



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Vítor de Souza Barcala

**Identificação Automática de Símbolos em Diagramas P&ID com Visão
Computacional e Geração Assistida de Projetos de Automação Industrial**

Florianópolis
2026

Vítor de Souza Barcala

Identificação Automática de Símbolos em Diagramas P&ID com Visão Computacional e Geração Assistida de Projetos de Automação Industrial

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Felipe Gomes de Oliveira Cabral, Dr.

Supervisor: Vinicius Batschauer, Eng.

Florianópolis

2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Barcala, Vítor de Souza

Identificação Automática de Símbolos em Diagramas P&ID
com Visão Computacional e Geração Assistida de Projetos de
Automação Industrial / Vítor de Souza Barcala ; orientador,
Felipe Gomes de Oliveira Cabral, coorientador, Vinicius
Batschauer, 2026.

118 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia de Controle e Automação,
Florianópolis, 2026.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Automação
Industrial. 3. Diagramas P&ID. 4. Visão Computacional. 5.
Engenharia de Automação. I. Cabral, Felipe Gomes de
Oliveira. II. Batschauer, Vinicius. III. Universidade
Federal de Santa Catarina. Graduação em Engenharia de
Controle e Automação. IV. Título.

Vítor de Souza Barcala

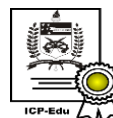
Identificação Automática de Símbolos em Diagramas P&ID com Visão Computacional e Geração Assistida de Projetos de Automação Industrial

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 03 de Agosto de 2026.

Prof. Marcelo de Lellis Costa de Oliveira, Dr.
Coordenador do Curso

Banca Examinadora:



Documento assinado digitalmente

Felipe Gomes de Oliveira Cabral

Data: 05/08/2026 12:29:15-0300

CPF: ***,807.927-**

Verifique as assinaturas em <https://v.ufsc.br>

Prof(a). Felipe Gomes de Oliveira Cabral, Dr(a).
Orientador(a)
UFSC/CTC/EAS

Vinicius Batschauer, Eng(a).
Supervisor(a)
GreyLogix Brasil



Documento assinado digitalmente

VINICIUS BATSCHAUER

Data: 05/08/2026 12:40:27-0300

CPF: ***,048.439-**

Verifique as assinaturas em <https://v.ufsc.br>

Prof(a). Diogo Ortiz Machado, Dr(a).
Avaliador(a)
UFSC/CTC/EAS

Prof. Hector Bessa Silveira, Dr.
Presidente da Banca
UFSC/CTC/EAS

Este trabalho é dedicado aos meus queridos pais e
minha amada.

AGRADECIMENTOS

Este trabalho não teria sido realizado sem o esforço e suporte de muitas pessoas, nas quais expresso minha profunda gratidão e dedico essas palavras de agradecimento. Primeiramente, gostaria de agradecer à minha família, especialmente aos meus pais, Rogério de Souza Barcala e Fabiana de Souza Barcala e à minha irmã, Beatriz de Souza Barcala, que desde sempre me trataram com muito amor e carinho, além de me apoiar de todas as formas possíveis ao longo da minha vida. Quero que saibam que cada momento que passei e irei passar juntamente com vocês foi especial, e que amo muito vocês.

À minha amada, Eduarda Bordignon Atuatti, que tenho o prazer de fazer parte da minha vida desde os meus 16 anos, e desde então me ajudou incontáveis vezes, incluindo a escolha deste curso e universidade. Você é meu exemplo de dedicação e perseverança inigualável, e me prova todos os dias que todos os esforços na vida serão recompensados. Obrigado por escolher me ter ao seu lado.

Academicamente, quero agradecer a todos os professores do departamento de Engenharia de Automação e Sistemas da UFSC, em especial ao prof. Felipe Gomes Cabral, que aceitou ser o orientador do meu trabalho e me guiou durante o seu desenvolvimento. Muito obrigado por todas as suas excelentes aulas, desde os esforços para lecionar Introdução a Engenharia de forma remota durante a pandemia, até as aulas extremamente técnicas, mas ainda intuitivas, de SEDs 2, e quero desejar sucesso e saúde a você e toda sua família. Faço menção também ao prof. Marcelo de Lellis Costa de Oliveira, que me recomendou como estagiário para a GreyLogix, e permitiu que eu iniciasse minha carreira profissional.

Profissionalmente, quero agradecer a toda a equipe da GreyLogix, em especial ao meu supervisor Vinicius Batschauer e o meu coordenador Adir Trevisan Jr. Seus conhecimentos técnicos e conselhos pessoais me ajudaram imensamente durante todo o período que faço parte da GreyLogix, e agradeço muito a oportunidade de crescer juntamente com essa excelente equipe de engenharia. Faço menção também ao meu ex-gerente de projetos, Guilherme Cornelli, que me ajudou a estruturar a ideia do meu trabalho e me acolheu no início da minha trajetória na empresa.

Quero agradecer aos meus amigos, que me ajudaram de todas as formas possíveis durante a minha trajetória na UFSC, desde sessões de estudo até momentos de celebração. Ninguém se forma sozinho, e fico feliz se pude ajudar vocês da forma como me ajudaram.

Por fim, quero agradecer a mim, por não ter desistido e ido até o fim com este trabalho e curso. Apesar das dificuldades, o esforço vale a pena, e agradeço a oportunidade de poder provar isso daqui em diante.

DECLARAÇÃO DE PUBLICIDADE

Florianópolis, 24 de Julho de 2026.

Na condição de representante da GreyLogix Brasil na qual o presente trabalho foi realizado, declaro não haver ressalvas quanto ao aspecto de sigilo ou propriedade intelectual sobre as informações contidas neste documento, que impeçam a sua publicação por parte da Universidade Federal de Santa Catarina (UFSC) para acesso pelo público em geral, incluindo a sua disponibilização *online* no Repositório Institucional da Biblioteca Universitária da UFSC. Além disso, declaro ciência de que o autor, na condição de estudante da UFSC, é obrigado a depositar este documento, por se tratar de um Trabalho de Conclusão de Curso, no referido Repositório Institucional, em atendimento à Resolução Normativa n° 126/2019/CUn.

Por estar de acordo com esses termos, subscrevo-me abaixo.

Documento assinado digitalmente
 VINICIUS BATSCHAUER
Data: 27/07/2026 11:36:15-0300
Verifique em <https://validar.iti.gov.br>

Vinicius Batschauer
GreyLogix Brasil

RESUMO

Projetos de automação industrial dependem da transferência consistente de informações entre a documentação técnica de processo e os ambientes de engenharia utilizados na configuração dos sistemas de controle. Entre esses documentos, os diagramas P&ID reúnem informações sobre instrumentos, equipamentos, atuadores e malhas de controle. Entretanto, a identificação dos elementos representados e a transcrição de seus rótulos (do inglês, *tags*) para as estruturas do projeto ainda são realizadas, em muitos fluxos de trabalho, de forma predominantemente manual, favorecendo a ocorrência de erros, inconsistências e atividades repetitivas. Nesse contexto, este trabalho desenvolveu e avaliou um sistema para a extração assistida de símbolos de instrumentação e de suas respectivas *tags* em diagramas P&ID, bem como para sua utilização na geração de estruturas iniciais de projetos de automação. A solução combinou técnicas de visão computacional, detecção de objetos, reconhecimento óptico de caracteres, representação estruturada dos dados, validação técnica e mapeamento das informações para objetos de engenharia. A arquitetura foi organizada de forma modular, mantendo separadas as etapas de processamento dos diagramas, revisão dos resultados e integração com o ambiente de destino. Como estudo de caso, a etapa de geração dos objetos foi implementada no Siemens TIA Portal, plataforma utilizada nos projetos analisados na GreyLogix. No conjunto de teste, o modelo de detecção alcançou precisão de 98,46%, revocação de 99,4% e mAP@0,5 de 99,47%. Nos dois diagramas completos utilizados na avaliação da solução, todos os 144 símbolos de interesse foram localizados e corretamente classificados, enquanto cinco detecções adicionais foram removidas durante a validação. O reconhecimento textual identificou corretamente 142 das 144 *tags*, sendo necessárias duas correções manuais. A integração foi verificada em um caso contendo 55 elementos, para os quais foram geradas e importadas as estruturas de engenharia previstas. Não foi realizada uma comparação quantitativa controlada entre o procedimento manual e o fluxo assistido; portanto, os possíveis ganhos de tempo e a redução de retrabalho são discutidos qualitativamente. Os resultados demonstraram a viabilidade do fluxo proposto e indicaram seu potencial para apoiar a padronização das informações e reduzir atividades repetitivas, preservando a validação técnica e a responsabilidade do engenheiro.

Palavras-chave: Automação industrial. Diagramas P&ID. Visão computacional. Engenharia de Automação.

ABSTRACT

Industrial automation projects depend on the consistent transfer of information between process documentation and the engineering environments used to configure control systems. Among these documents, P&ID diagrams provide information on instruments, equipment, actuators, and control loops. However, in many engineering workflows, the identification of the represented elements and the transcription of their tags into project structures are still predominantly performed manually, which may lead to errors, inconsistencies, and repetitive activities. In this context, this work developed and evaluated a system for the assisted extraction of instrumentation symbols and their corresponding tags from P&ID diagrams, as well as for their use in generating initial automation project structures. The solution combined computer vision techniques, object detection, optical character recognition, structured data representation, technical validation, and the mapping of extracted information to engineering objects. The architecture was organized in a modular manner, keeping diagram processing, result review, and integration with the target environment as separate stages. As a case study, the object generation stage was implemented in Siemens TIA Portal, the platform used in the projects analyzed at GreyLogix. On the test set, the detection model achieved a precision of 98.46%, a recall of 99.4%, and an mAP@0.5 of 99.47%. In the two complete diagrams used to evaluate the solution, all 144 symbols of interest were detected and correctly classified, while five additional detections were removed during validation. The text recognition stage correctly identified 142 of the 144 tags, requiring two manual corrections. Integration was verified in a case containing 55 elements, for which the expected engineering structures were generated and imported. No controlled quantitative comparison was performed between the manual procedure and the assisted workflow; therefore, potential time savings and reductions in rework are discussed qualitatively. The results demonstrated the feasibility of the proposed workflow and indicated its potential to support information standardization and reduce repetitive activities while preserving technical validation and the engineer's responsibility.

Keywords: Industrial automation. P&ID diagrams. Computer vision. Engineering automation.

LISTA DE FIGURAS

Figura 1 – Exemplo de diagrama P&ID com instrumentos, linhas de sinal e elementos de processo.	27
Figura 2 – Exemplo de um símbolo de instrumentação encontrado em um P&ID	30
Figura 3 – Exemplo de organização de objetos de engenharia em um projeto no TIA Portal.	34
Figura 4 – Representação gráfica de uma lógica de controle em LAD no ambiente TIA Portal.	36
Figura 5 – Representação textual de uma lógica de controle em SCL no ambiente TIA Portal.	36
Figura 6 – Exemplo de imagem anotada com caixas delimitadoras e classes de símbolos em P&ID.	41
Figura 7 – Arquitetura conceitual e fluxo sequencial da solução proposta. . . .	61
Figura 8 – Visão geral da implementação desenvolvida.	73
Figura 9 – Curvas de precisão e revocação do modelo de detecção.	80
Figura 10 – Matriz de confusão do modelo no conjunto de teste.	81
Figura 11 – Variação da métrica F1 em função do limiar de confiança.	82
Figura 12 – Exemplo de detecção dos símbolos de instrumentação em recorte de um diagrama P&ID.	96
Figura 13 – Exemplo da aplicação utilizada para a validação das detecções e das <i>tags</i> reconhecidas.	99
Figura 14 – Exemplos de objetos gerados e importados no projeto do TIA Portal.	100

LISTA DE TABELAS

Tabela 2 – Síntese parcial de símbolos para dispositivos de instrumentação em P&IDs, segundo a norma ANSI/ISA-5.1.	28
Tabela 3 – Identificação funcional de instrumentos conforme a ISA-5.1	29
Tabela 4 – Resultados possíveis na comparação entre predições e anotações de referência.	46
Tabela 5 – Síntese dos trabalhos relacionados à interpretação computacional de diagramas P&ID.	51
Tabela 6 – Requisitos funcionais da solução proposta	57
Tabela 7 – Requisitos não funcionais da solução proposta	58
Tabela 8 – Classes de símbolos de instrumentação consideradas pela solução proposta.	64
Tabela 9 – Principais campos da representação intermediária dos dados.	66
Tabela 10 – Distribuição das imagens entre os subconjuntos utilizados.	75
Tabela 11 – Distribuição das instâncias anotadas por classe nos subconjuntos utilizados.	76
Tabela 12 – Principais parâmetros utilizados no treinamento do modelo de detecção.	78
Tabela 13 – Resultados do modelo de detecção no conjunto de teste.	79
Tabela 14 – Principais parâmetros utilizados no reconhecimento textual.	85
Tabela 15 – Resultados do detector nos diagramas de aplicação analisados.	95
Tabela 16 – Resultados do reconhecimento das <i>tags</i> nos diagramas de aplicação.	97
Tabela 17 – Atendimento aos requisitos funcionais da solução.	105
Tabela 18 – Atendimento aos requisitos não funcionais da solução.	107

LISTA DE ABREVIATURAS E SIGLAS

AP	Precisão Média, do inglês <i>Average Precision</i>
CLP	Controlador Lógico Programável
CNN	Rede Neural Convolucional, do inglês <i>Convolutional Neural Network</i>
DB	Bloco de dados, do inglês <i>Data Block</i>
FB	Bloco de função, do inglês <i>Function Block</i>
FC	Função, do inglês <i>Function</i>
I/O	Entradas e saídas, do inglês <i>Input/Output</i>
IHM	Interface Humano-Máquina
IoU	Interseção sobre União, do inglês <i>Intersection over Union</i>
ISA	Sociedade Internacional de Automação, do inglês <i>International Society of Automation</i>
LAD	Diagrama Ladder, do inglês <i>Ladder Diagram</i>
LAPIX	Laboratório de Processamento de Imagens e Computação Gráfica
mAP	Precisão média das classes, do inglês <i>mean Average Precision</i>
OB	Bloco organizacional, do inglês <i>Organization Block</i>
OCR	Reconhecimento Óptico de Caracteres, do inglês <i>Optical Character Recognition</i>
P&ID	Diagrama de Tubulação e Instrumentação, do inglês <i>Piping and Instrumentation Diagram</i>
PFC	Projeto de Fim de Curso
SCL	Linguagem Estruturada de Controle, do inglês <i>Structured Control Language</i>
UDT	Tipo de dado definido pelo usuário, do inglês <i>User-Defined Data Type</i>
XML	Linguagem de Marcação Extensível, do inglês <i>Extensible Markup Language</i>
YOLO	Você Só Olha Uma Vez, do inglês <i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	GREYLOGIX MÁQUINAS BRASIL	15
1.2	CONTEXTUALIZAÇÃO E PROBLEMA DE PESQUISA	15
1.3	OBJETIVOS	17
1.4	PROPOSTA E METODOLOGIA DO PROJETO	18
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	PROJETOS DE AUTOMAÇÃO INDUSTRIAL E DOCUMENTAÇÃO TÉCNICA	22
2.1.1	Ciclo de desenvolvimento de projetos de automação	23
2.1.2	Documentação técnica em projetos de automação	23
2.1.3	Controladores lógicos programáveis em sistemas industriais	24
2.1.4	Bibliotecas de engenharia e padronização de projetos	25
2.2	DIAGRAMAS P&ID E SIMBOLOGIA DE INSTRUMENTAÇÃO	25
2.2.1	Conceito e finalidade dos diagramas P&ID	25
2.2.2	Elementos e simbologia em diagramas P&ID	26
2.2.3	Norma ISA-5.1 para símbolos e nomenclatura de tags	27
2.2.4	Relação entre P&IDs e projetos de automação	29
2.2.5	Desafios da leitura automática de P&IDs	31
2.3	SIEMENS TIA PORTAL E TIA PORTAL OPENNESS	32
2.3.1	Siemens TIA Portal como ambiente integrado de engenharia	32
2.3.2	Estrutura de programação no TIA Portal	33
2.3.3	Linguagens LAD e SCL em projetos de automação	35
2.3.4	Siemens TIA Portal Openness	37
2.3.5	SIMATIC ML	38
2.4	VISÃO COMPUTACIONAL APLICADA À EXTRAÇÃO DE INFORMA- ÇÕES EM DOCUMENTOS TÉCNICOS	39
2.4.1	Fundamentos de visão computacional	39
2.4.2	Processamento e pré-processamento de imagens	39
2.4.3	Aprendizado supervisionado, datasets e rotulagem de imagens	40
2.4.4	Deteção de objetos em imagens	42
2.4.5	Redes neurais convolucionais e modelos de aprendizado profundo	43
2.4.6	Modelos YOLO aplicados à detecção de símbolos	44
2.4.7	OCR em documentos técnicos e associação entre símbolos e tags	45
2.4.8	Métricas de avaliação do módulo de detecção	45
2.4.9	Trabalhos relacionados e posicionamento da contribuição	49
3	DESCRIÇÃO DO PROBLEMA E REQUISITOS TÉCNICOS	53
3.1	CONTEXTO TÉCNICO DE APLICAÇÃO	53

3.2	FLUXO ATUAL DE EXTRAÇÃO DE INFORMAÇÕES A PARTIR DA DOCUMENTAÇÃO TÉCNICA	54
3.3	CARACTERIZAÇÃO DO PROBLEMA TRATADO	55
3.4	REQUISITOS TÉCNICOS	56
4	SOLUÇÃO PROPOSTA E METODOLOGIA UTILIZADA	59
4.1	VISÃO GERAL DA SOLUÇÃO PROPOSTA	59
4.2	ESTRUTURA E ETAPAS DA SOLUÇÃO	62
4.2.1	Preparação dos documentos de entrada	62
4.2.2	Identificação dos símbolos de instrumentação	63
4.2.3	Reconhecimento de tags e associação símbolo-tag	64
4.2.4	Representação intermediária dos dados	65
4.2.5	Validação técnica pelo engenheiro	67
4.2.6	Mapeamento para objetos de engenharia	68
4.2.7	Integração com o ambiente de engenharia	68
4.3	CRITÉRIOS DE AVALIAÇÃO E VALIDAÇÃO DA SOLUÇÃO	69
5	DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS OBTIDOS .	71
5.1	VISÃO GERAL DA IMPLEMENTAÇÃO	71
5.2	DESENVOLVIMENTO E AVALIAÇÃO DO MÓDULO DE DETECÇÃO	74
5.2.1	Seleção e organização do conjunto de dados	74
5.2.2	Definição das classes e preparação das anotações	75
5.2.3	Configuração e treinamento do modelo YOLO	76
5.2.4	Avaliação do modelo de detecção	78
5.2.5	Exportação e aplicação do modelo na inferência	81
5.3	RECONHECIMENTO TEXTUAL E ASSOCIAÇÃO ENTRE SÍMBOLOS E TAGS	83
5.3.1	Definição das regiões de busca	83
5.3.2	Aplicação do módulo de OCR	84
5.3.3	Associação das tags aos símbolos detectados	85
5.4	ESTRUTURAÇÃO DOS DADOS E APLICAÇÃO DE VALIDAÇÃO . .	86
5.4.1	Geração da representação em JSON	86
5.4.2	Desenvolvimento da aplicação de validação	87
5.5	MAPEAMENTO E INTEGRAÇÃO NO ESTUDO DE CASO SIEMENS	88
5.5.1	Leitura e classificação das tags validadas	88
5.5.2	Desenvolvimento da aplicação em C#/.NET	90
5.5.3	Geração dos arquivos SIMATIC ML a partir de templates	91
5.5.4	Utilização das estruturas padronizadas da biblioteca LCFL	92
5.5.5	Importação dos objetos por meio do TIA Portal Openness	93
5.6	TESTES E RESULTADOS OBTIDOS	93
5.6.1	Comportamento do detector nos diagramas de aplicação	94

5.6.2	Resultados do reconhecimento e da associação das tags	96
5.6.3	Resultados da validação técnica	98
5.6.4	Resultados da geração de objetos no TIA Portal	98
5.7	ANÁLISE CRÍTICA DA SOLUÇÃO	99
6	CONCLUSÃO	104
6.1	SÍNTESE DO TRABALHO E ATENDIMENTO AOS OBJETIVOS . . .	104
6.2	IMPACTOS E LIMITAÇÕES DA SOLUÇÃO	108
6.3	POSSIBILIDADES DE APRIMORAMENTO E TRABALHOS FUTUROS	110
	REFERÊNCIAS	112

1 INTRODUÇÃO

1.1 GREYLOGIX MÁQUINAS BRASIL

A GreyLogix Brasil é uma empresa de engenharia atuante no setor de automação industrial, com atividades relacionadas ao desenvolvimento de soluções para plantas produtivas, incluindo engenharia de automação, fabricação de painéis elétricos, serviços de campo, consultoria técnica e integração de sistemas industriais. No contexto deste Projeto de Fim de Curso (PFC), a empresa constitui o ambiente de aplicação do problema estudado, uma vez que seu fluxo de trabalho envolve a interpretação de documentação técnica de processo e a posterior configuração de estruturas de automação em ambientes de engenharia (GreyLogix Brasil, 2026).

A atuação da empresa está inserida em um cenário de transformação da automação industrial, caracterizado pela digitalização dos processos de engenharia, pela integração entre sistemas de controle e supervisão e pela crescente necessidade de interoperabilidade entre diferentes ferramentas. Esse cenário está associado aos princípios da Indústria 4.0, que favorecem arquiteturas digitais orientadas à conectividade, à troca estruturada de dados e à integração entre as tecnologias de operação e de informação (Folgado et al., 2024; Grüner et al., 2023).

No âmbito dos projetos de automação desenvolvidos pela empresa, uma etapa relevante consiste em transformar informações contidas em documentos técnicos, especialmente em Diagrama de Tubulação e Instrumentação, do inglês *Piping and Instrumentation Diagram* (P&ID), em estruturas compatíveis com ambientes de engenharia. Na GreyLogix, essa transição envolve a identificação de ativos representados nos diagramas, como sensores, motores, válvulas e atuadores, e sua conversão em objetos de engenharia, tabelas de *tags*, estruturas de projeto e blocos funcionais aderentes aos padrões técnicos internos da empresa. Esse processo ocorre em um contexto no qual padronização, modularidade e reuso são relevantes para assegurar consistência, rastreabilidade e manutenção dos projetos (Vogel-Heuser et al., 2017).

Dessa forma, o fluxo de trabalho analisado apresenta uma oportunidade de aumentar a produtividade e a consistência da configuração inicial dos projetos de automação. O apoio computacional à identificação e à organização das informações presentes nos diagramas P&ID pode reduzir o tempo dedicado à interpretação e à transcrição manual, além de contribuir para mitigar omissões, duplicidades e divergências de nomenclatura.

1.2 CONTEXTUALIZAÇÃO E PROBLEMA DE PESQUISA

A execução de projetos de automação industrial depende da consistência entre a documentação técnica, os dispositivos de campo, os controladores, as interfaces

de operação e os ambientes de engenharia. Ao longo desse processo, informações descritas em documentos de processo e de detalhamento precisam ser interpretadas e convertidas em estruturas de software, como tabelas de *tags*, blocos de função, tipos de dados, telas de supervisão e tabelas de Entradas e saídas, do inglês *Input/Output* (I/O). A qualidade dessa transição influencia a rastreabilidade entre a documentação e a implementação, além de afetar atividades de manutenção, expansão, reutilização e integração ao longo do ciclo de engenharia (Vogel-Heuser et al., 2017; Grüner et al., 2023).

Entre os documentos utilizados em projetos industriais, os diagramas P&ID desempenham papel importante na representação de equipamentos, tubulações, instrumentos, válvulas, malhas de controle e relações funcionais do processo. Esses diagramas empregam convenções de simbologia e identificação definidas ou orientadas por normas técnicas, como a ANSI/ISA-5.1 e a ISO 10628, e constituem uma das principais fontes de informação para etapas posteriores de engenharia (International Society of Automation, 2024; International Organization for Standardization, 2014, 2012). Além de representarem o processo, os P&IDs podem fornecer dados reutilizáveis em tarefas de digitalização, simulação e geração assistida de estruturas de engenharia (Arroyo et al., 2016; Rahul et al., 2019; Paliwal et al., 2021).

Apesar da existência de normas, bibliotecas de engenharia e softwares especializados, a interpretação de P&IDs ainda envolve forte participação manual. Em muitos fluxos de trabalho, incluindo o observado na GreyLogix, o engenheiro precisa identificar visualmente os instrumentos, sensores, motores, válvulas e atuadores, associar esses elementos às respectivas *tags*¹ (termo em inglês utilizado para designar rótulos) e convertê-los em objetos compatíveis com o ambiente de desenvolvimento do projeto. Essa etapa é sensível a inconsistências, erros e atrasos no fluxo de engenharia, sobretudo em documentos densos, com ruído, variações de simbologia, classes visualmente semelhantes e grande volume de ativos (Rahul et al., 2019; Paliwal et al., 2021; Jamieson; Moreno-García; Elyan, 2024; Kim et al., 2022).

Esse problema tem motivado o desenvolvimento de métodos computacionais para a extração de informações em documentos técnicos. Estudos relacionados à digitalização de P&IDs mostram que essa tarefa não se limita ao reconhecimento isolado de símbolos. Ela pode envolver também o reconhecimento de textos, a associação entre símbolos e *tags*, a identificação de linhas e conexões e a reconstrução das relações estruturais do diagrama. Dessa forma, a interpretação computacional desses documentos exige a combinação de técnicas de visão computacional, Reconhecimento Óptico de Caracteres, do inglês *Optical Character Recognition* (OCR), regras de associação espacial e formas estruturadas de representação dos dados (Rahul et al., 2019; Paliwal

¹ Neste trabalho, o termo *tag* é utilizado para se referir aos rótulos de variáveis em Controladores Lógicos Programáveis (CLPs).

et al., 2021; Jamieson; Moreno-García; Elyan, 2024; Stürmer; Graumann; Koch, 2025).

No contexto deste trabalho, o problema de engenharia concentra-se na etapa inicial de configuração de projetos de automação a partir de diagramas P&ID. Nessa etapa, informações referentes principalmente a instrumentos e sensores precisam ser identificadas, organizadas e convertidas em estruturas iniciais no software de programação do Controlador Lógico Programável (CLP). Assim, busca-se investigar como técnicas de visão computacional podem ser articuladas a uma camada intermediária de mapeamento de dados e a uma interface de automação de engenharia, de modo a reduzir a dependência de procedimentos manuais na criação dessas estruturas.

Portanto, a automação assistida apresenta-se como uma alternativa para reduzir atividades repetitivas, aumentar a consistência das informações e apoiar a configuração inicial de projetos de automação industrial. Entretanto, para que essa abordagem seja tecnicamente aplicável, não basta reconhecer símbolos de forma isolada. É necessário integrar a detecção automática de elementos gráficos à organização estruturada dos dados e à geração de objetos compatíveis com o ambiente de engenharia utilizado pela empresa. Desse modo, a solução proposta deve combinar extração computacional de informações, mapeamento intermediário e integração com software de programação do CLP, utilizando automação de fluxos de trabalho. Essa abordagem se alinha à literatura que defende fluxos semiautomáticos, nos quais a extração computacional de informações reduz esforço repetitivo, mas a validação técnica permanece sob responsabilidade do especialista (Sierla et al., 2020; Grüner et al., 2023).

1.3 OBJETIVOS

O objetivo geral deste trabalho é desenvolver e avaliar um sistema integrado, baseado em técnicas de visão computacional e em recursos de automação de engenharia, para identificar símbolos de instrumentação e suas respectivas *tags* em diagramas P&ID e apoiar a geração de estruturas iniciais de projetos de automação.

Para alcançar esse propósito, foram definidos os seguintes objetivos específicos:

- Selecionar, preparar e organizar uma base de dados (do inglês, *dataset*) de símbolos presentes em diagramas P&ID, adequando suas classes e anotações ao treinamento e à avaliação do modelo de detecção
- Implementar um módulo de visão computacional capaz de detectar, localizar e classificar as classes de símbolos de instrumentação definidas no escopo do trabalho.
- Implementar um módulo de reconhecimento textual para extrair as *tags* associadas aos símbolos detectados e estabelecer a correspondência entre as informações gráficas e textuais.

- Desenvolver uma representação intermediária dos dados e uma aplicação de validação que permita ao engenheiro revisar, corrigir, excluir ou complementar os registros produzidos pelo processamento automático.
- Implementar uma aplicação em C# e .NET Framework integrada ao software da fabricante Siemens, como um protótipo e estudo de caso de aplicação, capaz de mapear os instrumentos validados para os padrões técnicos adotados e gerar estruturas iniciais compatíveis com o ambiente de engenharia.
- Avaliar a solução por meio de métricas de detecção e da verificação funcional das etapas de reconhecimento textual, validação dos dados, mapeamento e geração das estruturas iniciais no ambiente de engenharia adotado no estudo de caso.

1.4 PROPOSTA E METODOLOGIA DO PROJETO

A partir do problema apresentado, este trabalho propõe um sistema para a extração assistida de informações contidas em diagramas P&ID e sua utilização na geração de estruturas iniciais de projetos de automação. A proposta busca estabelecer um fluxo entre a documentação técnica de processo e os ambientes de engenharia, reduzindo a necessidade de transcrição manual das informações relacionadas aos instrumentos considerados no escopo. Essa abordagem está relacionada a trabalhos que investigam a transferência de informações entre a documentação de processo e as estruturas utilizadas no desenvolvimento de sistemas de automação (Koltun; Kolter; Vogel-Heuser, 2018; Papakonstantinou et al., 2019).

A solução foi organizada em módulos responsáveis pela preparação dos documentos, detecção dos símbolos, reconhecimento das *tags*, estruturação dos dados, validação técnica, mapeamento das informações e integração com o ambiente de engenharia. Os diagramas são preparados para a aplicação de um modelo de detecção de objetos, treinado para reconhecer as duas classes de símbolos de instrumentação selecionadas para o trabalho. A combinação entre detecção de elementos gráficos e reconhecimento de informações textuais é recorrente em abordagens destinadas à digitalização e à extração de informações de diagramas P&ID (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022).

As regiões correspondentes aos símbolos detectados são posteriormente encaminhadas a um módulo de reconhecimento textual, responsável pela leitura das identificações associadas aos instrumentos. A utilização das regiões localizadas pelo detector permite direcionar o reconhecimento aos elementos de interesse e estabelecer a associação entre os símbolos e suas respectivas *tags*. Abordagens semelhantes são empregadas em trabalhos que combinam detecção de objetos, reconhecimento textual e relações espaciais para interpretar documentos técnicos (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022).

As informações obtidas são organizadas em uma representação intermediária estruturada, contendo, para cada registro, a classe do símbolo, o valor de confiança da detecção, sua localização no documento, a página de origem e a *tag* associada. Essa representação separa os módulos de processamento visual das regras específicas do ambiente de automação, permite preservar a rastreabilidade das informações extraídas e estabelece uma interface entre as diferentes etapas do fluxo. A transferência estruturada de informações entre etapas e ferramentas de engenharia contribui para reduzir inconsistências e favorecer a continuidade do desenvolvimento (Regulin; Aicher; Vogel-Heuser, 2016; Papakonstantinou et al., 2019; Grüner et al., 2023).

Antes da integração com o ambiente de engenharia, os registros são submetidos a uma etapa de validação técnica. Para isso, foi desenvolvida uma aplicação que permite ao engenheiro examinar cada detecção, confirmar ou alterar sua classe, corrigir a *tag* reconhecida, excluir falsos positivos e inserir manualmente símbolos que não tenham sido identificados pelo modelo. Essa etapa impede que os resultados do processamento sejam incorporados diretamente ao projeto sem revisão e preserva a responsabilidade técnica do profissional. A manutenção da intervenção do especialista é compatível com abordagens semiautomáticas, nas quais os recursos computacionais apoiam a extração e a organização dos dados, sem substituir sua validação técnica (Sierla et al., 2020; Grüner et al., 2023).

Após a validação, os dados revisados são processados por uma camada de mapeamento, responsável por interpretar as *tags*, classificar os instrumentos e relacioná-los às estruturas definidas para o projeto de automação. A separação entre a extração das informações e as regras de geração permite modificar os padrões, bibliotecas e objetos adotados no ambiente de destino sem alterar os módulos responsáveis pela leitura dos diagramas. Essa organização está relacionada aos princípios de modularidade, reutilização e geração assistida empregados em ferramentas de engenharia de automação (Maga; Jazdi; Göhner, 2011; Koziolk et al., 2020a).

Como protótipo e estudo de caso, a etapa de integração foi implementada no Siemens TIA Portal, ambiente utilizado nos projetos analisados na GreyLogix. Essa escolha está relacionada ao contexto tecnológico da empresa e à disponibilidade de mecanismos que permitem a manipulação programática dos objetos de engenharia da plataforma (Siemens, 2026b, 2026a). A organização geral do fluxo, entretanto, permanece separada dos recursos específicos do fabricante. Dessa forma, sua adaptação a outros ambientes de engenharia depende da implementação de novas regras de mapeamento e de uma camada de integração compatível com as interfaces disponibilizadas pela plataforma de destino.

No estudo de caso desenvolvido, os dados validados são utilizados para gerar e importar tabelas de *tags*, blocos de função, blocos de configuração e blocos de dados no projeto de automação. Os detalhes das tecnologias, dos formatos de dados

e dos recursos específicos empregados nessa implementação são apresentados nos capítulos de metodologia e desenvolvimento.

A avaliação da solução foi dividida em diferentes níveis. O desempenho do modelo de detecção foi analisado por meio de métricas de precisão, revocação e precisão média, amplamente utilizadas na avaliação de algoritmos de detecção de objetos (Padilla; Netto; Silva, 2020). O reconhecimento textual e a associação entre símbolos e *tags* foram verificados nos diagramas utilizados como casos de aplicação. A validação técnica foi analisada a partir das correções, exclusões e inserções realizadas durante a revisão dos registros, enquanto a integração com o ambiente de engenharia foi verificada por meio da criação e da importação dos objetos definidos no escopo.

