

SISTEMA AUTOMATIZADO DE IDENTIFICAÇÃO DE PEÇAS

Mayara Vieira Morais¹ e Wesley Candido da Silva ²

Resumo

Este trabalho apresenta o desenvolvimento de um sistema automatizado para identificação de peças metálicas cortadas a laser, com o objetivo de eliminar falhas operacionais, reduzir o retrabalho e aumentar a eficiência no ambiente industrial. A proposta consiste na integração de um Controlador Lógico Programável (CLP), um robô cartesiano e uma impressora industrial, controlados por meio de dados extraídos automaticamente de arquivos de corte (.ncp) utilizando a biblioteca Pandas em Python. As coordenadas obtidas são processadas e enviadas ao CLP via a biblioteca Snap7, permitindo o posicionamento preciso do robô para aplicação de etiquetas identificadoras sobre as peças. O sistema foi testado com conjuntos de dados reais, demonstrando capacidade de leitura, filtragem e envio dos comandos ao CLP de forma confiável. O robô cartesiano, por sua vez, será responsável pelos movimentos e simulações de aplicação com precisão. Como resultado, a solução mostrou-se tecnicamente viável e alinhada aos princípios da Indústria 4.0, destacando-se como uma alternativa de baixo custo para pequenas e médias empresas que buscam modernizar processos de identificação de peças. O projeto também abre espaço para futuras expansões, como a aplicação prática em linha de produção.

Palavras-chave: Automação industrial. CLP. robô cartesiano. Python. Identificação.

Abstract

This work presents the development of an automated system for identifying laser-cut metal parts, with the aim of eliminating operational failures, reducing rework and increasing efficiency in the industrial environment. The proposal consists of integrating a Programmable Logic Controller (PLC), a Cartesian robot and an industrial printer, controlled by means of data automatically extracted from cutting files (.ncp) using the Pandas library in Python. The coordinates obtained are processed and sent to the PLC via the Snap7 library, allowing precise positioning of the robot for applying identification labels to the parts.

The system was tested with real data sets, demonstrating its ability to read, filter and send commands to the PLC reliably. The Cartesian robot, in turn, will be responsible for the movements and simulations of the application with precision. As a result, the solution proved to be technically viable and aligned with the principles of Industry 4.0, standing out as a low-cost alternative for small and medium-sized companies seeking to modernize part identification processes. The project also opens space for future expansions, such as practical application on the production line.

Keywords: Industrial automation. PLC. Cartesian robot. Python. Identification.

¹ UniSenaiPR e-mail: maay.v.m@gmail.com

² UniSenaiPR e-mail: Wesley.candido@sistemafiep.org.br

1 INTRODUÇÃO

A competitividade no setor industrial tem impulsionado empresas a buscarem constantemente maneiras de otimizar seus processos produtivos, reduzindo desperdícios e aumentando a eficiência operacional, sem comprometer a qualidade dos produtos. Esse movimento é especialmente perceptível na indústria metalúrgica, onde a produção em larga escala exige alto controle de qualidade e precisão nas operações. Segundo (SLACK; CHAMBERS; JOHNSTON, 2002), a melhoria contínua dos processos é fundamental para manter a competitividade, especialmente em ambientes com alta variabilidade de produtos e demanda por customização, como ocorre frequentemente na metalurgia.

Dentre as abordagens mais utilizadas para aprimorar processos industriais, destacam-se metodologias como Lean Manufacturing, Seis Sigma e a automação industrial, que visam eliminar gargalos, padronizar operações e garantir maior controle sobre a produção. A adoção de tecnologias inovadoras, como sistemas automatizados e softwares de controle, contribui diretamente para aumentar a rastreabilidade e reduzir falhas humanas nos processos produtivos.

Na empresa em que este trabalho foi desenvolvido, o processo de identificação das peças cortadas a laser é realizado manualmente. Esse processo envolve a impressão de etiquetas com identificadores únicos para cada peça, as quais são aplicadas uma a uma pelos operadores após o corte. As etiquetas são geradas a partir de uma planilha e dispostas em um rolo sequencial, sem seguir a ordem lógica de disposição das peças na chapa cortada, o que dificulta e atrasa a identificação correta. Esse cenário causa ociosidade de máquinas e retrabalho frequente, afetando diretamente a produtividade da linha. Segundo (MACHADO, 2020), a identificação automática de peças é essencial para garantir a rastreabilidade e a eficiência nos processos de manufatura, reduzindo erros e melhorando o controle de qualidade.

A identificação manual das peças tem se mostrado um gargalo significativo no processo produtivo da empresa. Quando não automatizado, esse procedimento apresenta alto risco de erros operacionais, como a troca de etiquetas, perda de rastreabilidade e, conseqüentemente, prejuízos à qualidade final do produto. (MONTGOMERY, 2019) destaca que a variabilidade dos processos industriais pode ser reduzida significativamente por meio da automação e do controle estatístico, resultando em maior confiabilidade e precisão na produção. Portanto, torna-se necessário implementar uma solução que reduza a interferência humana nesse processo, aumentando a produtividade e garantindo a rastreabilidade das peças.

Diante desse cenário, este trabalho visa implementar um sistema automatizado de identificação de peças cortadas a laser, com o uso de tecnologias como CLP (Controlador Lógico Programável), robô cartesiano e uma impressora de etiquetas. A proposta se fundamenta na integração de ferramentas de automação com metodologias de melhoria contínua, permitindo eliminar falhas humanas, reduzir desperdícios e aumentar a eficiência operacional. Além disso, o sistema fará uso da biblioteca Pandas, em Python, para processar dados extraídos dos planos de corte e automatizar a colagem das etiquetas com base na posição real das peças, proporcionando um processo mais ágil, seguro e rastreável.

A crescente demanda por qualidade, rastreabilidade e produtividade nos processos industriais impõe às empresas o desafio de reduzir falhas operacionais sem comprometer o ritmo de produção. No setor metalúrgico, onde a customização de peças e a diversidade de geometrias são frequentes, a identificação correta dos componentes torna-se essencial para garantir a continuidade dos processos produtivos e a satisfação do cliente final.

A realização manual da identificação das peças cortadas a laser tem gerado recorrentes falhas na

operação da empresa estudada, como a troca de etiquetas, retrabalhos e até o envio incorreto de peças aos clientes. Esses problemas acarretam desperdício de tempo, recursos e impactam negativamente a imagem da empresa.

Nesse contexto, a automação do processo de identificação surge como uma alternativa estratégica para mitigar os riscos operacionais, garantir a rastreabilidade das peças e aumentar a produtividade da linha. Com o uso de tecnologias acessíveis como CLP, sensores, impressoras industriais e scripts de análise de dados com a biblioteca Pandas, é possível criar uma solução personalizada e eficiente para os desafios enfrentados atualmente.

De forma geral o objetivo deste trabalho é implementar um sistema automatizado de identificação de peças cortadas a laser, com o uso de tecnologias de automação industrial e processamento de dados, visando melhorar a rastreabilidade, reduzir falhas operacionais e otimizar a produtividade no processo produtivo.

De modo a analisar o processo atual de identificação de peças na empresa e mapear os principais pontos de falha, desenvolver uma solução com base em automação industrial, utilizando CLP, sensores e impressora de etiquetas, aplicar uma solução para leitura e extração automatizada das coordenadas dos planos de corte, integrar o sistema de automação com os dados processados, visando garantir a impressão correta das etiquetas conforme a posição real das peças e avaliar os resultados obtidos com a implementação do sistema em termos de produtividade, redução de erros e tempo de operação.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo aborda os principais conceitos teóricos relacionados ao processo fabril tratado neste trabalho, que envolve o corte de chapas metálicas por meio de tecnologia a laser, seguido pela identificação automatizada das peças produzidas. Para viabilizar essa automação, utilizam-se controladores lógicos programáveis (CLPs), robô cartesiano e sistemas de comunicação industrial. Além disso, são empregadas bibliotecas da linguagem de programação Python, como o Pandas e o Snap7, com o objetivo de integrar dados de produção ao controle automatizado. Assim, este capítulo está estruturado da seguinte forma: a Seção 2.1 apresenta os fundamentos do corte a laser; a Seção 2.2 trata da linguagem Python e das bibliotecas utilizadas; a Seção 2.3 explora o funcionamento do CLP e sua aplicação na indústria; a Seção 2.4 aborda os principais protocolos de redes industriais; e, por fim, a Seção 2.5 descreve o sistema de coordenadas utilizado em robôs cartesianos.

2.1 Corte a Laser

O corte a laser é um processo amplamente utilizado na indústria para a fabricação de peças metálicas, plásticas e outros materiais, garantindo alta precisão e velocidade na produção. Esse método utiliza um feixe de laser de alta potência para fundir, queimar ou vaporizar o material, realizando cortes complexos com mínima interferência mecânica.

De acordo com Steen e Mazumder ([STEEN; MAZUMDER, 2010](#)), o corte a laser é considerado um dos métodos mais eficientes para o processamento de materiais, devido à precisão, repetibilidade e capacidade de cortar geometrias complexas com alta velocidade e baixa geração de rebarbas.

A utilização do corte a laser proporciona diversas vantagens para a fabricação industrial, incluindo: alta precisão e repetibilidade dos cortes, redução do desperdício de material devido à otimização do nesting (arranjo das peças na chapa), possibilidade de cortes complexos e detalhados e automação do processo, reduzindo a dependência de operadores para ajustes manuais.

Segundo Powell (POWELL, 2017), a evolução dos sistemas de corte a laser permitiu a integração com tecnologias digitais, aumentando a eficiência produtiva e garantindo maior controle de qualidade nas peças fabricadas.

