

Desafios na Educação em Engenharia: integrando o Pensamento Computacional e as Simulações 3D

Challenges in Engineering Education: Integrating Computational Thinking and 3D Simulations

Fernando Covolan Rosito¹
Eliana Maria Sacramento Soares²
Carine Geltrudes Webber³

Resumo

Este artigo explora os desafios e as oportunidades na educação em engenharia ao integrar o Pensamento Computacional e as simulações 3D como ferramentas pedagógicas. O principal objetivo é avaliar como essa combinação pode melhorar a formação de engenheiros, especialmente no contexto da Indústria 4.0, que exige competências avançadas em programação e automação. A pesquisa foi realizada por meio de uma oficina prática, onde estudantes de Engenharia de Controle e Automação foram provocados a realizar atividades que combinassem simulações 3D com a programação de Controladores Lógicos Programáveis (CLPs) utilizando a linguagem Ladder. Os principais desafios identificados incluem a complexidade de adaptação dos estudantes a novas tecnologias e a necessidade de recursos técnicos e pedagógicos adequados. Os resultados indicam que o uso de simulações 3D, aliado ao desenvolvimento do Pensamento Computacional, promove uma compreensão mais profunda dos conceitos de programação e uma experiência fidedigna de situações reais de engenharia. Perspectivas futuras apontam para a ampliação do uso dessas metodologias em diferentes disciplinas de engenharia a fim de avaliá-la em diversos contextos educacionais.

Palavras-chave: Educação; Engenharia; Pensamento computacional; Tecnologias; Indústria 4.0.

Abstract

This article explores the challenges and opportunities in engineering education by integrating Computational Thinking and 3D simulations as pedagogical tools. The main objective is to assess how this combination can enhance the training of engineers, especially in the context of Industry 4.0, which demands advanced skills in programming and automation. The research was conducted through a practical workshop, where Control and Automation Engineering students were challenged to carry out activities that combined 3D simulations with programming Programmable Logic Controllers (PLCs) using the Ladder language. The main challenges identified include the complexity of students' adaptation to new technologies and the need for appropriate technical and pedagogical resources. The results indicate that using 3D simulations, combined with the development of Computational Thinking, fosters a deeper

¹ Doutor em Educação pela Universidade de Caxias do Sul (UCS). Professor no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS). E-mail: fernando.rosito@farroupilha.ifrs.edu.br

² Doutora em Educação pela Universidade Federal de São Carlos (UFSCAR-SP). Professora na Universidade de Caxias do Sul (UCS). E-mail: emsoares@ucs.br

³ Doutora em Ciência da Computação pela Université de Grenoble, França. Professora na Universidade de Caxias do Sul (UCS). E-mail: cgwebber@ucs.br

understanding of programming concepts and provides an authentic experience of real-world engineering situations. Future perspectives point to expanding the use of these methodologies in different engineering disciplines to evaluate them in various educational contexts.

Keywords: Education; Engineering; Computational thinking; Technologies; Industry 4.0.

1. Introdução

A quarta revolução industrial, conhecida como Indústria 4.0, tem provocado transformações profundas nos processos produtivos e nas competências exigidas dos profissionais de engenharia, não se limitando à resolução de problemas e integração entre conhecimento e processos (SHEPPARD et al., 2006). Caracterizada pela digitalização e pela integração de tecnologias avançadas como a Internet das Coisas (do inglês *Internet of Thing* - IoT), Inteligência Artificial (IA) e sistemas ciberfísicos, a Indústria 4.0 demanda que engenheiros sejam proficientes em projetar, operar e programar sistemas complexos e interconectados. Nesse cenário, a habilidade de programação torna-se não apenas relevante, mas essencial para a formação de engenheiros capazes de enfrentar os desafios tecnológicos contemporâneos.

No entanto, o aprendizado de programação por engenheiros enfrenta diversas dificuldades, que vão desde a complexidade intrínseca dos conceitos computacionais até a falta de contextualização prática no processo educativo. A linguagem Ladder, amplamente utilizada na programação de Controladores Lógicos Programáveis (CLPs) — componentes fundamentais para a automação industrial —, apesar de ser projetada para ser intuitiva, apresenta barreiras significativas de compreensão e aplicação prática. Muitos estudantes encontram desafios em abstrair e decompor problemas complexos, desenvolver algoritmos eficientes e aplicar conceitos teóricos em situações reais de engenharia, o que pode comprometer sua formação e desempenho profissional.

Nesse contexto, o Pensamento Computacional (PC) emerge como uma competência central para superar essas dificuldades. Conceituado por Jeannette Marie Wing (2006), o PC envolve a capacidade de formular problemas e suas soluções de maneira que possam ser executados por um computador. Para engenheiros, desenvolver o PC significa aprimorar habilidades como decomposição de problemas, reconhecimento de padrões, abstração e elaboração de algoritmos, todas críticas para a programação eficaz de sistemas automatizados. Contudo, incorporar e fortalecer

essas habilidades no ensino de engenharia requer abordagens pedagógicas inovadoras que conectem teoria e prática de forma significativa.

As metodologias tradicionais de ensino de programação em engenharia, frequentemente baseadas em aulas teóricas e laboratórios presenciais, têm se mostrado insuficientes para atender às demandas da Indústria 4.0, ou seja, não proporcionam experimentos com sistemas de automação com a integração de diversos dispositivos, simulando processos reais. Essas abordagens muitas vezes não conseguem engajar os estudantes de maneira efetiva, nem proporcionar experiências de aprendizagem que reflitam a complexidade e a interdisciplinaridade dos ambientes industriais modernos. Consequentemente, há uma necessidade premente de estratégias educativas que facilitem o aprendizado ativo e contextualizado, promovendo o desenvolvimento de competências técnicas e cognitivas de forma integrada.