A proposta não busca substituir a análise técnica do engenheiro nem realizar a interpretação integral de todos os elementos funcionais do processo. Seu foco está na redução de atividades repetitivas da etapa inicial de configuração, oferecendo apoio à criação de estruturas de projeto a partir de informações identificadas nos diagramas P&ID e organizadas de modo padronizado. Dessa forma, o sistema é concebido como uma ferramenta de apoio à engenharia, e não como um mecanismo autônomo de geração de projetos.

A principal contribuição deste trabalho consiste na proposição e na avaliação de uma arquitetura modular para um fluxo integrado de engenharia, que encadeia a preparação dos documentos, a detecção e a classificação dos símbolos, o reconhecimento textual, a representação intermediária dos dados, a validação técnica, o mapeamento das informações e a geração assistida de objetos. O diferencial da proposta não está isoladamente no emprego de um modelo de detecção ou de um mecanismo de OCR, mas na articulação dessas etapas até a produção de estruturas utilizáveis em um ambiente de engenharia de automação. Embora o protótipo tenha sido implementado com ferramentas específicas e avaliado no Siemens TIA Portal, a separação entre os módulos reduz o acoplamento entre o processamento dos diagramas e a plataforma de destino, favorecendo a substituição de componentes e a adaptação da camada de integração a outros ambientes de engenharia.

ESTRUTURA DO DOCUMENTO

A partir desta introdução, o trabalho está organizado em cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica, abordando os conceitos necessários à compreensão do problema, incluindo projetos de automação industrial, diagramas P&ID, visão computacional, detecção de objetos, OCR, Siemens TIA Portal e TIA Portal Openness. O Capítulo 3 caracteriza o problema de engenharia no contexto da GreyLogix, descreve o fluxo atual de extração e transcrição das informações e apresenta os requisitos funcionais e não funcionais definidos para a solução. O Capítulo 4 apresenta a solução proposta e a abordagem metodológica adotada, detalhando sua arquitetura,

as etapas de preparação dos documentos, detecção dos símbolos, reconhecimento das *tags*, representação intermediária, validação técnica, mapeamento dos dados e integração com o Siemens TIA Portal. Também são definidos os critérios utilizados na avaliação da solução. O Capítulo 5 descreve a implementação dos módulos desenvolvidos e analisa os resultados obtidos. São apresentados a preparação do conjunto de dados, o treinamento e a avaliação do modelo de detecção, o reconhecimento textual, a aplicação de validação, o mapeamento dos instrumentos, a geração dos arquivos SIMATIC ML e os testes de integração com o TIA Portal. Por fim, o Capítulo 6 reúne as conclusões do projeto, suas limitações e as possibilidades de aprimoramento em desenvolvimentos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos teóricos necessários ao desenvolvimento do sistema proposto nesse PFC. Na Seção 2.1 são discutidos os conceitos relacionados a projetos de automação industrial e à documentação técnica utilizada nesse contexto. Em seguida, a Seção 2.2 aborda os diagramas P&ID e a simbologia de instrumentação, destacando sua importância para a identificação de ativos industriais. Na sequência, na seção 2.3 são apresentados o Siemens TIA Portal e o TIA Portal Openness, considerando sua aplicação na estruturação e automação de projetos de engenharia. Por fim, na Seção 2.4 são discutidos os fundamentos de visão computacional aplicados à extração de informações em documentos técnicos, incluindo pré-processamento de imagens, detecção de objetos, OCR e métricas de avaliação.

2.1 PROJETOS DE AUTOMAÇÃO INDUSTRIAL E DOCUMENTAÇÃO TÉCNICA

Esta seção apresenta os conceitos gerais relacionados ao desenvolvimento de projetos de automação industrial e ao papel da documentação técnica nesse processo. A discussão é importante para entender o processo geral de desenvolvimento de projetos de automação e compreender como informações extraídas de documentos de engenharia podem ser convertidas em estruturas utilizadas nos ambientes de automação.

A automação industrial compreende a integração de dispositivos de medição, controle, atuação, supervisão e comunicação, com o objetivo de operar processos produtivos de forma segura, repetível e aderente aos requisitos técnicos da planta. Nesse contexto, sensores, atuadores, controladores, interfaces de operação e sistemas supervisórios compõem a infraestrutura responsável por adquirir variáveis de processo, executar estratégias de controle, comandar equipamentos e disponibilizar informações necessárias às atividades de operação, manutenção e diagnóstico (Folgado et al., 2024; Vyatkin, 2013).

A evolução e digitalização dos sistemas industriais têm ampliado a necessidade de integração entre automação, documentação técnica e ambientes digitais de engenharia. De acordo com os princípios da Indústria 4.0, sistemas de controle e supervisão passam a ser organizados em arquiteturas mais conectadas, nas quais a consistência das informações presentes nos documentos de engenharia torna-se fundamental para garantir rastreabilidade, interoperabilidade e manutenção adequada dos sistemas automatizados (Folgado et al., 2024).

2.1.1 Ciclo de desenvolvimento de projetos de automação

O desenvolvimento de um projeto de automação industrial envolve a transformação dos requisitos de processo em estruturas implementáveis de controle e supervisão. De modo geral, esse fluxo abrange a análise de requisitos e da documentação técnica, o levantamento de instrumentos e sinais de campo, a definição de entradas e saídas, a estruturação de *tags*, a programação do CLP, a configuração da Interface Humano-Máquina (IHM), além de testes, integração e comissionamento do sistema (Jetley et al., 2013; Dürkop et al., 2015). Embora essas etapas possam ser apresentadas de forma sequencial, sua execução prática tende a ser iterativa e interdisciplinar, com sucessivos ajustes entre projeto, implementação e validação (Vogel-Heuser et al., 2014; Vyatkin, 2013).

Entre as atividades recorrentes desse ciclo, destaca-se a conversão de informações presentes em documentos de engenharia para estruturas equivalentes no ambiente de desenvolvimento. Essa etapa pode envolver a criação de tabelas de *tags*, a organização da árvore do projeto, o uso de bibliotecas padronizadas, a instanciamento de blocos funcionais e a associação entre ativos industriais e seus respectivos objetos de automação (Regulin; Aicher; Vogel-Heuser, 2016; Papakonstantinou et al., 2019; Vogel-Heuser et al., 2018). Quando realizada manualmente, essa transposição tende a aumentar o esforço de engenharia e o risco de inconsistências, sobretudo em projetos com grande volume de instrumentos, ativos e padrões internos de nomenclatura (Regulin; Aicher; Vogel-Heuser, 2016; Papakonstantinou et al., 2019; Koziolk et al., 2020a, 2020b).

Nessa perspectiva, a documentação técnica atua como elemento de coordenação entre o processo representado, os dispositivos de campo e as estruturas implementadas no ambiente de automação. Quando as informações de engenharia estão fragmentadas, ambíguas ou pouco padronizadas, cresce o esforço necessário para alinhar software, hardware e operação, especialmente nas etapas de integração e comissionamento (Dürkop et al., 2015; Winkler; Biffl, 2012). Por essa razão, abordagens orientadas à rastreabilidade e à integração entre modelos, documentos e implementação tendem a reduzir retrabalho e a tornar o desenvolvimento mais previsível (Vogel-Heuser et al., 2014; Koziolk et al., 2020b).

2.1.2 Documentação técnica em projetos de automação

A documentação técnica ocupa papel central em projetos de automação, pois organiza as informações necessárias para representar o processo, especificar os elementos de campo e orientar sua implementação nos sistemas de controle e supervisão. Documentos como fluxogramas de processo, P&IDs, listas de instrumentos, listas de I/O, diagramas elétricos, descritivos lógicos e descritivos de operação contribuem para

manter a coerência entre a concepção do processo, a definição dos ativos industriais e sua implementação no sistema de automação (Winkler; Biffl, 2012; Papakonstantinou et al., 2019).

Dentre esses documentos, os diagramas P&ID assumem relevância particular por reunirem, em uma mesma representação, informações sobre instrumentos, válvulas, equipamentos, linhas de processo, malhas de controle e relações funcionais importantes para a automação. Por concentrarem dados técnicos utilizados em diferentes etapas do projeto, esses diagramas servem de base para a interpretação dos ativos industriais e para a organização das informações que serão posteriormente utilizadas nos ambientes de engenharia (Koltun; Kolter; Vogel-Heuser, 2018; Arroyo et al., 2016).

2.1.3 Controladores lógicos programáveis em sistemas industriais

Os CLPs constituem uma das principais plataformas de controle empregadas em sistemas de automação industrial. Esses dispositivos são responsáveis pela leitura de sinais provenientes de sensores e instrumentos de campo, pelo processamento de lógicas de comando, intertravamento e sequenciamento, e pelo acionamento de equipamentos como motores, válvulas, atuadores e dispositivos de sinalização, conforme a lógica previamente programada pelo engenheiro. Desse modo, o CLP atua como elemento central entre a instrumentação instalada na planta e as estratégias de controle definidas no projeto de automação (Sehr et al., 2021).

A utilização de CLPs em ambientes industriais está associada à necessidade de confiabilidade operacional, modularidade e integração com diferentes dispositivos de campo. Esses controladores permitem organizar lógicas de controle, sequenciamentos, intertravamentos, tratamentos de falhas e rotinas de comando em estruturas programáveis e reutilizáveis, favorecendo a manutenção, a padronização e a rastreabilidade das soluções desenvolvidas. Do ponto de vista da engenharia de projeto, essa característica também viabiliza o reaproveitamento de estruturas previamente validadas, como blocos funcionais, tipos de dados, rotinas de diagnóstico, padrões de nomenclatura e bibliotecas corporativas. Essa possibilidade é particularmente importante em empresas integradoras, como a GreyLogix, nas quais equipamentos semelhantes tendem a ser implementados com padrões recorrentes de controle, supervisão, alarme e intertravamento (Vogel-Heuser et al., 2017, 2018).

No contexto deste PFC, os CLPs assumem papel central porque os ativos identificados nos diagramas P&ID precisam ser convertidos em estruturas compatíveis com o ambiente de automação. Sensores, motores, válvulas e atuadores representados nos diagramas devem ser associados a estruturas de engenharia equivalentes, juntamente com a lógica de controle desenvolvida no ambiente de programação do CLP. Dessa forma, a identificação automática de símbolos em P&ID somente se torna aplicável ao fluxo de automação quando articulada à forma como o CLP organiza, executa

e documenta a lógica de controle no ambiente de desenvolvimento (Koltun; Kolter; Vogel-Heuser, 2018; Papakonstantinou et al., 2019).

2.1.4 Bibliotecas de engenharia e padronização de projetos

A padronização por meio de bibliotecas de engenharia é uma prática importante em projetos de automação, pois favorece a reutilização de soluções previamente validadas e reduz variações desnecessárias entre projetos. Bibliotecas de blocos funcionais, *data types*, lógicas sequenciais e padrões de nomenclatura contribuem para maior uniformidade na implementação, facilitando a manutenção, a legibilidade, o comissionamento, a expansão e a revisão de sistemas industriais (Maga; Jazdi; Göhner, 2011; Vogel-Heuser et al., 2017, 2018).

Para a proposta desenvolvida nesse PFC, essa padronização é relevante porque os elementos extraídos dos diagramas P&ID precisam ser relacionados a estruturas previamente definidas no ambiente de engenharia da empresa. Dessa forma, o mapeamento entre ativos identificados, padrões de nomenclatura e bibliotecas internas torna-se uma etapa essencial para que os dados extraídos da documentação técnica possam ser utilizados de modo consistente na geração assistida do projeto (Maga; Jazdi; Göhner, 2011; Vogel-Heuser et al., 2018).

2.2 DIAGRAMAS P&ID E SIMBOLOGIA DE INSTRUMENTAÇÃO

Esta seção aborda os diagramas P&ID como documentos fundamentais para a representação de processos industriais e para a identificação de instrumentos, equipamentos, válvulas e malhas de controle. Esses conceitos são necessários para compreender a origem das informações utilizadas pelo sistema proposto nesse projeto.

2.2.1 Conceito e finalidade dos diagramas P&ID

Os P&IDs representam, de forma sistemática, os principais elementos de uma instalação de processo, incluindo equipamentos, tubulações, válvulas, instrumentos, malhas de controle e interligações relevantes à operação da planta. O P&ID reúne informações funcionais e de engenharia necessárias para o detalhamento, a automação, a operação e a manutenção de sistemas industriais. Por esse motivo, esses diagramas constituem um dos principais documentos para a transição entre a concepção do processo e a implementação de soluções de controle e automação (Koltun; Kolter; Vogel-Heuser, 2018).

O P&ID também desempenha papel importante na organização das informações técnicas do processo, permitindo compreender a relação entre equipamentos, instrumentos, atuadores, variáveis monitoradas e malhas de controle. Dessa forma, sua interpretação adequada é essencial para a elaboração de listas de instrumentos,

definição de sinais de entrada e saída, estruturação de *tags* e configuração de objetos de automação em ambientes de engenharia (Arroyo et al., 2016; Koltun; Kolter; Vogel-Heuser, 2018; Paliwal et al., 2021).

Assim, além de sua função documental, os P&IDs atuam como uma base técnica para diferentes etapas do projeto de automação. Para compreender essa relação, as subseções seguintes apresentam os principais elementos presentes nesses diagramas, os critérios de identificação de instrumentos e malhas de controle, a norma ISA-5.1 e os desafios associados à leitura automática dessas representações.

2.2.2 Elementos e simbologia em diagramas P&ID

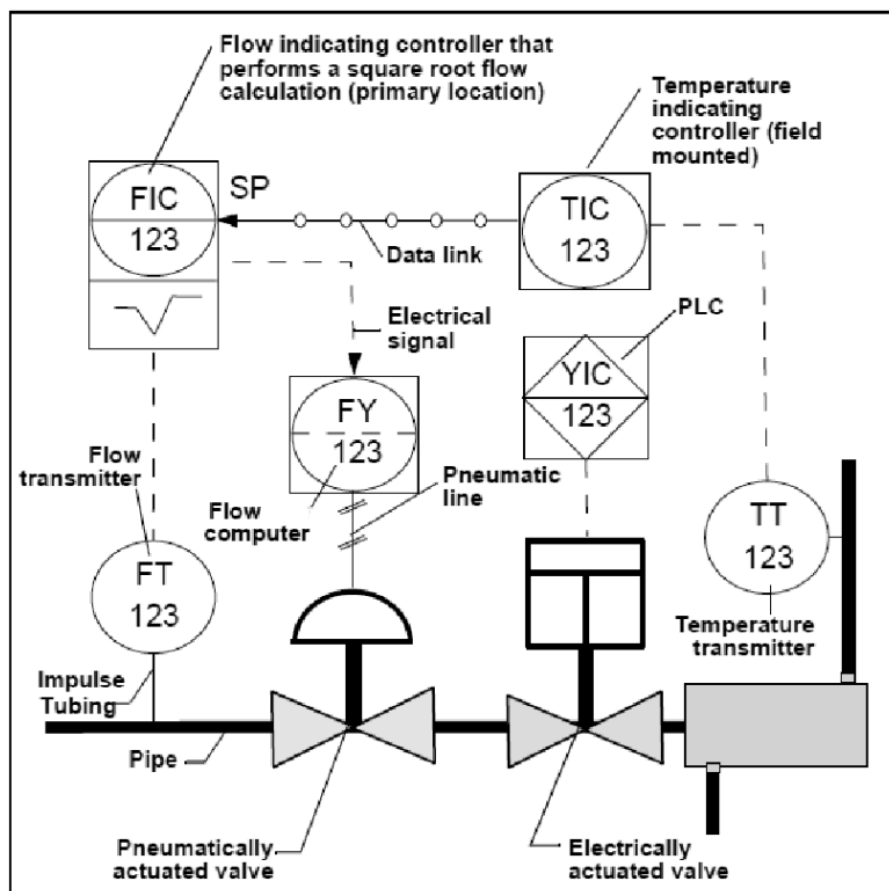
Os diagramas P&ID combinam símbolos gráficos e identificações textuais para representar a configuração de uma instalação de processo. Neles, equipamentos como tanques, bombas, vasos e trocadores de calor indicam os principais ativos físicos da planta, enquanto tubulações e conexões representam os caminhos de escoamento e a relação funcional entre diferentes partes do sistema. Válvulas, instrumentos, linhas de sinal e malhas de controle complementam essa representação ao indicar pontos de medição, atuação, comando e supervisão, aproximando o diagrama das necessidades do projeto de automação (International Organization for Standardization, 2014, 2012; International Society of Automation, 2024).

Além da simbologia gráfica, os P&IDs incorporam identificadores textuais que atribuem significado técnico aos elementos representados. Esses identificadores incluem *tags* de instrumentos, códigos de linha, identificação de equipamentos, referências a malhas de controle e anotações complementares de processo. Assim, a interpretação do diagrama depende da leitura conjunta entre o símbolo representado, sua posição no desenho, suas conexões gráficas e a identificação textual associada. Essa característica faz do P&ID um documento de elevada densidade informacional, pois combina informações visuais e códigos técnicos em uma mesma estrutura documental (International Society of Automation, 2024; Rahul et al., 2019; Kim et al., 2022; Stürmer; Graumann; Koch, 2025; Mani et al., 2020).

A Figura 1 exemplifica essa integração entre elementos gráficos e identificadores textuais em um diagrama P&ID, evidenciando a presença de instrumentos, linhas de sinal, atuadores, tubulações e referências funcionais associadas ao controle do processo.

Embora todos esses componentes sejam relevantes para a compreensão global do processo, este PFC concentra-se nos símbolos associados à instrumentação, especialmente sensores e instrumentos de medição, por constituírem os objetos de interesse da etapa de identificação automática. Nesse recorte, equipamentos, tubulações, válvulas e demais componentes são tratados principalmente como elementos de contextualização, enquanto os instrumentos representados no diagrama constituem

Figura 1 – Exemplo de diagrama P&ID com instrumentos, linhas de sinal e elementos de processo.



Fonte: PAControl.com (2006).

a principal fonte de informação para a extração de tags e posterior estruturação dos dados utilizados no fluxo de automação.

2.2.3 Norma ISA-5.1 para símbolos e nomenclatura de tags

A norma ISA-5.1, estabelecida pela Sociedade Internacional de Automação, do inglês *International Society of Automation* (ISA), define critérios para a identificação funcional de instrumentos e para a representação gráfica de sistemas de medição e controle em diagramas de engenharia. No contexto de projetos de automação, essa padronização fornece uma base importante para a interpretação dos elementos presentes nos P&IDs, pois relaciona símbolos gráficos, localização dos instrumentos, funções desempenhadas e identificações alfanuméricas utilizadas nas *tags* (International Society of Automation, 2024).

Do ponto de vista gráfico, a ISA-5.1 estabelece diferentes símbolos e convenções para representar instrumentos, funções de controle, localização dos dispositivos e formas de acesso à informação no processo. Como este PFC se concentra na identificação automática de instrumentos em diagramas P&ID, são destacados apenas

os símbolos mais diretamente relacionados a esse panorama. A Tabela 2 apresenta exemplos básicos de representações gráficas de instrumentos, incluindo variações associadas à localização e à forma de apresentação da informação no sistema de controle (International Society of Automation, 2024).

Tabela 2 – Síntese parcial de símbolos para dispositivos de instrumentação em P&IDs, segundo a norma ANSI/ISA-5.1.

No.	Display/controle compartilhado	Instrumento discreto	Localização e acessibilidade
1			Instrumento localizado em campo, normalmente visível e acessível ao operador.
2			Instrumento localizado na frente de painel principal, console ou sistema de operação, normalmente visível e acessível ao operador.
3			Instrumento localizado na parte traseira de painel principal ou em gabinete, normalmente não visível e não acessível ao operador.

Fonte: Adaptado de ANSI/ISA-5.1 (International Society of Automation, 2024).

Além da representação gráfica dos instrumentos, a ISA-5.1 também define critérios para sua identificação funcional por meio de códigos alfanuméricos, usualmente chamados de *tags*. Essas identificações permitem reconhecer, de forma sistemática, a variável associada ao instrumento, sua função na malha de controle e o número correspondente à malha ou ao ponto de medição. Em geral, a primeira letra da *tag* indica a variável de processo ou variável iniciadora, como “P” para pressão, “T” para temperatura e “F” para vazão. As letras seguintes especificam funções desempenhadas pelo instrumento, como indicação, transmissão, registro ou controle, e a numeração subsequente indica a malha de controle associada ao instrumento. Assim, uma *tag* como “PIT-100” pode indicar um instrumento associado à pressão, com funções de indicação e transmissão, pertencente à malha 100 (International Society of Automation, 2024; Paliwal et al., 2021). A Tabela 3 apresenta exemplos dessa lógica de identificação, conforme padrão empregado pela GreyLogix com base na norma ISA-5.1.

No escopo da identificação automática, a padronização proposta pela ISA-5.1 é relevante porque associa símbolos gráficos e estruturas de *tags* a significados funcionais previamente definidos. Essa organização favorece o tratamento computacional dos P&IDs, ao permitir que elementos visuais e textuais sejam relacionados a instrumentos, variáveis de processo e malhas de controle. Com isso, técnicas de visão computacional e reconhecimento óptico de caracteres podem ser empregadas para

Tabela 3 – Identificação funcional de instrumentos conforme a ISA-5.1

Letra	Primeira letra	Letras subsequentes		
		Função de informação ou passiva	Função final	Modificadora
A	Análise	Alarme		
B	Chama do queimador			
C			Controle	
D				Diferencial
E	Tensão elétrica	Elemento primário		
F	Vazão			Razão
G		Visor		
H	Comando manual			Alto (1)
I	Corrente elétrica	Indicador		
J	Potência			Varredura
K	Tempo		Estação de controle	Taxa de variação com o tempo
L	Nível	Lâmpada piloto		Baixo (2)
M	Umidade			Médio ou intermediário
N	A critério do cliente			
O		Orifício de restrição		
P	Pressão			
Q	Quantidade			Integração ou totalização
R	Radioatividade	Registrador ou impressora		
S	Velocidade	Segurança	Chave	
T	Temperatura		Transmissor	
U	Multivariável	Multifunção		
V	Vibração		Válvula	
W	Peso	Poço		
X	Válvula on-off			
Y	A critério do cliente		Relé de computação ou conversor, solenoide	
Z	Posição		Atuador para elemento final de controle, acionador	

Fonte: Adaptado de material interno da GreyLogix Brasil, elaborado com base na norma ISA-5.1.

identificar símbolos de instrumentação, extrair textos associados e estruturar os dados obtidos para etapas posteriores do fluxo de automação (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022; Stürmer; Graumann; Koch, 2025).

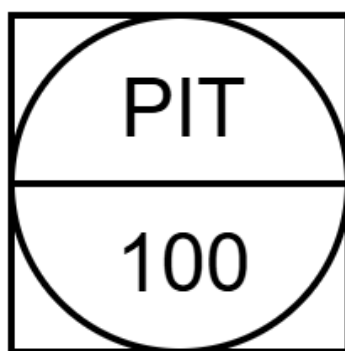
2.2.4 Relação entre P&IDs e projetos de automação

Em projetos de automação industrial, os P&IDs funcionam como uma das principais bases documentais para a passagem entre a representação do processo e sua implementação em sistemas de controle. As informações registradas nesses diagramas auxiliam na elaboração de documentos e estruturas de engenharia, como listas de instrumentos, listas de sinais, listas de I/O, descritivos lógicos, matrizes de intertravamento, listas de alarmes e especificações associadas ao controle e à supervisão do

processo (Koltun; Kolter; Vogel-Heuser, 2018; Winkler; Biffel, 2012).

Na prática, a identificação do símbolo de um instrumento no P&ID fornece informações que orientam sua implementação no sistema de automação. No exemplo da Figura 2, a identificação *PIT-100* representa um transmissor indicador de pressão associado à malha 100. A partir dessa informação, o engenheiro pode inferir a necessidade de um sinal de entrada no CLP, definir uma *tag* correspondente no ambiente de programação, associar essa variável a uma estrutura de dados e utilizá-la em telas de supervisão, alarmes, diagnósticos ou lógicas de controle. Dessa forma, o P&ID contribui para a organização dos objetos de engenharia utilizados no desenvolvimento do sistema automatizado, ao reunir informações funcionais do processo em uma representação estruturada (International Society of Automation, 2024; Paliwal et al., 2021).

Figura 2 – Exemplo de um símbolo de instrumentação encontrado em um P&ID



Fonte: Elaborado pelo autor.

Essa etapa de conversão evidencia a importância da padronização interna em projetos de automação. Embora o P&ID forneça a identificação inicial dos instrumentos, a forma como essas informações são transformadas em objetos de engenharia depende de regras de nomenclatura, tipos de dados, bibliotecas de blocos, padrões de programação e critérios adotados pela empresa integradora. Assim, um mesmo tipo de instrumento pode ser representado de formas distintas no ambiente de desenvolvimento, conforme a arquitetura do projeto, a plataforma utilizada e os padrões previamente definidos para *tags*, blocos funcionais e instâncias de dados (Maga; Jazdi; Göhner, 2011; Vogel-Heuser et al., 2018).

No contexto deste PFC, essa relação é particularmente relevante porque a leitura automática do P&ID compõe a etapa inicial para a geração assistida de estruturas no Siemens TIA Portal. Para isso, as informações extraídas do diagrama devem ser organizadas em formato estruturado, de modo a reduzir etapas manuais de interpretação e transcrição e apoiar a criação de objetos compatíveis com o ambiente de automação.

2.2.5 Desafios da leitura automática de P&IDs

A leitura automática de P&IDs apresenta desafios específicos porque esses documentos combinam, em uma mesma representação, símbolos gráficos, textos, linhas de processo, linhas de sinal, equipamentos, conexões e anotações técnicas. Em diagramas reais, esses elementos podem apresentar diferenças de escala, variações gráficas, sobreposição parcial, baixa resolução e elevada densidade visual. Além disso, símbolos estruturalmente semelhantes podem representar funções distintas, tornando a classificação automática dependente de bases de dados bem rotuladas e de modelos capazes de reconhecer diferenças visuais sutis (Rahul et al., 2019; Jamieson; Moreno-García; Elyan, 2024; Kim et al., 2022).

Outro fator que interfere na leitura automática é grau de aderência dos documentos às normas de simbologia e identificação. Embora normas como a ANSI/ISA-5.1 forneçam convenções para representação de instrumentos e *tags*, na prática, documentos industriais podem apresentar adaptações internas, símbolos simplificados, convenções próprias de empresas, inconsistências de nomenclatura ou representações herdadas de projetos anteriores. Essas variações reduzem a uniformidade visual e textual dos diagramas e podem dificultar a aplicação direta de modelos treinados com base em simbologias padronizadas. Assim, a adequação do documento às convenções normativas e aos padrões internos da empresa torna-se um fator relevante para o desempenho de sistemas de leitura automática de P&IDs (International Society of Automation, 2024; Jamieson; Moreno-García; Elyan, 2024).

Outro problema está na necessidade de associar corretamente símbolos, *tags* e relações espaciais. A simples detecção de um elemento gráfico não é suficiente para gerar informação útil ao projeto de automação, pois é necessário identificar qual texto corresponde ao símbolo detectado, qual função técnica é indicada pela *tag* e como esse instrumento se relaciona com os demais elementos do diagrama. Dessa forma, a interpretação automática de um P&ID envolve o reconhecimento visual dos elementos e a organização semântica das informações obtidas a partir de suas relações no documento (Paliwal et al., 2021; Kim et al., 2022; Stürmer; Graumann; Koch, 2025).

Por esse motivo, trabalhos voltados à digitalização de P&IDs frequentemente tratam o problema como um processo composto por diferentes etapas, como detecção de símbolos, OCR, extração de linhas e associação entre componentes. Essa abordagem integrada é necessária porque a informação relevante ao projeto não está contida isoladamente no símbolo, no texto ou na conexão, mas na relação estabelecida entre esses elementos (Rahul et al., 2019; Paliwal et al., 2021; Stürmer; Graumann; Koch, 2025). No contexto deste PFC, essa característica justifica a adoção de um fluxo de processamento voltado à identificação de símbolos de instrumentação, detecção do texto relacionado ao símbolo e à posterior estruturação das informações extraídas para uso em etapas de automação assistida.

2.3 SIEMENS TIA PORTAL E TIA PORTAL OPENNESS

Esta seção apresenta o Siemens TIA Portal como ambiente integrado de engenharia e discute os principais elementos utilizados na estruturação de projetos de automação. Também são abordados o TIA Portal Openness e o formato SIMATIC ML, por serem recursos diretamente relacionados à geração assistida de objetos de engenharia proposta nesse projeto.

2.3.1 Siemens TIA Portal como ambiente integrado de engenharia

O Siemens TIA Portal, sigla para Portal de Automação Totalmente Integrada (do inglês, *Totally Integrated Automation Portal*), é um ambiente integrado de engenharia voltado ao desenvolvimento, configuração e manutenção de sistemas de automação industrial. A plataforma reúne recursos para configuração de dispositivos, programação de CLPs, definição de redes industriais, parametrização de componentes e integração com outros elementos do sistema de automação. Dessa forma, o TIA Portal concentra diferentes etapas do projeto em uma estrutura comum de engenharia, reduzindo a fragmentação entre ferramentas e favorecendo a consistência das informações ao longo do desenvolvimento (Siemens, 2026b; Maga; Jazdi; Göhner, 2011; Vogel-Heuser et al., 2014).

No ecossistema de automação da Siemens, essa integração permite organizar, em uma mesma estrutura de projeto, controladores da linha SIMATIC S7, módulos de I/O, dispositivos de campo, redes de comunicação, variáveis, blocos de programação e recursos de diagnóstico. Essa característica é relevante em aplicações industriais porque aproxima a configuração de hardware, a definição de sinais e a programação do controlador, permitindo que diferentes elementos do sistema sejam tratados de forma articulada no ambiente de desenvolvimento (Siemens, 2026b; Siemens AG, 2026).

Em projetos desenvolvidos no TIA Portal, essas informações são organizadas por meio de objetos de engenharia, como tabelas de *tags*, tipos de dados, blocos de função e bibliotecas. Esses objetos permitem estruturar o projeto de forma modular e reutilizável, favorecendo a padronização de soluções recorrentes e a manutenção da coerência entre diferentes partes do sistema. Assim, o TIA Portal pode ser compreendido como um ambiente no qual os elementos de hardware, software e documentação técnica são relacionados ao longo do desenvolvimento do projeto (Vogel-Heuser et al., 2017, 2018; Siemens, 2026b).

No contexto deste PFC, o TIA Portal representa o ambiente de destino das informações extraídas dos diagramas P&ID. Os símbolos e *tags* identificados na etapa de leitura automática devem ser convertidos em dados estruturados capazes de apoiar a criação inicial de objetos de engenharia no projeto de automação. Com isso, a integração com o TIA Portal constitui a etapa em que as informações extraídas do

diagrama passam a ser relacionadas a objetos concretos do projeto de automação.

2.3.2 Estrutura de programação no TIA Portal

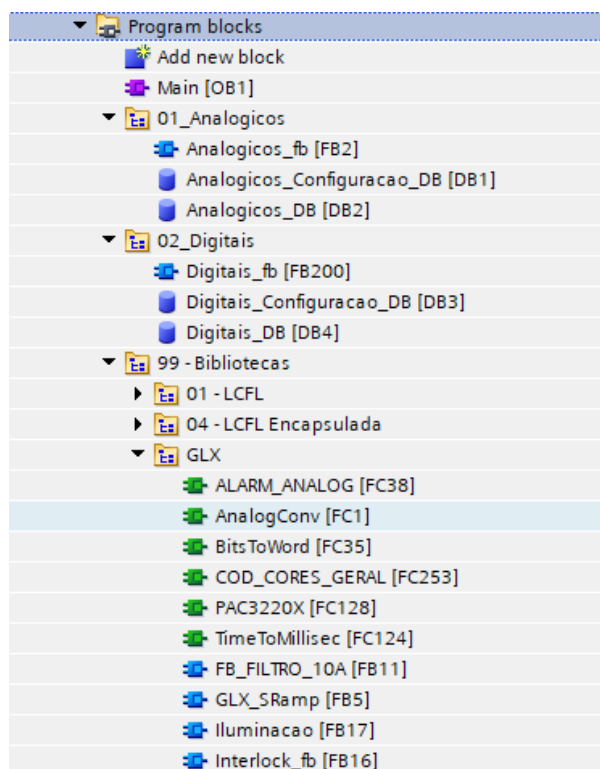
A estrutura de programação no TIA Portal é organizada de forma modular, articulando a configuração dos dispositivos de automação aos objetos de software executados pelo controlador. Em um projeto típico, a CPU do CLP, os módulos de I/O, as redes de comunicação, as tabelas de *tags*, os tipos de dados e os blocos de programa são configurados dentro de uma mesma estrutura de engenharia (Siemens AG, 2026; Siemens, 2026b).

Na programação de CLPs, a lógica de automação é organizada por meio de diferentes tipos de blocos, cada um com função específica na estrutura do programa. O Bloco organizacional, do inglês *Organization Block* (OB) define o ponto de entrada para a execução da lógica, podendo estar associados ao ciclo principal do controlador, à inicialização da CPU ou ao tratamento de eventos específicos. A Função, do inglês *Function* (FC) permite encapsular trechos de lógica reutilizáveis, sem armazenamento próprio de dados, sendo adequadas para operações, cálculos, comandos ou verificações que não precisam manter estado entre execuções. Já o Bloco de função, do inglês *Function Block* (FB) é utilizado quando a lógica necessita preservar dados associados à sua execução, como parâmetros, estados internos, temporizações, diagnósticos ou informações de instâncias de equipamentos e instrumentos (Siemens AG, 2026).

Os dados utilizados pelo programa são organizados em Bloco de dados, do inglês *Data Block* (DB), que podem armazenar informações globais do projeto ou dados associados à execução de blocos funcionais. No caso dos FBs, os blocos de dados de instância registram os valores internos de cada chamada do bloco, permitindo que uma mesma lógica seja reutilizada para diferentes instrumentos, motores, válvulas ou equipamentos, sem perda da individualização de seus parâmetros e estados. Essa estrutura favorece a modularidade, a reutilização de soluções e a padronização da lógica de controle em projetos industriais (Siemens AG, 2026).