Principais tipos de corte a laser: CO₂ e Fibra.

A principal diferença entre essas tecnologias está no tipo de laser utilizado e na forma como a energia é gerada e transmitida ao material.

O laser de CO₂ utiliza um gás como meio ativo para a geração do feixe de laser, composto principalmente por dióxido de carbono, nitrogênio e hélio. O feixe gerado possui comprimento de onda de aproximadamente 10,6 µm, sendo eficaz para cortar materiais não metálicos e metais mais espessos.

O laser de fibra utiliza uma fibra óptica dopada com elementos como érbio, ítrio ou neodímio para amplificar o feixe de laser. Esse feixe apresenta comprimento de onda menor (1,06 µm), possibilitando uma absorção mais eficiente em metais e sendo ideal para o corte de aço carbono, aço inoxidável e alumínio.

As principais características do Corte a Laser CO₂ e fibra são apresentados na Tabela 1:

Tabela 1 – Fibra X CO₂.

Laser CO ₂	Laser Fibra
Alta qualidade no acabamento das bordas. Melhor desempenho em cortes de chapas espessas.	Maior eficiência energética em comparação ao CO ₂ . Menos manutenção, pois não há troca de espelhos ou gás.
Aplicação em materiais orgânicos (madeira, acrílico, etc.) e metálicos. Maior necessidade de manutenção devido ao desgaste dos espelhos e do tubo de gás.	Alta velocidade de corte, principalmente em chapas finas. Melhor desempenho no corte de materiais altamente refletivos, como alumínio e cobre.

Fonte: Autoria própria (2025).

2.2 Ferramentas em Python

2.2.1 Pandas

O Pandas é uma biblioteca de código aberto desenvolvida para a manipulação e análise de dados em Python. Ela fornece estruturas de dados eficientes, como DataFrames e Series, permitindo a leitura, processamento e exportação de grandes volumes de informações de maneira otimizada (DADOS, 2025).

De acordo com McKinney (MCKINNEY, 2017), o Pandas é uma ferramenta essencial para a análise de dados, permitindo a manipulação eficiente e estruturada de grandes conjuntos de informações por meio de operações vetorizadas e indexação avançada. A estrutura fundamental do Pandas, o DataFrame, pode ser compreendida como uma tabela, similar às utilizadas em softwares como Excel ou bancos de dados SQL, onde os dados são organizados em linhas e colunas.

Além das funcionalidades básicas de manipulação de dados, a biblioteca Pandas oferece recursos avançados que foram explorados neste projeto para tornar o processo mais robusto e eficiente. Por meio da função `read_excel()`, foi possível importar diretamente as informações dos planos de corte, facilitando o acesso às coordenadas de posicionamento das peças.

Em conjunto com métodos como `dropna()` e `fillna()`, foi possível tratar dados inconsistentes ou ausentes, garantindo maior confiabilidade nas leituras. O projeto também se beneficiou da capacidade do Pandas de realizar filtragens condicionais, ordenações (`sort_values()`) e agrupamentos (`groupby()`),

permitindo segmentar dados conforme critérios definidos, como tipo de peça ou posição na chapa.

Essas operações são fundamentais para organizar as informações de forma lógica antes do envio ao sistema de automação. Segundo (MCKINNEY, 2017), o uso dessas operações vetorizadas e estruturadas é justamente o que torna o Pandas uma ferramenta poderosa na análise de grandes volumes de dados, facilitando tanto a clareza quanto a eficiência dos processos computacionais.

Outro aspecto importante foi a integração do Pandas com bibliotecas auxiliares, como OpenPyXL, utilizada para leitura de planilhas com múltiplas abas, e os/glob, que permitem a automação da busca por arquivos em diretórios específicos. Essa integração ampliou a flexibilidade da aplicação, tornando-a adaptável a diferentes formatos e fluxos de entrada.

Do ponto de vista prático, a adoção da biblioteca Pandas reduziu significativamente o tempo necessário para processar os arquivos e eliminou etapas manuais sujeitas a falhas humanas. Isso permitiu maior fluidez na comunicação entre o software e os dispositivos físicos, como o CLP e a impressora, fortalecendo a confiabilidade e a repetibilidade do processo de identificação automatizada.

Na automação industrial ou em contextos de processamento de dados, o conceito de “sistema de varredura” refere-se a um processo de leitura sistemática de informações, utilizado para detectar, extrair ou inspecionar dados de forma contínua ou conforme uma rotina predefinida. Neste trabalho, o sistema de varredura está relacionado à leitura automatizada de arquivos de corte (como arquivos .ncp) ou de tabelas (.xlsx), extraindo coordenadas automaticamente. A biblioteca Pandas é empregada como ferramenta para implementar esse sistema de varredura via software. A Tabela 2 apresenta uma correlação entre o conceito de sistema de varredura e sua implementação utilizando a biblioteca Pandas.

Tabela 2 – Sistema de Varredura X Pandas.

Conceito do Sistema de Varredura	Implementação com Pandas
Varredura dos dados (linhas, colunas)	Percorre o arquivo linha a linha (usando <code>.iterrows()</code> ou filtros)
Extração de coordenadas	Acessa colunas como <code>df['X']</code> , <code>df['Y']</code>
Deteção de padrões	Aplica filtros, condições (<code>df[df['X'] > 0]</code>)
Automatização de leitura e organização	Organiza os dados em DataFrames e trata com rapidez

Fonte: Autoria própria (2025).

2.2.2 Snap7

A biblioteca `python-snap7` é uma interface desenvolvida em Python para permitir a comunicação direta com Controladores Lógicos Programáveis (CLPs) da Siemens, tais como os modelos das séries S7-300, S7-400, S7-1200 e S7-1500. Ela se baseia na biblioteca em C++ chamada Snap7, cuja finalidade é fornecer uma comunicação nativa com esses dispositivos por meio do protocolo S7, amplamente utilizado em sistemas de automação industrial (MOLENAAR, 2025).

Dentro da `python-snap7`, destaca-se o módulo `snap7.util`, que disponibiliza funções auxiliares como `set_real`, `get_real`, `set_dword` e `get_int`, responsáveis por converter e manipular dados binários de forma estruturada, de acordo com os tipos de dados utilizados internamente pelos CLPs Siemens (PYTHON-SNAP7, 2025). Essas funções são fundamentais para garantir que os dados enviados para o CLP estejam devidamente formatados na memória, evitando inconsistências e falhas na leitura ou escrita de variáveis de processo.

Segundo (BRAGA, 2015), o uso de bibliotecas que promovem a comunicação entre linguagens de alto nível, como o Python, e CLPs industriais representa uma tendência crescente na automação, pois oferece maior flexibilidade e integração com sistemas supervisórios, bancos de dados e interfaces homem-máquina (IHMs). Nesse contexto, a `python-snap7` tem se consolidado como uma ferramenta

robusta e acessível para desenvolvedores e pesquisadores que buscam soluções em automação de baixo custo e com alto grau de personalização.

Além disso, seu uso em projetos acadêmicos e industriais é reforçado pela comunidade open source, que fornece exemplos e documentação contínua, como observado em diversos tutoriais e guias de integração ([Simply Automationized, 2016](#)).

2.3 Controlador Lógico Programável (CLP)

Segundo Oliveira ([OLIVEIRA, 2012b](#)), a automação consiste na utilização de sistemas eletrônicos para o controle automatizado de processos, com o objetivo de substituir a ação humana em tarefas perigosas ou de grande repetitividade, melhorando a execução das atividades.

O Controlador Lógico Programável (CLP) é um dispositivo eletrônico projetado para automatizar processos industriais, substituindo circuitos lógicos convencionais e sistemas baseados em relés eletromecânicos. Ele é programado para realizar operações sequenciais, monitorar sensores e controlar atuadores conforme uma lógica predefinida.

Sua linguagem de programação pode ser gráfica, baseada em lógica combinacional ou sequencial, denominada linguagem Ladder. De acordo com Oliveira ([OLIVEIRA, 2012b](#)), os CLPs também permitem a especificação de sistemas de controle em outras linguagens, como STL (Statement List) ou Grafset, sendo largamente utilizados devido à sua robustez para operar em ambientes hostis e à facilidade de uso.

O funcionamento do CLP baseia-se no ciclo de varredura (scan), no qual inicia-se com a leitura do estado das entradas, armazenando essas informações em variáveis temporárias na memória de dados. Em seguida, são executadas operações lógicas e aritméticas utilizando os dados das entradas para a tomada de decisões e definição dos estados das saídas.

Posteriormente, os valores lógicos das saídas são enviados para a interface física, comandando a execução do processo. Esse ciclo se repete continuamente até que o sistema seja desligado. As quatro etapas principais do ciclo de funcionamento do CLP são:

1. Leitura das entradas: O CLP lê os sinais provenientes de sensores, botões, interfaces homem-máquina (IHM) e outros dispositivos.
2. Execução do programa: Com base na lógica programada, o CLP processa as informações e toma decisões.
3. Atualização das saídas: Após o processamento, o CLP aciona atuadores como motores, válvulas, a impressora de etiquetas e o robô cartesiano, no caso deste trabalho.
4. Comunicação e diagnóstico: O CLP registra eventos, envia dados para supervisórios (SCADA) e diagnostica falhas no sistema.

2.4 Redes Industriais – Ethernet

As redes industriais desempenham um papel essencial na automação dos processos produtivos, permitindo a comunicação eficiente entre sensores, atuadores, controladores e sistemas supervisórios. Elas são projetadas para operar em ambientes exigentes, onde a robustez, a confiabilidade e o tempo de resposta são fatores críticos.