Para abordar esses desafios, este estudo propõe a integração da simulação em 3D como ferramenta pedagógica no ensino de programação de CLPs (linguagem Ladder). Essa proposta é fundamentada em teorias educacionais do Construcionismo de Seymour Papert e Teoria Sociocultural de Lev Semenovitch Vygotsky. O Construcionismo enfatiza o aprendizado por meio da construção ativa de conhecimento, onde os estudantes são incentivados a aprender experimentando e refletindo sobre suas práticas (PAPERT, 1985, 2008). A Teoria Sociocultural de Vygotsky destaca a importância da interação social e do contexto cultural no desenvolvimento cognitivo, ressaltando o papel da mediação e da colaboração no processo de aprendizagem (DUARTE *et al.*, 2019, VYGOTSKY, 1991).

A utilização de simulações em 3D proporciona um ambiente de aprendizagem imersivo e interativo, permitindo que os estudantes visualizem e manipulem sistemas automatizados complexos em um contexto virtual que replica situações reais da indústria (RÜßMANN *et al.*, 2015). Essa ferramenta facilita a compreensão de conceitos abstratos, promove o engajamento e possibilita a aplicação prática do conhecimento de forma segura e controlada. Além disso, as simulações em 3D favorecem a colaboração entre os alunos e o desenvolvimento de habilidades críticas, como a resolução de problemas e o pensamento analítico.

Dessa forma, este artigo investiga como o uso de simulações em 3D pode potencializar o ensino e o aprendizado da programação de CLPs, contribuindo para a

formação de engenheiros mais preparados para os desafios da Indústria 4.0. A hipótese central é que a integração dessa tecnologia educacional, aliada a abordagens pedagógicas construtivistas e socioculturais, melhore a compreensão e a proficiência dos alunos em programação, mas também promova um aprendizado mais profundo e significativo, conectando efetivamente teoria e prática e preparando os estudantes para a complexidade do ambiente industrial moderno.

2. Referencial teórico

A Indústria 4.0 trouxe consigo uma série de inovações tecnológicas que têm transformado profundamente os processos industriais, redefinindo as competências exigidas dos profissionais de engenharia (LEANDRO; FACIN; COSTA NETO, 2022). Essa revolução digital, caracterizada pela convergência de tecnologias emergentes, exige que os currículos de engenharia sejam constantemente atualizados (SILVA *et al.*, 2015). O objetivo é formar engenheiros capacitados para enfrentar os desafios do mercado de trabalho moderno, que demanda não apenas conhecimentos técnicos aprofundados, mas também habilidades analíticas, criativas e adaptativas.

Papert (1985, 2008), com sua teoria do Construcionismo, propôs uma abordagem educacional que se alinha diretamente com essas novas exigências da Indústria 4.0. Papert argumentou que o aprendizado ocorre de forma mais eficaz quando os estudantes são ativos na construção do conhecimento, seja por meio de criações físicas ou digitais. Ele enfatizou a importância de utilizar a tecnologia não apenas como uma ferramenta de ensino, mas como um ambiente que possibilita a exploração e manipulação interativa de conceitos. No contexto da educação em engenharia, especialmente no ensino de programação de CLPs, essa abordagem sugere que os alunos devem ser incentivados a experimentar e interagir ativamente com os sistemas que estão aprendendo a programar, facilitando um entendimento mais profundo e duradouro dos conceitos técnicos.

Complementando essa perspectiva, Vygotsky oferece uma visão sociocultural do aprendizado, que enfatiza a importância da interação social e da mediação no desenvolvimento cognitivo. De acordo com Vygotsky (1991), o aprendizado é um processo social mediado por instrumentos e signos, com a linguagem desempenhando um papel central. O conceito de Zona de Desenvolvimento Proximal

(ZDP), introduzido por Vygotsky, define o espaço entre o que o aluno pode realizar sozinho e o que ele pode alcançar com a ajuda de um mediador, como um professor ou colegas mais experientes. No ensino de programação, especialmente em ambientes mediados por tecnologia, essa interação é fundamental para que os alunos avancem em seu desenvolvimento cognitivo e adquiram habilidades de forma mais eficaz.

Diante das exigências da Indústria 4.0 e das abordagens pedagógicas propostas por Papert e Vygotsky, a Transposição Informática emerge como uma estratégia crucial para integrar essas teorias ao ensino de programação de CLPs. Segundo Nicolas Balacheff (1993), a Transposição Informática vai além da simples seleção de tecnologias; ela envolve a recontextualização dessas ferramentas digitais, transformando-as em instrumentos pedagógicos eficazes.

Portanto, a Transposição Informática não se resume à mera adaptação de conteúdo para um formato digital, mas envolve a reconfiguração estratégica do ensino, integrando tecnologias digitais de forma a enriquecer o processo de aprendizagem. Esse processo cria um ambiente educacional mais dinâmico, interativo e alinhado às exigências da Indústria 4.0. A combinação dessas abordagens teóricas com a prática no ensino de programação de CLPs, utilizando simulações em 3D, visa não apenas aprimorar a compreensão técnica dos alunos, mas também desenvolver as habilidades críticas e criativas essenciais para a atuação em ambientes industriais complexos e interconectados.

3. Materiais e métodos

Com o objetivo de investigar como as simulações em 3D podem ser utilizadas para enriquecer o ensino de programação de Controladores Lógicos Programáveis (CLPs), proporcionando um ambiente de aprendizagem alinhado às exigências da Indústria 4.0, foi concebida uma oficina prática voltada para estudantes de Engenharia de Controle e Automação. Esta oficina foi planejada para explorar a integração das práticas pedagógicas com as tecnologias da Indústria 4.0, baseando-se na resolução de problemas que mobilizassem o Pensamento Computacional⁴.