A Figura 3 apresenta um exemplo de organização de objetos de engenharia no TIA Portal, incluindo tabelas de *tags*, blocos de programa e Tipo de dado definido pelo usuário, do inglês *User-Defined Data Type* (UDT) utilizados pelo controlador.

Figura 3 – Exemplo de organização de objetos de engenharia em um projeto no TIA Portal.



Fonte: Elaborado pelo autor.

Além dos blocos de programa, o TIA Portal utiliza tabelas de *tags*, UDT e bibliotecas para estruturar e padronizar o projeto. As *tags* do controlador representam variáveis associadas a sinais físicos, dados internos ou parâmetros de controle, podendo estar relacionadas a entradas, saídas, memórias e estruturas de dados. Os UDTs permitem agrupar variáveis relacionadas em estruturas padronizadas, facilitando a representação de instrumentos, equipamentos ou funções recorrentes no projeto. Assim, um instrumento pode ser representado por um conjunto organizado de informações, como valor de processo, escala, unidade, limites de alarme, estado de comunicação e parâmetros de diagnóstico (Siemens AG, 2026).

As bibliotecas complementam essa organização ao reunir elementos reutilizáveis do projeto, como blocos, tipos de dados, estruturas de variáveis e padrões previamente definidos. Em conjunto, esses recursos favorecem a reutilização de soluções validadas, reduzem variações desnecessárias entre aplicações semelhantes e contribuem para manter coerência entre diferentes partes do sistema de automação (Siemens AG, 2026; Vogel-Heuser et al., 2017, 2018).

Na solução proposta por este PFC, os dados obtidos a partir dos diagramas P&ID precisam ser organizados conforme a estrutura utilizada no TIA Portal. A identificação de um instrumento envolve reconhecer o símbolo, extrair sua *tag* e relacionar essas informações a elementos do projeto, como *tags* do controlador, UDT, blocos

funcionais e DB. Essa relação é necessária para que o resultado da identificação automática possa ser utilizado como base para a geração assistida de estruturas iniciais de automação, respeitando os padrões de programação e nomenclatura adotados no projeto.

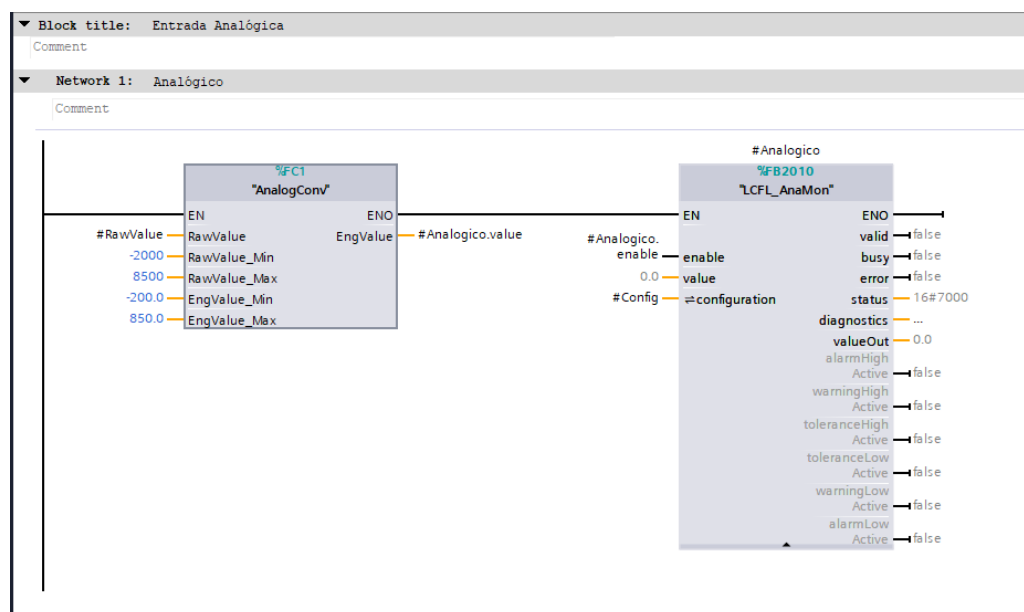
2.3.3 Linguagens LAD e SCL em projetos de automação

O TIA Portal oferece suporte a diferentes linguagens de programação para CLPs, cuja escolha depende da natureza da lógica implementada, dos padrões adotados pela equipe de engenharia e das características do controlador utilizado. Nesse PFC, destacam-se o Diagrama Ladder, do inglês *Ladder Diagram* (LAD) e a Linguagem Estruturada de Controle, do inglês *Structured Control Language* (SCL), por serem linguagens utilizadas nos projetos de automação da GreyLogix e por representarem duas formas distintas de estruturar a lógica executada pelo controlador: uma gráfica, voltada à visualização das relações de comando, e outra textual, adequada à implementação de rotinas estruturadas (PLCopen, 2025; Siemens AG, 2026).

O LAD é uma linguagem gráfica baseada na representação de circuitos elétricos de comando. Sua estrutura organiza a lógica em redes compostas por contatos, bobinas, blocos e operações lógicas, favorecendo a visualização das relações entre condições de entrada, intertravamentos, comandos e saídas acionadas. Por essa característica, o LAD é frequentemente empregado em lógicas discretas, comandos de motores e válvulas, sequências simples de operação e rotinas de intertravamento, especialmente quando a clareza visual da lógica é importante para atividades de desenvolvimento, diagnóstico, comissionamento e manutenção (Siemens, 2024; PLCopen, 2025).

A Figura 4 apresenta um exemplo simplificado de lógica representada em LAD no TIA Portal.

Figura 4 – Representação gráfica de uma lógica de controle em LAD no ambiente TIA Portal.



Fonte: Elaborado pelo autor a partir do Siemens TIA Portal.

A SCL, por sua vez, é uma linguagem textual de alto nível utilizada na programação de controladores SIMATIC. Por apresentar estrutura próxima à de linguagens algorítmicas, a SCL é adequada para trechos de código que envolvem cálculos, manipulação de variáveis, estruturas condicionais, laços de repetição e operações mais extensas sobre dados. Assim, enquanto o LAD favorece a leitura gráfica de lógicas de comando e intertravamento, a SCL oferece maior flexibilidade para a implementação de rotinas estruturadas e operações que seriam menos práticas em uma representação exclusivamente gráfica (Siemens, 2025; PLCopen, 2025).

A Figura 5 apresenta um exemplo simplificado de lógica escrita em SCL no TIA Portal.

Figura 5 – Representação textual de uma lógica de controle em SCL no ambiente TIA Portal.

```
Block title: Analog Conversion
Block to convert analogs from raw value to engineering units

Network 1: Conversion

1 #EngValue := SCALE_X(MIN := #EngValue_Min, VALUE := NORM_X(MIN := #RawValue_Min, VALUE := #RawValue, MAX := #RawValue_Max), MAX := #EngValue_Max);
```

Fonte: Elaborado pelo autor a partir do Siemens TIA Portal.

Em projetos de automação industrial, essas linguagens podem ser utilizadas de forma complementar dentro de uma mesma estrutura de controle. Determinadas rotinas podem ser desenvolvidas em LAD para facilitar a interpretação por equipes

de operação, manutenção e comissionamento, enquanto outras podem ser implementadas em SCL quando envolvem maior complexidade algorítmica ou manipulação de estruturas de dados. Essa combinação permite organizar o projeto conforme a função de cada bloco, mantendo equilíbrio entre legibilidade, padronização e eficiência na implementação da lógica de controle (Siemens AG, 2026; PLCopen, 2025; Sehr et al., 2021).

No contexto deste PFC, o LAD e a SCL são relevantes porque a geração assistida de estruturas no TIA Portal depende da forma como os objetos de automação são representados no ambiente de programação. Enquanto o LAD contribui para a compreensão visual de lógicas de comando e intertravamento, a SCL permite representar rotinas textuais e operações estruturadas. Dessa forma, essas linguagens ajudam a compreender como as informações extraídas dos diagramas P&ID podem ser posteriormente associadas a blocos, variáveis e rotinas no projeto de automação.

2.3.4 Siemens TIA Portal Openness

O TIA Portal Openness é uma interface de programação disponibilizada pela Siemens para a automação de tarefas de engenharia no ambiente TIA Portal. Essa interface pode ser acessada por aplicações externas desenvolvidas em ambiente *.NET*, como programas em *C#*, possibilitando a leitura, criação e manipulação de elementos de um projeto de automação, como dispositivos, tabelas de *tags*, blocos de programa, UDTs e demais objetos associados à configuração do sistema. Dessa forma, o Openness amplia o uso do TIA Portal para além da interação manual com a interface gráfica, permitindo que etapas repetitivas ou padronizadas do desenvolvimento sejam executadas de forma programática (Siemens, 2026a).

A utilização do Openness é relevante em projetos que demandam padronização, reutilização e geração assistida de estruturas de engenharia. Em vez de criar manualmente cada elemento no ambiente de desenvolvimento, uma aplicação externa pode empregar informações previamente organizadas para inserir, modificar ou atualizar objetos no projeto do TIA Portal. Essa abordagem pode contribuir para reduzir retrabalho, inconsistências de nomenclatura e erros decorrentes da transcrição manual de dados entre documentos de engenharia e o software de automação (Siemens, 2026a; Papakonstantinou et al., 2019; Koziolk et al., 2020a; Dürkop et al., 2015).

No escopo deste PFC, o TIA Portal Openness representa a interface entre as informações extraídas dos diagramas P&ID e a estrutura inicial do projeto de automação. Após a identificação de símbolos, *tags* e atributos relevantes, esses dados podem ser organizados em uma representação intermediária e utilizados por uma aplicação externa para apoiar a criação de objetos no TIA Portal. Com isso, os resultados obtidos nas etapas de visão computacional e reconhecimento de texto passam a ser incorporados a um fluxo de engenharia voltado à geração assistida de estruturas de automação

(Siemens, 2026a; Papakonstantinou et al., 2019; Koltun; Kolter; Vogel-Heuser, 2018).

Apesar desse potencial, o uso do TIA Portal Openness não elimina a necessidade de validação técnica por parte do engenheiro. As estruturas geradas dependem da qualidade das informações extraídas, da consistência das regras de nomenclatura e dos padrões definidos para cada projeto, e a lógica de controle ainda precisa ser implementada manualmente. Por isso, sua aplicação deve ser compreendida como um recurso de apoio à engenharia, capaz de acelerar e padronizar etapas do desenvolvimento, mas ainda sujeito à revisão, complementação e adequação conforme os requisitos específicos de cada aplicação industrial (Dürkop et al., 2015; Koziol et al., 2020a; Maga; Jazdi; Göhner, 2011).

2.3.5 SIMATIC ML

O SIMATIC ML, sigla do inglês *SIMATIC Markup Language*, é um formato baseado em Linguagem de Marcação Extensível, do inglês *Extensible Markup Language* (XML) utilizado no contexto do TIA Portal para representar dados de engenharia de maneira estruturada. Por meio desse formato, determinados elementos do projeto podem ser exportados, descritos externamente e posteriormente importados ou manipulados com o auxílio do TIA Portal Openness, mantendo compatibilidade com a organização interna do ambiente de engenharia (Siemens, 2026a).

A utilização de arquivos em SIMATIC ML é relevante porque permite descrever objetos de engenharia em uma estrutura legível por ferramentas computacionais. Blocos de programa, UDTs, tabelas de *tags* e outros elementos do projeto podem ser representados de forma hierárquica, por meio de atributos, identificadores e propriedades associados a cada objeto. Essa organização favorece a troca de informações entre aplicações externas e o TIA Portal, especialmente em fluxos de trabalho que envolvem geração, importação ou atualização programática de partes do projeto (Siemens, 2026a).

Para os objetivos deste PFC, o SIMATIC ML oferece um meio de aproximar os dados extraídos dos diagramas P&ID dos objetos utilizados no ambiente de automação. Após a identificação de símbolos, *tags* e informações associadas aos instrumentos, esses dados podem ser organizados em uma representação intermediária e utilizados como base para a geração assistida de elementos compatíveis com o TIA Portal. Dessa forma, o SIMATIC ML contribui para transformar informações provenientes da documentação de engenharia em estruturas reconhecidas pelo ambiente de desenvolvimento.

2.4 VISÃO COMPUTACIONAL APLICADA À EXTRAÇÃO DE INFORMAÇÕES EM DOCUMENTOS TÉCNICOS

Essa seção apresenta os fundamentos de visão computacional utilizados na extração de informações em documentos técnicos, com ênfase no processamento de imagens, na detecção de objetos e no reconhecimento óptico de caracteres. Esses conceitos fundamentam a etapa computacional responsável por identificar símbolos, textos e relações espaciais em diagramas P&ID.

2.4.1 Fundamentos de visão computacional

A visão computacional é uma área da computação voltada à extração, análise e interpretação de informações visuais a partir de imagens ou vídeos. Seu objetivo é permitir que sistemas computacionais processem dados visuais e reconheçam padrões, objetos, estruturas e relações presentes em uma cena ou documento. Para isso, combina técnicas de processamento de imagens, reconhecimento de padrões, aprendizado de máquina e, em aplicações mais recentes, modelos de aprendizado profundo (Szeliski, 2022; Gonzalez; Woods, 2018).

Em aplicações industriais, a visão computacional pode ser empregada em tarefas como inspeção visual, controle de qualidade, reconhecimento de objetos, monitoramento de processos e interpretação de documentos técnicos. No caso de documentos de engenharia, sua utilização permite converter informações representadas graficamente em dados passíveis de tratamento computacional. Essa característica é particularmente relevante para documentos como os P&IDs, nos quais símbolos, textos e conexões representam instrumentos, equipamentos e relações funcionais do processo (Rahul et al., 2019; Paliwal et al., 2021; Jamieson; Moreno-García; Elyan, 2024).

Para os objetivos deste PFC, a visão computacional constitui a base para a identificação automática de símbolos de instrumentação em diagramas P&ID. A partir da imagem do documento, o sistema busca localizar e classificar elementos gráficos de interesse, permitindo que os resultados sejam posteriormente associados a *tags* e organizados em informações estruturadas.

2.4.2 Processamento e pré-processamento de imagens

O processamento de imagens compreende um conjunto de técnicas utilizadas para representar, transformar e analisar imagens para análise computacional. Em aplicações de visão computacional, essas técnicas permitem realçar e ajustar a representação visual dos dados, reduzir interferências e preparar as imagens para etapas posteriores de reconhecimento ou detecção. Operações como ajuste de escala, normalização, filtragem, segmentação, recorte e padronização de dimensões podem ser

empregadas conforme a natureza do problema e a qualidade das imagens disponíveis (Gonzalez; Woods, 2018; Szeliski, 2022).

O pré-processamento corresponde às transformações aplicadas antes da etapa principal de análise, com o objetivo de tornar as imagens mais adequadas ao modelo computacional utilizado. Em documentos técnicos, símbolos, textos e linhas podem aparecer em escalas reduzidas, com elevada densidade visual e diferentes padrões de representação. Assim, a preparação adequada das imagens contribui para preservar detalhes importantes e reduzir variações que possam prejudicar a identificação dos elementos de interesse (Gonzalez; Woods, 2018; Rahul et al., 2019; Paliwal et al., 2021; Jamieson; Moreno-García; Elyan, 2024).

No caso de diagramas P&ID, essa preparação é relevante devido à extensão dos documentos e à pequena dimensão relativa dos símbolos de interesse em relação ao tamanho total do desenho. Por esse motivo, estratégias como conversão do documento para imagem, ajuste de resolução, padronização do formato e divisão do diagrama em regiões menores podem auxiliar a etapa de detecção, especialmente quando o modelo precisa localizar símbolos pequenos em representações extensas e visualmente densas (Rahul et al., 2019; Paliwal et al., 2021).

Na aplicação proposta, o processamento e o pré-processamento de imagens são tratados como etapas preparatórias para a detecção automática de símbolos de instrumentação. O objetivo é organizar sua representação visual em um formato compatível com o modelo de detecção utilizado, preservando e destacando as informações necessárias para a localização e classificação dos elementos de interesse.

2.4.3 Aprendizado supervisionado, datasets e rotulagem de imagens

O aprendizado supervisionado é uma abordagem de aprendizado de máquina na qual o modelo é treinado a partir de exemplos previamente anotados. Em tarefas de visão computacional, esses exemplos normalmente consistem em imagens associadas a informações de referência, como classes, regiões de interesse ou coordenadas dos objetos presentes. A partir desses dados, o modelo ajusta seus parâmetros para reconhecer padrões visuais e aplicar esse aprendizado a novas imagens, não utilizadas durante o treinamento (Szeliski, 2022).

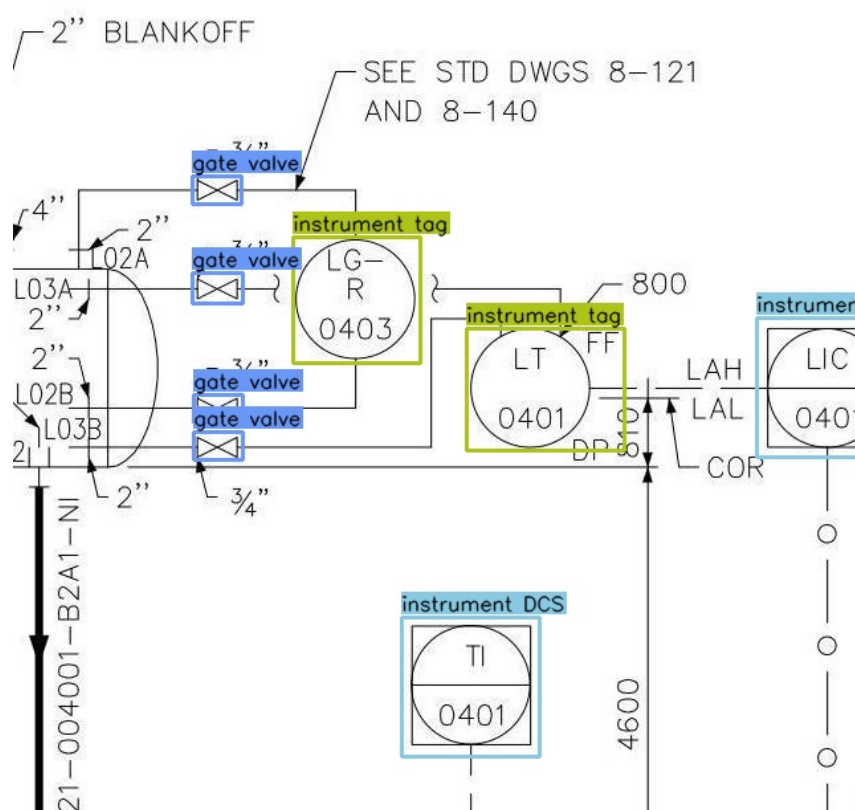
O conjunto de imagens anotadas constitui a base de dados, ou *dataset*, utilizada para o treinamento e a avaliação do modelo. Em tarefas de detecção de objetos, esse conjunto deve reunir exemplos representativos dos elementos que se deseja reconhecer, contemplando variações de escala, posição, qualidade visual e padrões gráficos. No caso de diagramas P&ID, as imagens podem conter símbolos de instrumentação, válvulas, sensores ou outros elementos de interesse, conforme o escopo definido para o sistema de detecção (Paliwal et al., 2021; Kim et al., 2022).

Na detecção de objetos, a anotação consiste no registro manual dos elementos

de interesse presentes em cada imagem. Cada ocorrência individual de um objeto é denominada instância anotada e recebe uma classe e uma caixa delimitadora. A classe indica o tipo de elemento representado, enquanto a caixa delimitadora corresponde à região retangular que define sua localização e sua extensão na imagem. Dessa forma, o conjunto de treinamento fornece ao modelo informações tanto para a classificação quanto para a localização dos objetos (Paliwal et al., 2021; Everingham et al., 2010).

A Figura 6 apresenta um exemplo de imagem anotada do *dataset* utilizado no treinamento, contendo caixas delimitadoras e classes associadas aos símbolos presentes no diagrama.

Figura 6 – Exemplo de imagem anotada com caixas delimitadoras e classes de símbolos em P&ID.



Fonte: Elaborado pelo autor com dados do *dataset* *PID Symbols*, disponível no Roboflow Universe (Roboflow Universe, 2023).

A qualidade do *dataset* influencia diretamente o desempenho da detecção, uma vez que erros de rotulagem, classes mal definidas, baixa diversidade de exemplos ou desequilíbrio entre categorias podem comprometer a capacidade de generalização do modelo. Por esse motivo, a construção do conjunto de dados deve considerar não apenas a quantidade de imagens, mas também a representatividade dos símbolos, a consistência das anotações e a presença de variações gráficas compatíveis com os documentos analisados (Everingham et al., 2010; Paliwal et al., 2021).

Além da rotulagem, é comum dividir o *dataset* em subconjuntos de treinamento, validação e teste. O conjunto de treinamento é utilizado para ajustar os parâmetros do modelo, enquanto o conjunto de validação auxilia no acompanhamento do desempenho durante o desenvolvimento e na escolha de configurações mais adequadas. O conjunto de teste, por sua vez, permite avaliar o modelo em imagens não utilizadas nas etapas anteriores, fornecendo uma estimativa mais independente de seu desempenho (Szeliski, 2022; Jamieson; Moreno-García; Elyan, 2024; Everingham et al., 2010).

Na abordagem adotada nesse PFC, o aprendizado supervisionado é empregado para que o modelo de detecção reconheça símbolos de interesse em diagramas P&ID a partir de exemplos previamente anotados. Assim, a rotulagem das imagens constitui uma etapa fundamental entre a preparação dos documentos técnicos e o treinamento do modelo, pois transforma os elementos visuais do diagrama em informações compreensíveis para o algoritmo de detecção.

2.4.4 Detecção de objetos em imagens

A detecção de objetos é uma área da visão computacional cujo objetivo é identificar a presença de determinados elementos em uma imagem, classificá-los e indicar sua localização espacial. Diferentemente da classificação de imagens, que atribui uma categoria à imagem como um todo, a detecção de objetos busca reconhecer múltiplos elementos em uma mesma imagem e delimitar a posição de cada um deles, geralmente por meio de caixas delimitadoras. Dessa forma, o modelo precisa responder simultaneamente quais objetos estão presentes e onde eles se encontram (Everingham et al., 2010; Zou et al., 2023; Szeliski, 2022).

Em documentos técnicos, a detecção de objetos pode ser aplicada à identificação de elementos gráficos específicos, como símbolos, componentes, conectores e regiões de interesse. No caso de diagramas P&ID, os objetos de interesse correspondem aos símbolos associados à instrumentação e aos demais elementos definidos no escopo do modelo. A localização desses símbolos permite transformar parte da informação visual do diagrama em dados estruturados, indicando a presença de determinado instrumento e sua posição no documento (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022; Jamieson; Moreno-García; Elyan, 2024).

A representação da localização por caixas delimitadoras é particularmente útil nesse tipo de aplicação, pois permite associar cada símbolo detectado a uma região específica da imagem. Essa informação espacial pode ser utilizada posteriormente para relacionar o símbolo a textos próximos, como *tags* e identificadores, além de apoiar a organização dos dados extraídos. Assim, a caixa delimitadora indica onde o símbolo foi encontrado e define uma região de referência para as etapas seguintes de leitura e associação das informações presentes no diagrama (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022).

Neste PFC, a detecção de objetos corresponde à etapa responsável por localizar e classificar automaticamente símbolos de instrumentação em diagramas P&ID. Essa etapa permite delimitar, na imagem, as regiões em que se encontram os elementos de interesse, criando uma base para a leitura de textos associados e para a posterior estruturação das informações extraídas.

2.4.5 Redes neurais convolucionais e modelos de aprendizado profundo

O aprendizado profundo corresponde a um conjunto de técnicas de aprendizado de máquina baseadas em redes neurais com múltiplas camadas, capazes de aprender representações complexas a partir dos dados de entrada. Em tarefas visuais, esses modelos passaram a ter grande relevância porque permitem extrair automaticamente características das imagens, reduzindo a dependência de descritores manuais e de regras fixas frequentemente utilizadas em abordagens clássicas de visão computacional (Goodfellow; Bengio; Courville, 2016; Szeliski, 2022; LeCun; Bengio; Hinton, G., 2015).

Em métodos tradicionais, a extração de características costuma depender de operações previamente definidas pelo engenheiro, como filtros, máscaras, descritores geométricos, limiares ou regras baseadas em contornos. Embora essas abordagens sejam úteis em diferentes aplicações, podem apresentar maior sensibilidade a variações de escala, orientação, resolução, ruído e forma dos objetos. Em contrapartida, modelos de aprendizado profundo aprendem representações visuais diretamente a partir dos dados de treinamento, o que os torna adequados a problemas nos quais os elementos de interesse apresentam variações gráficas ou aparecem em documentos com diferentes padrões visuais (Goodfellow; Bengio; Courville, 2016; Szeliski, 2022; LeCun; Bengio; Hinton, G., 2015).

Entre os modelos de aprendizado profundo aplicados à análise de imagens, destacam-se as redes neurais convolucionais, conhecidas como Rede Neural Convolucional, do inglês *Convolutional Neural Network* (CNN). Essas redes utilizam operações de convolução para analisar regiões locais da imagem e identificar padrões espaciais relevantes. Em camadas iniciais, a rede tende a aprender características visuais mais simples, como linhas, bordas e contrastes; em camadas mais profundas, essas informações são combinadas em representações mais abstratas, associadas a formas, partes de objetos e estruturas visuais de maior complexidade (LeCun et al., 1998; Krizhevsky; Sutskever; Hinton, G. E., 2017; Szeliski, 2022).

Essa capacidade de aprendizado hierárquico torna as CNNs adequadas para problemas de classificação, segmentação e detecção de objetos em imagens. Em vez de depender exclusivamente de regras fixas para reconhecer um símbolo, a rede aprende a partir de exemplos anotados e ajusta seus parâmetros para distinguir padrões associados às classes de interesse. Essa característica é importante em documentos técnicos, nos quais símbolos podem apresentar pequenas variações de escala,

traçado, orientação, resolução e qualidade gráfica (Krizhevsky; Sutskever; Hinton, G. E., 2017; Szeliski, 2022).

Na abordagem adotada neste PFC, os modelos de aprendizado profundo fornecem a base para a detecção automática de símbolos em diagramas P&ID. A partir de imagens anotadas, a CNN aprende padrões relacionados aos símbolos de instrumentação e utiliza esse aprendizado para localizar e classificar elementos semelhantes em novos diagramas. Essa lógica fundamenta o uso de modelos de detecção baseados em redes neurais, como os da família Você Só Olha Uma Vez, do inglês *You Only Look Once* (YOLO), apresentados na subseção seguinte (Rahul et al., 2019; Paliwal et al., 2021).

2.4.6 Modelos YOLO aplicados à detecção de símbolos

Os modelos da família YOLO correspondem a uma abordagem de detecção de objetos em etapa única. Diferentemente de métodos que separam a geração de regiões candidatas e a classificação dos objetos em etapas distintas, o YOLO realiza a predição das caixas delimitadoras, das classes e dos níveis de confiança em uma única passagem pela rede. Essa característica torna o modelo adequado para aplicações que demandam detecção simultânea de múltiplos objetos em uma mesma imagem (Redmon et al., 2016; Jiao et al., 2019).

Após o treinamento, a aplicação do modelo a uma nova imagem é denominada inferência. Durante esse processo, o YOLO analisa a imagem de entrada e estima, para diferentes regiões, a presença de objetos, suas respectivas classes e as coordenadas das caixas delimitadoras. O resultado consiste em um conjunto de detecções, cada uma associada a uma classe prevista, uma posição espacial e um valor de confiança. Esse valor expressa o grau de confiança atribuído pelo modelo à predição e pode ser utilizado para filtrar detecções de baixa confiabilidade antes das etapas posteriores do processamento (Redmon et al., 2016; Jiao et al., 2019; Zou et al., 2023).

Em diagramas P&ID, essa abordagem pode ser aplicada à identificação automática de símbolos de instrumentação, válvulas, sensores e demais elementos gráficos definidos no escopo do modelo. Como esses documentos podem conter múltiplos símbolos distribuídos em diferentes regiões da página, a capacidade de detectar vários objetos em uma única imagem torna os modelos YOLO compatíveis com a natureza visual dos diagramas técnicos. Além disso, a representação por caixas delimitadoras permite registrar a posição de cada símbolo detectado, informação necessária para etapas posteriores de associação com *tags* e organização dos dados extraídos (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022).

Na aplicação proposta por este PFC, o YOLO é utilizado para localizar e classificar símbolos de instrumentação em imagens de diagramas P&ID. Para isso, o modelo é treinado com um *dataset* de imagens previamente anotadas, nas quais cada sím-

bolo de interesse é identificado por uma classe e por uma caixa delimitadora. Após o treinamento, as detecções geradas fornecem as classes previstas, as posições dos símbolos e os respectivos valores de confiança. Essas informações constituem a base para as etapas seguintes do fluxo proposto, nas quais os símbolos detectados podem ser associados a identificadores textuais e convertidos em dados estruturados para apoio à geração inicial de objetos no ambiente de automação.

2.4.7 OCR em documentos técnicos e associação entre símbolos e tags

O OCR é uma técnica associada à visão computacional e ao processamento de documentos, utilizada para identificar e converter caracteres presentes em imagens em texto legível por computador. Em documentos técnicos, essa abordagem permite extrair informações textuais que complementam os elementos gráficos, como *tags*, identificadores de instrumentos, códigos de linha, nomes de equipamentos e demais anotações presentes no diagrama (Smith, 2007; Szeliski, 2022; Villena Toro; Wiberg; Tarkian, 2023).

Em diagramas P&ID, a identificação técnica de um instrumento geralmente depende do texto associado ao símbolo. A detecção visual de um símbolo de instrumentação, por exemplo, indica a presença de um elemento de interesse, mas a interpretação de sua função e medição depende da leitura da *tag* correspondente. Assim, a informação extraída do diagrama depende também do identificador textual acompanhado do símbolo detectado, tornando a aplicação de OCR relevante para a interpretação automática do diagrama (Rahul et al., 2019; Paliwal et al., 2021; Kim et al., 2022; International Society of Automation, 2024).

Na aplicação proposta, o OCR atua como uma etapa complementar à detecção de objetos. Enquanto o modelo de detecção fornece a classe e a localização espacial dos símbolos de instrumentação, o reconhecimento de texto permite extrair as *tags* e os identificadores associados a esses elementos. A combinação dessas informações possibilita estruturar dados mais completos sobre cada instrumento detectado, incluindo sua classe, sua posição no diagrama, sua função e sua identificação textual (Paliwal et al., 2021; Kim et al., 2022). Essas associações são necessárias para que os dados extraídos do P&ID possam ser relacionados a *tags*, tipos de dados e objetos de engenharia no TIA Portal.

2.4.8 Métricas de avaliação do módulo de detecção

A avaliação de um modelo de detecção de objetos é realizada pela comparação entre as predições produzidas pelo modelo e as anotações de referência do conjunto avaliado. Nesse contexto, cada ocorrência de um objeto marcada na imagem constitui uma instância anotada, caracterizada por uma classe e por uma caixa delimitadora que indica sua localização. Cada predição, por sua vez, contém uma classe estimada,

uma caixa delimitadora e um valor de confiança associado (Everingham et al., 2010; Szeliski, 2022; Padilla; Netto; Silva, 2020).

A detecção de objetos envolve simultaneamente a localização e a classificação dos elementos presentes na imagem. A localização consiste em determinar a região ocupada pelo objeto, enquanto a classificação corresponde à atribuição de uma categoria à região localizada. Portanto, uma predição somente pode ser considerada correta quando apresenta correspondência espacial suficiente com uma instância anotada e atribui a classe adequada ao objeto.

A correspondência entre uma predição e uma instância anotada é estabelecida com base em um limiar de sobreposição previamente definido. Quando a classe prevista está correta, a sobreposição atende ao limiar adotado e a instância ainda não foi associada a outra predição, o resultado é contabilizado como verdadeiro positivo (*VP*). Uma predição sem correspondência válida é contabilizada como falso positivo (*FP*), enquanto uma instância anotada que não tenha sido associada corretamente a nenhuma predição é contabilizada como falso negativo (*FN*). Quando um objeto é localizado, mas recebe uma classe incorreta, o resultado corresponde a um falso positivo para a classe prevista e a um falso negativo para a classe de referência (Everingham et al., 2010; Padilla; Netto; Silva, 2020).

A Tabela 4 sintetiza essas situações.

Tabela 4 – Resultados possíveis na comparação entre predições e anotações de referência.

Resultado	Interpretação
Verdadeiro positivo (<i>VP</i>)	A predição apresenta a classe correta, sobreposição espacial suficiente e correspondência com uma instância anotada ainda não associada a outra predição.
Falso positivo (<i>FP</i>)	A predição não possui correspondência válida com uma instância anotada, corresponde a uma detecção duplicada ou é atribuída a uma classe incorreta.
Falso negativo (<i>FN</i>)	Uma instância anotada não é associada corretamente a nenhuma predição.

Fonte: Elaborado pelo autor com base em (Everingham et al., 2010; Padilla; Netto; Silva, 2020).

Os verdadeiros negativos não são normalmente utilizados na avaliação de detectores de objetos, pois o fundo da imagem não corresponde a um conjunto finito de instâncias negativas anotadas. Diferentemente de uma tarefa convencional de classificação, uma imagem pode conter inúmeras regiões nas quais nenhum objeto está presente, tornando essa quantidade pouco informativa para a avaliação do detector (Padilla; Netto; Silva, 2020).