Segundo Barattella ([BARATTELLA, 2011](#)), essas redes proporcionam maior flexibilidade e modularidade aos sistemas, além de reduzir a complexidade do cabeamento e facilitar a implementação de diagnósticos remotos e manutenção preditiva. Dessa forma, contribuem diretamente para o aumento

da eficiência operacional e da disponibilidade dos equipamentos industriais.

Com o avanço das tecnologias de comunicação, protocolos baseados em Ethernet passaram a ser amplamente utilizados na indústria, promovendo maior integração entre os sistemas de automação e os ambientes corporativos.

A Ethernet industrial baseia-se nos mesmos princípios da Ethernet convencional, utilizando o modelo de camadas da ISO/OSI. No entanto, destaca-se o uso do conjunto de protocolos TCP/IP (Transmission Control Protocol / Internet Protocol), responsável por garantir a entrega confiável dos dados (no caso do TCP) e o roteamento eficiente entre diferentes dispositivos da rede (no caso do IP).

O TCP/IP constitui a base da comunicação em redes Ethernet industriais, encapsulando os dados da aplicação e garantindo sua entrega íntegra ao destino, mesmo em ambientes com ruídos e interferências. Para aplicações que exigem tempo real ou baixa latência, outros protocolos podem ser utilizados sobre Ethernet, como o UDP (User Datagram Protocol) ou protocolos proprietários otimizados para comunicação determinística.

No ambiente industrial, o uso do protocolo Ethernet permite a comunicação entre dispositivos de campo, controladores lógicos programáveis (CLPs) e sistemas de supervisão com alta confiabilidade e velocidade, além de compatibilidade com diversas tecnologias.

Barattella ([BARATTELLA, 2011](#)) também destaca que a principal vantagem da Ethernet na indústria está na sua capacidade de integração entre os níveis operacional e corporativo, possibilitando uma comunicação contínua desde os sensores de chão de fábrica até os servidores de gerenciamento e planejamento.

2.5 Sistema de coordenadas em robô cartesiano

As coordenadas cartesianas, conceito desenvolvido pelo filósofo francês René Descartes (1596–1650), constituem um sistema matemático utilizado em geometria analítica para localizar um ponto no espaço em relação a dois ou três eixos. Esse sistema é amplamente utilizado em diversas áreas, incluindo automação e robótica, especialmente na programação de robôs cartesianos.

Em um sistema bidimensional, há o eixo vertical, denominado eixo dos y , e o eixo horizontal, denominado eixo dos x . O ponto onde ambos os eixos se intersectam é denominado origem. Quando os valores de y são inferiores a zero, os pontos se encontram abaixo da origem; de forma análoga, valores negativos de x representam pontos à esquerda da origem. A localização de um ponto é feita pela interseção das perpendiculares aos dois eixos, que passam por esse ponto. A ordenada refere-se à distância ao eixo dos x , enquanto a abscissa é a distância ao eixo dos y .

A posição de qualquer ponto em um sistema de duas dimensões é representada numericamente pelos valores da abscissa e da ordenada, indicando sua posição no plano cartesiano, e é escrita como um par de números separados por vírgula dentro de parênteses, como por exemplo (x, y) .

Quando se trata de um sistema tridimensional, um terceiro eixo, denominado z , é introduzido, permitindo localizar um ponto em relação a esse novo eixo, além dos eixos x e y . Esse sistema tridimensional é fundamental para aplicações em robótica, como no caso de robôs cartesianos. De acordo com Santos ([SANTOS, 2016](#)), em robôs cartesianos a movimentação ocorre ao longo dos três eixos (x, y, z) , permitindo que o manipulador se mova de forma precisa em um espaço tridimensional, o que é especialmente útil em tarefas de automação industrial, como montagem e controle de processos.

O robô cartesiano é um tipo de robô industrial que utiliza um sistema de coordenadas cartesianas para controlar sua posição. Esse tipo de robô é frequentemente utilizado em tarefas de *pick and place*, onde objetos são movidos de um local para outro de forma precisa. Segundo Silva ([SILVA, 2017](#)), os

robôs cartesianos oferecem vantagens significativas em termos de custo e simplicidade de controle, tornando-os ideais para tarefas repetitivas que exigem precisão no posicionamento.

Esse sistema de coordenadas é fundamental também na programação de CLPs (Controladores Lógicos Programáveis), que controlam as movimentações do robô cartesiano em ambientes industriais. De acordo com Oliveira (OLIVEIRA, 2012a), por meio de um ciclo de varredura, o CLP monitora as posições do robô ao longo dos eixos cartesianos e ajusta os comandos enviados aos atuadores, garantindo a precisão desejada na execução das tarefas.

3 METODOLOGIA

A metodologia adotada neste trabalho foi organizada em etapas, de modo a permitir uma abordagem sistemática para análise, desenvolvimento e implementação do sistema de automação proposto. As etapas incluem o estudo do processo atual, a elaboração da solução automatizada, o desenvolvimento do sistema de processamento de dados, a integração dos componentes e a avaliação dos resultados obtidos.

3.1 Análise do Processo Atual

Atualmente, o processo de identificação de peças na etapa de corte a laser é realizado de forma manual. O operador da máquina é responsável por imprimir as etiquetas de acordo com o lote que será processado. Esses lotes costumam apresentar quantidades elevadas de peças, podendo ultrapassar 500 unidades, distribuídas em múltiplas chapas conforme o planejamento de produção.

O termo *nesting* refere-se ao agrupamento de peças em uma única chapa, visando a otimização do uso do material. Cada nesting pode conter dezenas de peças, o que eleva a complexidade do processo de identificação, especialmente quando as peças possuem geometrias semelhantes, porém com variações nas especificações. Na Figura 1, observa-se um exemplo de nesting com 25 peças distribuídas em uma única chapa.

Figura 1 – Exemplo de nesting com múltiplas peças.



Fonte: Autoria própria (2025).

O setor de planejamento de produção é responsável pela geração da lista de produção, na qual constam todas as ordens de fabricação programadas para o dia ou para a semana. Essa lista é organizada em lotes que podem ser divididos conforme a quantidade de peças, matéria-prima, máquina utilizada e setor de montagem, visando facilitar o fluxo produtivo.

No momento da montagem dos planos de corte, o operador insere manualmente uma coluna denominada “Posição” na planilha. Este campo representa o número identificador de cada peça dentro do nesting, servindo como referência para a etapa de etiquetagem posterior.

Durante o processo de geração dos nestings, o software de otimização organiza as peças de maneira a maximizar o aproveitamento da chapa, minimizando desperdícios — conhecidos como *scrap*. Contudo, essa organização não segue a ordem definida na coluna de “Posição” da planilha, o que resulta na ausência de uma sequência lógica entre o layout do nesting e a ordem das etiquetas.

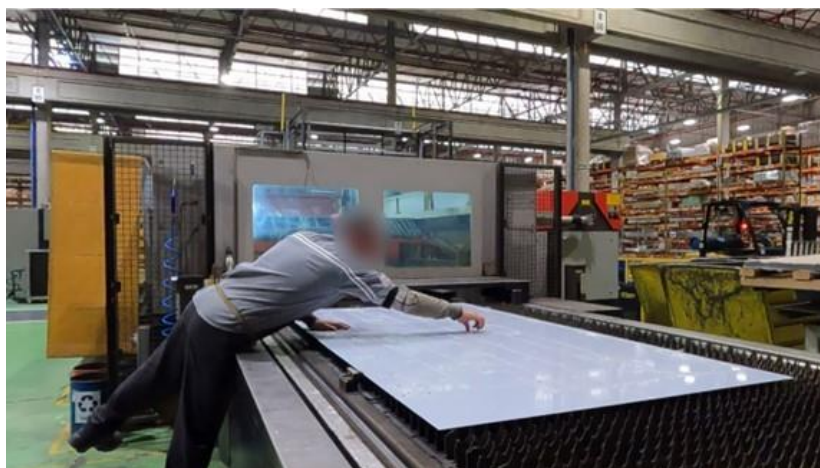
Após a finalização dos nestings, o técnico responsável comunica ao setor de planejamento, que, por sua vez, realiza a impressão dos arquivos PDF contendo os planos de corte, além do carregamento dos programas nas máquinas.

Cada peça, conforme demonstrado na Figura 1, possui seu identificador posicionado graficamente no centro da geometria dentro do PDF do nesting. Ao iniciar a operação de corte, o operador imprime as etiquetas correspondentes ao lote. Entretanto, as etiquetas são geradas sem uma ordem sequencial alinhada ao posicionamento das peças no nesting. Isso gera um entrave operacional, pois, ao término do corte de cada chapa, o operador precisa consultar os PDFs dos planos de corte, identificar visualmente cada peça e, manualmente, localizar no rolo de etiquetas aquela que corresponde à peça cortada. Esse procedimento é ilustrado na Figura 2.

Para garantir que não haja parada da máquina, o operador deve executar essa tarefa de forma ágil. As máquinas de corte a laser possuem duas mesas rotativas. Enquanto uma chapa está sendo cortada, o operador prepara a próxima chapa na segunda mesa. Assim que o corte da primeira chapa é concluído, ocorre a troca das mesas: a chapa cortada é deslocada para fora da máquina e a chapa seguinte inicia imediatamente o corte.

Nesse intervalo, o operador precisa identificar todas as peças da chapa recém-cortada, utilizando os PDFs dos nestings e localizando manualmente as etiquetas. Após a identificação, as peças são organizadas nas gôndolas, que serão posteriormente encaminhadas ao setor de dobra. Concluído esse

Figura 2 – Processo atual de identificação manual das peças.



Fonte: Autoria própria (2025).

processo, o operador remove o esqueleto da chapa (material residual), acondicionando-o na caçamba de sucata, e inicia a preparação da próxima chapa.