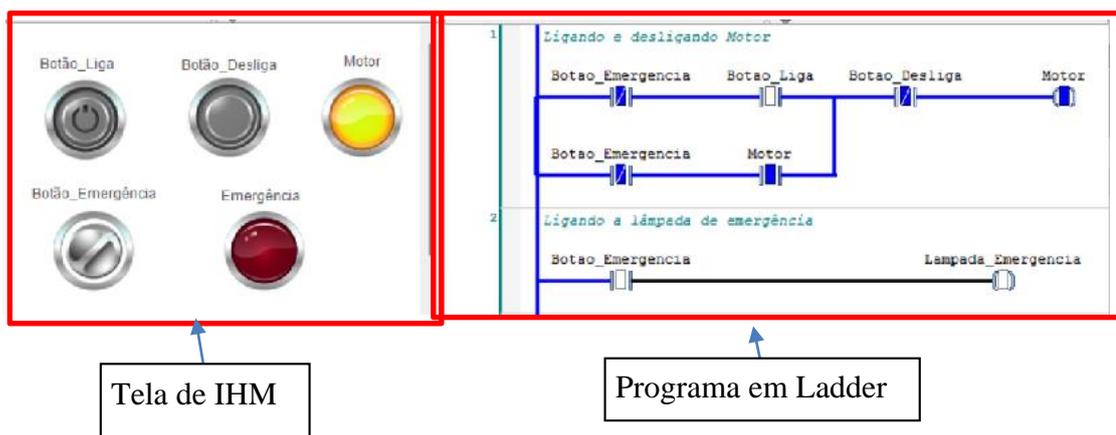
⁴ Este trabalho detalha um recorte de uma tese de doutorado. Dados referentes ao comitê de Ética foram omitidos para fins de avaliação anônima do artigo.

3.1 Contexto e planejamento da oficina

A oficina serviu como estudo de caso para avaliar a eficácia da simulação em 3D no desenvolvimento de competências técnicas e de Pensamento Computacional entre os alunos. Durante a oficina, os participantes foram desafiados a desenvolver códigos, criar interfaces homem-máquina (IHM) e projetar ambientes em 3D, todos aplicados à programação de CLPs pela linguagem Ladder. Esses elementos constituíram o *corpus* de dados analisado neste artigo.

Os ambientes de programação e simulação selecionados para o desenvolvimento das dinâmicas pedagógicas foram os softwares CODESYS e Factory I/O. O CODESYS permitiu a programação em linguagem *Ladder*, a criação de interfaces IHM, e a simulação de processos, como ilustrado na Figura 1.

Figura 1 – Ambiente de programação e simulação pela tela de IHM



Fonte: Elaborado pelos autores (desenvolvido no software CODESYS)

O Factory I/O, por sua vez, foi utilizado para simular processos industriais em 3D, conforme a Figura 2. As experiências computacionais concebidas tinham como intuito não só ajudar os estudantes a aprender a programar um CLP em linguagem *Ladder*, mas também a se apropriarem das tecnologias da Indústria 4.0, das quais a simulação em 3D é um dos pilares.

Figura 2 – Ambiente de simulação em 3D



Fonte: Elaborado pelo autor - desenvolvido no software Factory I/O.

3.2 Realização da oficina

A oficina ocorreu no formato de ensino a distância, utilizando a plataforma Google Meet. O ambiente virtual proporcionou um espaço para o compartilhamento de informações, execução de exercícios práticos, e realização de discussões colaborativas, todas essenciais para a resolução dos problemas propostos. Originalmente planejada para dez encontros, a oficina exigiu uma extensão para quinze encontros online, devido à complexidade e à dinâmica dos temas abordados. Participaram da oficina oito estudantes do curso de Engenharia de Controle e Automação de uma instituição de ensino superior.

Os temas abordados na oficina estiveram relacionados a problemas na área de automação de sistemas, com a utilização de simulação em 3D. Durante as atividades, o professor incentivou um ambiente de interação entre os alunos, seguindo a perspectiva de Vygotsky, que destaca a importância da mediação social no processo de aprendizagem. No total, foram desenvolvidas vinte atividades práticas, organizadas em sequência, com aumento gradual de dificuldade nos conceitos de programação em Ladder. Um exemplo de exercício é apresentado no Quadro 1.

Quadro 1 – Enunciado de um exercício

Título: Porta de banco.
Enunciado: Elabore um programa em Ladder, que controle a abertura de uma porta de entrada de um banco. A porta deve abrir quando qualquer um dos dois botões for pressionado. Se nenhum dos botões estiver pressionado, a porta deve permanecer fechada.
OBS: os botões estão em locais separados, não podendo ser acionados ao mesmo tempo. OBS: escreva os nomes das variáveis nos contatos utilizados no programa.

Fonte: Elaborado pelo autor.