Entre as métricas derivadas dessas ocorrências estão a precisão, do inglês *precision*, e a revocação, do inglês *recall*. A precisão indica a proporção de detecções

corretas em relação ao total de detecções produzidas pelo modelo, sendo definida por (Padilla; Netto; Silva, 2020):

$$Precision = \frac{VP}{VP + FP}. \quad (1)$$

Na Equação 1, uma precisão elevada indica que uma parcela reduzida das predições foi contabilizada como falso positivo. Essa métrica é afetada, por exemplo, pela indicação de objetos inexistentes, pelos erros de classificação e pelas detecções duplicadas de uma mesma instância.

A revocação indica a proporção das instâncias anotadas que foram corretamente identificadas pelo modelo, sendo definida por (Padilla; Netto; Silva, 2020):

$$Recall = \frac{VP}{VP + FN}. \quad (2)$$

Na Equação 2, uma revocação elevada indica que poucas instâncias anotadas deixaram de ser identificadas. Os falsos positivos não integram o denominador dessa métrica, pois a revocação avalia a recuperação dos objetos de referência, enquanto as predições indevidas são consideradas no cálculo da precisão.

Precisão e revocação devem ser analisadas conjuntamente. A adoção de um limiar de confiança elevado tende a restringir o conjunto de predições aceitas, podendo reduzir a ocorrência de falsos positivos, mas também aumentar o número de objetos não detectados. Em contrapartida, a redução desse limiar tende a recuperar mais instâncias, embora possa introduzir detecções incorretas. Dessa forma, as duas métricas descrevem aspectos complementares do comportamento do detector (Padilla; Netto; Silva, 2020).

A qualidade espacial das caixas delimitadoras é avaliada por meio da Interseção sobre União, do inglês *Intersection over Union* (IoU), que mede a sobreposição entre uma caixa prevista e a caixa de referência correspondente. Seu valor é obtido pela razão entre a área de interseção e a área de união das duas caixas (Everingham et al., 2010; Padilla; Netto; Silva, 2020):

$$IoU = \frac{\text{Área de interseção}}{\text{Área de união}}. \quad (3)$$

Na Equação 3, a área de interseção corresponde à região comum entre a caixa delimitadora prevista e a caixa de referência, enquanto a área de união representa a região total ocupada por ambas, considerando a sobreposição apenas uma vez. O valor da IoU varia entre 0 e 1: valores próximos de 0 indicam baixa correspondência espacial, enquanto valores próximos de 1 indicam maior coincidência entre as caixas.

Para definir se uma predição pode ser associada a uma instância anotada, estabelece-se um limiar mínimo de IoU. Uma predição com classe correta e valor de IoU igual ou superior ao limiar pode ser contabilizada como verdadeiro positivo. Caso

a sobreposição seja inferior ao valor estabelecido, a localização não é considerada suficientemente adequada para formar uma correspondência válida (Padilla; Netto; Silva, 2020).

Outra medida empregada na avaliação de detectores é a Precisão Média, do inglês *Average Precision* (AP), calculada separadamente para cada classe. Para sua obtenção, as predições são ordenadas de acordo com seus valores de confiança e o limiar de aceitação é variado, produzindo diferentes combinações de precisão e revocação. A AP sintetiza a área sob a curva precisão–revocação resultante desse processo (Everingham et al., 2010; Redmon et al., 2016; Padilla; Netto; Silva, 2020). De forma idealizada, essa relação pode ser representada por:

$$AP_c = \int_0^1 p_c(r) dr. \quad (4)$$

Na Equação 4, AP_c representa a precisão média da classe c , e $p_c(r)$ representa a precisão dessa classe em função da revocação r . Na prática, essa área é estimada a partir dos pontos da curva precisão–revocação obtidos durante a avaliação. Portanto, $p_c(r)$ não corresponde ao valor isolado da precisão calculado para um único limiar de confiança, mas ao comportamento da precisão ao longo dos diferentes níveis de revocação.

Para um conjunto com múltiplas classes, a Precisão média das classes, do inglês *mean Average Precision* (mAP) é obtida pela média das APs calculadas para cada categoria:

$$mAP = \frac{1}{C} \sum_{c=1}^C AP_c. \quad (5)$$

Na Equação 5, C representa o número total de classes avaliadas. A mAP sintetiza o desempenho médio do detector entre as categorias consideradas, permitindo comparar modelos, configurações de treinamento ou versões distintas de uma mesma arquitetura (Everingham et al., 2010; Redmon et al., 2016; Bochkovskiy; Wang; Liao, 2020).

Os valores de AP e mAP também dependem do limiar de IoU adotado para estabelecer a correspondência entre predições e anotações. A mAP@0,50, por exemplo, utiliza um limiar de IoU igual a 0,50. Nesse caso, uma predição somente pode ser contabilizada como verdadeiro positivo quando, além de apresentar a classe correta, sua sobreposição com a caixa de referência for igual ou superior a esse valor.

A mAP@0,50:0,95, por sua vez, corresponde à média dos valores de mAP calculados para dez limiares de IoU, variando de 0,50 a 0,95 em incrementos de 0,05 (Padilla; Netto; Silva, 2020):

$$mAP_{0,50:0,95} = \frac{1}{10} \sum_{t \in \{0,50,0,55,\dots,0,95\}} mAP_t. \quad (6)$$

Na Equação 6, mAP_t representa o valor de mAP calculado para o limiar de IoU t . Enquanto a $mAP@0,5$ utiliza um critério mais permissivo quanto à localização espacial, a $mAP@0,5:0,95$ incorpora limiares progressivamente mais exigentes, permitindo avaliar de forma mais rigorosa a capacidade do modelo de localizar e classificar os objetos.

No presente PFC, essas métricas são utilizadas especificamente para avaliar o módulo de detecção e classificação dos símbolos de instrumentação. A precisão e a revocação descrevem, respectivamente, a ocorrência de predições indevidas e a recuperação das instâncias anotadas; a IoU avalia a correspondência espacial entre caixas previstas e caixas de referência; e a mAP sintetiza o desempenho entre as classes e os limiares de sobreposição considerados. O reconhecimento das *tags*, a validação técnica, o mapeamento das informações e a integração com o ambiente de engenharia são avaliados separadamente, por meio da comparação com os diagramas de referência e da verificação funcional das estruturas produzidas.

2.4.9 Trabalhos relacionados e posicionamento da contribuição

Os trabalhos relacionados à interpretação computacional de diagramas P&ID podem ser diferenciados de acordo com a etapa do fluxo de engenharia que buscam automatizar. Uma parte das pesquisas concentra-se na conversão de documentos disponíveis como imagens em representações estruturadas, por meio da identificação de símbolos, textos, linhas e relações entre os elementos do diagrama. Outra parte utiliza informações já organizadas ou documentos em formatos processáveis para derivar modelos de planta, estruturas de software ou artefatos de engenharia. Essa distinção é relevante porque a extração das informações visuais e sua posterior utilização em ambientes de automação constituem problemas relacionados, mas com métodos e requisitos distintos.

Uma das primeiras abordagens de extração integrada de informações em diagramas P&ID foi apresentada por Rahul *et al.* (Rahul et al., 2019). O método combina técnicas tradicionais de visão computacional e modelos de aprendizado profundo para identificar códigos de tubulação, linhas, pontos de entrada e saída e símbolos presentes nos diagramas. Após a identificação, os elementos são associados às respectivas tubulações, e as relações obtidas são armazenadas em uma estrutura hierárquica destinada a representar a organização do processo. O trabalho, portanto, não se limita à classificação dos símbolos, pois também busca recuperar parte das relações entre os componentes representados.

A abordagem denominada Digitize-PID amplia esse tipo de processamento ao organizar a digitalização em três etapas: detecção, compreensão e reconciliação. Na primeira, são extraídos textos, linhas e símbolos; na segunda, esses elementos são associados para representar suas relações no diagrama; na terceira, são aplicadas regras de domínio para verificar e corrigir os dados resultantes. A saída é composta por tabelas que relacionam símbolos, textos e tubulações, além de uma representação das conexões entre linhas. O estudo também apresenta um conjunto sintético anotado de diagramas, desenvolvido para o treinamento e a avaliação dos métodos de extração propostos (Paliwal et al., 2021).

A reconstrução das relações entre os elementos foi tratada de forma mais abrangente por Kim *et al.* (Kim et al., 2022), que propuseram um processo para converter diagramas em formato de imagem em representações digitais. A abordagem compreende o reconhecimento de símbolos, textos e linhas, a reconstrução da topologia entre os objetos identificados e a geração de uma representação compatível com o formato DEXPI. Nesse caso, a saída não consiste apenas em uma relação de elementos detectados, mas em um modelo no qual as conexões entre os componentes do processo também são explicitamente representadas.

Em uma abordagem mais recente, Stürmer, Graumann e Koch formularam a digitalização como um problema de geração de grafos a partir da imagem. Em vez de executar separadamente a detecção dos símbolos e a reconstrução das linhas, o método emprega uma arquitetura baseada em transformadores para extrair conjuntamente os elementos e suas interconexões. O trabalho também apresenta uma base de avaliação com anotações em nível de grafo, permitindo analisar a recuperação da estrutura relacional do diagrama, e não apenas a identificação individual dos objetos (Stürmer; Graumann; Koch, 2025).

Outros estudos direcionam as informações extraídas dos P&IDs a aplicações específicas de engenharia. Arroyo *et al.* combinaram reconhecimento óptico e análise semântica para converter documentos legados em descrições de planta orientadas a objetos e, posteriormente, em modelos qualitativos de simulação. Nesse caso, a interpretação do diagrama é utilizada como base para uma representação computacional voltada ao apoio de atividades de validação e análise do processo (Arroyo et al., 2016).

No domínio da automação, Koltun, Kolter e Vogel-Heuser propuseram uma abordagem para segmentar as informações de um P&ID de acordo com a estrutura da ISA-88 e gerar software modular para CLPs. O protótipo desenvolvido produz estruturas de software no formato de intercâmbio padronizado PLCopen. Diferentemente dos trabalhos voltados à leitura de diagramas rasterizados, essa abordagem concentra-se na transformação das informações de um documento vetorial de engenharia em módulos de software de controle (Koltun; Kolter; Vogel-Heuser, 2018).

A Tabela 5 sintetiza o escopo e as principais saídas dos trabalhos discutidos.

Tabela 5 – Síntese dos trabalhos relacionados à interpretação computacional de diagramas P&ID.

Trabalho	Processamento realizado	Representação ou saída principal	Aplicação posterior
Rahul <i>et al.</i> (Rahul <i>et al.</i> , 2019)	Identificação de textos, símbolos, tubulações e pontos de entrada e saída	Estrutura hierárquica com elementos associados às tubulações	Digitalização e consulta das informações do diagrama
Paliwal <i>et al.</i> (Paliwal <i>et al.</i> , 2021)	Deteção de textos, símbolos e linhas, seguida de associação e aplicação de regras de domínio	Tabelas estruturadas e grafo de conectividade	Digitalização do P&ID
Kim <i>et al.</i> (Kim <i>et al.</i> , 2022)	Reconhecimento de símbolos, textos e linhas e reconstrução da topologia	P&ID digital compatível com o formato DEXPI	Utilização das informações em sistemas digitais de engenharia
Stürmer, Graumann e Koch (Stürmer; Graumann; Koch, 2025)	Extração conjunta de símbolos e interconexões por meio de transformadores	Grafo representando os elementos e suas relações	Compreensão estrutural do diagrama
Arroyo <i>et al.</i> (Arroyo <i>et al.</i> , 2016)	Reconhecimento óptico e análise semântica das informações do documento	Descrição de planta orientada a objetos	Geração de modelos qualitativos de simulação
Koltun, Kolter e Vogel-Heuser (Koltun; Kolter; Vogel-Heuser, 2018)	Segmentação de um P&ID vetorial conforme a ISA-88	Software modular de controle no formato PLCopen	Programação de CLPs
Este trabalho	Deteção de símbolos de instrumentação, reconhecimento de <i>tags</i> , estruturação e validação dos dados	Representação intermediária validada e mapeada para objetos de engenharia	Geração assistida de estruturas no Siemens TIA Portal

Fonte: Elaborado pelo autor.

A comparação evidencia que a detecção de símbolos, isoladamente, não constitui o principal diferencial deste trabalho. As pesquisas existentes já apresentam métodos para identificar símbolos, textos e linhas, reconstruir conexões e produzir representações digitais dos diagramas. Algumas abordagens alcançam escopo mais amplo de digitalização, incluindo a recuperação da topologia do processo, funcionalidade que não foi implementada na solução desenvolvida neste PFC.

A delimitação adotada neste trabalho é distinta. A etapa de visão computacional concentra-se em duas classes de símbolos de instrumentação e no reconhecimento das respectivas *tags*, sem buscar reconstruir integralmente tubulações, equipamentos

ou relações topológicas do diagrama. Em contrapartida, o fluxo prossegue além da extração documental: os resultados são armazenados em uma representação intermediária, submetidos à validação técnica, interpretados segundo regras de mapeamento e utilizados para gerar objetos em um ambiente de engenharia de automação.

Assim, a contribuição está na articulação modular entre a leitura assistida do documento e a configuração inicial do projeto de automação. No conjunto de trabalhos analisado, não foi identificada uma abordagem que reunisse, em um mesmo fluxo, o processamento de diagramas em formato PDF, a detecção de símbolos e o reconhecimento de suas *tags*, uma etapa explícita de validação pelo engenheiro, o mapeamento para padrões internos de uma empresa integradora e a geração de objetos no Siemens TIA Portal. Essa constatação refere-se ao recorte bibliográfico examinado e não implica afirmar a inexistência de outras soluções acadêmicas ou comerciais com características semelhantes.

Os conceitos apresentados neste capítulo estabelecem a base teórica necessária para o desenvolvimento da solução proposta. A partir da relação entre documentação técnica, diagramas P&ID, ambientes de engenharia e técnicas de visão computacional, o capítulo seguinte detalha o problema tratado na implementação e os procedimentos metodológicos adotados para desenvolver o fluxo de extração, organização e geração assistida de estruturas de automação.

3 DESCRIÇÃO DO PROBLEMA E REQUISITOS TÉCNICOS

Este capítulo apresenta a descrição do problema de engenharia tratado neste PFC, considerando o fluxo de desenvolvimento de projetos de automação industrial no ambiente da GreyLogix Brasil. A partir da contextualização apresentada na introdução, aprofunda-se a análise da transição entre a documentação técnica de processo e sua posterior organização em estruturas utilizadas nos ambientes de engenharia de automação. Como estudo de caso, considera-se a implementação dessa transição no Siemens TIA Portal, plataforma utilizada nos projetos analisados na GreyLogix. Também são discutidas as variações encontradas no recebimento e na elaboração da documentação, as dificuldades associadas à interpretação de diagramas P&ID e os requisitos técnicos que orientam a solução proposta.

3.1 CONTEXTO TÉCNICO DE APLICAÇÃO

A GreyLogix Brasil atua no desenvolvimento de soluções de engenharia voltadas à automação industrial, envolvendo atividades como engenharia, integração de sistemas, fabricação de painéis elétricos, serviços de campo e suporte técnico a projetos industriais (GreyLogix Brasil, 2026). Nesse cenário, a empresa pode participar de diferentes etapas do ciclo de um projeto. Em alguns casos, atua principalmente como integradora, recebendo documentos técnicos elaborados pelo cliente ou por terceiros e sendo responsável pela interpretação desses materiais, pelo desenvolvimento da programação, pela configuração dos sistemas de automação e pela integração final da solução. Em outros casos, também participa da engenharia de detalhamento, contribuindo para a elaboração ou revisão de documentos que servirão de base para as etapas posteriores do projeto, incluindo diagramas P&ID, listas de entradas e saídas, diagramas lógicos e demais documentos de apoio à automação.

Essa variação no escopo de atuação influencia diretamente o fluxo de trabalho do engenheiro de automação. Quando a documentação é desenvolvida ou detalhada internamente, há maior possibilidade de aderência aos padrões técnicos utilizados pela empresa, incluindo critérios de simbologia, nomenclatura, organização de *tags* e estruturação das informações. Por outro lado, quando a documentação é fornecida pelo cliente, os diagramas e listas podem apresentar diferentes níveis de padronização, revisão e completude. Em determinadas situações, o projeto pode chegar com um P&ID já definido; em outras, pode haver apenas informações parciais do processo, exigindo que a equipe interprete a demanda, complemente a documentação e derive os dados necessários para o desenvolvimento da automação.

Embora os diagramas P&ID sejam documentos centrais nesse processo, eles não são os únicos elementos utilizados na elaboração de um projeto de automação. Listas de I/Os, diagramas lógicos, memoriais descritivos, listas de instrumentos, es-

pecificações de equipamentos e padrões internos de programação também podem ser utilizados para definir as estruturas que serão implementadas no controlador. No entanto, esses documentos podem existir em diferentes níveis de detalhe ou mesmo estar ausentes em determinadas etapas do projeto. Nesses casos, o P&ID torna-se uma das principais fontes para identificação inicial de instrumentos, sensores e demais elementos associados ao desenvolvimento do sistema de automação.

Dessa forma, a passagem da documentação técnica para o ambiente de automação envolve uma etapa de interpretação e organização das informações de engenharia. O projetista precisa reconhecer os elementos representados nos diagramas, compreender sua função no processo e relacioná-los às estruturas que serão criadas no ambiente de automação, como *tags*, tipos de dados, blocos funcionais e tabelas de sinais. Assim, a configuração inicial do projeto depende da coerência entre a documentação disponível, os padrões técnicos adotados e a forma como esses elementos serão implementados no controlador.

3.2 FLUXO ATUAL DE EXTRAÇÃO DE INFORMAÇÕES A PARTIR DA DOCUMENTAÇÃO TÉCNICA

No fluxo atual de desenvolvimento de projetos de automação, as informações provenientes da documentação técnica precisam ser interpretadas e convertidas manualmente em estruturas utilizadas no ambiente de engenharia. Quando o conjunto documental recebido é completo, essa etapa envolve principalmente a conferência e adequação das informações aos padrões utilizados pela GreyLogix. Quando os documentos são incompletos ou apresentam inconsistências, o esforço inicial aumenta, pois a equipe precisa complementar, organizar e validar os dados antes de iniciar a configuração do projeto de automação.

Nos casos em que o P&ID é uma das principais fontes disponíveis, o engenheiro identifica manualmente os instrumentos e sensores representados no diagrama, interpreta as *tags* associadas e relaciona esses elementos às estruturas necessárias para o desenvolvimento do projeto. A partir dessa leitura, podem ser derivadas informações para listas de I/Os, definição de variáveis, seleção de blocos funcionais, organização de dados e posterior configuração no ambiente de engenharia adotado no projeto. No estudo de caso desenvolvido neste trabalho, essa etapa corresponde à configuração das estruturas no Siemens TIA Portal.

A natureza manual dessa etapa aumenta a possibilidade de inconsistências, especialmente em projetos com grande volume de instrumentos. Erros de leitura, omissão de símbolos, duplicidade de informações ou divergências de nomenclatura podem gerar retrabalho em fases posteriores do desenvolvimento. Além disso, quando diferentes projetistas participam do mesmo projeto ou de projetos semelhantes, a ausência de um fluxo inicial mais padronizado pode dificultar a uniformidade na criação de *tags*,

blocos e estruturas de dados.

A padronização tem papel importante nesse cenário porque facilita a geração inicial do projeto, sua manutenção e sua continuidade. Projetos organizados de forma consistente tendem a ser mais compreensíveis para outros engenheiros que atuem posteriormente na revisão, expansão, integração ou manutenção do sistema. Assim, reduzir a variabilidade na interpretação inicial da documentação contribui para melhorar a rastreabilidade das informações e a continuidade técnica do projeto ao longo de seu ciclo de vida.

3.3 CARACTERIZAÇÃO DO PROBLEMA TRATADO

O problema tratado neste PFC está relacionado à dificuldade de interpretar, organizar e converter informações presentes em diagramas P&ID em dados estruturados que possam apoiar a configuração inicial de projetos de automação em ambientes de engenharia. No caso analisado na GreyLogix, a etapa de integração é implementada no Siemens TIA Portal, plataforma adotada para o desenvolvimento do protótipo. Essa dificuldade é influenciada pela variabilidade da documentação técnica utilizada nos projetos, uma vez que os diagramas podem apresentar diferentes padrões gráficos, níveis de detalhamento, revisões, qualidade visual e convenções de nomenclatura.

Embora o P&ID não seja o único documento utilizado no desenvolvimento de projetos de automação, ele frequentemente assume papel relevante nas etapas iniciais, especialmente quando listas de I/Os, diagramas lógicos, listas de instrumentos ou outros documentos complementares ainda estão incompletos ou indisponíveis. Nessas situações, a leitura do diagrama contribui para a identificação preliminar dos instrumentos e sensores que serão posteriormente tratados no projeto de automação.

A identificação manual desses elementos representa uma etapa sensível do processo. Em documentos extensos ou visualmente densos, a leitura e a transcrição das informações podem demandar tempo significativo e estar sujeitas a falhas, como omissão de instrumentos, leitura incorreta de *tags* ou divergências de nomenclatura. Esses problemas podem gerar retrabalho em etapas posteriores, exigindo correções em tabelas de variáveis, ajustes de identificação e revisão de estruturas já criadas no projeto.

O problema de engenharia envolve apoiar a extração inicial de informações relevantes, principalmente relacionadas a sensores e instrumentos de medição, de modo que esses dados possam ser organizados com maior consistência e utilizados como base para a criação inicial de estruturas no ambiente de engenharia. No protótipo desenvolvido, essas estruturas são geradas para o Siemens TIA Portal, conforme os recursos e os padrões adotados no estudo de caso. A proposta, portanto, não substitui a análise técnica do engenheiro. Seu objetivo é apoiar as atividades de interpretação e transcrição das informações, favorecendo maior consistência e padronização na

configuração inicial do projeto e reduzindo a exposição a omissões, duplicidades e divergências de nomenclatura.

A partir dessa caracterização, a solução a ser desenvolvida deve considerar um fluxo assistido, no qual as informações extraídas do P&ID sejam apresentadas de forma estruturada e passível de validação. Essa abordagem preserva a responsabilidade técnica do projetista e, ao mesmo tempo, oferece suporte computacional para uma etapa repetitiva, trabalhosa e relevante para a organização inicial do projeto de automação.

3.4 REQUISITOS TÉCNICOS

A partir da caracterização do problema, foram definidos requisitos técnicos para orientar o desenvolvimento da solução proposta. Esses requisitos foram elaborados considerando as etapas necessárias ao fluxo assistido de extração de informações em diagramas P&ID, incluindo a entrada das imagens, a detecção dos símbolos de instrumentação, a associação com informações textuais, a organização dos dados extraídos, a validação pelo projetista e a posterior integração com um ambiente de engenharia. Para o estudo de caso desenvolvido, também são considerados os requisitos específicos necessários à integração com o Siemens TIA Portal. Os requisitos funcionais, apresentados na Tabela 6, descrevem as ações que a solução deve executar. Já os requisitos não funcionais, apresentados na Tabela 7, definem condições de qualidade, restrições técnicas e critérios de uso que devem ser respeitados durante o desenvolvimento e a aplicação da solução.

Tabela 6 – Requisitos funcionais da solução proposta

Código	Descrição
RF01	A solução deve permitir a entrada de imagens derivadas de diagramas P&ID para processamento computacional.
RF02	A solução deve identificar automaticamente símbolos de instrumentação definidos no escopo do projeto.
RF03	A solução deve retornar, para cada símbolo detectado, sua classe, sua posição na imagem e um valor de confiança associado à detecção.
RF04	A solução deve permitir a extração ou associação de informações textuais relacionadas aos símbolos detectados, especialmente <i>tags</i> de instrumentos.
RF05	A solução deve organizar as informações extraídas em uma estrutura intermediária de dados, contendo ao menos classe do símbolo, posição, <i>tag</i> associada e demais atributos relevantes.
RF06	A solução deve permitir a revisão técnica das informações extraídas antes de sua utilização no ambiente de engenharia.
RF07	A solução deve possibilitar o mapeamento dos elementos identificados para padrões de nomenclatura, tipos de dados ou objetos de engenharia utilizados no projeto de automação.
RF08	A solução deve fornecer dados estruturados em formato adequado às etapas posteriores de integração com ambientes de engenharia de automação.
RF09	A solução deve apoiar a criação assistida de estruturas iniciais no ambiente de engenharia a partir dos dados extraídos e validados.
RF10	No estudo de caso, a solução deve gerar e importar as estruturas de engenharia definidas no escopo para o Siemens TIA Portal.

Fonte: Elaborado pelo autor.

Tabela 7 – Requisitos não funcionais da solução proposta

Código	Descrição
RNF01	A solução deve preservar a rastreabilidade entre o símbolo detectado, sua localização no diagrama, a <i>tag</i> associada e o dado estruturado gerado.
RNF02	A solução deve manter a possibilidade de validação humana, sem substituir a análise técnica do engenheiro responsável pelo projeto.
RNF03	A estrutura intermediária de dados deve ser legível, organizada e adequada para uso em etapas posteriores do fluxo de automação.
RNF04	A solução deve ser modular, separando as etapas de preparação das imagens, detecção de símbolos, reconhecimento textual, associação símbolo- <i>tag</i> , estruturação dos dados, mapeamento das informações e integração específica com o ambiente de engenharia.
RNF05	A solução deve adotar a ANSI/ISA-5.1 como referência para a padronização dos símbolos e das <i>tags</i> de instrumentação considerados nos diagramas P&ID.
RNF06	A solução deve favorecer a padronização das informações utilizadas na criação inicial de projetos de automação.
RNF07	O fluxo desenvolvido deve ser reproduzível a partir dos mesmos dados de entrada e das mesmas configurações utilizadas no processamento.
RNF08	A camada de integração implementada no estudo de caso deve ser compatível com o Siemens TIA Portal e com os mecanismos de automação de engenharia disponibilizados pela plataforma.

Fonte: Elaborado pelo autor.

4 SOLUÇÃO PROPOSTA E METODOLOGIA UTILIZADA

Este capítulo apresenta a solução proposta para apoiar a extração de informações de diagramas P&ID e sua utilização na configuração inicial de projetos de automação industrial. A partir do problema e dos requisitos técnicos definidos no capítulo anterior, são descritos os componentes funcionais da proposta, o fluxo de tratamento das informações e os critérios adotados para organizar a transição entre a documentação de processo e os ambientes de engenharia de automação. Como estudo de caso, a etapa de geração dos objetos é implementada no Siemens TIA Portal, plataforma utilizada nos projetos analisados na GreyLogix.

A principal contribuição da proposta está na organização dessas etapas em uma arquitetura modular, na qual a preparação dos documentos, a extração das informações visuais e textuais, a representação intermediária dos dados, a validação técnica, o mapeamento para estruturas de engenharia e a integração com a plataforma de destino são tratados como módulos distintos, conectados por interfaces de dados bem definidas. Essa arquitetura permite apresentar e analisar o fluxo em nível conceitual antes de detalhar as tecnologias específicas empregadas em sua implementação.

A solução foi concebida como um fluxo assistido, no qual técnicas de visão computacional e reconhecimento textual são empregadas para identificar símbolos de instrumentação e *tags* presentes nos diagramas. Os dados obtidos são organizados em uma representação intermediária, submetidos à validação do engenheiro e posteriormente relacionados aos padrões necessários à geração de estruturas de engenharia. Dessa forma, a proposta busca reduzir etapas manuais de interpretação e transcrição, preservando a revisão técnica antes da integração com o ambiente de automação. Os detalhes de implementação, as configurações utilizadas e os resultados obtidos são apresentados no capítulo seguinte.

4.1 VISÃO GERAL DA SOLUÇÃO PROPOSTA

A solução proposta tem como entrada principal diagramas P&ID contendo símbolos de instrumentação e informações textuais associadas. Esses documentos são submetidos a um conjunto de etapas destinadas a localizar os símbolos pertencentes às classes definidas no escopo do projeto, reconhecer suas *tags* e organizar os resultados em uma representação estruturada. A partir dessa representação, os dados podem ser revisados pelo engenheiro, relacionados aos padrões adotados no projeto e utilizados como base para a criação inicial de objetos no ambiente de engenharia.

O fluxo foi organizado de forma modular, de modo que cada etapa desempenhe uma função específica no tratamento dos dados. Inicialmente, os documentos passam por uma etapa de preparação, responsável por adequar sua representação visual às condições necessárias para o processamento. Em seguida, um modelo de detecção

de objetos localiza e classifica os símbolos de instrumentação, retornando, para cada ocorrência, a classe prevista, a posição na imagem e o valor de confiança associado à detecção.

A partir das regiões delimitadas pelo detector, o OCR é utilizado para reconhecer as *tags* e os identificadores textuais associados aos símbolos. Como a informação gráfica isolada não permite identificar individualmente o instrumento, os resultados da detecção e do reconhecimento textual são reunidos para representar cada ocorrência por sua classe, localização e identificação textual.

Após a extração das informações, os resultados são organizados em uma estrutura intermediária. Essa camada separa o processamento visual das regras específicas utilizadas na geração dos objetos de automação, evitando que a saída dos módulos computacionais seja incorporada diretamente ao ambiente de engenharia. A representação intermediária também permite preservar a origem dos dados, os valores de confiança e a posição de cada elemento identificado.

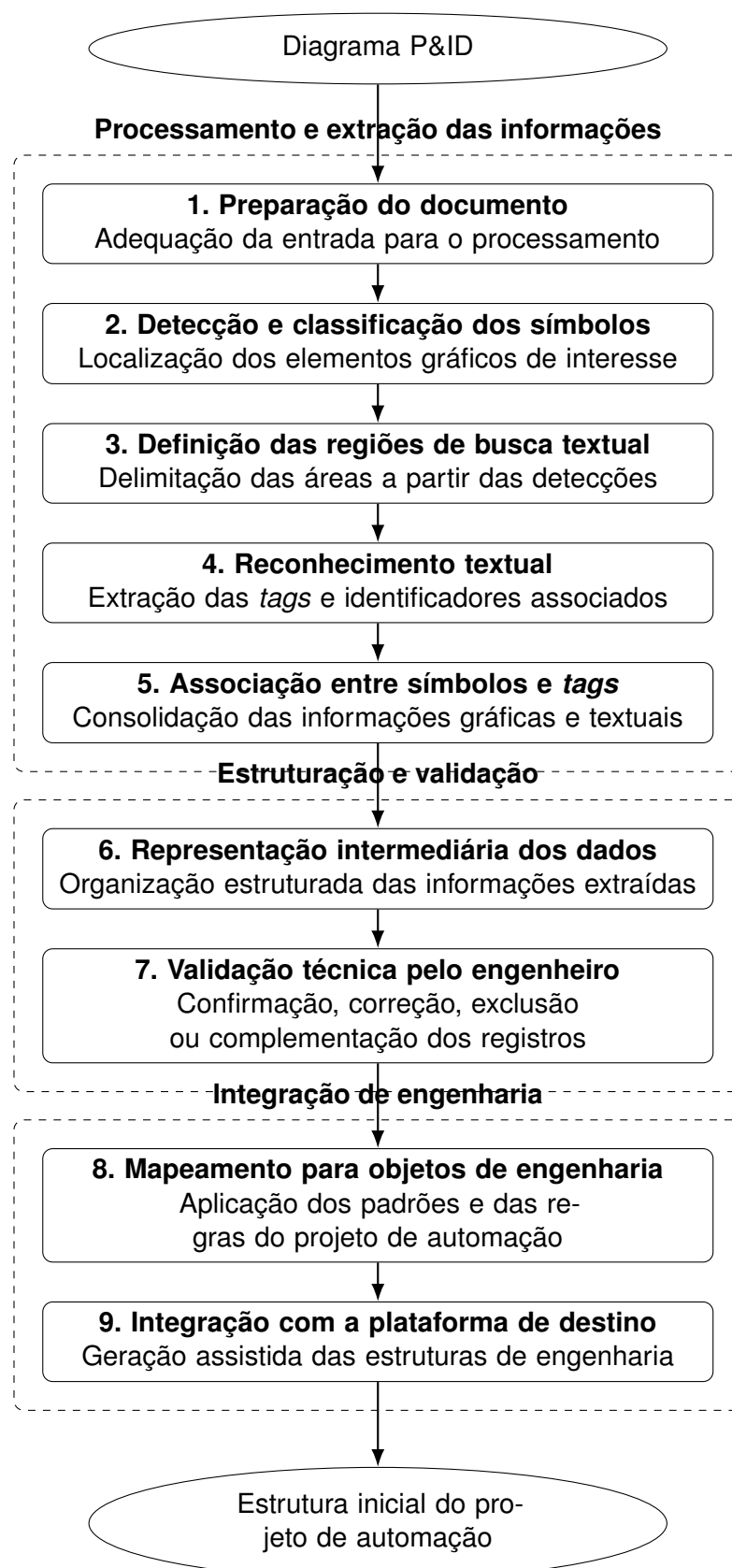
Antes da geração dos objetos, os dados extraídos são submetidos à revisão técnica. Essa etapa permite corrigir detecções incorretas, complementar informações ausentes e verificar a correspondência entre os símbolos e suas respectivas *tags*. A manutenção desse ponto de validação caracteriza a proposta como um sistema de apoio ao engenheiro, e não como um mecanismo autônomo de geração de projetos.

Após a validação, uma camada de mapeamento relaciona os elementos identificados no diagrama aos padrões adotados no projeto de automação. Essa etapa pode envolver a adequação de nomenclaturas, a definição de tipos de dados e a associação a tabelas de *tags*, estruturas ou blocos funcionais previamente estabelecidos.

Os dados resultantes são então encaminhados à camada responsável pela integração com o ambiente de engenharia. Como estudo de caso, essa etapa foi implementada no Siemens TIA Portal por meio dos mecanismos de automação disponibilizados pela plataforma. A separação dessa camada em relação aos módulos anteriores permite que a arquitetura geral seja mantida, embora a adaptação para outro ambiente exija novas regras de mapeamento e uma interface de integração compatível.

A Figura 7 apresenta a arquitetura conceitual da solução e o encadeamento sequencial das informações entre seus módulos. O fluxo é dividido em três conjuntos principais: processamento e extração das informações, estruturação e validação dos dados e integração de engenharia. Essa representação descreve as funções desempenhadas por cada etapa sem vinculá-las às tecnologias específicas adotadas na implementação do estudo de caso.