Todo esse fluxo, desde a identificação das peças até a organização nas gôndolas e a remoção do scrap, deve ser realizado dentro do tempo de corte da segunda chapa. Caso contrário, há parada de máquina, impactando diretamente na eficiência produtiva.

Inicialmente, foi realizado o levantamento e o mapeamento detalhado do processo manual de identificação das peças cortadas a laser. Essa análise teve como objetivo identificar os principais gargalos, falhas operacionais e fontes de desperdício que comprometem tanto a produtividade quanto a qualidade do processo.

A coleta de dados foi conduzida por meio de observação direta das atividades de fabricação e identificação, entrevistas informais com os operadores do turno A e C e análise documental dos nestings. As informações levantadas serviram como base para o desenvolvimento da proposta de automação do processo de identificação, apresentada nos tópicos subsequentes.

3.2 Desenvolvimento do Sistema de Automação

Com base nas informações levantadas na etapa de análise do processo, foi desenvolvida uma solução baseada na automação do processo de identificação de peças. Essa solução é composta por um Controlador Lógico Programável (CLP), um robô cartesiano, uma impressora de etiquetas industrial, um sistema de extração de dados dos nestings e ajustes no processo operacional.

No processo anterior, os programas das peças enviados para o nesting não continham informações de coordenadas específicas para a identificação das peças. Na proposta desenvolvida, os programas passam a incorporar uma coordenada no eixo XY, definida pelo programador. Essa escolha permite alocar a etiqueta em uma região da peça que não comprometa operações futuras, como cortes, furações ou dobras. As coordenadas são extraídas automaticamente por meio de um sistema desenvolvido em Python.

Durante os testes de bancada, foi estabelecida a comunicação entre o sistema de extração das coordenadas, utilizando a biblioteca Snap7, e o CLP Siemens. A comunicação apresentou desempenho satisfatório. No ambiente de programação TIA Portal, foi desenvolvido um bloco de dados (DB) específico para o armazenamento das informações extraídas dos nestings, sendo criadas duas variáveis de memória — COORD_X e COORD_Y — destinadas, respectivamente, às coordenadas nos eixos X e Y.

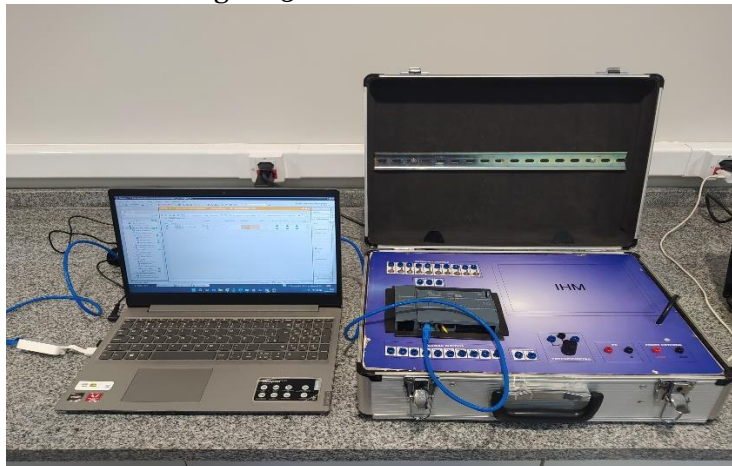
Na figura 3 temos a ligação da bancada de teste, uma maleta didática da instituição com CLP Siemens modelo S7-1200.

Nos testes, foi considerado um tempo médio de 1 segundo para a atualização das coordenadas, tendo em vista que, em uma única chapa, existem diversas peças a serem identificadas. O tempo de acionamento da impressão é controlado pelo próprio CLP, que libera o sinal de impressão apenas após a confirmação da aplicação da etiqueta anterior.

O CLP é responsável por gerenciar toda a lógica operacional, incluindo os movimentos do robô cartesiano e o acionamento da impressora de etiquetas. O robô foi projetado para realizar o posicionamento preciso das etiquetas sobre as peças, de forma sincronizada com a impressora, garantindo assertividade e fluidez no processo.

O desenvolvimento da estrutura mecânica, bem como a programação do robô cartesiano, foi realizado com foco em garantir alta precisão e agilidade na aplicação das etiquetas.

Figura 3 – Bancada de Teste.



Fonte: Autoria própria (2025).

3.3 Processamento dos Dados com Python

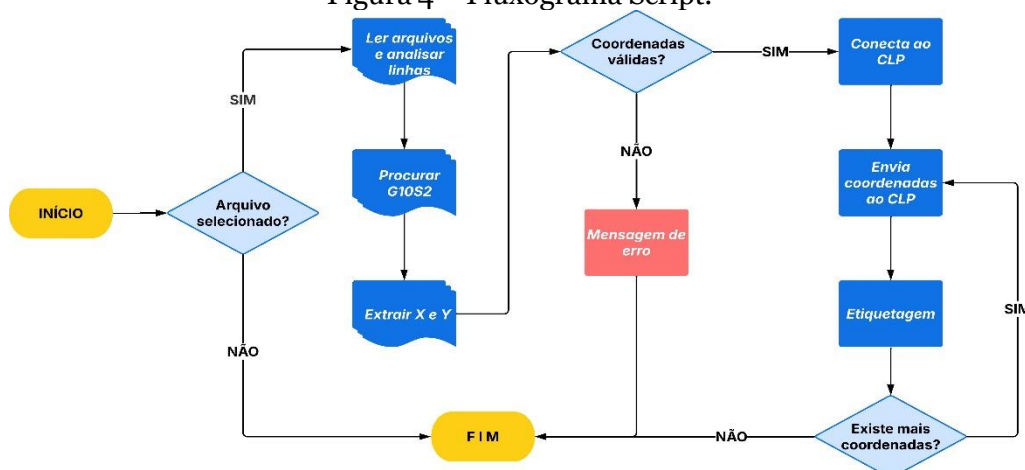
Para viabilizar a geração automatizada das etiquetas, foi desenvolvido um script em linguagem Python, utilizando as bibliotecas *Pandas* para tratamento e manipulação de dados, e *Snap7* para a comunicação com o Controlador Lógico Programável (CLP).

O script tem como função principal realizar a leitura dos arquivos de plano de corte (NCP), efetuando a extração das coordenadas de identificação das peças e estruturando as informações necessárias para o processo de impressão das etiquetas. Dessa forma, garante-se que cada etiqueta gerada corresponda corretamente à posição real da peça na chapa após o corte.

O processamento dos dados contempla etapas como filtragem das informações relevantes, organização dos dados em uma sequência lógica e posterior exportação para comunicação com o CLP. Essa integração permite que o sistema opere de forma sincronizada com o robô cartesiano e a impressora de etiquetas, assegurando a correta identificação das peças no processo produtivo.

De forma geral a figura 4 representa o fluxo do funcionamento do sistema de extração de coordenadas.

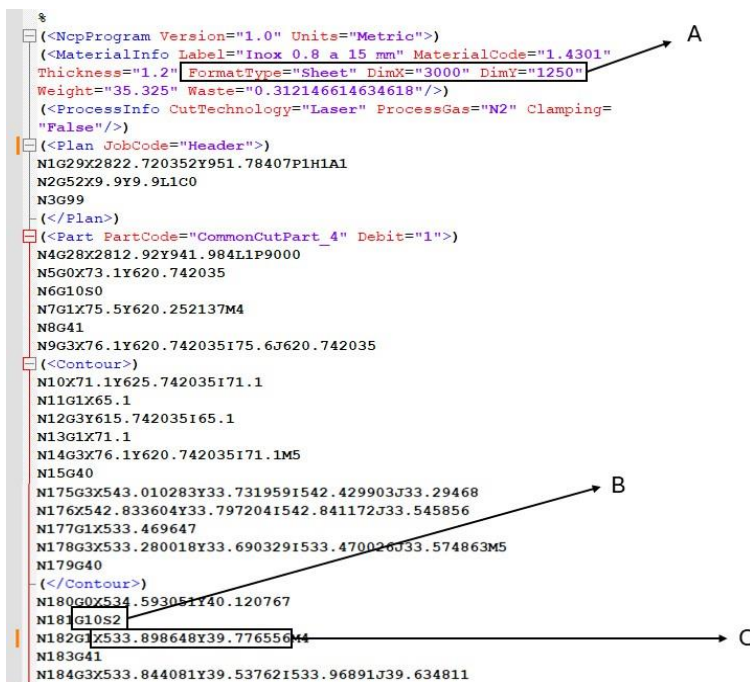
Figura 4 – Fluxograma Script.



Fonte: Autoria própria (2025).

Nos testes iniciais, o script desenvolvido em Python era responsável apenas pela extração direta das coordenadas presentes nos arquivos de plano de corte (NCP), sem a implementação de rotinas de validação. Entretanto, durante o desenvolvimento e os testes, foram identificadas algumas inconsistências que poderiam comprometer a operação do sistema.