3.3 Critérios de avaliação e acompanhamento

Para avaliar o impacto da intervenção pedagógica e acompanhar o progresso dos alunos, foram estabelecidos critérios específicos de avaliação. Os critérios utilizados para a análise detalhada do desempenho dos estudantes ao longo da oficina foram os seguintes:

- 1. Organização do Código:** Avalia a capacidade dos alunos de estruturar e organizar seus códigos de forma clara e lógica, incluindo a utilização de boas práticas como a adição de comentários, a separação das etapas do processo de controle, e a manutenção de um código limpo e legível.
- 2. Desenvolvimento Lógico:** Refere-se à habilidade dos alunos em construir sequências lógicas eficazes e em aplicar o raciocínio lógico na programação de CLPs. Este critério mede a progressão na capacidade de formular e implementar soluções lógicas que resolvam problemas específicos de automação.
- 3. Eficiência do Código:** Analisa a eficiência dos códigos desenvolvidos, incluindo a otimização dos algoritmos, a eliminação de redundâncias e a simplificação das operações de controle. Este critério é fundamental para avaliar a capacidade dos alunos de criar programas que não apenas funcionem corretamente, mas também sejam eficientes em termos de processamento e execução.
- 4. Pensamento Algorítmico:** Avalia o desenvolvimento do pensamento algorítmico dos alunos, ou seja, sua capacidade de planejar, decompor problemas em etapas lógicas e executar algoritmos de forma eficaz. Este critério é crucial para medir a habilidade dos alunos em abordar e resolver problemas de programação de forma estruturada.
- 5. Inovação e Criatividade:** Mede a disposição dos alunos para explorar soluções novas e criativas, bem como sua capacidade de pensar de forma inovadora ao programar CLPs. Este critério observa como os alunos aplicam sua criatividade na resolução de problemas e na programação de sistemas automatizados.
- 6. Engajamento e Interação:** Refere-se ao nível de participação dos alunos nas atividades da oficina, incluindo sua interação com os colegas e com o professor. Este critério também avalia o engajamento dos alunos em discussões colaborativas e sua motivação para explorar diferentes funcionalidades das simulações em 3D.
- 7. Resolução de Problemas:** Observa a capacidade dos alunos de identificar, diagnosticar e corrigir erros em seus programas de CLP. Este critério avalia a independência dos alunos na resolução de problemas e sua habilidade em aplicar soluções eficazes.
- 8. Aplicação Prática dos Conceitos:** Avalia como os alunos aplicaram os conceitos teóricos de programação em cenários práticos, utilizando as simulações em 3D para testar e validar suas soluções. Este critério mede a habilidade dos alunos em conectar teoria com prática, essencial para o sucesso na programação de sistemas automatizados.

3.4 Procedimentos de coleta e análise de dados

Durante a oficina, os alunos participaram de atividades práticas que exigiram a aplicação dos conceitos de programação de CLPs em ambientes simulados em 3D. A coleta de dados foi contínua ao longo dos encontros, e o progresso dos alunos foi registrado com base nos oito critérios de avaliação definidos, permitindo uma visão abrangente do desenvolvimento das habilidades de cada aluno.

Os dados coletados foram analisados para identificar padrões de comportamento e progresso entre os alunos. A partir dessa análise, foi possível identificar *clusters*⁵ de comportamento, que refletem diferentes trajetórias de aprendizagem e adaptação à metodologia pedagógica empregada. Esses *clusters* foram utilizados para categorizar os alunos e compreender melhor como cada grupo respondeu à intervenção pedagógica.

4. Resultados

A intervenção pedagógica realizada durante a oficina prática de programação trouxe à tona uma série de resultados significativos em termos de desenvolvimento das habilidades de Pensamento Computacional dos alunos. Inicialmente, foram observadas dificuldades variadas entre os estudantes, especialmente no que diz respeito à organização do código e à aplicação prática dos conceitos de programação. Esses desafios iniciais destacaram a necessidade de um apoio contínuo e de ferramentas pedagógicas eficazes, como as simulações em 3D, para ajudar os alunos a superar suas dificuldades e avançar em seu aprendizado.

Os resultados indicaram que, ao longo da oficina, a maioria dos alunos demonstrou uma evolução notável em termos de compreensão e aplicação dos conceitos de programação. A análise detalhada do desempenho dos alunos revelou padrões distintos de progresso, os quais foram agrupados em *clusters*.

A análise dos dados coletados permitiu a identificação de três *clusters*, refletindo similaridades entre as dificuldades e as iniciativas dos alunos, considerando os oito critérios avaliados. Cada *cluster* representa uma trajetória diferente de aprendizagem e adaptação às metodologias empregadas:

⁵ A definição de *Cluster* para este artigo sugere o agrupamento de dados similares.

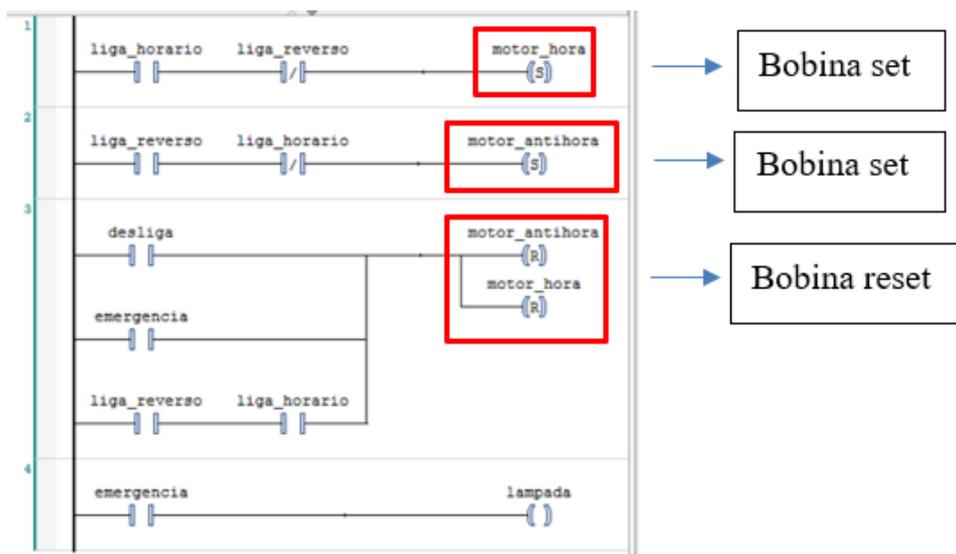
- **Cluster 1: Nível consistente** - Este grupo foi caracterizado por alunos que demonstraram uma organização exemplar desde o início da oficina. Eles apresentaram um desempenho consistente em todas as atividades, com uma lógica de programação bem estruturada e uma eficiente aplicação prática dos conceitos. Além do desempenho consistente ao longo de toda a oficina, apresentaram soluções criativas para os problemas propostos.
- **Cluster 2: Nível em evolução** - Os alunos deste *cluster* enfrentaram dificuldades iniciais, especialmente na organização e na lógica do código, mas mostraram uma evolução significativa ao longo da oficina. Com o apoio das simulações em 3D, esses alunos conseguiram melhorar suas habilidades de pensamento algorítmico e eficiência no código, adaptando-se às exigências do curso.
- **Cluster 3: Nível iniciante** - Este *cluster* compreende alunos que enfrentaram grandes desafios desde o início, como a falta de organização e erros frequentes na lógica dos códigos. No entanto, com o suporte contínuo e o uso de simulações em 3D, esses alunos conseguiram melhorar seu desempenho, embora a um ritmo mais lento em comparação com os outros *clusters*.