Figura 7 – Arquitetura conceitual e fluxo sequencial da solução proposta.



Fonte: Elaborado pelo autor.

4.2 ESTRUTURA E ETAPAS DA SOLUÇÃO

A solução foi organizada em etapas com funções distintas, abrangendo a preparação dos documentos, a extração de informações visuais e textuais, a validação dos resultados e sua posterior conversão em dados compatíveis com objetos de engenharia. Essa organização permite acompanhar o percurso dos dados ao longo do sistema e compreender como cada etapa contribui para o atendimento aos requisitos técnicos definidos no capítulo anterior.

A separação entre os módulos reduz o acoplamento entre as técnicas de visão computacional, os procedimentos de reconhecimento textual e as regras específicas do projeto de automação. Dessa forma, alterações no modelo de detecção, no mecanismo de OCR, nos padrões de mapeamento ou na camada de integração podem ser realizadas sem exigir a reformulação completa do fluxo. Essa organização também permite distinguir os componentes gerais da solução dos recursos específicos empregados na implementação do estudo de caso. As subseções seguintes descrevem a função de cada etapa.

4.2.1 Preparação dos documentos de entrada

A preparação dos documentos de entrada constitui a primeira etapa do fluxo de processamento. Sua finalidade é adequar os diagramas a uma representação visual compatível com os métodos de detecção e OCR, preservando os elementos gráficos e textuais necessários à identificação dos instrumentos.

Os diagramas utilizados em projetos de automação podem ter diferentes origens, podendo ser elaborados pela própria GreyLogix, recebidos de clientes ou fornecidos por terceiros envolvidos no projeto. Como consequência, esses documentos podem apresentar diferenças de revisão, organização, nível de detalhamento, qualidade visual e aderência às convenções de simbologia. No escopo deste trabalho, são considerados como referência os símbolos e as identificações de instrumentação definidos pela ANSI/ISA-5.1, conforme a delimitação estabelecida para a solução proposta (International Society of Automation, 2024).

Diante dessas variações, o pré-processamento consiste na adequação da representação visual dos documentos às condições de entrada exigidas pelas etapas computacionais. Quando o diagrama possui grandes dimensões ou elevada densidade de elementos, essa preparação pode incluir sua divisão em regiões menores. Essa estratégia busca melhorar a resolução relativa de símbolos e textos que ocupariam uma parcela reduzida da página completa, evitando que o redimensionamento do documento comprometa a identificação dos elementos de interesse.

A preparação não tem como objetivo modificar o conteúdo técnico do diagrama, corrigir inconsistências de engenharia ou reinterpretar as informações representadas.

Sua função é produzir uma entrada adequada ao processamento, preservando os elementos visuais e textuais necessários à identificação dos instrumentos. Ao final dessa etapa, as imagens preparadas são encaminhadas ao módulo de detecção de símbolos, cujos resultados orientam posteriormente o reconhecimento das *tags*.

4.2.2 Identificação dos símbolos de instrumentação

Após a preparação dos documentos de entrada, a etapa seguinte consiste na identificação dos símbolos de instrumentação presentes nos diagramas P&ID. Sua função é localizar os elementos gráficos pertencentes às classes definidas no escopo do projeto e atribuir a cada ocorrência uma classificação compatível com a representação visual do símbolo. Essa etapa concentra-se nas representações associadas a instrumentos de campo e a funções de indicação ou controle compartilhadas, conforme a delimitação estabelecida para a solução.

Para realizar essa tarefa, foi adotada uma abordagem de detecção de objetos baseada em modelos da família YOLO. Esse tipo de modelo permite localizar e classificar múltiplos elementos em uma mesma imagem, produzindo como saída a classe prevista, as coordenadas da caixa delimitadora e um valor de confiança para cada detecção. Essa estrutura de saída é adequada à leitura de diagramas técnicos, nos quais diferentes símbolos podem estar distribuídos ao longo da página e ocupar regiões reduzidas em relação às dimensões totais do documento (Redmon et al., 2016; Bochkovskiy; Wang; Liao, 2020).

A escolha de um modelo da família YOLO foi precedida por uma tentativa inicial com técnicas clássicas de visão computacional. Essa abordagem não apresentou resultados satisfatórios para a localização dos símbolos nos diagramas analisados, especialmente pela dificuldade de definir regras e descritores suficientemente robustos para lidar com a variação gráfica, a densidade visual e a pequena dimensão relativa dos elementos de interesse. Diante disso, optou-se pelo uso do YOLO, por integrar localização e classificação em um único modelo e permitir a detecção simultânea de múltiplos símbolos em uma mesma imagem (Redmon et al., 2016; Bochkovskiy; Wang; Liao, 2020).

Para delimitar o escopo do módulo de identificação, foram selecionadas duas classes de símbolos de instrumentação entre as representações discutidas anteriormente. A escolha considerou a compatibilidade dessas classes com os instrumentos tratados nas etapas posteriores do fluxo e a necessidade de avaliar a solução completa, desde a detecção até a geração dos objetos de engenharia. A Tabela 8 apresenta as classes consideradas pela solução proposta.

Tabela 8 – Classes de símbolos de instrumentação consideradas pela solução proposta.

Classe	Denominação	Representação	Descrição
instrument tag	Instrumento em campo		Instrumento localizado em campo, normalmente visível e acessível ao operador.
instrument DCS	Display ou controle compartilhado		Função de indicação ou controle compartilhada, localizada na frente de painel principal, console ou sistema de operação, normalmente visível e acessível ao operador.

Fonte: Elaborado pelo autor com base na ANSI/ISA-5.1 (International Society of Automation, 2024).

A delimitação a essas duas classes constituiu uma decisão metodológica do trabalho. Embora o conjunto de dados original incluísse outras categorias de elementos presentes em diagramas P&ID, foram mantidas somente as representações compatíveis com o recorte de instrumentação adotado e com os objetos previstos nas etapas de reconhecimento textual, validação, mapeamento e integração. Essa restrição permitiu concentrar a avaliação no funcionamento do fluxo integrado, sem pretender abranger todas as simbologias previstas pela ANSI/ISA-5.1.

Como premissa de aplicação, considera-se que os documentos de entrada utilizem representações gráficas compatíveis com as duas classes contempladas pelo conjunto de treinamento. Símbolos não pertencentes às classes selecionadas, bem como variações gráficas significativamente distintas das utilizadas no treinamento, não fazem parte do escopo avaliado e podem não ser identificados corretamente. Esses casos devem ser tratados pela ampliação futura do conjunto de classes ou pela etapa de revisão técnica.

Ao final dessa etapa, cada símbolo detectado é representado por sua classe, localização e confiança associada. Essas informações ainda não caracterizam completamente o instrumento, pois sua identificação técnica depende da leitura da *tag* presente no diagrama. Por esse motivo, os resultados da detecção são encaminhados à etapa de reconhecimento textual e associação entre símbolos e *tags*, descrita na subseção seguinte.

4.2.3 Reconhecimento de tags e associação símbolo-tag

Após a identificação dos símbolos de instrumentação, é necessário reconhecer a *tag* correspondente a cada detecção. A classe prevista pelo modelo indica o tipo de representação gráfica encontrada, mas não identifica individualmente o instrumento no projeto. Essa identificação depende da leitura da *tag*, que reúne informações sobre

a variável medida, a função desempenhada e o identificador numérico associado à malha ou ao ponto de medição.

Para essa etapa, as caixas delimitadoras obtidas pelo modelo YOLO são utilizadas como referência para definir as regiões submetidas ao OCR. Cada região é formada a partir da posição do símbolo detectado e pode ser ampliada para incluir os caracteres localizados em seus limites. Essa estratégia restringe o reconhecimento textual à área associada ao instrumento, reduzindo a interferência de textos pertencentes a outras partes do diagrama.

Como cada recorte é gerado a partir de uma detecção específica, o resultado do reconhecimento textual permanece diretamente vinculado ao símbolo que originou a região. Dessa forma, a associação entre símbolo e *tag* não depende de uma busca posterior por textos distribuídos em toda a página, mas da relação já estabelecida entre a detecção e a área encaminhada ao módulo de OCR.

Ainda assim, podem ocorrer falhas quando a caixa delimitadora não envolve adequadamente todos os caracteres, quando a *tag* está sobreposta ao símbolo ou quando outros elementos gráficos interferem na região analisada. Por esse motivo, os resultados permanecem passíveis de correção na etapa de validação técnica.

Ao final do processamento, cada instrumento pode ser representado por sua classe, posição, valor de confiança e *tag* candidata. Esse conjunto de informações é encaminhado à etapa de estruturação dos dados, na qual os resultados visuais e textuais são organizados em um formato intermediário adequado às etapas posteriores do fluxo.

4.2.4 Representação intermediária dos dados

Após a detecção dos símbolos e o reconhecimento das *tags*, as informações extraídas precisam ser organizadas em uma estrutura que possa ser utilizada pelas etapas posteriores da solução. Para isso, foi adotado o formato *JSON* como representação intermediária dos dados. Essa representação atua como uma camada de comunicação entre os módulos de visão computacional, OCR, validação, mapeamento e integração com o ambiente de engenharia.

A utilização de uma representação intermediária evita o acoplamento direto entre a saída dos modelos computacionais e os objetos do ambiente de engenharia. Com isso, os resultados da detecção e do reconhecimento textual podem ser armazenados, inspecionados e corrigidos antes de serem utilizados na criação de estruturas de automação. Essa separação também permite que alterações em uma etapa, como a substituição do modelo de detecção ou o ajuste das regras de associação entre símbolo e *tag*, não exijam a reformulação integral dos demais componentes da solução.

Além de apoiar a integração implementada no estudo de caso, a adoção de uma representação intermediária em *JSON* amplia as possibilidades de reutilização dos

dados extraídos. Por se tratar de um formato estruturado e independente da aplicação de destino, esses dados podem ser posteriormente empregados em outras rotinas de engenharia, como geração de planilhas, elaboração de listas de instrumentos, organização de relatórios, alimentação de bases de dados ou desenvolvimento de novas automações. Essas aplicações não fazem parte do escopo implementado neste trabalho, mas indicam a extensibilidade da solução e reduzem sua dependência de um único ambiente de engenharia.

O arquivo *JSON* registra informações gerais sobre o documento processado e uma lista dos símbolos identificados. Para cada detecção, são armazenados a classe prevista, o valor de confiança, a *tag* associada, as coordenadas da caixa delimitadora e a página de origem. As dimensões da imagem também são preservadas, permitindo interpretar as coordenadas absolutas e relacionar cada detecção à sua posição no documento analisado.

A Tabela 9 apresenta os principais campos definidos para essa representação.

Tabela 9 – Principais campos da representação intermediária dos dados.

Campo	Escopo	Descrição
<code>source_file</code>	Documento	Identificação do arquivo de origem utilizado no processamento.
<code>image_width</code> e <code>image_height</code>	Documento	Dimensões da imagem analisada, expressas em pixels.
<code>symbols</code>	Documento	Lista contendo os símbolos identificados no diagrama.
<code>symbol_class</code>	Símbolo	Classe atribuída ao símbolo pelo modelo de detecção.
<code>confidence</code>	Símbolo	Valor de confiança associado à classe prevista pelo modelo.
<code>tag</code>	Símbolo	Texto reconhecido e associado ao símbolo detectado como <i>tag</i> candidata.
<code>bbox_abs</code>	Símbolo	Coordenadas absolutas da caixa delimitadora na imagem.
<code>page</code>	Símbolo	Página do documento na qual o elemento foi identificado.

Fonte: Elaborado pelo autor.

A representação intermediária reúne em um mesmo registro os dados visuais, textuais e espaciais referentes a cada elemento identificado. Antes da integração com o ambiente de engenharia, esses registros são submetidos à validação técnica e às regras de mapeamento definidas para o projeto de automação.

No escopo deste trabalho, as coordenadas são preservadas para manter a

rastreabilidade dos registros, permitir a recuperação da região correspondente no documento e apoiar a etapa de validação técnica. Entretanto, essas informações não são utilizadas para interpretar a posição relativa entre os elementos nem para reconstruir as conexões gráficas representadas no diagrama.

4.2.5 Validação técnica pelo engenheiro

A validação técnica constitui uma etapa intermediária entre a extração automática das informações e sua utilização na geração dos objetos de engenharia. Sua inclusão no fluxo é necessária porque a detecção de símbolos, o reconhecimento de *tags* e a associação entre símbolo e texto podem apresentar erros, especialmente em regiões com elevada densidade gráfica, baixa legibilidade ou representações diferentes das previstas no conjunto de treinamento. Por esse motivo, os resultados obtidos não são encaminhados diretamente à etapa de geração dos objetos, mas passam antes pela análise do engenheiro responsável.

Para apoiar essa revisão, foi incorporada à solução uma aplicação de validação que apresenta individualmente cada detecção realizada. Para cada elemento identificado, a interface exibe a região correspondente do diagrama, a classe atribuída pelo modelo e a *tag* reconhecida pelo módulo de OCR. Essa visualização permite comparar o resultado gerado automaticamente com a representação original do instrumento, antes que os dados sejam utilizados nas etapas de mapeamento e integração.

Quando necessário, o engenheiro pode alterar a classe atribuída ao símbolo ou corrigir o texto reconhecido como *tag*. Também é possível excluir detecções consideradas incorretas, evitando que falsos positivos sejam mantidos na representação intermediária dos dados. Essas operações permitem corrigir falhas das etapas automáticas sem a necessidade de repetir todo o processamento do documento.

Além da revisão das detecções existentes, a aplicação permite inserir manualmente elementos que não tenham sido identificados pelo modelo. Nesse procedimento, o engenheiro seleciona a região correspondente no diagrama, informa a classe do símbolo e atribui a *tag* associada. Essa funcionalidade permite complementar os resultados da detecção automática e registrar instrumentos omitidos pelo modelo.

Ao término da validação, os registros corrigidos, excluídos ou adicionados manualmente são utilizados para atualizar a representação intermediária dos dados. Somente após essa revisão as informações são encaminhadas às etapas de mapeamento e integração com o ambiente de engenharia. Dessa forma, a solução mantém o caráter assistido do processo, combinando a extração automática com a avaliação técnica do engenheiro e reduzindo a possibilidade de que erros das etapas anteriores sejam incorporados ao projeto de automação.

4.2.6 Mapeamento para objetos de engenharia

Após a validação das informações extraídas, os símbolos identificados e suas respectivas *tags* são interpretados para definir quais elementos devem ser representados no projeto de automação. Nessa etapa, a classe do símbolo descreve o tipo de representação gráfica encontrada, enquanto a *tag* fornece informações sobre a variável medida, a função associada ao instrumento e sua identificação no processo.

A interpretação das *tags* utiliza como referência as convenções estabelecidas pela ANSI/ISA-5.1. De acordo com esse padrão, as letras que compõem a identificação funcional permitem reconhecer, entre outros aspectos, a variável de processo e a função desempenhada pelo instrumento. Assim, instrumentos associados a diferentes medições e funções podem ser direcionados a categorias específicas de objetos, conforme as regras definidas para o fluxo de mapeamento.

Com base nessa interpretação, a camada de mapeamento define quais estruturas devem ser preparadas para o ambiente de engenharia. Para um instrumento associado à medição de vazão, por exemplo, pode ser prevista a criação de uma *tag* correspondente e sua associação a uma estrutura padronizada para o tratamento de sinais analógicos. A correspondência exata depende do tipo de instrumento, da natureza do sinal, da biblioteca disponível e dos critérios adotados para a organização do projeto.

Além da criação das *tags* e da associação com estruturas de programação, o mapeamento deve considerar os tipos de dados, as bibliotecas e os padrões definidos para o projeto. Essa padronização permite que instrumentos semelhantes sejam representados de forma consistente, utilizando interfaces e parâmetros previamente estabelecidos.

A separação entre a extração das informações e as regras de mapeamento permite ajustar a representação dos instrumentos sem modificar o modelo de visão computacional. Dessa forma, as mesmas classes e *tags* identificadas no diagrama podem ser relacionadas a estruturas distintas, conforme o ambiente de engenharia, a biblioteca e os padrões de programação utilizados. No estudo de caso deste trabalho, as regras foram definidas para os objetos e as bibliotecas disponíveis no Siemens TIA Portal. Os detalhes dos tipos de dados, dos blocos e dos objetos efetivamente gerados são apresentados no capítulo de desenvolvimento.

4.2.7 Integração com o ambiente de engenharia

A integração com o ambiente de engenharia corresponde à etapa final da arquitetura proposta. Nela, os dados extraídos, validados e mapeados são utilizados para apoiar a criação das estruturas iniciais do projeto de automação. A implementação dessa etapa depende dos objetos, formatos de intercâmbio, bibliotecas e interfaces

disponibilizados pela plataforma de destino.

No estudo de caso, a integração foi definida para o Siemens TIA Portal, ambiente utilizado nos projetos analisados na GreyLogix. A camada responsável por essa operação recebe a representação intermediária validada e utiliza as regras de mapeamento para determinar quais estruturas devem ser preparadas para cada instrumento. Dessa forma, a saída do processamento documental não é transferida diretamente ao ambiente de engenharia, mas passa previamente pela validação e pela interpretação das informações conforme os padrões do projeto.

A geração dos objetos considera a categoria atribuída ao instrumento, sua *tag*, os tipos de dados e as estruturas padronizadas disponíveis no projeto. No escopo do estudo de caso, foram previstas tabelas de *tags*, blocos de configuração, blocos de função e blocos de dados associados aos instrumentos identificados. A correspondência entre essas estruturas e os registros validados é estabelecida pelas regras definidas na camada de mapeamento.

Para realizar a integração, foi prevista uma aplicação externa desenvolvida em C# e .NET Framework, responsável pela geração de arquivos no formato SIMATIC ML e por sua importação no projeto por meio do TIA Portal Openness. Os detalhes relativos à implementação da aplicação, à geração dos arquivos, às estruturas da biblioteca utilizada e às operações executadas no ambiente Siemens são apresentados no capítulo de desenvolvimento.

A separação dessa camada em relação aos módulos de detecção, reconhecimento textual e validação reduz o acoplamento entre o processamento dos diagramas e a plataforma utilizada como destino. Entretanto, a adaptação da solução a outro ambiente de engenharia exigiria a definição de novas regras de mapeamento e a implementação de uma interface compatível com os formatos e mecanismos disponibilizados pela plataforma correspondente.

4.3 CRITÉRIOS DE AVALIAÇÃO E VALIDAÇÃO DA SOLUÇÃO

A avaliação da solução foi definida de acordo com as etapas que compõem o fluxo proposto. Para o modelo de detecção, foram consideradas métricas próprias de detecção de objetos, como precisão, revocação, IoU e mAP. Essas métricas permitem avaliar a capacidade do modelo de localizar e classificar os símbolos pertencentes às classes definidas no escopo do projeto.

Além da avaliação do modelo, foi realizada uma verificação funcional das informações extraídas. Essa análise considerou a correspondência entre os símbolos identificados, as *tags* reconhecidas pelo módulo de OCR e os elementos representados nos diagramas. Os resultados também foram examinados por meio da aplicação de validação, permitindo verificar detecções incorretas, leituras incompletas e símbolos não localizados automaticamente.

A camada de integração foi avaliada de forma funcional no ambiente adotado como estudo de caso, considerando a criação dos objetos definidos no escopo, a aplicação dos padrões de nomenclatura e a utilização dos tipos de dados previstos pela biblioteca selecionada. Essa verificação buscou confirmar a continuidade do fluxo entre os dados extraídos do diagrama, a etapa de validação, o mapeamento das informações e a geração das estruturas no Siemens TIA Portal.

Devido ao período disponível para o desenvolvimento, não foi realizada uma comparação quantitativa controlada entre o procedimento manual e o fluxo assistido. Dessa forma, as contribuições relacionadas à redução de etapas manuais, à organização das informações e à padronização do processo são discutidas qualitativamente, com base nas funcionalidades implementadas e nas características do fluxo de engenharia analisado.

Este capítulo apresentou a solução proposta para apoiar a extração de informações de diagramas P&ID e sua utilização na configuração inicial de projetos de automação. Foram descritas as etapas de preparação dos documentos, identificação dos símbolos, reconhecimento e associação das *tags*, organização dos dados em formato intermediário, validação técnica, mapeamento para objetos de engenharia e integração com o ambiente de automação. Também foi apresentada a implementação da camada de integração no Siemens TIA Portal como estudo de caso e foram definidos os critérios adotados para avaliar o desempenho do modelo e verificar funcionalmente o fluxo desenvolvido. A partir dessa estrutura, o capítulo seguinte apresenta os detalhes da implementação, os testes realizados e a análise dos resultados obtidos.

5 DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS OBTIDOS

Este capítulo apresenta o desenvolvimento da solução proposta e a análise dos resultados obtidos em sua implementação. São descritas as etapas de construção do fluxo, incluindo o módulo de detecção de símbolos, o reconhecimento e a associação de *tags*, a geração da representação intermediária dos dados, a aplicação de validação e o mapeamento das informações para objetos de engenharia. A implementação da camada de integração é apresentada a partir do estudo de caso desenvolvido no Siemens TIA Portal, plataforma utilizada nos projetos analisados na GreyLogix.

A análise dos resultados considera tanto o desempenho do modelo de detecção quanto o funcionamento integrado das demais etapas da solução. Para isso, são examinadas as métricas obtidas no treinamento e na avaliação do modelo, bem como as verificações funcionais realizadas sobre o reconhecimento textual, a validação dos registros, o mapeamento dos instrumentos e a geração dos objetos no ambiente de engenharia adotado no estudo de caso.

5.1 VISÃO GERAL DA IMPLEMENTAÇÃO

A implementação foi organizada em dois conjuntos principais de módulos. O primeiro, desenvolvido em Python, concentra o pré-processamento dos diagramas, incluindo a conversão dos documentos, a divisão das imagens em recortes, a detecção dos símbolos, o reconhecimento das *tags*, a geração da representação intermediária e a validação técnica dos resultados. Essas etapas produzem dados estruturados sem manipular diretamente o ambiente de programação do controlador. O segundo conjunto corresponde à camada específica de geração e integração adotada no estudo de caso. Desenvolvida em C# e .NET Framework, essa camada utiliza os dados validados para classificar os instrumentos, gerar arquivos no formato SIMATIC ML e importá-los no Siemens TIA Portal por meio da interface TIA Portal Openness. A comunicação entre os dois conjuntos ocorre por meio de arquivos em formato *JSON*.

O fluxo tem início com o recebimento de um diagrama P&ID em formato PDF. Como o modelo de detecção utilizado opera sobre imagens, cada página do documento é convertida para uma imagem com resolução adequada à preservação dos símbolos e dos textos. Em seguida, as páginas são divididas em recortes parcialmente sobrepostos, permitindo que os elementos de interesse ocupem uma área maior na imagem fornecida ao modelo e reduzindo a perda de detalhes associada ao redimensionamento da página completa.

Os recortes são processados por um modelo da família YOLO, treinado para identificar as duas classes de símbolos definidas no escopo do trabalho. Para cada ocorrência, o modelo retorna a classe prevista, o valor de confiança e as coordenadas da caixa delimitadora. Como um mesmo símbolo pode aparecer em mais de um recorte

nas regiões de sobreposição, as coordenadas locais são convertidas para o sistema de referência da página original e as detecções duplicadas são removidas antes da continuidade do processamento.

Após a consolidação das detecções, as caixas delimitadoras orientam a etapa de reconhecimento textual. Para cada símbolo identificado, é definida uma região de busca ao seu redor, posteriormente encaminhada ao módulo de OCR. O texto reconhecido nessa região é tratado como candidato à *tag* do instrumento, associando a localização do símbolo à sua identificação textual no diagrama.

Os resultados da detecção e do reconhecimento textual são armazenados em um arquivo *JSON*. Para cada elemento identificado, são registrados a classe do símbolo, o valor de confiança, a *tag* reconhecida, as coordenadas absolutas da caixa delimitadora e a página de origem. Essa representação intermediária separa o processamento visual das rotinas de engenharia e estabelece a interface entre os módulos desenvolvidos em Python e a aplicação desenvolvida em C#.

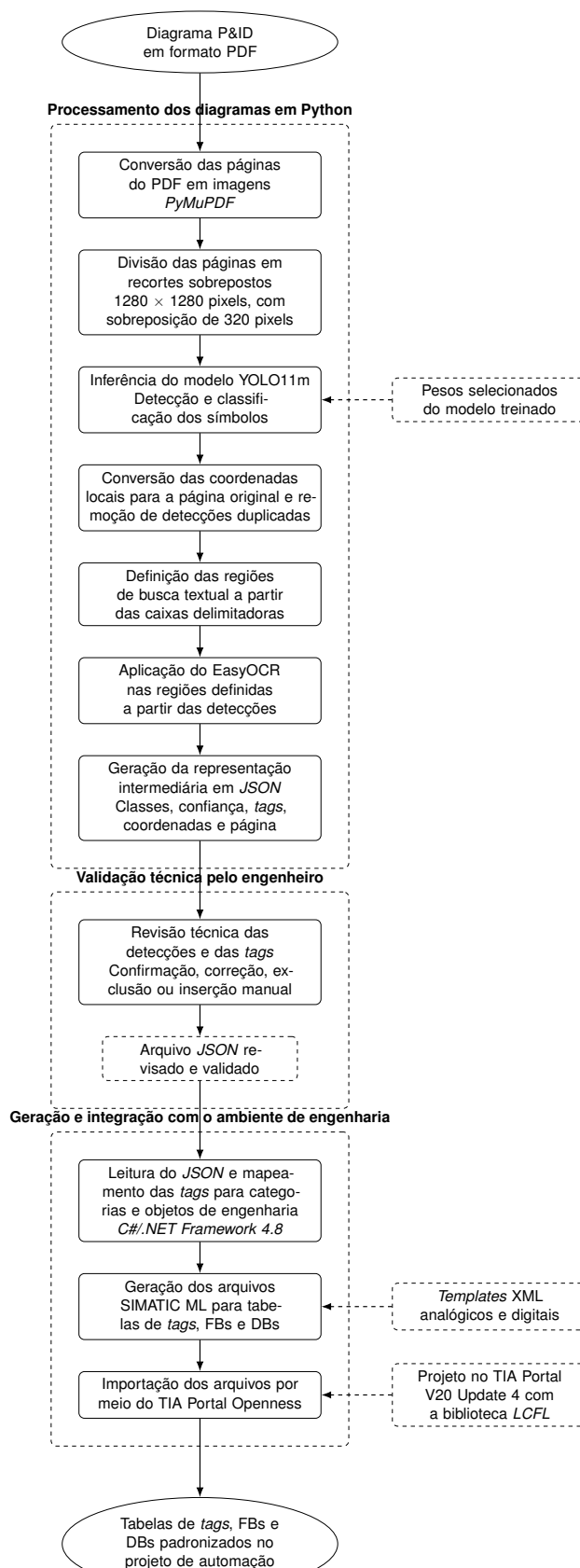
Antes da geração dos objetos no TIA Portal, os registros extraídos passam por uma etapa de validação técnica. A aplicação de revisão apresenta cada detecção juntamente com a região correspondente do diagrama, permitindo confirmar o registro, corrigir a *tag*, alterar a classe, excluir identificações incorretas ou inserir manualmente símbolos não localizados pelo modelo. Ao término da revisão, é gerada uma nova versão do arquivo *JSON*, contendo o conjunto de dados validado pelo engenheiro.

A aplicação desenvolvida em C# e .NET Framework lê o arquivo *JSON* revisado, interpreta as *tags* dos instrumentos e classifica os elementos conforme as categorias previstas na implementação. A partir dessa classificação, são selecionados modelos de arquivos XML utilizados como base para a geração das tabelas de *tags*, dos blocos de configuração, dos FBs e dos respectivos DBs. Esses arquivos seguem o formato SIMATIC ML e fazem referência a blocos e tipos de dados padronizados da biblioteca LCFL, previamente disponíveis no projeto utilizado para a integração.

Por fim, a aplicação estabelece comunicação com o Siemens TIA Portal V20 por meio do TIA Portal Openness. Os arquivos SIMATIC ML são importados para o projeto, resultando na criação das estruturas previstas para os instrumentos detectados e validados. Essa etapa materializa a camada de integração definida para o estudo de caso da GreyLogix. Dessa forma, o fluxo implementado transforma as informações extraídas do diagrama em uma estrutura inicial de programação compatível com a plataforma Siemens, mantendo a validação humana antes da incorporação dos resultados ao projeto.

A Figura 8 detalha a implementação da arquitetura conceitual, apresentando as bibliotecas, os parâmetros de processamento, os formatos de dados e os recursos empregados no protótipo desenvolvido para o estudo de caso Siemens.

Figura 8 – Visão geral da implementação desenvolvida.



Fonte: Elaborado pelo autor.

5.2 DESENVOLVIMENTO E AVALIAÇÃO DO MÓDULO DE DETECÇÃO

O módulo de detecção foi desenvolvido para localizar e classificar os símbolos de instrumentação definidos no escopo deste trabalho. Seu desenvolvimento envolveu a seleção do conjunto de dados, a adequação das classes e anotações, a configuração e o treinamento do modelo YOLO, além da avaliação de seu desempenho e de sua aplicação nos diagramas utilizados nos testes.

5.2.1 Seleção e organização do conjunto de dados

Para o treinamento do modelo de detecção, foi utilizado um conjunto de dados público disponibilizado na plataforma Roboflow Universe, composto por imagens extraídas de diagramas P&ID e anotações de diferentes símbolos técnicos (Roboflow Universe, 2023). A adoção de uma base pública previamente rotulada permitiu reduzir o tempo destinado à preparação dos dados e concentrar o desenvolvimento no treinamento e na integração do modelo. A construção de um conjunto próprio exigiria a seleção de diagramas industriais, a anonimização de informações confidenciais e a rotulagem manual dos símbolos, etapas que demandariam um período elevado dentro do cronograma do trabalho.

O conjunto original abrangia diferentes categorias de elementos presentes em diagramas P&ID. Entretanto, como o trabalho foi delimitado à identificação das duas representações de instrumentação utilizadas nas etapas posteriores do fluxo, foram mantidas somente as imagens e anotações relacionadas às classes selecionadas. Essa filtragem tornou o conjunto de dados compatível com o escopo do estudo de caso e permitiu concentrar o desenvolvimento na avaliação da cadeia completa de detecção, reconhecimento textual, validação e geração dos objetos de engenharia.

Após a adaptação do *dataset*, a base utilizada reuniu 2378 imagens, distribuídas entre os subconjuntos de treinamento, validação e teste. O subconjunto de treinamento foi empregado no ajuste dos parâmetros internos do modelo, enquanto o conjunto de validação permitiu acompanhar seu comportamento ao longo das épocas de treinamento. As imagens de teste foram mantidas separadas e utilizadas somente na avaliação final, de modo a verificar o desempenho do modelo sobre dados que não participaram de seu ajuste. A distribuição das imagens é apresentada na Tabela 10.

Tabela 10 – Distribuição das imagens entre os subconjuntos utilizados.

Subconjunto	Quantidade de imagens	Proporção aproximada
Treinamento	1664	70,0%
Validação	482	20,0%
Teste	232	10,0%
Total	2378	100,0%

Fonte: Elaborado pelo autor.

Para o treinamento, os arquivos foram organizados conforme a estrutura esperada pela biblioteca Ultralytics, com diretórios separados para imagens e anotações em cada subconjunto. As anotações foram mantidas em arquivos de texto no formato utilizado pelo YOLO, e o arquivo `data.yaml` foi configurado com os caminhos dos subconjuntos e os nomes das classes consideradas. Também foi desenvolvido um script auxiliar para visualizar as imagens com suas respectivas caixas delimitadoras, permitindo conferir a correspondência entre os símbolos representados e as anotações fornecidas ao modelo antes do treinamento.

5.2.2 Definição das classes e preparação das anotações

A partir do conjunto de dados original, foram selecionadas as duas classes de símbolos compatíveis com o escopo definido para a solução. As classes `instrument tag` e `instrument DCS`, bem como sua correspondência com as representações da ANSI/ISA-5.1, foram apresentadas anteriormente na Tabela 8. Para preservar a estrutura original do conjunto de dados, os nomes das classes foram mantidos no arquivo `data.yaml` e nos arquivos de anotação, sendo a classe `instrument tag` associada ao índice 0 e a classe `instrument DCS` ao índice 1.

As anotações foram mantidas no formato utilizado pelo YOLO, no qual cada imagem possui um arquivo de texto correspondente. Cada linha desse arquivo representa uma instância anotada e contém o identificador da classe, as coordenadas normalizadas do centro da caixa delimitadora e suas dimensões relativas à largura e à altura da imagem. A estrutura de cada registro é dada por:

```
classe    x_centro    y_centro    largura    altura
```

Os valores das coordenadas e das dimensões são normalizados no intervalo entre 0 e 1, o que permite utilizar as anotações independentemente da resolução original da imagem. Também foi verificada a correspondência entre os arquivos de imagem e seus respectivos arquivos de anotação nos subconjuntos de treinamento, validação e teste.

Para conferir a consistência das anotações, foi utilizado um script auxiliar que realiza a leitura do arquivo `data.yaml`, converte as coordenadas normalizadas para as dimensões reais da imagem e apresenta visualmente as caixas delimitadoras acompanhadas dos nomes das classes. Essa inspeção permitiu verificar se as caixas envolviam adequadamente os símbolos e se os identificadores atribuídos correspondiam às representações presentes nas imagens.

Como uma mesma imagem pode conter mais de um símbolo, a composição do conjunto de dados foi analisada com base no número de instâncias anotadas, e não apenas na quantidade de imagens. Essa análise permite avaliar a representatividade de cada classe nos subconjuntos utilizados durante o treinamento e a avaliação do modelo. A Tabela 11 apresenta a distribuição das instâncias entre treinamento, validação e teste.

Tabela 11 – Distribuição das instâncias anotadas por classe nos subconjuntos utilizados.

Classe	Treinamento	Validação	Teste	Total
instrument tag	1183	391	156	1730
instrument DCS	3137	889	416	4442
Total	4320	1280	572	6172

Fonte: Elaborado pelo autor.