Figura 5 – Trecho do arquivo NCP em linguagem G-code, utilizado na extração das coordenadas.



```

%
(<NcpProgram Version="1.0" Units="Metric">)
(<MaterialInfo Label="Inox 0.8 a 15 mm" MaterialCode="1.4301"
Thickness="1.2" FormatType="Sheet" DimX="3000" DimY="1250"
Weight="35.325" Waste="0.312146614634618"/>)
(<ProcessInfo CutTechnology="Laser" ProcessGas="N2" Clamping=
"False"/>)
(<Plan JobCode="Header">)
N1G29X2822.720352Y951.78407P1H1A1
N2G52X9.9Y9.9L1C0
N3G99
(</Plan>)
(<Part PartCode="CommonCutPart_4" Debit="1">)
N4G28X2812.92Y941.984L1P9000
N5G0X73.1Y620.742035
N6G10S0
N7G1X75.5Y620.252137M4
N8G41
N9G3X76.1Y620.742035I75.6J620.742035
(<Contour>)
N10X71.1Y625.742035I71.1
N11G1X65.1
N12G3Y615.742035I65.1
N13G1X71.1
N14G3X76.1Y620.742035I71.1M5
N15G40
N175G3X543.010283Y33.731959I542.429903J33.29468
N176X542.833604Y33.797204I542.841172J33.545856
N177G1X533.469647
N178G3X533.280018Y33.690329I533.470028J33.574863M5
N179G40
(</Contour>)
N180G0X534.59305Y140.120767
N181G10S2
N182G1X533.898648Y39.776556
N183G41
N184G3X533.844081Y39.53762I533.96891J39.634811
  
```

Fonte: Autoria própria (2025).

Foram realizadas simulações com diferentes cenários de dados inválidos, tais como:

Coordenadas negativas, que poderiam indicar erros de digitação ou exportação;

Inserção de caracteres especiais dentro dos valores numéricos, que causavam a leitura incorreta dos dados;

Coordenadas fora dos limites físicos da chapa, que poderiam gerar falhas no posicionamento do robô.

Na Figura 5 é apresentado um trecho do arquivo NCP, escrito na linguagem de máquina G-code, utilizado na geração dos planos de corte.

No ponto A, é possível identificar as dimensões da chapa, informações essenciais para que o script verifique se existe algum dado fora dos limites aceitáveis, funcionando como uma análise de *outliers*.

No ponto B, observa-se o termo G10S2, que foi adotado como referência para localizar no arquivo as coordenadas de interesse. As informações de posição demonstradas no ponto C, expressas pelos valores de X e Y imediatamente abaixo desse comando, são extraídas e enviadas ao CLP, permitindo a correta identificação e etiquetagem das peças.

3.4 Integração dos Sistemas

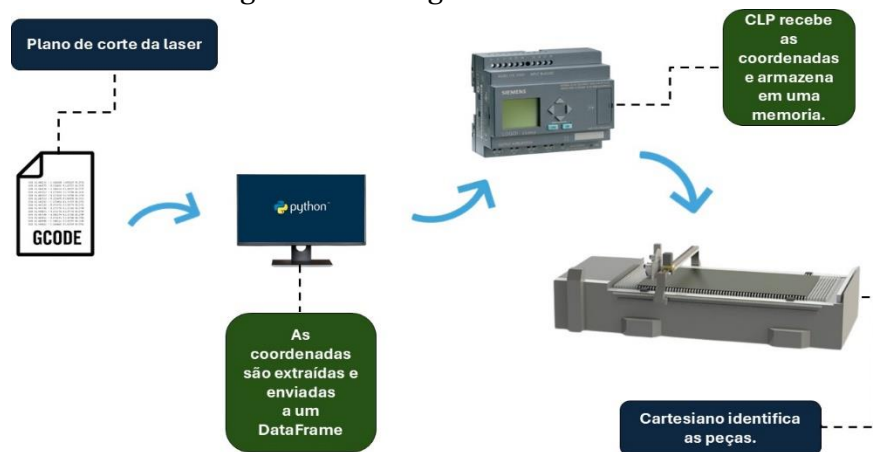
Após o desenvolvimento individual dos componentes, foi realizada a integração entre o sistema de automação e o sistema de processamento de dados.

O Controlador Lógico Programável (CLP) foi programado para interpretar as informações recebidas do script desenvolvido em Python e, a partir disso, comandar o robô cartesiano e a impressora de etiquetas, seguindo a sequência de identificação das peças estabelecida.

Essa integração viabilizará a automatização completa do processo de identificação, reduzindo significativamente a interferência humana e, conseqüentemente, minimizando a ocorrência de erros operacionais que anteriormente impactavam a produtividade e a qualidade do processo.

A Figura 6 ilustra o fluxo de integração entre os sistemas de automação e processamento de dados. Cada plano de corte (nesting) gera um arquivo em linguagem G-code que é enviado para a máquina de corte a laser. Por meio do script em Python, esses arquivos são acessados e processados, realizando a extração das coordenadas X e Y referentes à posição das etiquetas. Essas informações são então enviadas ao CLP, que por sua vez gerencia os movimentos do robô cartesiano e sincroniza a impressão e aplicação das etiquetas.

Figura 6 – Fluxograma do Processo.



Fonte: Autoria própria (2025).

Por fim, foi realizada uma avaliação de desempenho do sistema desenvolvido, com a comparação entre os métodos manual e automatizado. Foram analisados indicadores-chave, como tempo de operação, taxa de erros na identificação e produtividade. A análise comparativa possibilitou não apenas validar a eficácia da solução proposta, mas também identificar potenciais oportunidades de aperfeiçoamento. Diante desse contexto, a próxima seção apresenta a Discussão dos Resultados, onde são detalhados os ganhos operacionais, as melhorias obtidas e os impactos da implementação do sistema no processo produtivo.

4 APRESENTAÇÃO DE DISCUSSÃO DOS RESULTADOS

A implementação do sistema proposto teve como principal objetivo validar sua eficiência na identificação automatizada das peças cortadas a laser, visando reduzir erros operacionais e aumentar a produtividade no processo produtivo da empresa. Nesta seção, são apresentados os resultados obtidos por meio dos testes realizados, bem como a análise comparativa entre os dados gerados pelo código desenvolvido, os valores recebidos pelo CLP e os desvios identificados durante a validação do sistema.

Foram realizados testes no sistema em 105 nestings resultando na extração de 1116 conjuntos de coordenadas, sendo 5 desses nestings colocados com desvios para validar o sistema, como pode se notar na Figura 7.

Dos nestings testados, foram identificados os 5 que continham erros, e as coordenadas extraídas somente dos arquivos válidos, mostrando uma coerência no sistema.

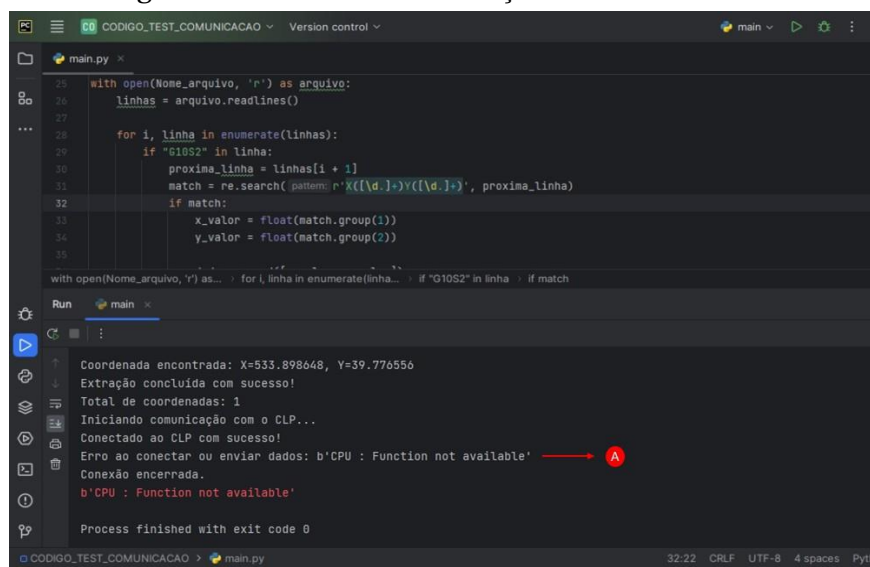
Figura 7 – Análise do sistema de extração de coordenadas.



Fonte: Autoria própria (2025).

No teste inicial, temos o envio de apenas uma coordenada para validar a comunicação com o CLP Siemens, a coordenada foi extraída e apresentada no console na Figura 8, a comunicação foi estabelecida no entanto é possível notar no ponto A que houve uma falha no envio dos dados.

Figura 8 – Teste de Comunicação com CLP - Falha.



```

25 with open(Nome_arquivo, 'r') as arquivo:
26     linhas = arquivo.readlines()
27
28     for i, linha in enumerate(linhas):
29         if "G10S2" in linha:
30             proxima_linha = linhas[i + 1]
31             match = re.search(pattern: r'X([\d.]+)Y([\d.]+)', proxima_linha)
32             if match:
33                 x_valor = float(match.group(1))
34                 y_valor = float(match.group(2))
35
36 with open(Nome_arquivo, 'r') as ... for i, linha in enumerate(linha... if "G10S2" in linha if match

```

Run main

```

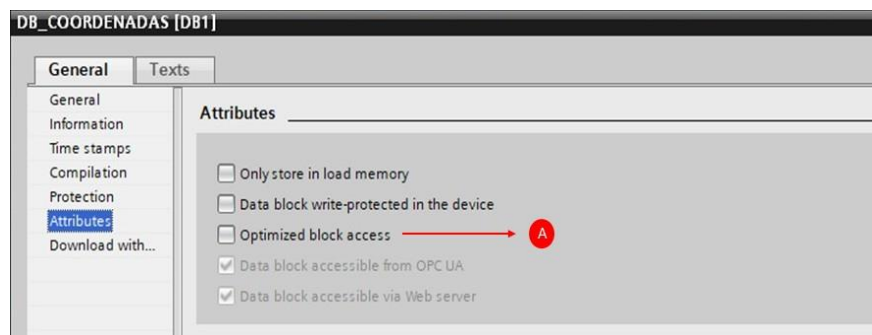
Coordenada encontrada: X=533.898648, Y=39.776556
Extração concluída com sucesso!
Total de coordenadas: 1
Iniciando comunicação com o CLP...
Conectado ao CLP com sucesso!
Erro ao conectar ou enviar dados: b'CPU : Function not available'
Conexão encerrada.
b'CPU : Function not available'
Process finished with exit code 0

```

Fonte: Autoria própria (2025).