Para ilustrar a análise dos *clusters*, dois alunos foram escolhidos como exemplos representativos: aluno A e o aluno B. O aluno A apresentou, desde o início da atividade, uma forte organização e clareza em seus códigos, como exemplificado no exercício 1 do 7º encontro, em que foi solicitado para o aluno desenvolver o código em Ladder e simulá-lo por uma tela IHM (painel de controle). Ele seguiu as exigências do enunciado de maneira eficaz, estruturando o código de forma lógica e utilizando a IHM para facilitar a execução do processo e validar o sistema com sucesso. O desempenho consistente e eficiente do aluno A ao longo da oficina o coloca no **Cluster 1**, destacando-se como um dos melhores desempenhos do curso.

O exercício 1, mencionado, consistia em resolver um desafio sobre girar um motor em dois sentidos. O enunciado sinalizava que ao pressionar um botão, o motor iria girar para um lado. Ao pressionar outro botão, o motor iria começar a girar para o outro lado. Caso o botão de desligar fosse pressionado, o motor iria parar de girar. Caso o botão de emergência fosse acionado, o motor iria parar de girar e uma lâmpada iria acender avisando o operador.

A Figura 3 apresenta a solução do aluno A, em que ele resolveu o problema corretamente utilizando as funções da bobina set e da bobina reset.

Figura 3 – Código em Ladder do exercício 1 do encontro 7 (1/7) (aluno A)



Fonte: Corpus da pesquisa com anotações do autor.

O aluno A também elaborou a IHM, conforme a Figura 4. Destaca-se que o aluno identificou todos os botões da IHM, facilitando a execução do processo. Ele, além de alterar a cor do fundo da IHM, também colocou uma animação com setas, indicando sentido do giro do motor. Essas funcionalidades não foram apresentadas na oficina, indicando que o aluno buscou novas maneiras de resolver o problema por meio da mediação pelo software.

Figura 4 – Telas de IHM do exercício 1 do encontro 7 (1/7) (aluno A)



Fonte: Corpus da pesquisa.

No final da oficina, o aluno A desenvolveu corretamente o exercício 1 do encontro 10. Para isso, ele teve que utilizar o ambiente 3D que ele mesmo desenvolveu no exercício 1 do encontro 9 (vide Figura 5). Então, a partir do processo industrial de seleção de caixas pequenas e grandes em uma esteira da Figura 5, o aluno teve que desenvolver a lógica para o exercício 1 do encontro 10. O enunciado mencionava que ao pressionar o botão de ligar, a lâmpada (que fica no próprio botão)

deveria ser acionada e o sistema deveria ser ativado, ou seja, a esteira seria ligada e começariam a surgir caixas no alimentador.

Figura 5 – Ambiente de simulação do exercício 1 do encontro 9 (1/9) (aluno A)