Observa-se uma predominância da classe `instrument DCS`, que corresponde a aproximadamente 72% das instâncias anotadas, enquanto a classe `instrument tag` representa cerca de 28%. Essa diferença não provém de um desbalanceamento artificial realizado durante a preparação dos dados, mas sim da própria composição dos diagramas P&ID, nos quais símbolos associados a funções compartilhadas de indicação e controle aparecem com maior frequência. A proporção entre as classes permanece semelhante nos subconjuntos de treinamento, validação e teste, indicando que a divisão preservou a distribuição geral do conjunto. Esse desbalanceamento foi considerado na análise dos resultados, especialmente na comparação das métricas obtidas para cada classe.

5.2.3 Configuração e treinamento do modelo YOLO

O treinamento foi realizado em um ambiente JupyterLab executado em servidor disponibilizado com o apoio do Laboratório de Processamento de Imagens e Computação Gráfica (LAPIX) da UFSC, utilizando a biblioteca Ultralytics em Python. Como ponto de partida, foi selecionado o modelo pré-treinado `yolo11m.pt`, correspondente à variante média da arquitetura YOLO11. Essa configuração foi adotada por oferecer maior capacidade de representação do que as variantes menores, mantendo um

custo computacional compatível com os recursos disponíveis e com a necessidade de identificar símbolos de pequenas dimensões nos diagramas.

A inicialização com pesos previamente treinados permitiu aplicar transferência de aprendizado, aproveitando características visuais já aprendidas pelo modelo e adaptando sua camada de detecção às duas classes definidas para o trabalho. Durante a inicialização, a saída originalmente configurada para um conjunto maior de categorias foi substituída por uma saída correspondente às classes `instrument tag` e `instrument DCS`.

As imagens foram processadas com resolução de entrada de 1280×1280 pixels. Esse valor foi adotado para preservar detalhes dos símbolos, que frequentemente ocupam uma pequena região dos diagramas, e foi mantido posteriormente na etapa de inferência. O treinamento foi configurado para até 150 épocas, com parada antecipada caso não houvesse melhora nas métricas de validação durante 20 épocas consecutivas. O critério de parada antecipada não foi acionado, de modo que o processo completou as 150 épocas previstas.

O tamanho do lote foi definido pelo valor -1, permitindo que a biblioteca Ultralytics realizasse o ajuste automático de acordo com a memória disponível na GPU. O treinamento utilizou o otimizador AdamW, com taxa de aprendizado inicial de 0,001, decaimento de pesos de 0,0005 e precisão mista automática habilitada. Também foram aplicadas técnicas de aumento de dados, incluindo pequenas rotações, translações, alterações de escala e composição em mosaico. Os espelhamentos horizontal e vertical foram desabilitados, evitando a introdução de orientações que não correspondessem à organização original dos símbolos e textos nos diagramas. Os principais parâmetros utilizados são apresentados na Tabela 12.

Tabela 12 – Principais parâmetros utilizados no treinamento do modelo de detecção.

Parâmetro	Configuração
Modelo inicial	yolo11m.pt
Número de classes	2
Resolução de entrada	1280 × 1280 pixels
Número máximo de épocas	150
Paciência para parada antecipada	20 épocas
Tamanho do lote	Ajuste automático
Otimizador	AdamW
Taxa de aprendizado inicial	0,001
Dcaimento de pesos	0,0005
Precisão mista automática	Habilitada
Aumento de dados	Rotação, translação, escala e mosaico
Espelhamento horizontal e vertical	Desabilitado

Fonte: Elaborado pelo autor.

O processamento foi executado com Python 3.11.10, Ultralytics 8.3.33 e PyTorch 2.5.1, utilizando uma GPU NVIDIA H100 com 80 GB de memória, disponibilizada por meio da infraestrutura computacional do LAPIX. O treinamento das 150 épocas foi concluído em aproximadamente 1 hora e 25 minutos. Ao final do processo, foram preservados os pesos correspondentes ao melhor desempenho no conjunto de validação. O arquivo `best.pt` foi selecionado para as etapas posteriores de avaliação e utilização no fluxo de inferência.

5.2.4 Avaliação do modelo de detecção

Após a conclusão do treinamento, o modelo foi avaliado sobre o subconjunto de teste, composto por 232 imagens e 572 instâncias anotadas das duas classes consideradas. Esse subconjunto não participou do ajuste dos parâmetros do modelo, sendo utilizado para verificar seu desempenho em imagens reservadas para a avaliação final.

A avaliação foi realizada com as métricas de precisão, revocação, $mAP@0,5$ e $mAP@0,5:0,95$, conforme os critérios apresentados na fundamentação teórica. Os resultados obtidos para o conjunto de teste são apresentados na Tabela 13.

Tabela 13 – Resultados do modelo de detecção no conjunto de teste.

Classe	Imagens	Instâncias	Precisão	Revocação	mAP@0,5	mAP@0,5:0,95
Geral	232	572	0,9846	0,9940	0,9947	0,9866
instrument tag	123	156	0,979	1,000	0,995	0,983
instrument DCS	206	416	0,990	0,988	0,994	0,990

Fonte: Elaborado pelo autor.

A quantidade de imagens indicada para cada classe corresponde ao número de imagens nas quais essa classe está presente. Como uma mesma imagem pode conter símbolos das duas categorias, a soma desses valores é superior ao total de imagens do conjunto de teste.

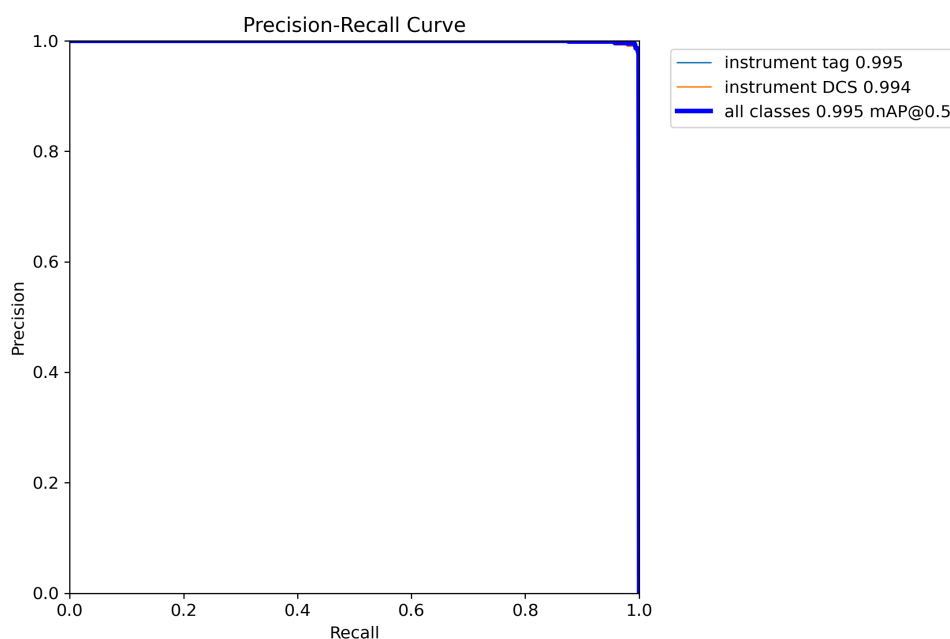
Considerando o desempenho geral, o modelo alcançou precisão de 98,46% e revocação de 99,40%. Esses valores indicam baixa ocorrência de detecções incorretas e pequena perda de instâncias anotadas. A mAP@0,5 foi de 99,47%, enquanto a métrica calculada entre os limiares de 0,5 e 0,95 atingiu 98,66%, indicando desempenho consistente também sob critérios mais rigorosos de localização das caixas delimitadoras.

Os resultados por classe apresentaram comportamento semelhante. A classe *instrument tag*, embora menos representada no conjunto de dados, obteve revocação de 100,00% e mAP@0,5 de 99,50%. Para a classe *instrument DCS*, foram obtidas precisão de 99,00%, revocação de 98,80% e mAP@0,5 de 99,40%. Dessa forma, o desbalanceamento observado na distribuição das instâncias não resultou, no conjunto de teste, em uma diferença expressiva de desempenho entre as classes.

A proximidade entre os valores de mAP@0,5 e mAP@0,5:0,95 sugere que o modelo apresentou boa correspondência espacial entre as caixas previstas e as anotações de referência. Esse comportamento é relevante para a aplicação proposta, pois a localização adequada dos símbolos é necessária para orientar as etapas posteriores de reconhecimento textual e associação com *tags*.

A Figura 9 apresenta as curvas de precisão e revocação para as duas classes avaliadas. As curvas indicam que o modelo manteve elevada precisão mesmo quando o limiar foi ajustado para recuperar uma parcela maior dos símbolos anotados, sem aumento expressivo da ocorrência de falsos positivos no conjunto de teste.

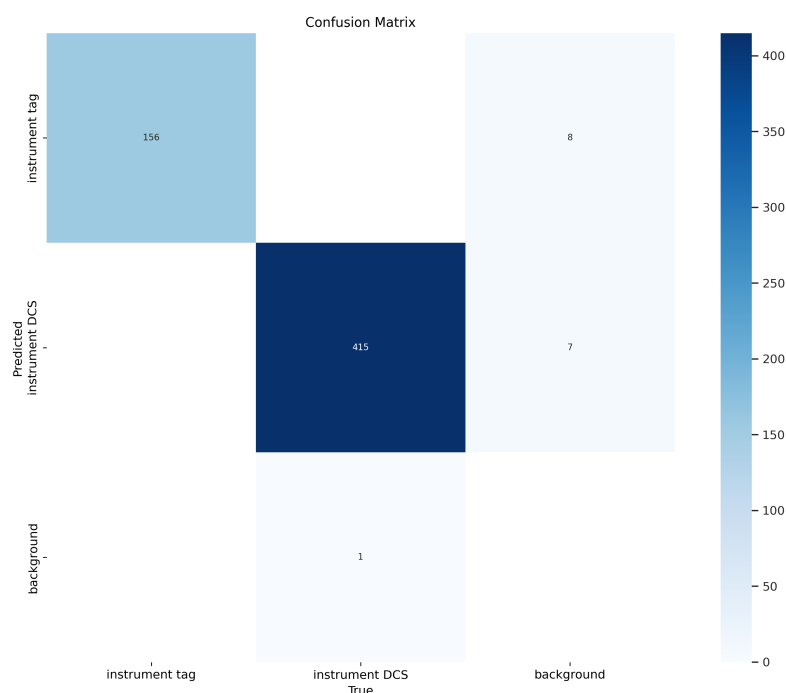
Figura 9 – Curvas de precisão e revocação do modelo de detecção.



Fonte: Elaborado pelo autor.

A matriz de confusão apresentada na Figura 10 permite observar a distribuição dos acertos e erros no conjunto avaliado. Foram corretamente identificadas 156 instâncias da classe `instrument tag` e 415 instâncias da classe `instrument DCS`. Não foram observadas trocas de classificação entre as duas categorias, indicando que os erros não estiveram associados à confusão direta entre os símbolos selecionados. As ocorrências incorretas concentraram-se nas relações com o fundo da imagem, com oito registros associados à classe `instrument tag`, sete à classe `instrument DCS` e uma instância de `instrument DCS` não detectada.

Figura 10 – Matriz de confusão do modelo no conjunto de teste.



Fonte: Elaborado pelo autor.

Embora os resultados no conjunto de teste indiquem desempenho elevado, sua interpretação deve considerar que as imagens avaliadas pertencem à mesma base utilizada no desenvolvimento do modelo. A aplicação em diagramas completos pode introduzir variações adicionais de resolução, escala, simbologia, densidade visual e organização. Por esse motivo, o comportamento do detector em documentos da GreyLogix é analisado posteriormente nos testes integrados da solução.

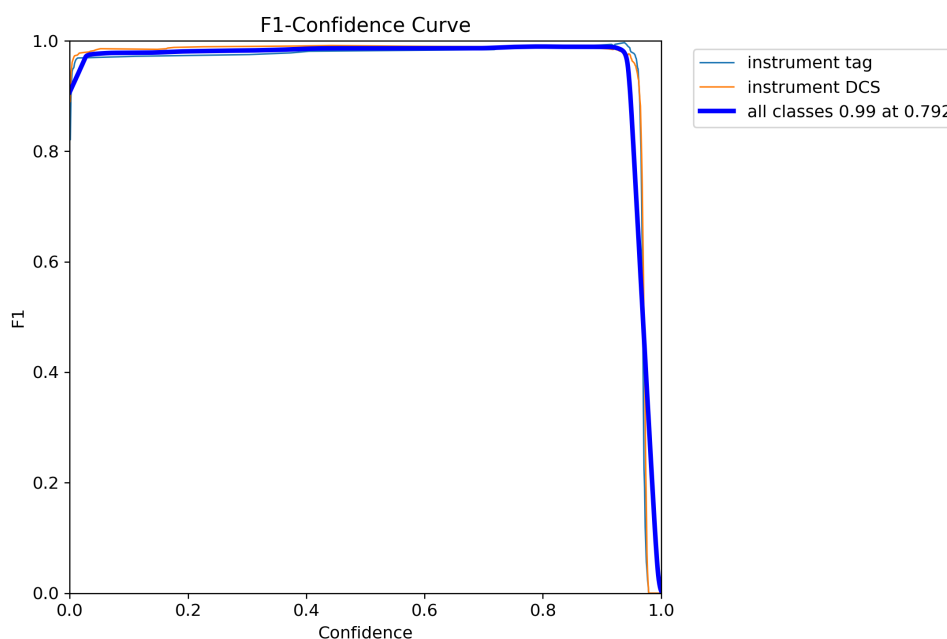
5.2.5 Exportação e aplicação do modelo na inferência

Após a avaliação, os pesos armazenados no arquivo `best.pt` foram exportados para o formato OpenVINO. Essa exportação foi adotada em função do ambiente utilizado nas etapas posteriores de desenvolvimento e teste do fluxo completo. Embora o treinamento tenha sido realizado com apoio da infraestrutura computacional do LAPIX, a aplicação integrada foi executada em um notebook com capacidade gráfica limitada. Assim, a conversão para OpenVINO teve como objetivo favorecer a execução da inferência em CPU, tornando o uso do modelo mais compatível com os recursos disponíveis nesse ambiente.

Na aplicação de inferência, o modelo foi carregado com a mesma resolução de entrada utilizada no treinamento, definida em 1280×1280 pixels. Foram definidos um limiar mínimo de confiança de 0,8 e um limiar de IoU de 0,5 para o tratamento das caixas sobrepostas durante a predição realizada pelo modelo.

A definição do limiar de confiança foi apoiada pela curva F1 apresentada na Figura 11. O maior valor de F1, aproximadamente 0,99, foi obtido para uma confiança de 0,792. Por esse motivo, adotou-se o valor de 0,8, próximo ao ponto de maior equilíbrio entre precisão e revocação observado na avaliação. Detecções com confiança inferior a esse valor são desconsideradas antes das etapas seguintes do processamento.

Figura 11 – Variação da métrica F1 em função do limiar de confiança.



Fonte: Elaborado pelo autor.

Para aplicar o modelo em diagramas completos, cada página do arquivo PDF é inicialmente convertida em imagem com resolução de 300 dpi. Em seguida, as páginas são divididas em recortes de 1280×1280 pixels, com sobreposição de 320 pixels entre regiões adjacentes. Essa estratégia reduz a necessidade de redimensionar a página completa para a entrada do modelo e contribui para preservar detalhes de símbolos que ocupam pequenas regiões do diagrama. Para cada recorte, o modelo retorna a classe prevista, o valor de confiança e as coordenadas da caixa delimitadora.

Como as coordenadas retornadas pelo modelo são relativas a cada recorte, elas são convertidas para o sistema de referência da página original por meio da adição dos deslocamentos horizontal e vertical correspondentes. Esse procedimento permite reunir as detecções obtidas em diferentes recortes e reconstruir suas posições no diagrama completo.

A sobreposição entre os recortes reduz a possibilidade de que símbolos localizados nas bordas sejam parcialmente excluídos durante o processamento. Por outro lado, essa mesma estratégia pode fazer com que um símbolo seja detectado em mais de um recorte. Para consolidar os resultados, foi aplicada uma etapa adicional de supressão de não máximos entre as detecções da mesma página, utilizando limiar de

IoU de 0,4. Nos casos em que duas caixas apresentaram sobreposição suficiente para representar o mesmo símbolo, foi mantida a detecção com maior valor de confiança.

Após a consolidação, cada símbolo detectado passa a ser representado por sua classe, valor de confiança, página de origem e coordenadas absolutas da caixa delimitadora. Essas informações são encaminhadas às etapas de reconhecimento textual e associação das *tags*, descritas na seção seguinte.

5.3 RECONHECIMENTO TEXTUAL E ASSOCIAÇÃO ENTRE SÍMBOLOS E TAGS

Após a localização dos símbolos de instrumentação, foi necessário identificar a *tag* associada a cada detecção. Para essa finalidade, foi utilizado um módulo de OCR orientado pelas caixas delimitadoras fornecidas pelo modelo YOLO. Em vez de realizar o reconhecimento textual sobre toda a página, o processamento foi aplicado às regiões associadas aos símbolos detectados. Essa estratégia restringe a leitura às áreas de interesse e reduz a interferência de textos pertencentes a tubulações, equipamentos, notas e outros elementos presentes no diagrama.

Essa etapa foi organizada em três operações principais: definição das regiões encaminhadas ao OCR, reconhecimento dos caracteres e tratamento do texto obtido para geração de uma *tag* candidata. Como cada região de busca é derivada de uma detecção específica, o resultado textual pode ser associado diretamente à classe, ao valor de confiança e às coordenadas do símbolo correspondente.

5.3.1 Definição das regiões de busca

As caixas delimitadoras produzidas pelo detector foram utilizadas como referência para definir as regiões submetidas ao reconhecimento textual. Para cada símbolo identificado, as coordenadas absolutas da caixa foram recuperadas no sistema de referência da página original, permitindo recortar a área correspondente diretamente na imagem do diagrama.

Para reduzir o risco de perda de caracteres nas bordas do recorte, foi acrescentada uma margem de 10 pixels em cada lado da caixa delimitadora. Essa ampliação compensa pequenas diferenças entre os limites previstos pelo modelo e a área efetivamente ocupada pelo texto associado ao instrumento. Considerando uma caixa original definida pelas coordenadas (x_1, y_1, x_2, y_2) , a região encaminhada ao OCR foi definida por:

$$\begin{aligned} x'_1 &= \max(0, x_1 - 10), & y'_1 &= \max(0, y_1 - 10), \\ x'_2 &= \min(W, x_2 + 10), & y'_2 &= \min(H, y_2 + 10), \end{aligned}$$

em que W e H representam, respectivamente, a largura e a altura da imagem. A limitação das coordenadas pelas dimensões da página impede que a região ampli-

ada ultrapasse os limites válidos da imagem, especialmente em símbolos localizados próximos às bordas do diagrama.

Cada recorte foi extraído da imagem original e convertido para escala de cinza antes de ser encaminhado ao módulo de OCR. Essa conversão reduz informações de cor que não contribuem diretamente para a leitura dos caracteres, preservando os contornos e as diferenças de intensidade entre o texto e o fundo.

A definição de uma região individual para cada símbolo também simplifica a associação entre detecção e texto reconhecido. Como o recorte é gerado a partir da própria caixa delimitadora, o resultado do OCR já permanece vinculado ao símbolo que originou a região de busca. Com isso, reduz-se a necessidade de relacionar posteriormente textos detectados em toda a página a diferentes elementos gráficos do diagrama.

5.3.2 Aplicação do módulo de OCR

O reconhecimento dos caracteres foi realizado com a biblioteca EasyOCR. O leitor foi configurado com o modelo de idioma inglês, pois as *tags* consideradas são compostas por letras do alfabeto latino, algarismos e caracteres separadores, sem depender da interpretação de palavras em língua portuguesa. A execução do módulo utiliza processamento gráfico quando há uma GPU compatível disponível e, nos demais casos, é realizada pela CPU.

Cada região definida na etapa anterior foi encaminhada individualmente ao método de reconhecimento da biblioteca. Foi mantida a saída detalhada do OCR, de modo que, além do texto identificado, fossem obtidos o valor de confiança e a localização de cada fragmento reconhecido. Essa configuração é relevante porque os caracteres de uma mesma *tag* podem aparecer distribuídos em mais de uma linha no interior do símbolo, fazendo com que uma única região de busca produza múltiplos fragmentos textuais.

A busca foi restringida aos caracteres esperados nas identificações dos instrumentos: letras maiúsculas, algarismos, ponto, hífen e espaço. Essa limitação reduz a ocorrência de caracteres sem uso nas *tags* consideradas pela solução. A lista de caracteres permitidos foi definida como:

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.-_

Também foi utilizado um fator de ampliação de 2,5 durante o reconhecimento, com o objetivo de favorecer a leitura de caracteres de pequenas dimensões presentes nos diagramas. O ajuste automático de contraste foi habilitado, e o limiar interno utilizado para a identificação das regiões textuais foi definido em 0,5. Os principais parâmetros empregados são apresentados na Tabela 14.

Tabela 14 – Principais parâmetros utilizados no reconhecimento textual.

Parâmetro	Configuração
Biblioteca	EasyOCR
Imagem de entrada	Escala de cinza
Caracteres permitidos	Letras maiúsculas, algarismos, ponto, hífen e espaço
Fator de ampliação	2,5
Ajuste de contraste	Habilitado
Limiar de identificação textual	0,5
Saída detalhada	Texto, confiança e coordenadas

Fonte: Elaborado pelo autor.

Quando nenhum conteúdo textual era identificado na região analisada, o campo destinado à *tag* permanecia vazio para posterior verificação pelo engenheiro. Nos demais casos, os fragmentos reconhecidos e seus respectivos valores de confiança eram encaminhados à etapa de tratamento textual, responsável por formar uma *tag* candidata e associá-la ao símbolo detectado.

5.3.3 Associação das tags aos símbolos detectados

Como as regiões encaminhadas ao OCR foram definidas a partir das caixas delimitadoras dos símbolos detectados, os textos reconhecidos em cada recorte permaneceram vinculados à respectiva detecção. Assim, a associação entre símbolo e texto foi realizada pelo próprio recorte utilizado no reconhecimento, sem a necessidade de uma busca posterior por proximidade em toda a página.

Em uma mesma região, o EasyOCR pode retornar mais de um fragmento textual, especialmente quando a identificação alfabética e a numeração do instrumento aparecem separadas ou distribuídas em linhas distintas. Por esse motivo, os fragmentos reconhecidos em cada recorte foram concatenados em uma única cadeia de caracteres, respeitando a ordem retornada pelo reconhecedor e inserindo um espaço entre eles. Dessa forma, por exemplo, os fragmentos PIT e 02101.1 passam a formar a cadeia PIT 02101.1.

Após essa concatenação, foi aplicada uma expressão regular para localizar sequências compatíveis com o formato esperado para as *tags* utilizadas em soluções da GreyLogix. O padrão adotado foi:

$$([A-Z]\{1,4\})?[\s\-_]*\d\{2,5\}(?:[\.\-_]\d\{1,4\})^*$$

Esse padrão admite um prefixo opcional formado por uma a quatro letras, seguido por espaços, hífen ou sublinhado e por uma sequência principal de dois a cinco

algarismos. Também são aceitos segmentos numéricos adicionais separados por ponto ou hífen. Com isso, podem ser reconhecidas identificações compostas por letras e números, incluindo variações com subdivisões numéricas.

Além da compatibilidade com o padrão definido, foi considerado o valor de confiança retornado pelo módulo de OCR. Para cada região, foi calculada a média aritmética das confianças atribuídas aos fragmentos reconhecidos. A *tag* candidata foi aceita somente quando uma correspondência com a expressão regular foi encontrada e a confiança média foi superior a 0,4. O valor armazenado corresponde ao trecho da cadeia que apresentou correspondência com o padrão, desconsiderando caracteres adicionais eventualmente reconhecidos na mesma região.

Quando o módulo não identificava fragmentos, a expressão regular não encontrava uma sequência compatível ou a confiança média não atendia ao limite estabelecido, a *tag* era mantida como uma cadeia vazia. Esses casos não eram descartados, pois o símbolo detectado continuava registrado com sua classe, confiança e coordenadas. A ausência ou eventual incorreção da *tag* poderia ser corrigida posteriormente na etapa de validação técnica pelo engenheiro.

Ao final do processamento, cada detecção passou a reunir a classe atribuída pelo modelo, o valor de confiança, as coordenadas absolutas, a página de origem e a *tag* reconhecida. Esse registro conjunto estabelece a ligação entre a informação gráfica e a identificação textual do instrumento, formando a base para a representação intermediária em formato *JSON*.

5.4 ESTRUTURAÇÃO DOS DADOS E APLICAÇÃO DE VALIDAÇÃO

As informações produzidas pelos módulos de detecção e reconhecimento textual foram organizadas em uma representação intermediária antes de serem utilizadas nas etapas de engenharia. Essa separação evita que a aplicação responsável pelo processamento das imagens manipule diretamente o projeto de automação e permite que os resultados sejam revisados e corrigidos antes de sua importação para o TIA Portal.

5.4.1 Geração da representação em JSON

Os resultados consolidados foram armazenados em um arquivo no formato *JSON*. Esse formato foi adotado por permitir a representação estruturada dos dados extraídos, manter uma organização legível e possibilitar sua utilização tanto pela aplicação desenvolvida em Python quanto pelo módulo de integração implementado em C#.

O arquivo gerado contém informações gerais sobre o documento processado e uma lista com os símbolos identificados. No cabeçalho, são registrados o nome do

arquivo de origem, as dimensões da imagem processada e o conjunto de detecções obtidas. Para cada símbolo, são armazenados a classe atribuída pelo modelo, o valor de confiança, a *tag* reconhecida pelo módulo de OCR, as coordenadas absolutas da caixa delimitadora e a página do documento em que o elemento foi localizado.

As coordenadas da caixa delimitadora são representadas pelos valores x_1 , y_1 , x_2 e y_2 , correspondentes aos limites superior esquerdo e inferior direito da região detectada. O uso de coordenadas absolutas preserva a posição do símbolo no sistema de referência da página original, permitindo que a aplicação de validação recupere posteriormente a região correspondente no diagrama.

Quando uma *tag* não é reconhecida ou não atende aos critérios definidos na etapa de OCR, o campo *tag* permanece vazio. O registro do símbolo, entretanto, é mantido no arquivo juntamente com sua classe, confiança e localização, possibilitando que a identificação seja preenchida ou corrigida posteriormente durante a validação técnica.

Essa representação intermediária estabelece a interface entre a etapa de extração e as etapas posteriores de validação, mapeamento e integração. Dessa forma, os dados extraídos do diagrama deixam de depender diretamente da saída bruta dos modelos computacionais e passam a ser organizados em um formato adequado ao fluxo de engenharia desenvolvido.

5.4.2 Desenvolvimento da aplicação de validação

A aplicação de validação foi desenvolvida para permitir a revisão técnica dos resultados produzidos pelos módulos de detecção e OCR antes de sua utilização nas etapas de engenharia. Essa etapa permite que o engenheiro possa corrigir falhas de classificação, leituras incorretas de *tags* e detecções indevidas antes da geração dos objetos no TIA Portal.

A aplicação foi implementada em Python, utilizando recursos do OpenCV para exibição das imagens e seleção de regiões, combinados com comandos apresentados no terminal. Inicialmente, são carregados o documento processado e o arquivo *JSON* gerado pelo fluxo automático. Em seguida, cada detecção é apresentada individualmente, acompanhada de uma região ampliada do diagrama, permitindo visualizar o símbolo e os elementos próximos necessários à sua interpretação.

Para cada registro, são exibidas a classe atribuída pelo modelo, a confiança da detecção e a *tag* obtida pelo módulo de OCR. A partir da inspeção visual, o usuário pode confirmar o registro, corrigir a *tag*, alterar a classe do símbolo ou excluir a detecção. A exclusão é utilizada para remover falsos positivos, enquanto a correção da *tag* e da classe permite ajustar erros produzidos pelas etapas automáticas. As modificações são realizadas diretamente sobre a estrutura carregada do arquivo *JSON*, preservando as coordenadas e demais informações associadas ao símbolo.

Após a revisão das detecções automáticas, a aplicação permite acrescentar manualmente símbolos que não tenham sido localizados pelo modelo. Para isso, a página completa é apresentada ao usuário, que seleciona a região correspondente ao símbolo ausente e informa sua classe e sua *tag*. O novo registro é então inserido na lista de detecções com coordenadas absolutas e página de origem, mantendo o mesmo padrão dos elementos identificados automaticamente.

Ao término do processo, os registros confirmados, corrigidos e adicionados manualmente são gravados em um novo arquivo *JSON*. A geração de uma versão separada preserva o arquivo original produzido pelo fluxo automático e permite distinguir os dados brutos dos dados validados pelo engenheiro. A estrutura do arquivo revisado permanece compatível com a representação definida anteriormente, permitindo que o módulo desenvolvido em C# utilize os resultados sem alterações adicionais.

5.5 MAPEAMENTO E INTEGRAÇÃO NO ESTUDO DE CASO SIEMENS

Esta seção apresenta a implementação da camada de mapeamento e integração adotada no estudo de caso desenvolvido na GreyLogix. Após a validação dos resultados extraídos do diagrama, as informações registradas no arquivo *JSON* revisado foram utilizadas como entrada para a geração dos objetos do projeto de automação. Essa etapa foi implementada em uma aplicação desenvolvida em C# para o ambiente .NET Framework, responsável por interpretar as *tags* validadas, classificá-las conforme as regras definidas para o projeto e produzir arquivos compatíveis com o Siemens TIA Portal.

A escolha dessa plataforma está relacionada ao ambiente tecnológico utilizado nos projetos analisados na empresa e à disponibilidade de mecanismos programáticos para automação de tarefas de engenharia. O fluxo de integração foi estruturado em quatro operações principais: leitura e classificação das *tags*, geração dos arquivos SIMATIC ML a partir de modelos XML, utilização das estruturas padronizadas da biblioteca LCFL e importação dos objetos por meio do TIA Portal Openness. Com isso, os dados obtidos a partir do diagrama são convertidos em tabelas de *tags*, blocos de configuração, FBs e DBs, mantendo a etapa de visão computacional separada das rotinas específicas da plataforma Siemens (Siemens, 2026b, 2026a).

5.5.1 Leitura e classificação das tags validadas

A aplicação recebe como entrada o arquivo *JSON* produzido ao término da validação técnica. Inicialmente, o conteúdo do arquivo é convertido em objetos internos da aplicação, em um processo de desserialização. Essa conversão permite acessar individualmente a *tag*, a classe do símbolo, o valor de confiança, as coordenadas e a página de origem de cada registro. Para o mapeamento dos objetos de engenharia,

entretanto, a principal informação utilizada é a identificação textual do instrumento.

A classificação é realizada a partir das letras presentes na *tag*, considerando a identificação funcional de instrumentos apresentada na Tabela 3. Essa identificação, baseada na ANSI/ISA-5.1, distingue a primeira letra, utilizada para indicar a variável de processo ou variável iniciadora, das letras subsequentes, que representam as funções desempenhadas pelo instrumento. Dessa forma, o significado de uma letra depende da posição ocupada na identificação funcional. Na implementação desenvolvida, essa interpretação posicional foi utilizada para direcionar os instrumentos aos grupos de mapeamento previstos na aplicação.

As *tags* associadas a variáveis e funções tratadas como sinais contínuos foram direcionadas ao grupo de instrumentos analógicos. Nesse grupo foram incluídas, entre outras, identificações relacionadas a medições de temperatura, pressão, nível, vazão e corrente, bem como instrumentos cuja função indicada na *tag* está associada à indicação ou transmissão de sinais. Essa classificação permite selecionar os modelos de blocos, tipos de dados e estruturas correspondentes ao tratamento de sinais analógicos no TIA Portal.

As identificações relacionadas a estados discretos, chaves e posições foram direcionadas ao grupo de instrumentos digitais. Nesse caso, foram considerados principalmente registros associados a funções de chaveamento, segurança, posição ou acionamento. A separação entre instrumentos analógicos e digitais foi adotada para orientar a geração dos objetos correspondentes nas etapas seguintes da integração.

Também foram previstas regras para reconhecer *tags* relacionadas a funções de controle, elementos finais de controle e funções de saída, com base nas letras subsequentes C, V e Y. No caso da letra Y, sua ocorrência entre as letras subsequentes foi interpretada como uma função de saída, conforme a identificação funcional apresentada na Tabela 3. Na convenção de mapeamento adotada pela GreyLogix, essa condição foi utilizada para identificar e direcionar as saídas analógicas ao grupo correspondente. Essa regra não se aplica à ocorrência da letra Y na primeira posição, cujo significado é definido pelo usuário conforme a ANSI/ISA-5.1. Entretanto, na versão implementada, a geração automática foi concluída apenas para os grupos analógico e digital. Os demais grupos foram mantidos como possibilidades de expansão do mapeamento desenvolvido.

É importante observar que a classe gráfica atribuída pelo modelo YOLO não determina diretamente o objeto criado no TIA Portal. As classes `instrument tag` e `instrument DCS` descrevem a forma de representação do símbolo no diagrama, enquanto a definição do objeto de engenharia depende da interpretação funcional da *tag*. Assim, símbolos pertencentes à mesma classe visual podem resultar em objetos distintos, conforme a variável de processo, a função indicada e as regras de mapeamento adotadas.

Ao término da classificação, as *tags* validadas são agrupadas em listas de instrumentos analógicos e digitais. Essas listas constituem a entrada para as rotinas responsáveis pela geração das tabelas de *tags*, dos blocos de configuração, dos FBs e dos respectivos DBs, descritas nas subseções seguintes.

5.5.2 Desenvolvimento da aplicação em C#/.NET

A etapa de geração dos objetos de engenharia foi implementada em C#, utilizando o .NET Framework 4.8. Essa tecnologia foi adotada por ser compatível com as bibliotecas disponibilizadas pelo TIA Portal Openness, permitindo que uma aplicação externa acesse e manipule projetos do TIA Portal por meio das classes fornecidas pela Siemens.