Para garantir a compatibilidade da comunicação entre o software desenvolvido em Python e o CLP Siemens, é necessário desabilitar a opção "Optimized block access" (acesso otimizado a blocos) no TIA Portal, na Figura 9 é mostrado dentro das configurações do bloco DB em atributos a opção que deve ser desmarcada. Isso ocorre porque a biblioteca Snap7 realiza a leitura e escrita de dados diretamente por meio dos endereços físicos de memória do bloco de dados (DB). Quando o acesso otimizado está habilitado, o CLP pode reorganizar internamente as variáveis para melhorar o desempenho, o que torna os endereços não determinísticos e, portanto, incompatíveis com o método de acesso utilizado pela biblioteca Snap7.

Figura 9 – Configuração do Bloco DB.

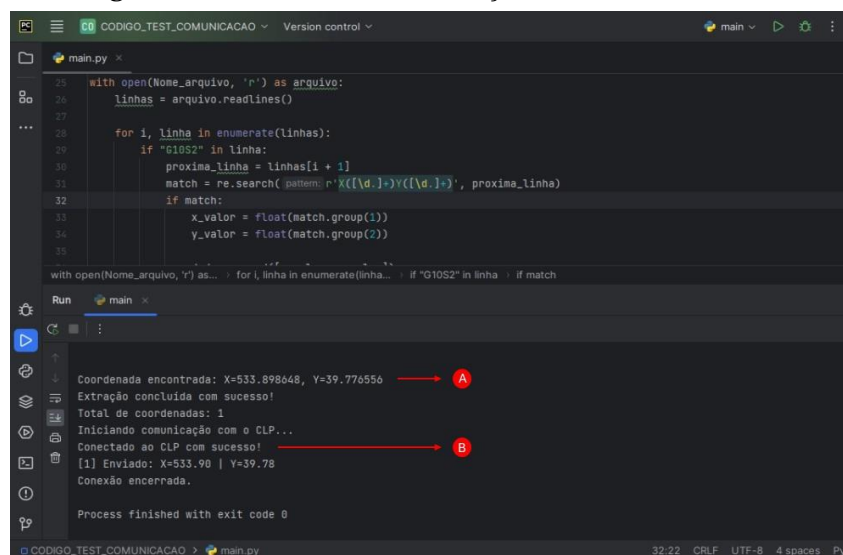


Fonte: Autoria própria (2025).

Dessa forma, ao desmarcar essa opção, os dados permanecem em posições fixas, garantindo a integridade da comunicação e o correto mapeamento das variáveis durante o intercâmbio de informações entre o CLP e o sistema em Python.

Após configurar corretamente o bloco, foi realizado novamente o teste de comunicação e desta vez conforme Figura 10, a conexão e o envio da coordenada foram realizados com sucesso.

Figura 10 – Teste de Comunicação com CLP - Sucesso



Fonte: Autoria própria (2025).

Os testes foram conduzidos utilizando diferentes conjuntos de dados, simulando situações reais do processo de corte e identificação das peças. Além disso, foram inclusos testes específicos com dados propositalmente inválidos, a fim de avaliar a robustez do script quanto ao tratamento de erros.

Os resultados desses testes revelaram que, inicialmente, o sistema apresentava algumas fragilidades na validação dos dados. Coordenadas negativas, por exemplo, eram simplesmente ignoradas pelo script, não sendo enviadas ao CLP. Embora isso impedisse erros operacionais diretos, essa falha não gerava nenhum aviso ao operador, o que poderia dificultar o diagnóstico de problemas no plano de corte, e se não houver atenção extrema do operador, facilmente poderia passar uma peça sem identificação.

Na Figura 11 é possível verificar os testes descritos a seguir, foram contadas apenas quatro coordenadas pois a coordenada negativa não é reconhecida e por este motivo foi desconsiderada da contagem como pode se notar no ponto B da figura, e consequentemente não foi enviada ao CLP como descrito anteriormente.

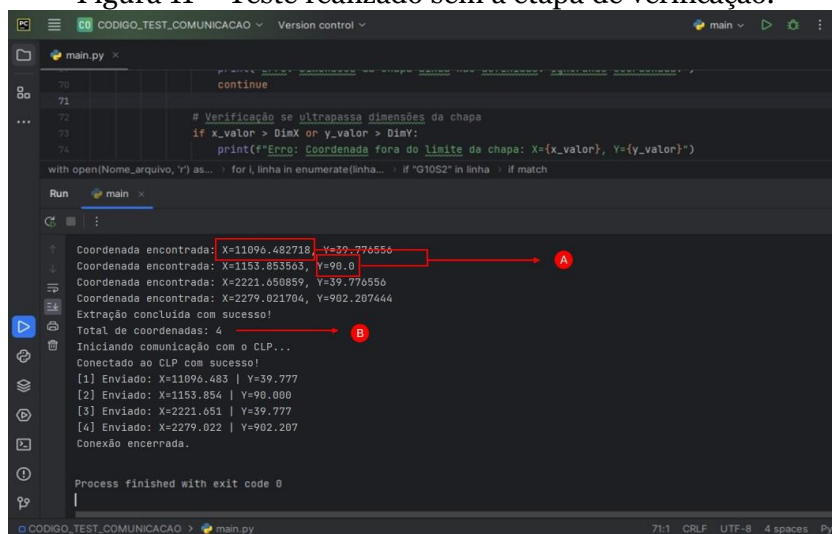
No ponto A outro problema identificado foi relacionado à presença de caracteres especiais indevidos nos valores das coordenadas, que fazia com que a leitura fosse truncada. Por exemplo, no caso de uma coordenada $Y = 902.234$, se houvesse um caractere especial inserido no meio (como $Y = 90\$2.234$), o sistema reconhecia apenas o valor anterior ao caractere inválido ($Y = 90$), o que comprometia diretamente o posicionamento da etiqueta e da peça.

Adicionalmente, coordenadas que excediam os limites físicos da chapa eram aceitas como válidas, o que representava um risco operacional, podendo ocasionar movimentos incorretos do robô cartesiano, colisões ou falhas no processo.

Diante desses resultados, foram realizadas melhorias no script, incorporando rotinas robustas de verificação e validação dos dados. O sistema passou a identificar automaticamente qualquer coordenada inválida, seja por valor negativo, presença de caracteres não numéricos ou excedente aos limites físicos da chapa. Nesses casos, uma mensagem de erro é gerada, impedindo o envio das informações incorretas ao CLP e garantindo assim a integridade e a segurança da operação.

A Tabela 3 apresenta os resultados de cinco testes realizados para validação do sistema. Na coluna **Código**, estão os valores das coordenadas X e Y extraídos diretamente do plano de corte (*nesting*). Na coluna **CLP**, constam os valores efetivamente enviados à memória do controlador.

Figura 11 – Teste realizado sem a etapa de verificação.



```

70 continue
71
72 # Verificação se ultrapassa dimensões da chapa
73 if x_valor > DimX or y_valor > DimY:
74     print(f"Erro: Coordenada fora do limite da chapa: X={x_valor}, Y={y_valor}")
75
76 with open(Nome_arquivo, "w") as f:
77     for i, linha in enumerate(linha...):
78         if "G10S2" in linha:
79             if match

```

Run main

```

Coordenada encontrada: X=11096.482718, Y=39.776556
Coordenada encontrada: X=1153.863563, Y=90.0
Coordenada encontrada: X=2221.650859, Y=39.776556
Coordenada encontrada: X=2279.021704, Y=902.207444
Exatidão concluída com sucesso!
Total de coordenadas: 4
Iniciando comunicação com o CLP...
Conectado ao CLP com sucesso!
[1] Enviado: X=11096.483 | Y=39.777
[2] Enviado: X=1153.864 | Y=90.000
[3] Enviado: X=2221.651 | Y=39.777
[4] Enviado: X=2279.022 | Y=902.207
Conexão encerrada.

```

Process finished with exit code 0

Fonte: Autoria própria (2025).

Tabela 3 – Teste de Validação do código.

	Código		CLP - Sem Validação		CLP - Com Validação	
	X	Y	X	Y	X	Y
Teste 1	-533,898648	39,776556			Mensagem Erro	39,776
Teste 2	11096,482718	39,776556	11096,483	39,777	Mensagem Erro	39,776
Teste 3	1153,853563	90\$2,207444	1153,854	90,000	1153,853	Mensagem Erro
Teste 4	2221,650859	39,776556	2221,651	39,777	2221,650	39,776
Teste 5	2279,021704	902,207444	2279,022	902,207	2279,021704	902,207

Fonte: Autoria própria (2025).