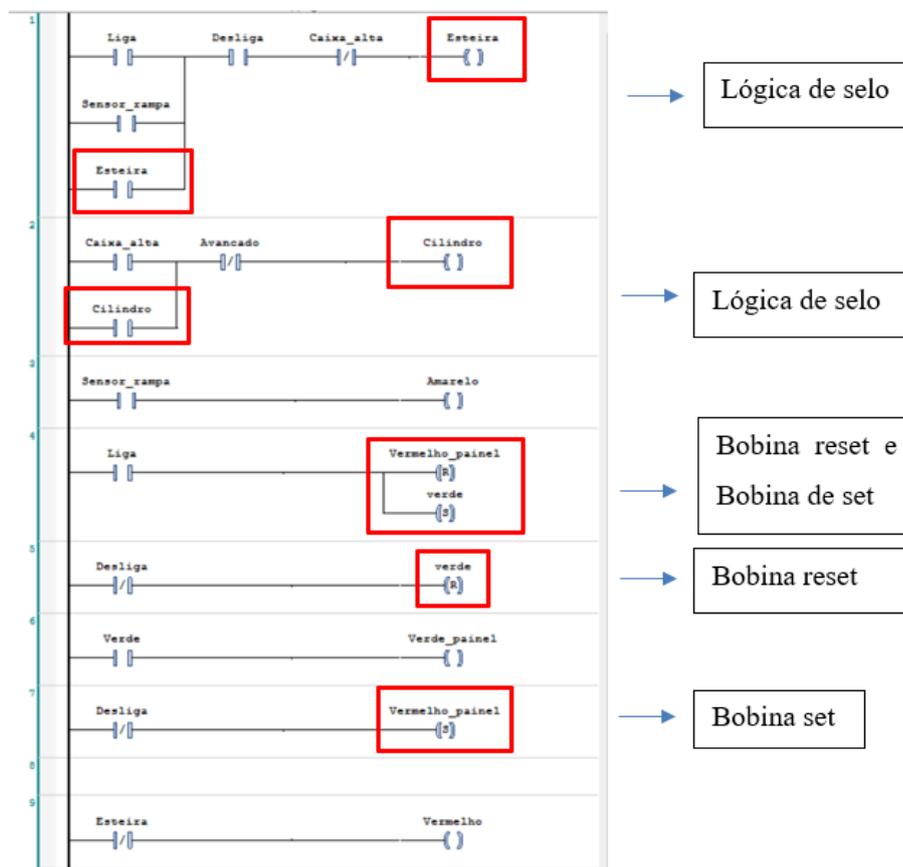


Fonte: *Corpus da pesquisa.*

Ao pressionar o botão de *stop*, a lâmpada de *stop* (que fica no próprio botão) deveria ser acionada e o sistema deveria parar. Essa lâmpada deve ficar ligada sempre que o sistema estiver parado. Assim, sempre que o sensor 1 (próximo ao cilindro) detectasse uma caixa alta, o cilindro deveria avançar empurrando a caixa para a rampa do meio. As caixas baixas (que não são detectadas pelo sensor 1) deveriam ser direcionadas para o final da esteira, sendo detectadas pelo sensor 2. Quando o sensor 2 detectasse uma caixa baixa, uma lâmpada no sinalizador deveria ser acionada (a lâmpada iria permanecer ligada enquanto o sensor estivesse detectando a caixa).

O aluno A desenvolveu o código corretamente, de forma organizada, conforme Figura 6, utilizando as funções da lógica de selo, da bobina set e da bobina reset, aprendidas em exercícios anteriores. A nomenclatura das variáveis como “Liga”, “Desliga”, “Sensor_rampa”, “Caixa_alta”, etc. facilitam o entendimento do código. Observou-se que o aluno A desde os exercícios iniciais obteve um ótimo desempenho, construindo solução claras e inovando.

Figura 6 – Código em *Ladder* do exercício 1 do encontro 10 (1/10) (aluno A)



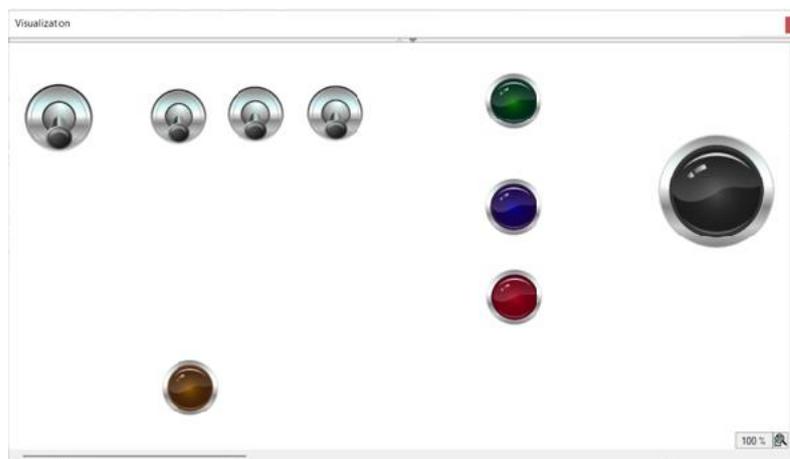
Fonte: *Corpus* da pesquisa com anotações do autor.

Em outro caso a analisado, o aluno B começou a oficina com algumas dificuldades em termos de organização do código e desenvolvimento lógico, como observado no exercício 4 (do 5º encontro), onde a sequência de operações foi executada de forma confusa e com problemas na nomenclatura das variáveis, dificultando a simulação. O referido exercício solicitava que fosse programado um processo de fabricação de painéis por estampagem. Neste caso o aluno deveria criar o código em *Ladder* e uma tela IHM (um painel de controle) para facilitar simular o processo, não era solicitado a criação do sistema em 3D. O processo envolvia uma sequência específica de operações, começando com o operador posicionando uma chapa metálica sob uma prensa vertical (acionada por um cilindro hidráulico). Um sensor deveria detectar a presença da chapa, enquanto outro sensor deveria verificar se o operador estava a uma distância segura para permitir o início do processo. Após a confirmação de que o operador estava fora da zona de perigo, o cilindro hidráulico deveria ser acionado automaticamente, descendo para estampar a chapa. Um terceiro

sensor era responsável por detectar quando o cilindro atingisse seu ponto final, momento em que o cilindro deveria subir novamente à sua posição inicial de forma automática. Finalmente, o operador retiraria a panela recém estampada e colocaria uma nova chapa, reiniciando o ciclo do processo. Além dessas instruções, os alunos também foram solicitados a desenvolver uma tela de IHM para facilitar a execução e monitoramento do processo.

Na solução apresentada pelo aluno B para este exercício 4, surgiram alguns problemas. Em primeiro lugar, a tela de IHM desenvolvida pelo aluno (vide Figura 7) não incluía nomenclaturas nos contatos, o que dificultou a simulação e a interpretação do processo. Isso é um aspecto crítico, pois a ausência de nomenclaturas pode tornar o controle do sistema confuso e propenso a erros durante a operação. Além disso, ao analisar o código em *Ladder* desenvolvido pelo aluno B (Figura 8), foram identificados erros que comprometeram a funcionalidade do programa. Durante a simulação, verificou-se que o cilindro não realizava o movimento de subida conforme esperado e que a lâmpada de emergência, incluída pelo aluno, não era acionada corretamente. Embora o aluno B tenha demonstrado iniciativa ao incluir uma lógica de erro e uma lâmpada de emergência – elementos não solicitados no enunciado original –, a execução dessas adições não foi eficiente e acabou complicando a lógica do sistema.

