

CAPACITAÇÃO AVANÇADA EM ROBÓTICA INDUSTRIAL

FERRAMENTAS ACADÊMICAS PARA SOLUÇÕES INDUSTRIAIS



ORGANIZADORES

Jeanne Moreira de Sousa

José Pinheiro Queiroz Neto

José Cavalcante Lacerda Junior



Editora Poisson



Jeanne Moreira de Sousa
José Pinheiro Queiroz Neto
José Cavalcante Lacerda Junior
Organizadores

Capacitação avançada em robótica industrial: ferramentas acadêmicas para soluções industriais

1ª Edição

Belo Horizonte
Editora Poisson
2024

Editor Chefe: Dr. Darly Fernando Andrade

Conselho Editorial

Dr. Antônio Artur de Souza – Universidade Federal de Minas Gerais
MSc. Davilson Eduardo Andrade

Dra. Elizângela de Jesus Oliveira – Universidade Federal do Amazonas
MSc. Fabiane dos Santos

Dr. José Eduardo Ferreira Lopes – Universidade Federal de Uberlândia

Dr. Otaviano Francisco Neves – Pontifícia Universidade Católica de Minas Gerais

Dr. Luiz Cláudio de Lima – Universidade FUMEC

Dr. Nelson Ferreira Filho – Faculdades Kennedy

MSc. Valdiney Alves de Oliveira – Universidade Federal de Uberlândia

Dados Internacionais de Catalogação na Publicação (CIP)

C236

Capacitação avançada em robótica industrial:
ferramentas acadêmicas para soluções
industriais/ Organizadores: Jeanne Moreira de
Sousa, José Pinheiro Queiroz Neto, José
Cavalcante Lacerda Junior Belo Horizonte-MG:
Editora Poisson, 2024

Formato: PDF

ISBN: 978-65-5866-378-2

DOI: 10.36229/978-65-5866-378-2

Modo de acesso: World Wide Web

Inclui bibliografia

1. Robótica 2. Educação. II. Sousa, Jeanne
Moreira de II. Queiroz Neto, José Pinheiro
III. Lacerda Junior, José Cavalcante IV. Título.

CDD-620

Sônia Márcia Soares de Moura - CRB 6/1896



O conteúdo deste livro está licenciado sob a Licença de Atribuição Creative Commons 4.0.

Com ela é permitido compartilhar o livro, devendo ser dado o devido crédito, não podendo ser utilizado para fins comerciais e nem ser alterada.

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos seus respectivos autores

www.poisson.com.br
contato@poisson.com.br

Organizadores

Jeanne Moreira de Sousa: Possui Licenciatura, Bacharelado e Mestrado em Matemática pela Universidade Federal do Amazonas; Doutorado em Clima e Ambiente pelo Instituto Nacional de Pesquisas da Amazônia - INPA/UEA; Pós-doutorado em Modelagem Numérica e Previsão de Tempo e Clima no Laboratório de Modelagem Climática - LMC/INPA.

José Pinheiro Queiroz Neto: Formado em Engenharia Elétrica e com Especialização em Engenharia de Produção pela Universidade Federal do Amazonas, desenvolveu os cursos de Mestrado e Doutorado em Ciência da Computação na Universidade Federal de Minas Gerais e Pós-doutorado em Educação Profissional na Tampere University of Applied Science (TAMK) / Finlândia.. Como profissional, atuou em empresas do Parque Industrial de Manaus, desenvolvendo atividades na área de Engenharia Industrial, tendo sido Chefe de Engenharia Industrial, na Gradiente Eletrônica S/A. Na área acadêmica, foi coordenador dos cursos Superiores do IFAM/Campus Distrito Industrial, Professor colaborador dos Mestrados em Ciência da Computação (UFAM) e Engenharia Elétrica (UFAM), Diretor Geral do Campus Distrito Industrial e Pró-Reitor de Pesquisa, Pós-Graduação e Inovação do do Instituto Federal de Educação, Ciência e Tecnológica do Amazonas (IFAM). Foi General Chair do XIX SIBGRAPI (Brazilian Symposium on Computer Graphics) em 2006. Participou como bolsista CNPq do Programa Professores para o Futuro - Finlândia. Atualmente é professor pesquisador do IFAM, professor do Mestrado/Doutorado PPGET, Membro da Associação Polo Digital de Manaus (APDM) e do Conselho Superior do IFAM. Na área de pesquisa, participa de grupos voltados a área de Ensino, com particular interesse em ferramentas modernas de ensino e educação, e grupo de pesquisa na área de Ciência da Computação/Automação, com particular interesse em Visão Computacional, Robótica e Sistemas Embarcados.

José Cavalcante Lacerda Junior: Graduado em Filosofia, Psicologia e Pedagogia. Especialista em Psicologia Jurídica e Saúde Mental. Mestre em Educação em Ciências na Amazônia. Doutor em Ciências do Ambiente e Sustentabilidade na Amazônia. Pesquisador e Líder do Grupo de Pesquisa Tecnologia, Educação e Meio Ambiente - TEMA. Possui interesse em assuntos relacionados a infância e adolescência, educação ambiental na cidade, divulgação científica, filosofia no contexto amazônico e educação profissional e tecnológica. Atua como professor no Instituto Federal de Educação, Ciência e Tecnologia no Amazonas.