A aplicação foi desenvolvida separadamente dos módulos de visão computacional executados em Python. A comunicação entre essas partes ocorre por meio do arquivo *JSON* revisado, de modo que a aplicação em C# não depende diretamente das bibliotecas utilizadas para detecção de símbolos ou reconhecimento textual. Essa separação permite executar o processamento do diagrama e a geração dos objetos de engenharia em ambientes distintos, desde que o arquivo intermediário preserve a estrutura definida. Essa organização concentra a dependência da plataforma Siemens na camada de integração. Assim, os módulos responsáveis pelo processamento dos diagramas, pelo reconhecimento textual e pela validação dos resultados não precisam manipular diretamente objetos do TIA Portal. A adaptação para outro ambiente de engenharia exigiria uma nova aplicação ou um novo componente de integração, além da adequação das regras de mapeamento e dos formatos utilizados pela plataforma de destino.

A execução da aplicação inicia-se com o carregamento do arquivo *JSON* revisado e com o estabelecimento da conexão com o TIA Portal. O conteúdo do arquivo é convertido em estruturas internas equivalentes aos campos definidos na representação intermediária, permitindo acessar individualmente os símbolos validados e suas respectivas *tags*. Em seguida, as identificações são submetidas às regras de classificação descritas anteriormente e distribuídas entre os grupos analógico e digital, que orientam a seleção dos modelos utilizados na geração dos arquivos SIMATIC ML.

A partir dessa classificação, a aplicação produz separadamente os arquivos correspondentes às tabelas de *tags*, aos blocos de configuração, aos FBs e aos DBs de cada categoria de instrumento. A organização dessas operações em métodos distintos mantém separadas as responsabilidades de leitura dos dados, classificação dos instrumentos, manipulação dos modelos XML e comunicação com o TIA Portal. Essa estrutura facilita a manutenção do código e permite acrescentar posteriormente novas categorias de instrumentos ou novos modelos de objetos sem modificar as etapas de detecção e validação.

Para utilizar o TIA Portal Openness, foram adicionadas ao projeto as referências às bibliotecas de engenharia fornecidas com a instalação do Siemens TIA Portal. Por meio dessas referências, a aplicação pode iniciar ou acessar uma instância do ambiente de engenharia, abrir o projeto e utilizar os serviços de importação disponíveis. Na implementação desenvolvida, essa integração foi configurada para o Siemens TIA Portal V20 Update 4.

5.5.3 Geração dos arquivos SIMATIC ML a partir de templates

Após a classificação das *tags* validadas, a aplicação utiliza modelos de arquivos XML previamente estruturados para gerar os objetos correspondentes aos instrumentos analógicos e digitais. Esses modelos seguem a organização do SIMATIC ML, formato empregado pelo TIA Portal para representar objetos de engenharia em operações de importação e exportação. A utilização do SIMATIC ML pertence à implementação específica da camada de integração. Em uma adaptação para outra plataforma, essa etapa precisaria ser substituída pelo formato ou pela interface de criação de objetos disponibilizada pelo respectivo ambiente de engenharia.

A utilização de *templates* evita que toda a estrutura XML seja construída diretamente pelo código. Cada modelo contém os elementos fixos necessários para representar determinada categoria de objeto, enquanto a aplicação substitui ou acrescenta os trechos que variam conforme as *tags* identificadas. Dessa forma, permanecem previamente definidos aspectos como a organização geral do documento, os tipos de objetos representados e as referências utilizadas pelo projeto, enquanto os nomes e as instâncias dos instrumentos são gerados dinamicamente.

Para cada grupo de instrumentos, foram utilizados modelos específicos para a geração das tabelas de *tags*, dos blocos de configuração, dos FBs e dos respectivos DBs. Assim, a aplicação percorre as listas de instrumentos analógicos e digitais formadas na etapa de classificação e produz arquivos SIMATIC ML separados para cada categoria. Na implementação desenvolvida, foram gerados arquivos destinados às tabelas de *tags*, aos blocos de configuração, aos blocos funcionais e aos blocos de dados de instância dos instrumentos analógicos e digitais.

Nas tabelas de *tags*, a aplicação também atribui endereços provisórios aos instrumentos. Para os sinais analógicos, a atribuição tem início no endereço %IW500, com incremento de dois bytes entre os registros. Para os sinais digitais, a sequência começa em %I10.0 e avança de forma individual entre os bits disponíveis.

Esses endereços não representam necessariamente as posições físicas definitivas dos sinais no controlador. Como o diagrama P&ID não contém as informações completas de endereçamento das entradas e saídas, os valores são utilizados como referências provisórias para permitir a criação inicial dos objetos. Sua substituição deve ser realizada posteriormente, com base na lista de I/Os e na configuração de hardware

definida para o projeto.

A geração separada dos arquivos analógicos e digitais mantém a organização adotada no projeto e facilita sua importação para os grupos correspondentes no TIA Portal. Antes de serem incorporados ao ambiente de engenharia, os documentos gerados permanecem armazenados como arquivos XML independentes, permitindo sua inspeção e preservando a separação entre a etapa de geração e a modificação do projeto.

Os *templates* utilizados também contêm as referências necessárias às estruturas padronizadas empregadas pela GreyLogix. Assim, a aplicação não implementa internamente toda a lógica de tratamento dos instrumentos, mas gera objetos que utilizam blocos e tipos de dados já disponíveis na biblioteca LCFL, conforme detalhado na subseção seguinte.

5.5.4 Utilização das estruturas padronizadas da biblioteca LCFL

Os arquivos SIMATIC ML gerados pela aplicação foram estruturados para utilizar a biblioteca LCFL, disponibilizada pela Siemens e já incorporada ao projeto utilizado na integração. Essa biblioteca reúne blocos funcionais, tipos de dados e estruturas padronizadas para aplicações de controle e monitoramento em projetos de automação. No contexto deste trabalho, a LCFL foi utilizada como referência para a geração dos objetos associados aos instrumentos analógicos e digitais no TIA Portal.

A utilização da biblioteca LCFL constitui uma escolha específica do estudo de caso e pressupõe que suas estruturas estejam previamente disponíveis no projeto-base. Em outros projetos ou plataformas, o mapeamento precisaria ser adaptado às bibliotecas, aos blocos e aos tipos de dados adotados pela organização responsável pelo desenvolvimento.

A utilização dessa biblioteca permite que a aplicação não precise recriar internamente toda a lógica de tratamento dos instrumentos. Em vez disso, o módulo desenvolvido em C# gera as chamadas, instâncias e registros de configuração necessários para utilizar estruturas já disponíveis no projeto. Dessa forma, os objetos criados permanecem alinhados aos padrões previamente adotados no ambiente de engenharia.

A seleção das estruturas é realizada de acordo com a classificação obtida a partir das *tags* validadas. Para cada instrumento, a aplicação insere uma instância do bloco padronizado, associa a *tag* de entrada e cria os elementos necessários nos blocos de configuração e nos DBs. Com isso, instrumentos pertencentes a uma mesma categoria são gerados segundo uma organização comum.

Essa abordagem contribui para manter a consistência entre os objetos gerados e as interfaces utilizadas no projeto. Ao instanciar estruturas previamente definidas, a aplicação reduz variações na implementação e facilita a continuidade do desenvolvi-

mento pelo engenheiro.

Após a importação, a configuração gerada deve ser revisada pelo engenheiro. Permanecem sob sua responsabilidade a verificação das instâncias criadas, o ajuste dos endereços provisórios e a complementação das configurações específicas de cada instrumento.

5.5.5 Importação dos objetos por meio do TIA Portal Openness

A importação dos arquivos gerados corresponde à etapa final do fluxo de integração. Para realizá-la, a aplicação em C# estabelece comunicação com o TIA Portal V20 por meio da interface TIA Portal Openness e abre o projeto definido para a execução. Esse projeto já contém a configuração inicial do controlador e as estruturas da biblioteca LCFL necessárias para resolver as referências presentes nos arquivos SIMATIC ML.

Após a abertura do projeto, a aplicação localiza o controlador e acessa o respectivo software de programação. A partir desse objeto, são recuperadas as coleções utilizadas para organizar as tabelas de *tags* e os blocos de programa, permitindo direcionar cada arquivo SIMATIC ML ao local correspondente na estrutura do projeto.

Para manter a separação adotada nas etapas anteriores, os arquivos são importados em grupos distintos para instrumentos analógicos e digitais. As tabelas de *tags* são incorporadas à coleção de tabelas do controlador, enquanto os arquivos referentes aos blocos de configuração, aos FBs e aos DBs são importados para as coleções de blocos correspondentes. Dessa forma, cada arquivo é processado conforme o tipo de objeto que representa.

As operações realizadas são registradas no terminal da aplicação, permitindo acompanhar a abertura do projeto, a localização do controlador e a importação dos arquivos. Caso algum item não possa ser processado, a ocorrência é informada para que o usuário verifique a estrutura do arquivo, as dependências do projeto ou possíveis conflitos com objetos já existentes.

Após a conclusão das importações, o projeto é salvo com os objetos incorporados. O resultado corresponde a uma configuração inicial composta por tabelas de *tags*, blocos de configuração, FBs e DBs associados aos instrumentos analógicos e digitais identificados no fluxo. Essa configuração ainda deve ser revisada pelo engenheiro, especialmente quanto aos endereços provisórios, às configurações específicas dos instrumentos e às conexões dependentes da arquitetura de hardware.

5.6 TESTES E RESULTADOS OBTIDOS

Após o desenvolvimento dos módulos, a solução foi submetida a testes para verificar o funcionamento do fluxo completo, desde a aplicação do detector nos diagra-

mas P&ID até a geração dos objetos no ambiente de engenharia adotado no estudo de caso. A análise considerou a detecção dos símbolos, o reconhecimento e a associação das *tags*, a validação técnica dos dados extraídos e a integração implementada no Siemens TIA Portal.

Nesta etapa, diferentemente da avaliação realizada sobre o conjunto de teste do modelo, foram utilizados diagramas completos. Essa avaliação permitiu observar o comportamento da solução em documentos com variações de escala, densidade visual, quantidade de páginas e organização gráfica, aproximando os testes das condições encontradas na aplicação do fluxo desenvolvido.

5.6.1 Comportamento do detector nos diagramas de aplicação

O detector foi aplicado a diferentes diagramas P&ID disponibilizados pela Grey-Logix, contendo entre 3 e 15 páginas e diferentes níveis de complexidade para diferentes aplicações. Como parte desses documentos contém informações não públicas, sua utilização foi restrita à verificação interna da solução. Para a apresentação dos resultados, foram selecionados dois casos de aplicação nos quais foi possível contabilizar manualmente os símbolos existentes e comparar essas ocorrências com as detecções produzidas pelo modelo.

No primeiro caso, o diagrama continha 55 símbolos pertencentes às classes consideradas pela solução. Todos os elementos foram localizados pelo detector e classificados corretamente, sem ocorrência de símbolos não detectados ou trocas entre as classes *instrument tag* e *instrument DCS*. Além das detecções válidas, foi registrada uma ocorrência adicional considerada falso positivo. Em uma das detecções corretas, a caixa delimitadora não envolveu adequadamente toda a região do símbolo, e embora esse comportamento não tenha impedido a identificação e a classificação do elemento, ele interferiu posteriormente na região encaminhada ao módulo de OCR.

O segundo teste foi realizado com um diagrama composto por sete páginas, contendo 89 símbolos de interesse. O processamento produziu 93 detecções, das quais 89 correspondiam aos símbolos existentes e quatro foram consideradas falsos positivos. Nenhum símbolo deixou de ser localizado e todas as detecções válidas foram atribuídas à classe correta. As ocorrências adicionais estiveram associadas à identificação de elementos que não pertenciam às classes do modelo e à duplicação de uma detecção sobre o mesmo símbolo.

A Tabela 15 apresenta uma síntese dos resultados obtidos nos dois casos analisados. A partir da contagem manual dos símbolos existentes nos documentos, as detecções válidas foram contabilizadas como verdadeiros positivos, as ocorrências adicionais como falsos positivos e os símbolos não identificados como falsos negativos. Com base nessas ocorrências, foram calculadas a precisão e a revocação para cada caso e para o conjunto dos dois diagramas.

Tabela 15 – Resultados do detector nos diagramas de aplicação analisados.

Caso	Símbolos	VP	FP	FN	Precisão	Revocação
Caso 1	55	55	1	0	98,21%	100,00%
Caso 2	89	89	4	0	95,70%	100,00%
Total	144	144	5	0	96,64%	100,00%

Fonte: Elaborado pelo autor.

Considerando conjuntamente os dois casos, os 144 símbolos existentes foram localizados e corretamente classificados, resultando em 144 verdadeiros positivos e nenhum falso negativo. A revocação foi, portanto, de 100,00%, indicando que não houve omissão de símbolos pertencentes às classes avaliadas nos documentos analisados.

As cinco ocorrências adicionais foram contabilizadas como falsos positivos, resultando em precisão de 96,64% para o conjunto dos dois diagramas. No primeiro caso, a precisão foi de 98,21%, enquanto no segundo foi de 95,70%. A diferença decorre da maior proporção de detecções adicionais no segundo documento, no qual quatro das 93 predições produzidas não correspondiam a símbolos válidos das classes consideradas.

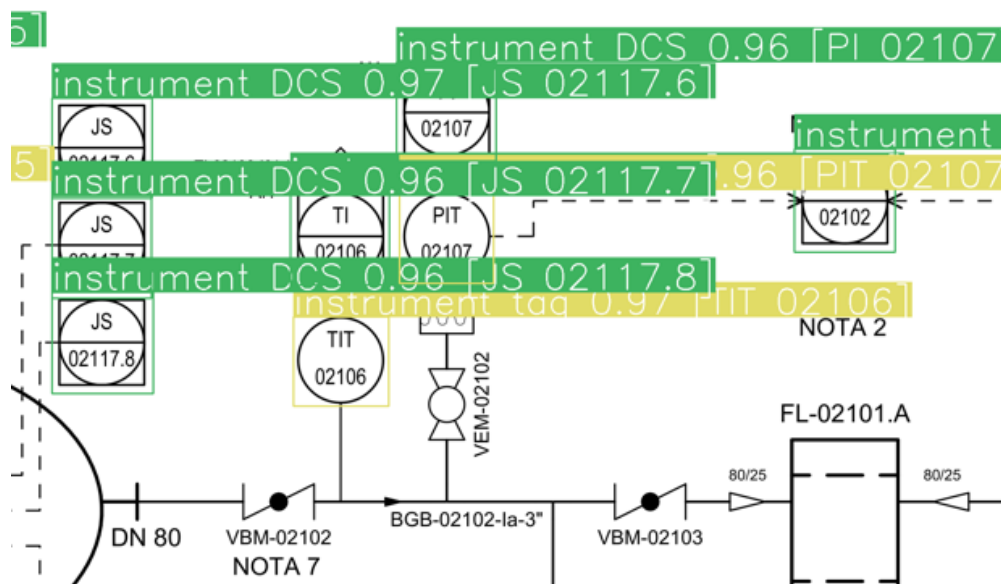
Esses resultados mostram que o detector preservou elevada capacidade de recuperação nos dois diagramas completos, embora tenha produzido ocorrências adicionais que precisaram ser removidas durante a validação técnica. Dessa forma, a etapa de revisão permaneceu relevante mesmo diante da ausência de falsos negativos nos casos analisados.

A ausência de omissões é relevante para o fluxo proposto, pois símbolos não detectados precisam ser inseridos manualmente durante a validação. Por outro lado, os falsos positivos observados puderam ser identificados e excluídos diretamente na aplicação de revisão. Dessa forma, o comportamento apresentado pelo detector reduziu o volume de intervenção necessário antes da continuidade do processamento.

Não foram calculadas as métricas AP e mAP para os diagramas completos, pois esses documentos não foram preparados como um conjunto de avaliação com caixas delimitadoras de referência formalmente anotadas. A comparação manual permitiu contabilizar os símbolos identificados, as ocorrências adicionais e as omissões, mas não avaliar de forma padronizada a correspondência espacial das caixas em diferentes limiares de IoU.

A Figura 12 apresenta um recorte de um dos diagramas utilizados nos testes, com as caixas delimitadoras geradas pelo modelo. Foi utilizado apenas um trecho do documento, de modo a ilustrar o resultado da detecção sem expor integralmente as informações do projeto.

Figura 12 – Exemplo de detecção dos símbolos de instrumentação em recorte de um diagrama P&ID.



Fonte: Elaborado pelo autor a partir de recorte anonimizado de diagrama disponibilizado pela GreyLogix.

Também foi realizado um teste exploratório com um diagrama cuja simbologia apresentava diferenças em relação às representações da ANSI/ISA-5.1 adotadas no treinamento. Nesse caso, observou-se uma elevada quantidade de detecções incorretas e valores de confiança inferiores aos obtidos nos diagramas compatíveis com o padrão considerado pela solução.

Esse comportamento indica que o desempenho do detector depende da semelhança entre os símbolos presentes no documento processado e aqueles representados no conjunto de treinamento. Quando a geometria, os elementos internos ou a forma de representação dos instrumentos diferem das classes aprendidas pelo modelo, aumenta a possibilidade de falsos positivos, omissões ou detecções com menor confiança.

Esse resultado não caracteriza uma falha em relação ao escopo definido, uma vez que a solução foi desenvolvida para as representações apresentadas na Tabela 8. Entretanto, evidencia que a aplicação em documentos com convenções de simbologia distintas pode exigir adaptação do conjunto de dados, inclusão de novos exemplos de treinamento ou uma revisão manual mais ampla dos resultados.

5.6.2 Resultados do reconhecimento e da associação das tags

O desempenho do módulo de reconhecimento textual foi analisado nos mesmos dois casos utilizados para verificar o comportamento do detector. Foram consideradas apenas as detecções correspondentes a símbolos efetivamente presentes nos diagra-

mas. As detecções classificadas como falsos positivos não foram contabilizadas como erros de OCR, pois não possuíam uma *tag* de referência válida e foram removidas posteriormente durante a validação técnica.

No primeiro caso, dos 55 símbolos corretamente identificados pelo detector, 53 tiveram suas *tags* reconhecidas integralmente. As duas ocorrências que exigiram revisão decorreram de condições distintas: em uma delas, a caixa delimitadora não abrangeu toda a região necessária para a leitura, resultando em uma *tag* incompleta; na outra, parte do texto estava sobreposta à representação gráfica do próprio símbolo, dificultando a distinção dos caracteres pelo módulo de OCR. Em ambos os casos, os símbolos e suas classes foram identificados corretamente, e as *tags* foram corrigidas durante a validação técnica.

No segundo caso, as *tags* dos 89 símbolos válidos foram reconhecidas corretamente. As quatro detecções adicionais produzidas pelo detector não correspondiam a instrumentos válidos e, portanto, foram desconsideradas na avaliação do reconhecimento textual e excluídas na etapa de revisão. Não foram observados erros de leitura ou de associação entre os símbolos efetivamente presentes nesse diagrama. A Tabela 16 apresenta a síntese dos resultados obtidos.

Tabela 16 – Resultados do reconhecimento das *tags* nos diagramas de aplicação.

Caso	Símbolos avaliados	Tags corretas	Tags revisadas	Proporção correta
Caso 1	55	53	2	96,4%
Caso 2	89	89	0	100,0%
Total	144	142	2	98,6%

Fonte: Elaborado pelo autor.

Considerando conjuntamente os dois casos, 142 das 144 *tags* associadas a símbolos válidos foram reconhecidas corretamente, correspondendo a 98,6% do total avaliado. Também não foram observadas associações entre uma *tag* e um símbolo diferente daquele que originou sua região de busca, pois cada recorte encaminhado ao OCR permaneceu vinculado à respectiva detecção.

Os dois erros observados tiveram origens distintas. O primeiro evidencia a dependência entre a qualidade da caixa delimitadora e o reconhecimento textual, uma vez que um recorte incompleto pode excluir parte dos caracteres. O segundo demonstra a influência da própria organização gráfica do diagrama, especialmente quando textos e elementos do símbolo se sobrepõem. Em ambos os casos, a etapa de validação permitiu corrigir diretamente a *tag*, sem necessidade de excluir ou inserir novamente o símbolo.

Os resultados indicam que a aplicação do OCR em regiões orientadas pelas detecções foi adequada para os diagramas avaliados. As correções necessárias fi-

caram restritas a duas ocorrências, enquanto as demais *tags* foram reconhecidas e associadas corretamente aos respectivos símbolos.

5.6.3 Resultados da validação técnica

A validação técnica permitiu revisar os resultados produzidos pelos módulos de detecção e reconhecimento textual antes de sua utilização nas etapas de engenharia. Cada ocorrência foi apresentada juntamente com um recorte do diagrama, a classe atribuída, a confiança da detecção e a *tag* reconhecida, possibilitando confirmar, corrigir ou excluir o registro.

Nos dois casos analisados, foram revisados 149 registros iniciais. Ao todo, duas *tags* precisaram ser corrigidas devido às limitações descritas na subseção anterior, e cinco detecções adicionais foram excluídas por não corresponderem a símbolos válidos ou por representarem duplicações. Não foram necessárias correções de classe nem inserções manuais, uma vez que todos os 144 símbolos presentes nos diagramas haviam sido localizados corretamente.

Ao término da revisão, foram gerados arquivos contendo somente os 144 registros validados. Esses resultados mostram que a aplicação cumpriu sua função de impedir que erros pontuais fossem encaminhados às etapas seguintes, mantendo reduzida a quantidade de intervenções necessárias. A Figura 13 apresenta um exemplo da interface utilizada nesse processo.

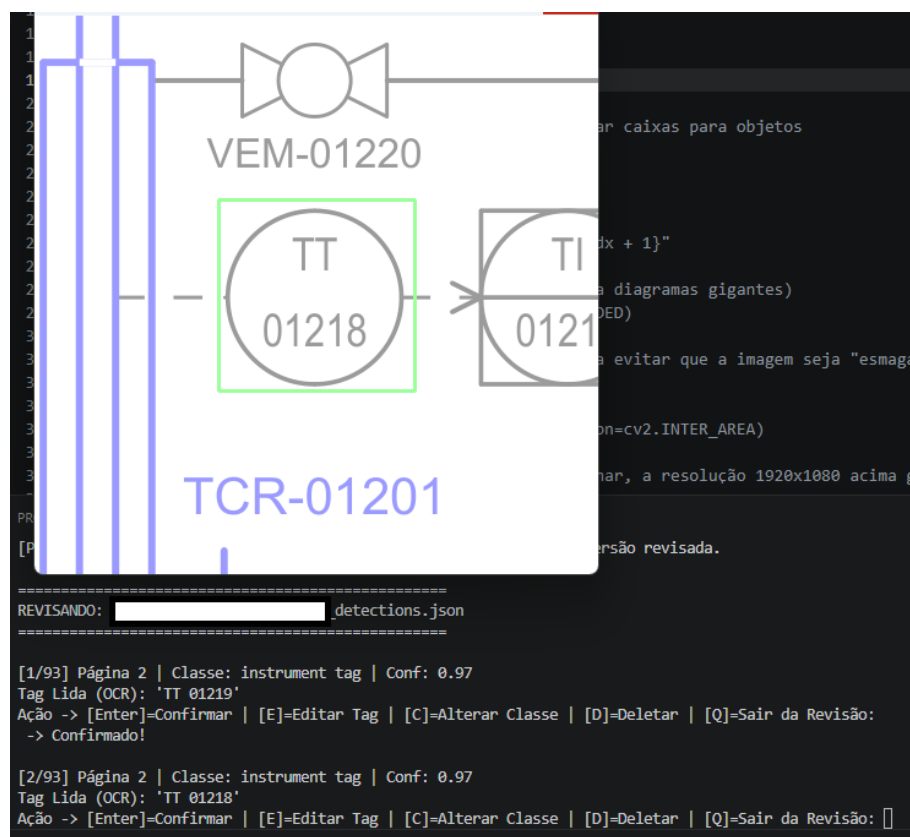
5.6.4 Resultados da geração de objetos no TIA Portal

A etapa de geração e integração foi verificada utilizando o arquivo *JSON* revisado correspondente ao caso de aplicação com 55 elementos. A aplicação realizou a leitura e a classificação das *tags*, gerou os oito arquivos SIMATIC ML previstos e efetuou sua importação no projeto por meio do TIA Portal Openness.

No teste realizado, os arquivos foram importados sem erros e os objetos foram inseridos nos grupos 01_Analogicos e 02_Digitais, conforme a classificação atribuída a cada *tag*. Foram geradas as tabelas de *tags*, os blocos de configuração, os FBs e os DBs associados aos instrumentos. As referências às estruturas da biblioteca LCFL também foram resolvidas pelo TIA Portal, uma vez que os blocos e tipos de dados necessários já estavam disponíveis no projeto.

A verificação no ambiente de engenharia confirmou que os nomes dos instrumentos presentes no arquivo validado foram preservados nos objetos gerados e que a organização definida nos *templates* foi mantida após a importação. Os endereços inseridos nas tabelas de *tags* também foram criados conforme a sequência provisória adotada pela aplicação. Esses valores, entretanto, devem ser substituídos posteriormente pelos endereços físicos definidos na lista de I/Os e na configuração de hardware do projeto.

Figura 13 – Exemplo da aplicação utilizada para a validação das detecções e das tags reconhecidas.



Fonte: Elaborado pelo autor a partir de recorte anonimizado de diagrama disponibilizado pela GreyLogix.

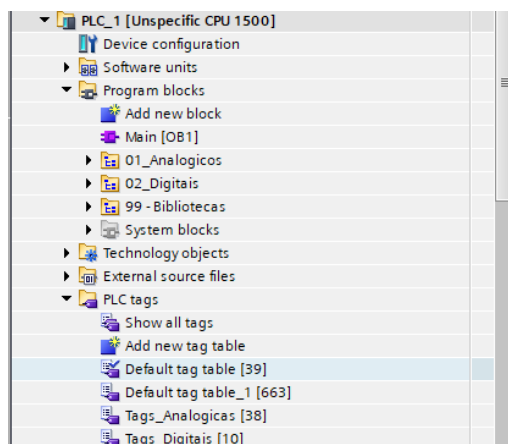
A Figura 14 apresenta exemplos dos objetos incorporados ao projeto após a importação dos arquivos SIMATIC ML. Os recortes mostram a organização dos grupos no TIA Portal, a tabela de tags gerada para os instrumentos analógicos e um bloco funcional contendo instâncias associadas aos instrumentos identificados no diagrama.

O teste confirmou funcionalmente a continuidade do fluxo entre o arquivo JSON validado e o ambiente de engenharia adotado no estudo de caso, demonstrando que as informações extraídas do diagrama puderam ser convertidas em uma estrutura inicial de programação no Siemens TIA Portal. O conjunto de 55 elementos foi utilizado como caso de verificação da integração e não representa um limite de capacidade da aplicação. Após a importação, permanecem necessárias a revisão dos objetos pelo engenheiro, a substituição dos endereços provisórios e a complementação das configurações específicas do projeto.

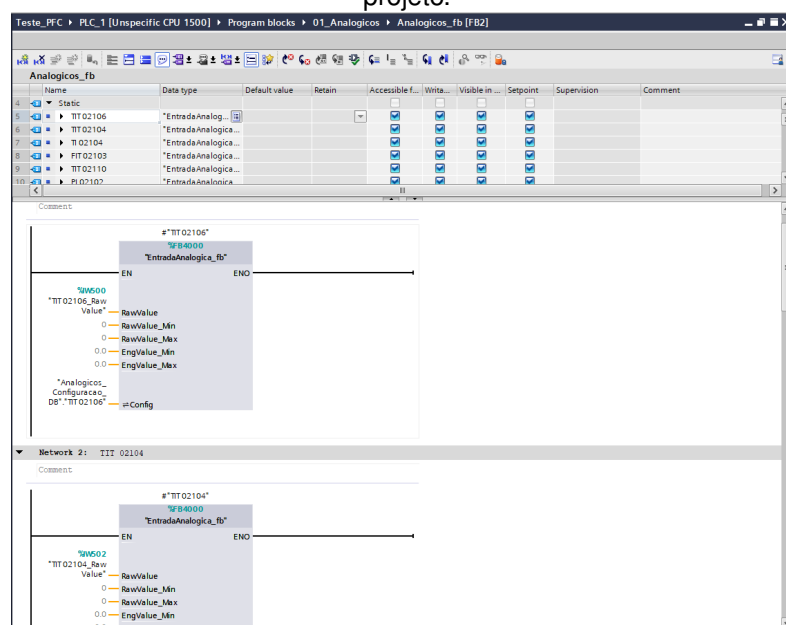
5.7 ANÁLISE CRÍTICA DA SOLUÇÃO

Os resultados obtidos demonstraram que a solução desenvolvida executou o fluxo proposto, desde a identificação dos símbolos nos diagramas P&ID até a geração e importação dos objetos no TIA Portal. A análise desta seção reúne as principais

Figura 14 – Exemplos de objetos gerados e importados no projeto do TIA Portal.



(a) Organização dos grupos e tabelas no projeto.



(b) Bloco funcional com instâncias associadas aos instrumentos.

Tags_Analogicas									
	Name	Data type	Address	Retain	Acces...	Write...	Visibl...	Supervision	Comment
1	TIT 02106_RawValue	Int	%IW500		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
2	TIT 02104_RawValue	Int	%IW502		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
3	TI 02104_RawValue	Int	%IW504		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
4	FIT 02103_RawValue	Int	%IW506		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
5	TIT 02110_RawValue	Int	%IW508		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
6	PI 02102_RawValue	Int	%IW510		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
7	TIT 02105_RawValue	Int	%IW512		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
8	TI 02105_RawValue	Int	%IW514		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
9	AI 02113_RawValue	Int	%IW516		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
10	PI 02101_2_RawValue	Int	%IW518		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
11	TI 02106_RawValue	Int	%IW520		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
12	TI 02110_RawValue	Int	%IW522		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
13	PI 02116_RawValue	Int	%IW524		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
14	FDI 02102_RawValue	Int	%IW526		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC
15	PI 02111_RawValue	Int	%IW528		☒	☒	☒		Gerado automaticamente pelo Codigo do PFC

(c) Tabela de tags analógicas gerada.
Fonte: Elaborado pelo autor.

vantagens observadas durante os testes e as limitações associadas ao escopo, aos dados utilizados e às escolhas de implementação.

A principal vantagem da solução está na integração entre etapas que, no fluxo convencional, dependem de atividades manuais sucessivas. A combinação entre detecção de símbolos, reconhecimento textual, representação intermediária em *JSON*, validação técnica e mapeamento das informações permitiu transformar dados presentes no diagrama em registros estruturados e revisados. No estudo de caso Siemens, esses registros também foram utilizados para gerar uma configuração inicial de projeto. Nos casos analisados, o detector localizou os 144 símbolos existentes, sem erros de classificação entre as classes consideradas, enquanto as intervenções ficaram restritas à exclusão de cinco falsos positivos e à correção de duas *tags*.

A etapa de validação também se mostrou relevante, pois permite que o engenheiro confirme, corrija ou exclua os registros antes da integração com o ambiente de programação. Com isso, erros provenientes da detecção ou do OCR podem ser corrigidos antes de serem incorporados ao projeto, preservando a responsabilidade técnica sobre os dados gerados.

A representação intermediária em *JSON* favoreceu a modularidade da implementação, separando os componentes responsáveis pelo processamento dos diagramas da aplicação em C# utilizada na geração dos arquivos SIMATIC ML e na comunicação com o TIA Portal. Essa separação facilita ajustes no detector, no reconhecimento textual ou nas regras de mapeamento sem exigir a reformulação integral do fluxo. Também permite que os dados validados sejam utilizados por outras aplicações, embora a adaptação para um ambiente de engenharia diferente exija o desenvolvimento de uma nova camada de integração.

Outro aspecto positivo foi a geração dos objetos a partir de *templates* e de estruturas padronizadas da biblioteca LCFL. Essa abordagem permitiu criar tabelas de *tags*, blocos de configuração, FBs e DBs de forma coerente com o projeto, reduzindo a necessidade de inserção individual dos instrumentos no ambiente de engenharia. Como não foi realizada uma comparação cronometrada entre o procedimento manual e o fluxo assistido, não é possível quantificar a redução de tempo proporcionada pela solução. Ainda assim, os resultados indicam potencial de ganho operacional, uma vez que o fluxo reduz etapas de transcrição manual, permite corrigir inconsistências antes da integração com o TIA Portal e gera estruturas de forma padronizada, diminuindo a possibilidade de retrabalho em etapas posteriores do projeto.

Entre as limitações identificadas está a dependência da simbologia utilizada nos diagramas. O modelo foi treinado para reconhecer duas representações específicas, definidas com base na ANSI/ISA-5.1. O teste com um diagrama que apresentava convenções de símbolos distintas resultou em elevada quantidade de detecções incorretas e menores valores de confiança, indicando que a aplicação em outros padrões pode

exigir ampliação ou adaptação do conjunto de treinamento.

O reconhecimento textual também depende da qualidade gráfica do documento e da região fornecida pelo detector. Textos parcialmente sobrepostos aos símbolos, baixa legibilidade ou caixas delimitadoras incompletas podem comprometer a leitura da *tag*. Embora esses casos possam ser corrigidos durante a validação, eles indicam que o desempenho do OCR depende das características visuais do diagrama e da qualidade da detecção anterior.

Embora a representação intermediária preserve a página de origem e as coordenadas absolutas de cada símbolo, o uso dessas informações ficou restrito à rastreabilidade e à recuperação das regiões apresentadas na aplicação de validação. A implementação não utiliza a posição relativa dos instrumentos, as linhas de processo e de sinal ou as conexões com equipamentos e tubulações para inferir relações funcionais entre os elementos do diagrama. Consequentemente, a solução gera objetos individuais a partir das classes e das *tags* identificadas, mas não reconstrói a topologia do processo nem deriva automaticamente a estrutura funcional das malhas representadas no P&ID.