Figura 7 – Tela de IHM do exercício 4 do encontro 5 (4/5) (aluno B)



Fonte: *Corpus* da pesquisa.

Ao examinar a sequência de operações programadas (veja Figura 8), observou-se que a ordem das etapas estava incorreta em relação ao que foi solicitado. O enunciado especificava que o cilindro deveria descer, atingir seu ponto final, e então subir de volta à posição inicial, após o qual o operador deveria remover a peça e

reiniciar o processo. No entanto, o código do aluno B implementou essa sequência de forma desordenada, o que comprometeu a clareza e a eficiência do processo automatizado.

Figura 8 – Código em Ladder do exercício 4 do encontro 5 (4/5) (aluno B)



Fonte: *Corpus* da pesquisa com anotações do autor.

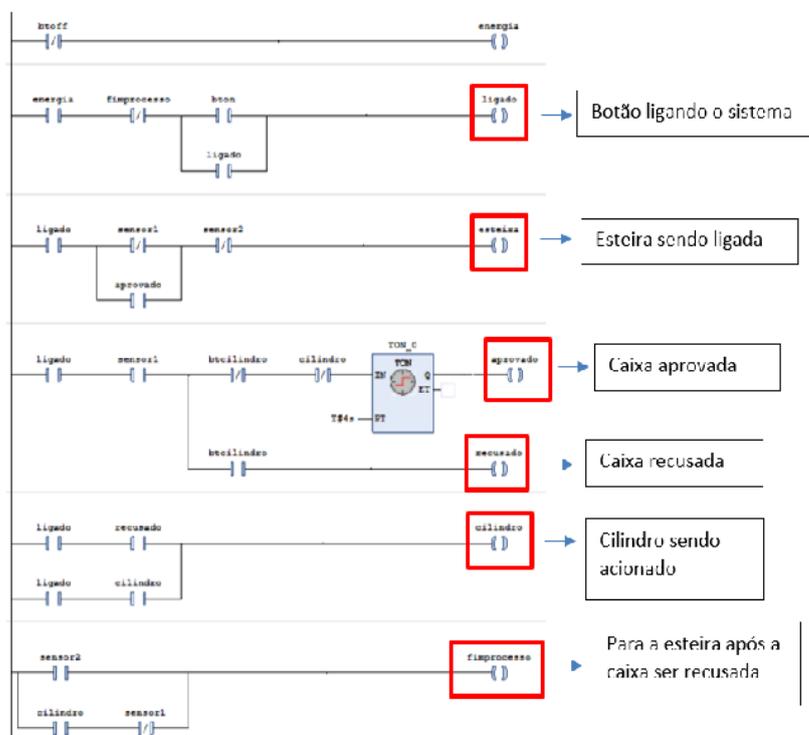
A simulação foi extremamente relevante neste cenário. Acompanhar a execução das ações, de forma visual possibilitou a compreensão dos pontos incorretos e a correção do código. Sucessivos testes e melhorias conduziram a solução correta. A repetição, aliada à simulação, permitiu ao aluno B uma evolução notável.

Na resolução de um exercício posterior (exercício 1 do 11º encontro – vide Figura 9), ele conseguiu estruturar o código de maneira lógica e sequencial, utilizando corretamente as nomenclaturas e facilitando o entendimento e a simulação do processo. No referido exercício o aluno deveria criar o código em Ladder e utilizá-lo no ambiente em 3D desenvolvido pelo próprio aluno. O objetivo era programar uma esteira de transporte de caixas pelo seguinte cenário: o operador deveria pressionar um botão para ligar a esteira. Uma vez em funcionamento, a caixa seria movida até um sensor, que ao detectá-la, deveria interromper o movimento da esteira por quatro segundos, tempo durante o qual um funcionário poderia inspecionar a caixa.

Durante esse intervalo, se a caixa for rejeitada, o funcionário iria pressionar outro botão, acionando um cilindro que empurraria a caixa para uma rampa lateral de descarte. A caixa passaria então por um segundo sensor, que acionaria a parada da esteira. Se a caixa for aprovada, ao término dos quatro segundos, a esteira retomaria seu movimento, transportando a caixa até o final do sistema, onde o processo de análise continuará. O aluno B foi capaz de programar esta solução. Tal evolução permitiu inferir que o aluno B se encaixa no *Cluster 2*, demonstrando uma curva de aprendizado positiva ao longo do curso.

No exercício 1 do 11º encontro, a magnitude do projeto a ser automatizado se torna evidente. Montar fisicamente a estrutura em 3D, representada na Figura 10, requer um investimento significativo. Nesse contexto, a simulação do processo é essencial para garantir que o projeto atenderá às exigências da empresa antes mesmo de adquirir os equipamentos. A simulação em 3D desempenha um papel crucial no processo de tomada de decisões relacionadas a investimentos em equipamentos e na inovação dentro da empresa.

Figura 9 – Código em *Ladder* para o exercício 1/11 (aluno B)



Fonte: *Corpus* da pesquisa com anotações do autor.

Para ambos os alunos, a simulação em 3D desempenhou um papel crucial na validação dos sistemas que eles programaram. Para o aluno B, a simulação ajudou a

identificar e corrigir problemas na lógica do código, contribuindo para sua evolução ao longo da oficina. A simulação foi uma ferramenta para confirmar que o sistema funcionava conforme planejado, reforçando sua confiança e compreensão dos conceitos.

Figura 10 – Ambiente de simulação em 3D do exercício 1 do encontro 11 (1/11) (aluno B)



Fonte: *Corpus da pesquisa* (2021).

