/2023). Mai. de 2022. 68 pp.
- [49] Amy J. Morgan e Scott A. Inks. «Technology and the Sales Force: Increasing Acceptance of Sales Force Automation». eng. Em: *Industrial marketing management* 30.5 (2001), pp. 463–472. ISSN: 0019-8501. (Acesso em: 19/06/2023).
- [50] T.G. Cummings e C.G. Worley. *Organization Development and Change*. Cengage Learning, 2014. ISBN: 9781305143036. URL: <https://books.google.com.br/books?id=IuXKAgAAQBAJ>.
- [51] N Kristi. «Increase your odds for automation success». Em: *Sales and Marketing Strategies and News* 5 (1995), pp. 1–2.
- [52] Microsoft. *Função DDERequest*. (Acesso em: 02/06/2023). URL: <https://support.microsoft.com/pt-br/office/fun%C3%A7%C3%A3o-dderequest-be787695-67ec-4f26-a31b-1d55ec18d117>.
- [53] Kelvin T Erickson. «Programmable Logic Controllers: An emphasis on design and application». Em: (2016).
- [54] Smar - Technology Company. *SIS - Safety Instrumented Systems - A practical view*. (Acesso em: 14/06/2023). URL: <https://www.smar.com/en/technical-article/sis-safety-instrumented-systems-a-practical-view-part-1>.
- [55] Million In\$ights. *Productivity App Market Size, Share and Trends Analysis Report By Marketplace*. (Acesso em: 18/06/2023). URL: <https://www.millioninsights.com/snapshots/productivity-apps-market-report#:~:text=The%20global%20productivity%20app%20market,productivity%20apps%20across%20the%20globe..>

- [56] Million In\$ights. *Productivity Apps: An Insight Into The Fast Growing Market's Trends*. (Acesso em: 18/06/2023). URL: <https://www.digitalinformationworld.com/2022/04/productivity-apps-insight-into-fast.html>.
- [57] Marek Bekier e Brett R. C Molesworth. «Altering user' acceptance of automation through prior automation exposure». eng. Em: *Ergonomics* 60.6 (2017), pp. 745–753. ISSN: 0014-0139.
- [58] D. Cao et al. «Acceptance of automation manufacturing technology in China: an examination of perceived norm and organizational efficacy». Em: *Production Planning & Control* 31.8 (2020), pp. 660–672. DOI: 10.1080/09537287.2019.1669091. eprint: <https://doi.org/10.1080/09537287.2019.1669091>. URL: <https://doi.org/10.1080/09537287.2019.1669091>.
- [59] Ladder Logix World. *PLC Manufacturers: The Latest PLC Brands, Rankings & Revenues*. (Acesso em: 18/06/2023). URL: <https://ladderlogicworld.com/plc-manufacturers/>.