A geração de objetos no TIA Portal foi implementada somente para os grupos de instrumentos analógicos e digitais. Embora as regras de classificação reconheçam outras funções, como controladores e elementos finais de controle, essas categorias ainda não foram incorporadas à geração automática. Além disso, os endereços inseridos nas tabelas de *tags* são provisórios, pois o diagrama P&ID não contém as informações necessárias para determinar os endereços físicos das entradas e saídas.

Outra limitação está na especificidade da camada de integração implementada. A aplicação desenvolvida utiliza o .NET Framework, o TIA Portal Openness, o formato SIMATIC ML e estruturas da biblioteca LCFL, estando configurada para o Siemens TIA Portal V20 Update 4. Dessa forma, o módulo de geração não pode ser utilizado diretamente em ambientes de outros fabricantes. Uma adaptação exigiria novas regras de mapeamento, formatos de representação e mecanismos de comunicação compatíveis com a plataforma de destino. Entretanto, a organização modular permite preservar as etapas anteriores de detecção, reconhecimento textual, estruturação e validação, desde que o escopo dos diagramas permaneça compatível.

A principal contribuição da solução está na articulação das etapas de processamento documental e de engenharia em um fluxo integrado e modular. A detecção dos símbolos e o reconhecimento das *tags* não constituem resultados isolados, pois suas saídas são organizadas em uma representação intermediária, submetidas à validação técnica, interpretadas por regras de mapeamento e utilizadas na geração assistida de objetos. Essa organização diferencia a proposta de abordagens voltadas exclusivamente à digitalização dos diagramas e estabelece uma continuidade entre a documentação de processo e a configuração inicial do ambiente de automação.

Apesar das limitações identificadas, os testes demonstraram a viabilidade funcional da arquitetura dentro do escopo avaliado. Nos diagramas de aplicação, as informações de instrumentação foram extraídas, estruturadas e submetidas à validação, enquanto, no estudo de caso Siemens, os dados revisados foram convertidos em estruturas iniciais do projeto de automação. A solução não elimina a participação do engenheiro, mas transfere parte das operações de identificação, transcrição e criação individual de objetos para um processo assistido de revisão, complementação e adequação das informações.

6 CONCLUSÃO

Este capítulo reúne as principais conclusões do trabalho, relacionando os resultados obtidos aos objetivos estabelecidos. Também são discutidos os impactos e as limitações da solução desenvolvida, bem como as possibilidades de aprimoramento e continuidade do projeto.

6.1 SÍNTESE DO TRABALHO E ATENDIMENTO AOS OBJETIVOS

O desenvolvimento de projetos de automação industrial envolve a interpretação de documentos técnicos e a transferência de informações para ambientes de engenharia que nem sempre possuem integração direta. Entre essas atividades, a identificação de instrumentos em diagramas P&ID, a transcrição de suas *tags* e a criação inicial de estruturas de programação constituem etapas repetitivas, sujeitas a erros de leitura, preenchimento e padronização.

Diante desse problema, este trabalho propôs uma solução assistida para identificar símbolos de instrumentação em diagramas P&ID, reconhecer suas *tags*, organizar os dados extraídos e utilizá-los na geração de estruturas iniciais de projetos de automação. O fluxo desenvolvido integrou visão computacional, reconhecimento textual, representação intermediária em *JSON*, validação técnica, mapeamento das informações e geração de objetos de engenharia. Como estudo de caso, a camada de integração foi implementada no Siemens TIA Portal, ambiente utilizado nos projetos analisados na GreyLogix.

A principal contribuição deste trabalho consiste na proposição e na avaliação de uma arquitetura modular que encadeia a preparação dos documentos, a detecção e a classificação dos símbolos, o reconhecimento textual, a representação intermediária dos dados, a validação técnica, o mapeamento das informações e a geração assistida de objetos de engenharia. A contribuição, portanto, não está na aplicação isolada de uma técnica de visão computacional ou de reconhecimento textual, mas na integração dessas etapas em um fluxo capaz de transformar informações extraídas de diagramas P&ID em dados revisados e utilizáveis na configuração inicial de um projeto de automação.

O objetivo geral foi atendido por meio da implementação de um fluxo funcional entre o documento de entrada e o ambiente de engenharia. Para isso, foi adaptado e organizado um conjunto de dados com 2378 imagens e 6172 instâncias anotadas, utilizado no treinamento de um modelo YOLO11m para as duas classes de instrumentação consideradas. O modelo apresentou desempenho elevado no conjunto de teste e, nos dois diagramas completos utilizados como casos de aplicação, localizou e classificou corretamente os 144 símbolos existentes.

O reconhecimento textual também apresentou resultado satisfatório para o es-

copo avaliado. Das 144 *tags* associadas aos símbolos válidos, 142 foram reconhecidas corretamente, enquanto as duas ocorrências restantes foram corrigidas na etapa de validação técnica. A aplicação de validação permitiu ainda excluir detecções adicionais e gerar arquivos revisados contendo apenas os registros confirmados pelo engenheiro.

A etapa de integração demonstrou a continuidade do fluxo no ambiente adotado como estudo de caso. A aplicação desenvolvida em C# realizou a leitura das *tags* validadas, a geração de arquivos SIMATIC ML a partir de *templates* e sua importação por meio do TIA Portal Openness. No caso utilizado para verificar essa integração, contendo 55 elementos, foram criadas tabelas de *tags*, blocos de configuração, FBs e DBs associados aos instrumentos analógicos e digitais, utilizando estruturas da biblioteca LCFL previamente disponíveis no projeto-base.

Dessa forma, foram contemplados os objetivos específicos relacionados à preparação dos dados, à detecção dos símbolos, ao reconhecimento textual, à representação estruturada das informações, à validação técnica e ao mapeamento dos elementos identificados. A implementação no Siemens TIA Portal permitiu ainda verificar a geração assistida dos objetos definidos no escopo do estudo de caso. A solução mantém a participação do engenheiro, mas desloca parte do esforço de identificação, transcrição e criação inicial dos objetos para um processo de revisão e complementação das informações geradas automaticamente.

Além do atendimento aos objetivos, os resultados foram confrontados com os requisitos técnicos estabelecidos na Seção 3.4. Para essa análise, um requisito foi considerado atendido quando sua implementação e sua verificação foram demonstradas dentro do escopo definido para o estudo de caso. A classificação como parcialmente atendido foi utilizada nos casos em que a funcionalidade foi implementada, mas sua comprovação permaneceu qualitativa ou não incluiu uma verificação específica de desempenho. As Tabelas 17 e 18 apresentam a rastreabilidade entre os requisitos, as evidências obtidas e as respectivas seções do trabalho.

Tabela 17 – Atendimento aos requisitos funcionais da solução.

Código	Evidência de atendimento	Seção	Situação
RF01	Os diagramas em formato PDF foram convertidos em imagens e divididos em recortes adequados ao processamento computacional.	5.1 e 5.2.5	Atendido
RF02	O modelo YOLO11m foi treinado para identificar as duas classes definidas. Nos diagramas de aplicação, os 144 símbolos de referência foram localizados e corretamente classificados.	5.2.4 e 5.6.1	Atendido

Tabela 17 – continuação

Código	Evidência de atendimento	Seção	Situação
RF03	Para cada detecção, foram produzidos a classe prevista, o valor de confiança, as coordenadas da caixa delimitadora e a página de origem.	5.2.5 e 5.4.1	Atendido
RF04	O módulo de reconhecimento textual foi associado às regiões definidas pelo detector. Das 144 <i>tags</i> de referência, 142 foram reconhecidas automaticamente, e duas foram corrigidas durante a validação.	5.3 e 5.6.2	Atendido
RF05	As informações extraídas foram armazenadas em uma representação intermediária em formato <i>JSON</i> , contendo classe, confiança, <i>tag</i> , coordenadas e página.	4.2.4 e 5.4.1	Atendido
RF06	A aplicação de validação permitiu confirmar, corrigir, excluir e inserir registros antes de sua utilização nas etapas de engenharia.	5.4.2 e 5.6.3	Atendido
RF07	As identificações validadas foram interpretadas e relacionadas às categorias, aos tipos de dados e aos objetos previstos para os instrumentos analógicos e digitais do estudo de caso.	5.5.1 e 5.5.4	Atendido
RF08	O arquivo <i>JSON</i> revisado foi utilizado como entrada pela aplicação em C#, estabelecendo a transferência estruturada dos dados entre os módulos de extração e integração.	5.4.1 e 5.5.2	Atendido
RF09	Os dados extraídos, validados e mapeados foram utilizados para gerar tabelas de <i>tags</i> , blocos de configuração, FBs e DBs.	5.5 e 5.6.4	Atendido

Tabela 17 – continuação

Código	Evidência de atendimento	Seção	Situação
RF10	No estudo de caso, os arquivos SIMA-TIC ML correspondentes a 55 elementos foram gerados e importados no Siemens TIA Portal por meio do TIA Portal Openness.	5.5.5 e 5.6.4	Atendido

Fonte: Elaborado pelo autor.

Tabela 18 – Atendimento aos requisitos não funcionais da solução.

Código	Evidência de atendimento	Seção	Situação
RNF01	O arquivo intermediário mantém o documento e a página de origem, as coordenadas do símbolo, sua classe e a <i>tag</i> associada, permitindo recuperar o registro e sua região no diagrama.	5.4.1 e 5.4.2	Atendido
RNF02	Os dados somente são encaminhados à etapa de geração após a revisão pelo engenheiro, que pode confirmar, corrigir, excluir ou complementar os registros.	5.4.2 e 5.6.3	Atendido
RNF03	A representação em <i>JSON</i> organiza os dados em campos definidos e foi lida tanto pela aplicação em Python quanto pelo módulo desenvolvido em C#.	4.2.4, 5.4.1 e 5.5.2	Atendido
RNF04	A implementação separa preparação, detecção, reconhecimento textual, estruturação, validação, mapeamento e integração, utilizando o arquivo <i>JSON</i> como interface entre os módulos principais.	4.1, 4.2 e 5.1	Atendido
RNF05	A ANSI/ISA-5.1 foi utilizada como referência para relacionar as classes selecionadas às representações de instrumentação e para interpretar as identificações funcionais consideradas no mapeamento.	2.2.3, 4.2.2 e 5.5.1	Atendido

Tabela 18 – continuação

Código	Evidência de atendimento	Seção	Situação
RNF06	A utilização de <i>templates</i> , regras de nomenclatura e estruturas da biblioteca LCFL produziu objetos organizados de forma uniforme. Entretanto, não foi realizada uma comparação quantitativa da padronização em relação ao procedimento convencional.	5.5.3, 5.5.4 e 5.6.4	Parcialmente atendido
RNF07	As versões das ferramentas, os parâmetros de treinamento, os limiares e as configurações de processamento foram documentados. Entretanto, não foi realizado um teste independente de reprodução integral do fluxo por outro usuário ou ambiente computacional.	5.2.3, 5.2.5 e 5.3.2	Parcialmente atendido
RNF08	A aplicação foi executada com o TIA Portal V20 Update 4 e utilizou o TIA Portal Openness e o formato SIMATIC ML para gerar e importar os objetos definidos no estudo de caso.	5.5.2 a 5.5.5 e 5.6.4	Atendido
Fonte: Elaborado pelo autor.			

A análise evidencia que os requisitos funcionais foram implementados e verificados dentro do escopo estabelecido para o protótipo. Os requisitos não funcionais relacionados à rastreabilidade, à validação humana, à organização dos dados, à modularidade, à referência normativa e à compatibilidade com o ambiente Siemens também foram atendidos. O favorecimento da padronização e a reprodutibilidade foram classificados como parcialmente atendidos, pois, embora tenham sido incorporados à implementação, não foram submetidos, respectivamente, a uma comparação quantitativa com o procedimento convencional e a um teste independente de reprodução integral.

6.2 IMPACTOS E LIMITAÇÕES DA SOLUÇÃO

Do ponto de vista tecnológico, o trabalho demonstrou a viabilidade de combinar técnicas de visão computacional e reconhecimento textual com recursos de automação de engenharia. A adoção de uma representação intermediária em *JSON* permitiu separar os módulos desenvolvidos em Python da aplicação responsável pela geração dos

objetos, favorecendo a modularidade do fluxo e permitindo que as etapas de extração, validação, mapeamento e integração sejam ajustadas de forma independente.

Para a GreyLogix, a principal contribuição está no apoio à padronização da configuração inicial dos projetos de automação. No estudo de caso, a geração baseada em *templates* e em estruturas da biblioteca LCFL reduziu a necessidade de inserir individualmente os instrumentos no TIA Portal e permitiu manter maior uniformidade na criação de tabelas de *tags*, blocos e instâncias. Embora não tenha sido realizada uma comparação cronometrada entre o procedimento convencional e o fluxo assistido, os resultados indicam potencial para reduzir erros de transcrição, retrabalho e atividades repetitivas.

A manutenção da validação técnica pelo engenheiro também representa um aspecto relevante da solução. Os dados produzidos automaticamente não são incorporados diretamente ao ambiente de engenharia, permanecendo sujeitos à conferência profissional antes da geração dos objetos. Dessa forma, a automação proposta atua como apoio ao desenvolvimento, sem transferir ao sistema a responsabilidade pelas decisões técnicas do projeto.

Como premissa de aplicação, a solução considera diagramas cujas representações de instrumentação sejam compatíveis com as classes utilizadas no treinamento do detector.

O módulo de detecção foi treinado e avaliado para apenas duas classes de símbolos de instrumentação. Essa delimitação foi adequada ao estudo de caso e permitiu avaliar o fluxo integrado, mas restringe a aplicação direta da solução a diagramas que utilizem representações compatíveis com essas classes. Além disso, a capacidade de generalização para outros padrões gráficos, convenções empresariais e características documentais não foi avaliada de forma abrangente. A ampliação para válvulas, atuadores, equipamentos e outras categorias de instrumentos exigirá novas anotações, treinamento adicional e revisão das regras de mapeamento. O reconhecimento das *tags*, por sua vez, permanece sensível à qualidade gráfica dos diagramas, à precisão das caixas delimitadoras e à sobreposição entre textos e elementos gráficos. Documentos com baixa resolução, convenções textuais distintas ou representações diferentes das presentes nos casos avaliados podem exigir ajustes no pré-processamento, nos parâmetros do reconhecimento textual ou nas regras utilizadas para validar as identificações.

Na etapa de engenharia, a geração automática foi concluída apenas para os grupos de instrumentos analógicos e digitais. Controladores, válvulas e outros elementos finais de controle ainda dependem da criação de novos *templates*, regras de mapeamento e rotinas de geração. Além disso, os endereços inseridos nas tabelas de *tags* são provisórios, pois o diagrama P&ID não contém as informações necessárias para determinar a posição física dos sinais no controlador.

Outra limitação está relacionada à camada de integração implementada. A aplicação desenvolvida utiliza C#, .NET Framework, SIMATIC ML, TIA Portal Openness e estruturas da biblioteca LCFL, estando configurada para o Siemens TIA Portal V20 Update 4. Dessa forma, o módulo de geração não pode ser utilizado diretamente em ambientes de outros fabricantes. Uma adaptação exigiria novas regras de mapeamento e uma camada de integração compatível com os formatos, as interfaces e os modelos de projeto disponibilizados pela plataforma de destino.

A integração também depende de um projeto-base compatível, contendo previamente os blocos e tipos de dados da biblioteca LCFL referenciados pelos arquivos gerados. A implementação não foi avaliada sistematicamente em outras versões do TIA Portal, diferentes modelos de controlador ou estruturas alternativas de projeto.

6.3 POSSIBILIDADES DE APRIMORAMENTO E TRABALHOS FUTUROS

Como continuidade do trabalho, o conjunto de dados pode ser ampliado com exemplos provenientes de diferentes padrões gráficos, resoluções e formas de representação. Essa ampliação permitiria avaliar a capacidade de generalização do detector e reduzir sua dependência das duas classes utilizadas no treinamento inicial. Também podem ser incorporadas novas categorias de símbolos, incluindo controladores, válvulas, atuadores e outros elementos relevantes para a geração dos projetos de automação.

O reconhecimento textual pode ser aprimorado por meio da avaliação de diferentes mecanismos de OCR, técnicas adicionais de pré-processamento e métodos específicos para separar caracteres de linhas ou elementos gráficos sobrepostos. Também pode ser estudada a utilização conjunta do reconhecimento visual com a extração direta de textos em documentos PDF que preservem informações vetoriais.

A detecção de objetos também pode ser utilizada independentemente da camada de integração implementada no estudo de caso. A identificação automática de instrumentos, sensores e outros elementos presentes nos diagramas P&ID pode apoiar a elaboração ou a conferência de documentos auxiliares de engenharia, como listas de instrumentos, listas de I/Os, relatórios de componentes identificados e levantamentos preliminares de sinais. Essa aplicação permitiria utilizar parte do fluxo desenvolvido mesmo em projetos nos quais a geração automática de objetos de programação não fosse adotada.

Outra possibilidade de aprimoramento consiste em ampliar a análise dos próprios diagramas P&ID. A partir da identificação dos símbolos, das *tags* e das relações espaciais entre os elementos, podem ser desenvolvidas rotinas para verificar a completude das informações presentes no documento, identificar inconsistências de nomenclatura, detectar duplicidades ou apontar instrumentos sem informações suficientes para as etapas posteriores do projeto. Dessa forma, a solução também poderia atuar

como ferramenta de apoio à revisão documental.

A utilização das informações espaciais e topológicas dos diagramas constitui outra possibilidade de continuidade. A identificação de tubulações, linhas de sinal, equipamentos e pontos de conexão permitiria representar o P&ID como uma estrutura relacional ou um grafo, no qual os elementos identificados estariam associados conforme suas conexões no processo. Essa ampliação poderia apoiar a organização dos instrumentos por equipamentos e malhas, a verificação de consistência das informações extraídas e a geração de estruturas de engenharia mais completas (Paliwal et al., 2021; Kim et al., 2022; Stürmer; Graumann; Koch, 2025).

Na aplicação de validação, podem ser acrescentados mecanismos de navegação por página, filtros por classe ou confiança, registro das alterações realizadas e apresentação simultânea de múltiplas detecções. Essas funcionalidades podem reduzir o tempo de revisão em projetos maiores e fornecer maior controle sobre as correções efetuadas.

Na camada de integração, uma evolução relevante consiste em utilizar informações provenientes da lista de I/O ou da configuração de hardware para substituir automaticamente os endereços provisórios. A geração também pode ser estendida aos demais grupos de instrumentos e adaptada a diferentes bibliotecas, estruturas de projetos e versões do ambiente de engenharia.

O desenvolvimento de novas camadas de integração para plataformas de outros fabricantes também constitui uma possibilidade de continuidade. Nesse caso, as etapas de detecção, reconhecimento textual, estruturação e validação poderiam ser preservadas, enquanto as regras de mapeamento, os formatos de representação e os mecanismos de comunicação com o ambiente de destino precisariam ser implementados conforme os recursos disponibilizados por cada plataforma.

Além disso, a incorporação de outros documentos técnicos ao fluxo pode permitir uma geração mais completa das estruturas de automação. A interpretação conjunta de diagramas P&ID, listas de instrumentos, listas de I/Os, diagramas lógicos, memoriais descritivos e documentação elétrica poderia fornecer informações complementares sobre endereçamento, tipos de sinais, intertravamentos, alarmes e estratégias de controle. Com o apoio de outras técnicas de análise documental, essa integração poderia ampliar o escopo da solução para além da identificação inicial dos instrumentos, aproximando o fluxo de uma geração mais abrangente do projeto de automação.

A avaliação da solução pode ser complementada por testes com maior quantidade de projetos, diferentes usuários e níveis variados de complexidade. Uma comparação cronometrada entre o procedimento manual e o fluxo assistido permitiria quantificar os ganhos de produtividade, enquanto a análise dos erros, das intervenções e do retrabalho poderia fornecer uma estimativa mais precisa de seus impactos operacionais e financeiros.

REFERÊNCIAS

- ARROYO, Esteban; HOERNICKE, Mario; RODRÍGUEZ, Pablo; FAY, Alexander. Automatic Derivation of Qualitative Plant Simulation Models from Legacy Piping and Instrumentation Diagrams. **Computers & Chemical Engineering**, v. 92, p. 112–132, 2016. DOI: 10.1016/j.compchemeng.2016.04.040.
- BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. YOLOv4: Optimal Speed and Accuracy of Object Detection. **arXiv preprint arXiv:2004.10934**, 2020. Disponível em: <https://arxiv.org/abs/2004.10934>.
- DÜRKOP, Lars; WISNIEWSKI, Lukasz; HEYMANN, Sascha; LÜCKE, Benedikt; JASPERNEITE, Jürgen. Analyzing the engineering effort for the commissioning of industrial automation systems. *In: 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. [S. l.: s. n.], 2015. p. 1–4. DOI: 10.1109/ETFA.2015.7301577.
- EVERINGHAM, Mark; VAN GOOL, Luc; WILLIAMS, Christopher K. I.; WINN, John; ZISSERMAN, Andrew. The PASCAL Visual Object Classes (VOC) Challenge. **International Journal of Computer Vision**, v. 88, n. 2, p. 303–338, 2010. DOI: 10.1007/s11263-009-0275-4.
- FOLGADO, Francisco Javier; CALDERÓN, David; GONZÁLEZ, Isaías; CALDERÓN, A. J. Review of Industry 4.0 from the Perspective of Automation and Supervision Systems: Definitions, Architectures and Recent Trends. **Electronics**, v. 13, n. 4, p. 782, 2024. DOI: 10.3390/electronics13040782.
- GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing**. 4. ed. [S. l.]: Pearson, 2018.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge: MIT Press, 2016.
- GREYLOGIX BRASIL. **GreyLogix: evoluindo indústrias com tecnologia e inovação**. [S. l.: s. n.], 2026. Disponível em: <https://greylogix.com.br/>.
- GRÜNER, Sten; HOERNICKE, Mario; STARK, Katharina; SCHOCH, Nicolai; ESKANDANI, Nafise; PRETLOVE, John. Towards asset administration shell-based continuous engineering in process industries. **at - Automatisierungstechnik**, v. 71, n. 8, p. 689–708, 2023. DOI: 10.1515/auto-2023-0012.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 10628-1:2014: Diagrams for the Chemical and Petrochemical Industry – Part 1: Specification of Diagrams**. [S. l.: s. n.], 2014. Disponível em: <https://www.iso.org/standard/51840.html>.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 10628-2:2012: Diagrams for the Chemical and Petrochemical Industry – Part 2: Graphical Symbols**. [S. l.: s. n.], 2012. Disponível em: <https://www.iso.org/standard/51841.html>.

INTERNATIONAL SOCIETY OF AUTOMATION. **ANSI/ISA-5.1-2024: Instrumentation and Control – Symbols and Identification**. [S. l.: s. n.], 2024. Disponível em: <https://www.isa.org/products/ansi-isa-5-1-2024-instrumentation-and-control-symb>.

JAMIESON, Laura; MORENO-GARCÍA, Carlos Francisco; ELYAN, Eyad. A review of deep learning methods for digitisation of complex documents and engineering diagrams. **Artificial Intelligence Review**, v. 57, n. 6, p. 136, 2024. DOI: 10.1007/s10462-024-10779-2.

JETLEY, Raoul; NAIR, Anil; CHANDRASEKARAN, Prakash; DUBEY, Alpana. Applying software engineering practices for development of industrial automation applications. *In: 2013 11th IEEE International Conference on Industrial Informatics (INDIN)*. [S. l.: s. n.], 2013. p. 558–563. DOI: 10.1109/INDIN.2013.6622945.

JIAO, Licheng; ZHANG, Fan; LIU, Fang; YANG, Shuyuan; LI, Lingling; FENG, Zhixi; QU, Rong. A Survey of Deep Learning-Based Object Detection. **IEEE Access**, v. 7, p. 128837–128868, 2019. DOI: 10.1109/ACCESS.2019.2939201.

KIM, Byung Chul; KIM, Hyunki; MOON, Yoochan; LEE, Gwang; MUN, Duhwan. End-to-end digitization of image format piping and instrumentation diagrams at an industrially applicable level. **Journal of Computational Design and Engineering**, v. 9, n. 4, p. 1298–1326, 2022. DOI: 10.1093/jcde/qwac056.

KOLTUN, Gennadiy; KOLTER, Moritz; VOGEL-HEUSER, Birgit. Automated Generation of Modular PLC Control Software from P&ID Diagrams in Process Industry. *In: 2018 IEEE International Systems Engineering Symposium (ISSE)*. [S. l.: s. n.], 2018. p. 1–8. DOI: 10.1109/SysEng.2018.8544401.

KOZIOLEK, Heiko; BURGER, Andreas; PLATENIUS-MOHR, Marie; JETLEY, Raoul. A classification framework for automated control code generation in industrial automation. **Journal of Systems and Software**, v. 166, p. 110575, 2020a. DOI: 10.1016/j.jss.2020.110575.

KOZIOLEK, Heiko; BURGER, Andreas; PLATENIUS-MOHR, Marie; RÜCKERT, Julius; ABUKWAIK, Hadil; JETLEY, Raoul; ABDULLA, P. P. Rule-based code generation in industrial automation: four large-scale case studies applying the CAYENNE method. *In: PROCEEDINGS of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice*. Seoul, South Korea: Association for Computing Machinery, 2020b. (ICSE-SEIP '20), p. 152–161. DOI: 10.1145/3377813.3381354.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet classification with deep convolutional neural networks. **Communications of the ACM**, v. 60, n. 6, p. 84–90, 2017. DOI: 10.1145/3065386.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep Learning. **Nature**, v. 521, p. 436–444, 2015. DOI: 10.1038/nature14539.

LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. DOI: 10.1109/5.726791.

MAGA, Camelia Rodica; JAZDI, Nasser; GÖHNER, Peter. Requirements on engineering tools for increasing reuse in industrial automation. *In: 2011 IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA)*. [S. l.: s. n.], 2011. p. 1–7. DOI: 10.1109/ETFA.2011.6058976.

MANI, Shouvik; HADDAD, Michael A.; CONSTANTINI, Dan; DOUHARD, Willy; LI, Qiwei; POIRIER, Louis. Automatic Digitization of Engineering Diagrams using Deep Learning and Graph Search. *In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S. l.: s. n.], 2020. p. 673–679. DOI: 10.1109/CVPRW50498.2020.00096.

PACONTROL.COM. **Instrumentation and Control: Process Control Fundamentals**. [S. l.: s. n.], 2006. Disponível em: https://faculty.ksu.edu.sa/sites/default/files/process_control_fundamentals.pdf.

PADILLA, Rafael; NETTO, Sergio L.; SILVA, Eduardo A. B. da. A Survey on Performance Metrics for Object-Detection Algorithms. *In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. [S. l.: s. n.], 2020. p. 237–242. DOI: 10.1109/IWSSIP48289.2020.9145130.

PALIWAL, Shubham; JAIN, Arushi; SHARMA, Monika; VIG, Lovekesh. **Digitize-PID: Automatic Digitization of Piping and Instrumentation Diagrams**. [S. l.: s. n.], 2021. arXiv preprint. DOI: 10.48550/arXiv.2109.03794. Disponível em: <https://arxiv.org/abs/2109.03794>.

PAPAKONSTANTINOU, Nikolaos; KARTTUNEN, Juha; SIERLA, Seppo; VYATKIN, Valeriy. Design to automation continuum for industrial processes: ISO 15926 – IEC 61131 versus an industrial case. *In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. [S. l.: s. n.], 2019. p. 1207–1212. DOI: 10.1109/ETFA.2019.8869325.

PLCOPEN. **IEC 61131-3 and PLCopen**. [S. l.: s. n.], 2025. Disponível em: <https://www.plcopen.org/standards/logic/iec-61131-3/>.

RAHUL, Rohit; PALIWAL, Shubham; SHARMA, Monika; VIG, Lovekesh. Automatic Information Extraction from Piping and Instrumentation Diagrams. *In: PROCEEDINGS of the 8th International Conference on Pattern Recognition Applications and Methods – ICPRAM*. [S. l.]: SciTePress, 2019. p. 163–172. DOI: 10.5220/0007376401630172.

REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. You Only Look Once: Unified, Real-Time Object Detection. *In: PROCEEDINGS of the IEEE Conference on Computer Vision and Pattern Recognition*. [S. l.: s. n.], 2016. p. 779–788. DOI: 10.1109/CVPR.2016.91.

REGULIN, Daniel; AICHER, Thomas; VOGEL-HEUSER, Birgit. Improving Transferability Between Different Engineering Stages in the Development of Automated Material Flow Modules. **IEEE Transactions on Automation Science and Engineering**, v. 13, n. 4, p. 1422–1432, 2016. DOI: 10.1109/TASE.2016.2576022.

ROBOFLOW UNIVERSE. **PID Symbols Dataset**. [S. l.: s. n.], 2023. Roboflow Universe. Disponível em: <https://universe.roboflow.com/maetee-lorprajuksiri-ue9no/pid-symbols-y1ew4>. Acesso em: 20 jun. 2026.

SEHR, Martin A.; LOHSTROH, Marten; WEBER, Mathew; UGALDE, Ines; WITTE, Martin; NEIDIG, Joerg; HOEME, Stephan; NIKNAMI, Mehrdad; LEE, Edward A. Programmable Logic Controllers in the Context of Industry 4.0. **IEEE Transactions on Industrial Informatics**, v. 17, n. 5, p. 3523–3533, 2021. DOI: 10.1109/TII.2020.3007764.

SIEMENS. **Programming language SCL – STEP 7**. [S. l.: s. n.], 2025. Disponível em: <https://docs.tia.siemens.cloud/r/en-us/v21/creating-scl-programs/basics-of-scl/programming-language-scl>.

SIEMENS. **The LAD and FBD programming languages**. [S. l.: s. n.], 2024. Disponível em:

<https://docs.tia.siemens.cloud/r/it-it/v20/simotion-scout-tia-device-configuration/getting-started-simotion-scout-tia-sample-project-simotion-d435-2/programming-the-simotion-application/lad/fbd-ladder-logic/function-block-diagram/the-lad-and-fbd-programming-languages>.

SIEMENS. **TIA Portal Openness: Automation of engineering workflows**. [S. l.: s. n.], 2026a. Disponível em:

<https://support.industry.siemens.com/cs/document/109792902/tia-portal-openness-automation-of-engineering-workflows?dti=0&lc=en-WW>.

SIEMENS. **TIA Portal: One platform for efficient engineering**. [S. l.: s. n.], 2026b. Disponível em: <https://www.siemens.com/us/en/products/automation/industry-software/automation-software/tia-portal.html>.

- SIEMENS AG. **SIMATIC STEP 7 Basic/Professional and WinCC: System Manual**. [S. l.], 2026. Disponível em: <https://support.industry.siemens.com/cs/document/109977280/simatic-step-7-basic-professional-und-simatic-wincc>.
- SIERLA, S.; SORSAMÄKI, L.; AZANGOO, Mohammad; VILLBERG, Antti; HYTÖNEN, E.; VYATKIN, V. Towards Semi-Automatic Generation of a Steady State Digital Twin of a Brownfield Process Plant. **Applied Sciences**, v. 10, n. 19, p. 6959, 2020. DOI: 10.3390/app10196959.
- SMITH, Ray. An Overview of the Tesseract OCR Engine. *In: PROCEEDINGS of the Ninth International Conference on Document Analysis and Recognition*. [S. l.: s. n.], 2007. p. 629–633. DOI: 10.1109/ICDAR.2007.4376991.
- STÜRMER, Jan Marius; GRAUMANN, Marius; KOCH, Tobias. From Engineering Diagrams to Graphs: Digitizing P&IDs with Transformers. *In: 2025 IEEE 12th International Conference on Data Science and Advanced Analytics (DSAA)*. [S. l.: s. n.], 2025. DOI: 10.1109/DSAA65442.2025.11248012.
- SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. 2. ed. [S. l.]: Springer, 2022.
- VILLENA TORO, Javier; WIBERG, Anton; TARKIAN, Mehrdad. Optical character recognition on engineering drawings to achieve automation in production quality control. **Frontiers in Manufacturing Technology**, v. 3, p. 1154132, 2023. DOI: 10.3389/fmtec.2023.1154132.
- VOGEL-HEUSER, Birgit; FISCHER, Juliane; FELDMANN, Stefan; ULEWICZ, Sebastian; RÖSCH, Susanne. Modularity and Architecture of PLC-Based Software for Automated Production Systems: An Analysis in Industrial Companies. **Journal of Systems and Software**, v. 131, p. 35–62, 2017. DOI: 10.1016/j.jss.2017.05.051.
- VOGEL-HEUSER, Birgit; FISCHER, Juliane; NEUMANN, Eva-Maria; DIEHM, Sebastian. Key maturity indicators for module libraries for PLC-based control software in the domain of automated Production Systems. **IFAC-PapersOnLine**, v. 51, n. 11, p. 1610–1617, 2018. DOI: 10.1016/j.ifacol.2018.08.261.
- VOGEL-HEUSER, Birgit; SCHÜTZ, Daniel; FRANK, Timo; LEGAT, Christoph. Model-driven engineering of Manufacturing Automation Software Projects – A SysML-based approach. **Mechatronics**, v. 24, n. 7, p. 883–897, 2014. DOI: 10.1016/j.mechatronics.2014.05.003.
- VYATKIN, Valeriy. Software Engineering in Industrial Automation: State-of-the-Art Review. **IEEE Transactions on Industrial Informatics**, v. 9, n. 3, p. 1234–1249, 2013. DOI: 10.1109/TII.2013.2258165.

WINKLER, Dietmar; BIFFL, Stefan. Improving Quality Assurance in Automation Systems Development Projects. *In*: SAVSAR, Mehmet (ed.). **Quality Assurance and Management**. [S. l.]: IntechOpen, 2012. DOI: 10.5772/33487.

ZOU, Zhengxia; CHEN, Keyan; SHI, Zhenwei; GUO, Yuhong; YE, Jieping. Object Detection in 20 Years: A Survey. **Proceedings of the IEEE**, v. 111, n. 3, p. 257–276, 2023. DOI: 10.1109/JPROC.2023.3238524.