O aluno B demonstrou uma curva de aprendizagem ascendente, evoluindo para se tornar mais organizado e eficiente, enquanto o aluno A manteve uma consistência forte e eficiente ao longo de todo o curso. A simulação em 3D foi um recurso valioso para ambos, facilitando o aprendizado prático e a consolidação do Pensamento Computacional.

Os resultados da oficina indicam que a intervenção pedagógica com o uso de simulações em 3D foi eficaz em promover a evolução das habilidades dos alunos na programação de CLPs. A análise dos *clusters* revelou diferentes trajetórias de aprendizado, desde alunos que mantiveram um alto nível de desempenho ao longo do curso, até aqueles que, apesar das dificuldades iniciais, conseguiram melhorar significativamente. A simulação em 3D emergiu como uma ferramenta essencial para facilitar a compreensão dos conceitos de Pensamento Computacional e para conectar teoria e prática de maneira integrada.

O desempenho dos alunos A e B ilustra claramente como diferentes perfis de aprendizagem podem se beneficiar das metodologias aplicadas. Enquanto o aluno B demonstrou uma evolução adaptativa e progressiva, o aluno A manteve uma consistência notável em seu desempenho. Esses resultados sugerem que

abordagens pedagógicas personalizadas, que considerem as necessidades individuais dos alunos, podem maximizar o impacto da educação em engenharia no contexto da Indústria 4.0.

5. Considerações finais

O presente estudo demonstrou que a integração de simulações em 3D no ensino de CLPs com a linguagem Ladder pode ser uma abordagem promissora para o desenvolvimento do Pensamento Computacional entre estudantes de Engenharia de Controle e Automação. A utilização de simulações em 3D permitiu que os alunos visualizassem e manipulassem sistemas automatizados em um ambiente controlado, facilitando a compreensão de conceitos complexos e a aplicação prática do aprendizado em situações reais de engenharia.

No entanto, é importante reconhecer que a implementação de tecnologias avançadas, como simulações em 3D, ainda enfrenta desafios significativos em muitas instituições de ensino, particularmente no que diz respeito à infraestrutura tecnológica, à formação docente e à disponibilidade de recursos. Nem todas as instituições estão preparadas para integrar essas ferramentas de maneira eficaz, o que limita a aplicabilidade imediata das práticas descritas neste estudo a um número restrito de contextos educacionais.

Adicionalmente, é importante destacar que este trabalho é preliminar e deve ser considerado como um primeiro passo na exploração do uso de simulações em 3D no ensino de programação de CLPs. Além disso, a pesquisa destacou a importância do papel do professor como mediador no processo de aprendizagem, conforme as teorias de Vygotsky e Papert, complementadas pelo conceito de transposição informática. A abordagem construcionista de Papert, que vê o aluno como protagonista do seu próprio aprendizado e o computador como uma ferramenta para isso, revelou-se particularmente eficaz no contexto da simulação em 3D. Da mesma forma, a perspectiva sociocultural de Vygotsky, que enfatiza a interação social como um motor para o desenvolvimento cognitivo, contribuiu para a criação de um ambiente de aprendizagem dinâmico e colaborativo.

Concluimos que a automatização de processos por meio de práticas pedagógicas ordenadas e integradas com simulações em 3D pode efetivamente promover o desenvolvimento de habilidades essenciais do Pensamento

Computacional. Este estudo sugere que a utilização de recursos tecnológicos da Indústria 4.0, como a simulação em 3D, é um caminho promissor para aprimorar a educação em engenharia, preparando os alunos para os desafios de um mercado de trabalho cada vez mais orientado pela tecnologia.

Assim, a aplicação dessas práticas não apenas fortalece a compreensão técnica dos alunos, mas também estimula o desenvolvimento de competências críticas e criativas, essenciais para a atuação no cenário contemporâneo da engenharia, onde a inovação e a capacidade de adaptação são fundamentais.

Referências

- BALACHEFF, N. La transposition informatique, un nouveau problème pour la didactique. In: **VINGT ANS DE DIDACTIQUE DES MATHÉMATIQUES EN FRANCE**, Grenoble: La Pensée Sauvage, p. 364-370, jun. 1993.
- DUARTE, A. E. P. *et al.* Vygotsky: suas contribuições no campo educacional. In: VI CONEDU, 2019, Campina Grande. **Anais...** Campina Grande: Realize Editora, 2019.
- LEANDRO, C. C.; FACIN, A. L. F.; COSTA NETO, P. L. O. Indústria 4.0: **Competências requeridas para a formação de engenheiros**. In: XLIII ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO (ENEGEP), 2023, Fortaleza. **Anais...**Ceará: ABEPRO, 2023.
- PAPERT, S. **Mindstorms: children, computers and powerful ideas**. New York: Basic Books, 1980.
- PAPERT, S. **A máquina das crianças: repensando a escola na era da informática**. Tradução de Sandra Costa. Porto Alegre: Artes Médicas, 2008.
- PAPERT, S. **LOGO: computadores e educação**. Tradução de José Armando Valente, Beatriz Bitelman e Afira Vianna Ripper. São Paulo: Editora Brasiliense, 1985.
- RÜßMANN, M. *et al.* Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. **Boston Consulting Group (BCG)**, 09 abril 2015.
- SILVA, A. M. *et al.* O ENSINO DE ENGENHARIA EM FACE ÀS COMPETÊNCIAS PROFISSIONAIS EXIGIDAS PELA INDÚSTRIA 4.0. In: XLVIII Congresso Brasileiro de Educação em Engenharia (COBENGE) e III Simpósio Internacional de Educação em Engenharia da ABENGE, 2020, Brasília. **Anais...** Brasília: ABENGE, 2020.
- VYGOTSKY, L. S. **A Formação Social da Mente**. 4. ed. São Paulo: Martins Fontes, 1991.
- WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.