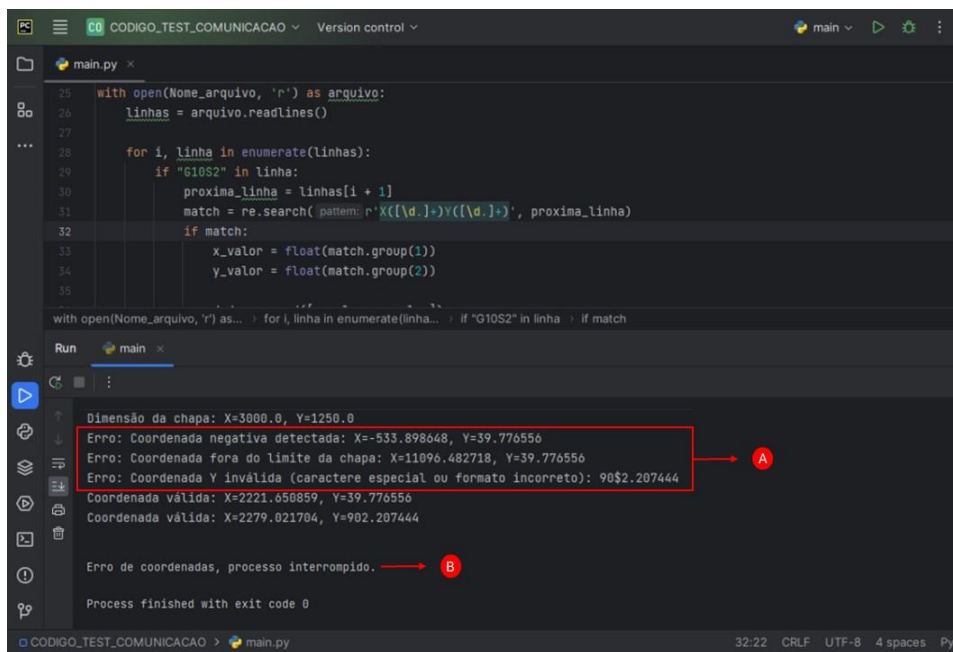
Os testes 1, 2 e 3 foram elaborados especificamente para verificar o comportamento do sistema frente a dados inconsistentes. No **Teste 1**, uma coordenada com valor negativo foi utilizada para avaliar se o sistema descartaria corretamente essa informação, uma vez que uma posição negativa é fisicamente inviável para o robô cartesiano. O script, de fato, bloqueou o envio da coordenada, funcionando corretamente após as melhorias implementadas.

No **Teste 2**, foi utilizada uma coordenada com valor superior ao limite físico da chapa. Na primeira versão do script, esse dado foi aceito e enviado ao CLP, o que poderia ocasionar falhas operacionais. Após a implementação das rotinas de validação, esse tipo de inconsistência passou a ser identificado e bloqueado.

O **Teste 3** teve como objetivo avaliar o comportamento do sistema diante da inserção de um caractere especial no meio do valor da coordenada. Antes das melhorias, o sistema truncava o valor, resultando na perda de precisão (por exemplo, a coordenada Y = 902.207444 foi lida como Y = 90.000). Com a atualização do código, esse erro passou a ser detectado, gerando uma mensagem de falha e bloqueando o envio ao CLP.

Na Figura 12, podemos verificar no ponto A que as coordenadas que contem algum tipo de incongruência foram devidamente reportadas como mensagem de Erro, e no ponto B a mensagem de processo interrompido, pois um plano de corte que apresente qualquer tipo de erro nas coordenadas de identificação não podem seguir, mesmo que algumas estejam certas, pois certamente as peças correspondentes as coordenadas que apresentam erros ficariam sem identificação.

Figura 12 – Erros Reportados - Sistema de Validação.



```

25 with open(Nome_arquivo, 'r') as arquivo:
26     linhas = arquivo.readlines()
27
28     for i, linha in enumerate(linhas):
29         if "G10S2" in linha:
30             proxima_linha = linhas[i + 1]
31             match = re.search(pattern=r'X([0-9.]+)Y([0-9.]+)', proxima_linha)
32             if match:
33                 x_valor = float(match.group(1))
34                 y_valor = float(match.group(2))
35
with open(Nome_arquivo, 'r') as...  for i, linha in enumerate(linha...  if "G10S2" in linha  if match

```

Run main x

Dimensão da chapa: X=3000.0, Y=1250.0

Erro: Coordenada negativa detectada: X=-533.898048, Y=39.776556

Erro: Coordenada fora do limite da chapa: X=11096.482718, Y=39.776556

Erro: Coordenada Y inválida (caractere especial ou formato incorreto): 90\$2.207444

Coordenada válida: X=2221.650859, Y=39.776556

Coordenada válida: X=2279.021704, Y=902.207444

Erro de coordenadas, processo interrompido.

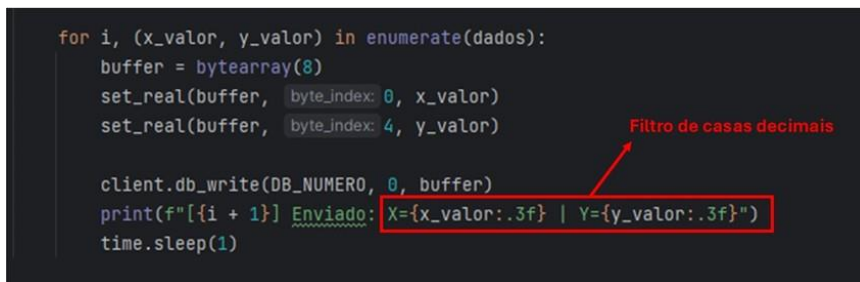
Process finished with exit code 0

CODIGO_TEST_COMUNICACAO > main.py 32:22 CRLF UTF-8 4 spaces Pyth

Fonte: Autoria própria (2025).

Além dos testes específicos, se faz notória uma diferença na precisão dos valores transmitidos. Enquanto os dados extraídos do plano de corte possuem até seis casas decimais, os valores enviados ao CLP foram adequados ao padrão de três casas decimais, conforme Figura 13, considerando as limitações de processamento do controlador e garantindo a precisão necessária para a operação do robô.

Figura 13 – Ajuste de precisão.



```

for i, (x_valor, y_valor) in enumerate(dados):
    buffer = bytearray(8)
    set_real(buffer, byte_index=0, x_valor)
    set_real(buffer, byte_index=4, y_valor)

    client.db_write(DB_NUMERO, 0, buffer)
    print(f"[{i + 1}] Enviado: X={x_valor:.3f} | Y={y_valor:.3f}")
    time.sleep(1)

```

Filtro de casas decimais

Fonte: Autoria própria (2025).

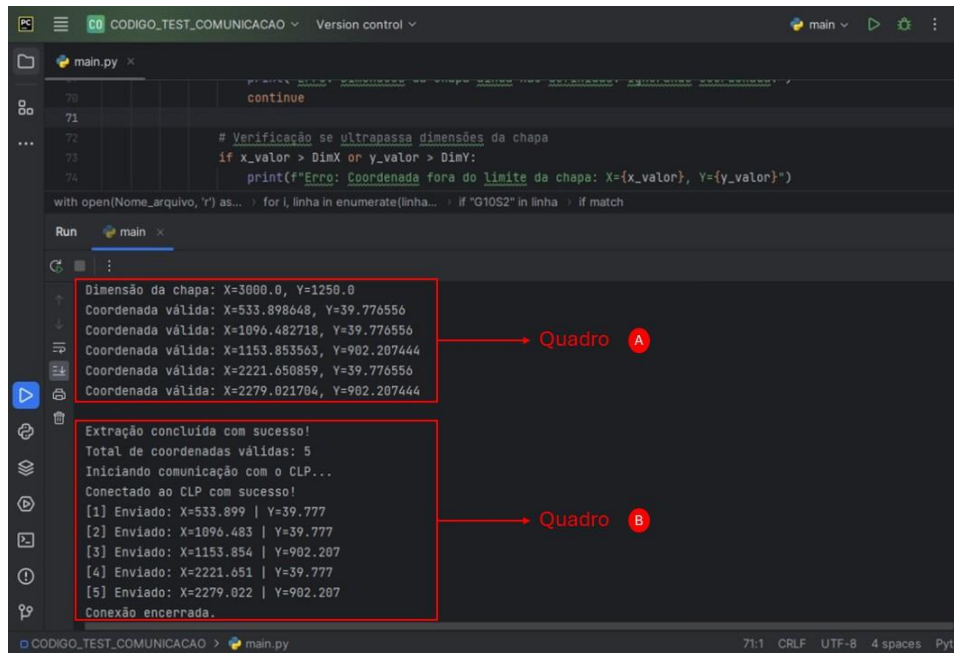
Esses resultados comprovam que a implementação das rotinas de validação foram essenciais para garantir a robustez, a segurança e a confiabilidade do sistema.

Na figura 14 temos a coleta das 5 coordenadas de teste após a passagem pelo sistema de validação. No Quadro A temos as coordenadas agora validadas e no Quadro B as mesmas sendo enviadas ao CLP.

Na Figura 15 temos as coordenadas recebidas do Sistema, pode-se notar no ponto A da figura o monitoramento das coordenadas na interface gráfica de simulação do software Tia Portal. Nos testes executados, as coordenadas foram alterando conforme a leitura do código, sendo imputadas a cada 1 segundo para visualização.

A seguir, são apresentados os demais resultados operacionais, incluindo o comparativo entre os tempos de execução dos processos manual e automatizado, além dos ganhos observados em termos de

Figura 14 – Resultado.



```

70 continue
71
72 # Verificação se ultrapassa dimensões da chapa
73 if x_valor > DimX or y_valor > DimY:
74     print(f"Erro: Coordenada fora do limite da chapa: X={x_valor}, Y={y_valor}")

```

Run main x

Dimensão da chapa: X=3000.0, Y=1250.0
 Coordenada válida: X=533.898648, Y=39.776556
 Coordenada válida: X=1096.482718, Y=39.776556
 Coordenada válida: X=1153.853563, Y=902.207444
 Coordenada válida: X=2221.650859, Y=39.776556
 Coordenada válida: X=2279.021704, Y=902.207444

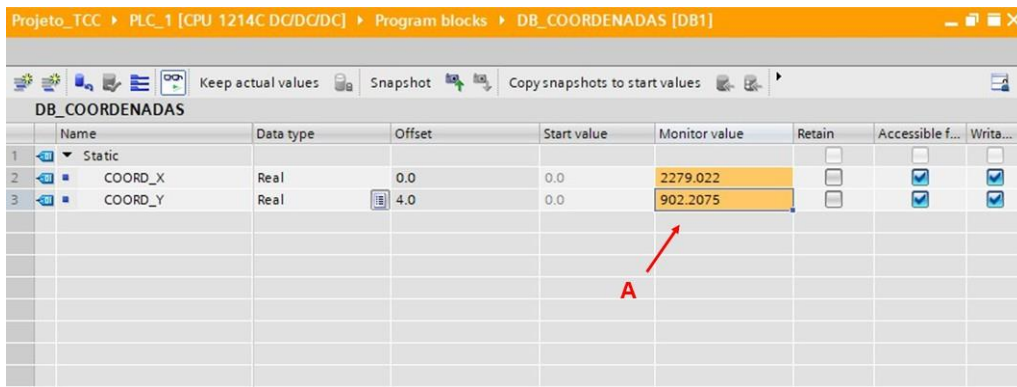
Exatidão concluída com sucesso!
 Total de coordenadas válidas: 5
 Iniciando comunicação com o CLP...
 Conectado ao CLP com sucesso!
 [1] Enviado: X=533.899 | Y=39.777
 [2] Enviado: X=1096.483 | Y=39.777
 [3] Enviado: X=1153.854 | Y=902.207
 [4] Enviado: X=2221.651 | Y=39.777
 [5] Enviado: X=2279.022 | Y=902.207
 Conexão encerrada.

Quadro A

Quadro B

Fonte: Autoria própria (2025).

Figura 15 – CLP.



Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Writa...
Static							
COORD_X	Real	0.0	0.0	2279.022		☒	☒
COORD_Y	Real	4.0	0.0	902.2075		☒	☒

A

Fonte: Autoria própria (2025).

produtividade e redução de erros.

Como referência, utilizou-se uma empresa parceira que possui um sistema semelhante já implementado. Nesse sistema, verifica-se que, em média, são necessários 20 segundos para identificar 6 peças e retornar ao ponto zero, resultando em aproximadamente 3,33 segundos por peça.

A Tabela 4 apresenta os dados obtidos na crono análise do processo manual de etiquetagem, considerando cinco amostras.

A média geral do tempo por peça no processo manual foi de aproximadamente 8,549 segundos.

Com base nos dados obtidos, aplicou-se a Equação 1 para calcular o ganho percentual proporcionado pela automação do processo de identificação. A equação considera a diferença entre o tempo médio do processo manual e o tempo médio do processo automatizado, de acordo com a seguinte formulação:

$$\text{Ganho percentual} = \frac{A \text{ Tempo}_{Operador} - \text{Tempo}_{Robo}^B}{\text{Tempo}_{Operador}} \times 100 \quad (1)$$

Tabela 4 – Tempo de etiquetagem manual.

	Tempo em (s)	Quantidade peças	Tempo em (s)/peças
Amostra 1	73	8	9,125.
Amostra 2	99	10	9,900.
Amostra 3	80	10	8,000.
Amostra 4	105	18	5,833.
Amostra 5	89	9	9,889.

Fonte: Autoria própria (2025).

O resultado obtido demonstra que a implementação do sistema automatizado proporcionou uma redução de 61,01% no tempo de execução da atividade de identificação de peças, quando comparado ao processo manual. Esse ganho reflete diretamente na melhora da produtividade, na redução de paradas operacionais e no aumento da eficiência do setor de corte.

Tabela 5 – Comparativo entre o tempo de etiquetagem manual e automatizado.

Método	Tempo médio por peça (s)
Manual	8,549
Automatizado	3,333

Fonte: Autoria própria (2025).

A Tabela 5 apresenta o comparativo entre o tempo médio por peça no processo manual e no processo automatizado. Observa-se uma redução de 61,01% no tempo médio por peça após a implementação do sistema de automação.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema proposto para identificação automatizada de peças metálicas cortadas a laser demonstrou que a integração entre tecnologias como Python, CLP Siemens, robô cartesiano e impressora industrial pode oferecer uma solução eficiente e precisa para processos industriais que ainda dependem de métodos manuais e suscetíveis a falhas.

A utilização da biblioteca Pandas mostrou-se eficaz para a leitura e tratamento das coordenadas extraídas dos arquivos de corte, permitindo a automação de uma etapa que anteriormente era feita com grande chance de erro humano. A comunicação entre o código Python e o CLP via Snap7 foi fundamental para garantir a transmissão confiável dos dados, enquanto o robô cartesiano, comandado pelo CLP, executará com precisão o posicionamento e a aplicação das etiquetas.

Durante o projeto, foram realizados testes simulados que evidenciaram o potencial de ganho em produtividade, rastreabilidade e organização do fluxo de trabalho. Embora a implementação prática em ambiente fabril ainda dependa de ajustes e validações adicionais, os resultados obtidos atestam a viabilidade técnica da proposta, além de indicarem um caminho promissor para futuras expansões.

Dessa forma, conclui-se que a automação da identificação de peças representa uma contribuição relevante para a modernização de processos industriais, mostrando - se vantajoso quando comparado ao processo manual, alinhando-se aos princípios da Indústria 4.0 ao utilizar tecnologias acessíveis para resolver problemas reais de chão de fábrica. Este trabalho não apenas comprovou a funcionalidade da solução, como também abriu portas para melhorias contínuas.

Com este projeto espera-se uma redução significativa no número de peças identificadas incorretamente, evitando assim retrabalhos, insatisfação dos clientes, custos extras, entre outras questões que são vivenciadas na indústria.

Como proposta para trabalhos futuros, sugere-se a implementação completa do sistema desenvolvido, integrando todas as etapas em um ambiente industrial real, desde a extração dos dados do plano de corte até o acionamento automatizado do robô cartesiano e da impressora de etiquetas. Essa aplicação prática permitirá validar o desempenho do sistema e avaliar ganhos reais em produtividade e rastreabilidade no processo fabril.

Outra possibilidade relevante é o desenvolvimento de uma solução inteligente para a definição automática da coordenada de identificação em cada peça, de forma a evitar sobreposição com furações, recortes ou áreas críticas do componente. Essa funcionalidade pode ser viabilizada por meio da aplicação de visão computacional ou algoritmos de análise geométrica, capazes de identificar regiões seguras e otimizadas para posicionamento das etiquetas.

Além disso, propõe-se a integração do sistema com bancos de dados industriais ou sistemas de gestão (ERP/MES), permitindo que os dados de identificação gerados sejam armazenados e utilizados para fins de controle de produção, rastreabilidade, inspeção de qualidade ou logística.

Também se vislumbra a possibilidade de empregar técnicas de inteligência artificial para melhorar a tomada de decisão quanto ao posicionamento da etiqueta, priorizando critérios como área útil, estética, e facilidade de leitura. A utilização de redes neurais treinadas com imagens de peças e seus respectivos planos pode agregar ainda mais autonomia ao sistema, ampliando suas aplicações para diferentes tipos de peças e layouts de corte.

Referências

- BARATTELLA, P. R. *Redes Industriais*. São Paulo: Érica, 2011. Citado na página 6.
- BRAGA, N. C. *Automação Industrial: Controladores Lógicos Programáveis*. São Paulo: Érica, 2015. Citado na página 5.
- DADOS, J. *Pandas: Guia Rápido de Manipulação de Dados*. 2025. <<https://jornadadados.com/conheca-a-biblioteca-pandas-analise-de-dados/>>. Acesso em: 26 abr. 2025. Citado na página 4.
- MACHADO, C. e. a. Rastreabilidade em processos de manufatura. *Revista de Engenharia*, v. 28, n. 4, p. 45–56, 2020. Citado na página 2.
- MCKINNEY, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2. ed. Sebastopol: O'Reilly Media, 2017. Citado na página 4.
- MOLENAAR, G. *python-snap7*. 2025. <<https://github.com/gijzelaerr/python-snap7>>. Acesso em: 21 maio 2025. Citado na página 5.
- MONTGOMERY, D. C. *Introdução ao Controle Estatístico da Qualidade*. 7. ed. [S.l.]: LTC, 2019. Citado na página 2.
- OLIVEIRA, J. A. d. *Automação e Controle com CLPs*. São Paulo: Érica, 2012. Citado na página 7.
- OLIVEIRA, J. P. A. d. *Automação e Controle de Processos Industriais*. São Paulo: Érica, 2012. Citado na página 5.
- POWELL, J. *CO2 Laser Cutting*. 3. ed. London: Springer, 2017. Citado na página 3.
- PYTHON-SNAP7. *Documentation: API – Util Module*. 2025. <<https://python-snap7.readthedocs.io/en/stable/API/util.html>>. Acesso em: 21 maio 2025. Citado na página 5.
- SANTOS, F. *Introdução à Robótica Industrial*. Rio de Janeiro: LTC, 2016. Citado na página 7.
- SILVA, C. E. *Fundamentos de Robótica: Aplicações Industriais*. São Paulo: Blucher, 2017. Citado na página 7.
- Simply Automationized. *Python Snap7 S7-1200 Simple Example*. 2016. <<https://simplyautomationized.blogspot.com/2016/02/python-snap7-s7-1200-simple-example.html>>. Acesso em: 21 maio 2025. Citado na página 5.
- SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. 2. ed. [S.l.]: Atlas, 2002. Citado na página 2.
- STEEN, W. M.; MAZUMDER, J. *Laser Material Processing*. 4. ed. London: Springer, 2010. Citado na página 3.