SUMÁRIO

Capítulo 1: Características do curso capacitação avançada em robótica industrial... 06

Jeanne Moreira de Sousa, José Cavalcante Lacerda Junior, Vitor Bremgartner da Frota

DOI: 10.36229/978-65-5866-378-2.CAP.01

Capítulo 2: Disciplina de múltiplas competências e *hands on* 13

José Pinheiro de Queiroz Neto, Micila Sumária Medeiros Pereira

DOI: 10.36229/978-65-5866-378-2.CAP.02

Capítulo 3: Introdução á robotica industrial..... 22

Ricardo Augusto Medeiros de Oliveira, Benjamim Batista de Oliveira Neto

DOI: 10.36229/978-65-5866-378-2.CAP.03

Capítulo 4: Disciplina de programação para aplicações robóticas..... 41

Lenildo Marcos da Mota Silva

DOI: 10.36229/978-65-5866-378-2.CAP.04

Capítulo 5: Disciplina de programação em CLP 67

Hugo Alves Veloso

DOI: 10.36229/978-65-5866-378-2.CAP.05

Capítulo 6: Disciplina de programação em IHM 81

Vanderson de Lima Reis

DOI: 10.36229/978-65-5866-378-2.CAP.06

Capítulo 7: Disciplina de visão computacional..... 94

Marcelo Chamy Machado

DOI: 10.36229/978-65-5866-378-2.CAP.07

Capítulo 8: Construção coletiva: parcerias 110

José Cavalcante Lacerda Junior

DOI: 10.36229/978-65-5866-378-2.CAP.08

Características do curso capacitação avançada em robótica industrial

Jeanne Moreira de Sousa¹, José Cavalcante Lacerda Junior², Vitor Bremgartner da Frota³

Ao longo da História, o mundo sempre passou por transformações tecnológicas. Desde os primórdios, na Idade da Pedra, com as primeiras invenções, como a roda, até os dias atuais, com a Indústria 4.0 e sua automação inteligente. Essas transformações ocorreram não somente pela habilidade criativa do ser humano, mas também pelas necessidades que cada época apresentava. Hoje, com a constâncias das novas tecnologias, o ser humano encontra-se ainda com desafios a serem superados envolto a um contínuo processo de transformação e adaptação.

Esta perspectiva não é diferente em nosso contexto amazônico. O desenvolvimento de tecnologias, a implementação de processos cada vez mais eficientes e o diálogo com os elementos da bioeconomia (Nobre; Nobre, 2019) incidem sobre a Amazônia a demanda de novos produtos nos quais as empresas procuram métodos de inovação para os seus produtos. Neste contexto, a automação dos processos na indústria e o incremento robótico são métodos usados para a agilidade e qualidade no desenvolvimento de produtos e oferta de serviços. Em Manaus, capital do estado do Amazonas, temos o Polo Industrial o qual é composto por diversas empresas, nacionais e multinacionais, que necessitam de agilidade na fabricação de novos produtos (Oliveira, 2021).

Para isso, é necessário que as pessoas que trabalham nessas indústrias tenham conhecimento das demais ferramentas e dos processos tecnológicos, mantendo-se sempre atualizados com tecnologias de ponta para que continuem firmes e atuantes neste mundo do trabalho (Borges, 2015). Uma das ferramentas largamente utilizadas nas indústrias, mas que ainda não possui uma grande difusão na região amazônica é o braço robótico e suas tecnologias associadas. Nota-se uma carência de profissionais locais que tenham experiência nesta área.

É nesse contexto que o Curso de Capacitação Avançada de Robótica Industrial emerge na parceria entre o público e o privado, isto é, entre o Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM) e a LG *Electronics*. Seu escopo é a capacitação, por intermédio de recursos humanos e materiais, qualificada para o mundo do trabalho. O curso é materialização da interface indústria e ensino superior, onde as urgências do contemporâneo - trabalho e inovação, por exemplo - interpõem a necessidade do diálogo.

¹ Doutora em Clima e Ambiente. Coordenadora do Curso Capacitação Avançada em Robótica Industrial. Professora do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

² Doutor em Ciências do Ambiente e Sustentabilidade na Amazônia. Apoio Pedagógico do Curso de Capacitação Avançada de Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

³ Doutor em Informática. Orientador de alunos do Curso Capacitação Avançada em Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

Nesse sentido, o principal objetivo deste curso é promover e desenvolver estratégias de ensino e aprendizado, por meio da capacitação na área de robótica industrial, com enfoque em tecnologias emergentes conectadas às necessidades e potencialidades do Polo Industrial de Manaus. Desta maneira, suas ações especificam sua finalidade, como: desenvolvimento de estratégias para aproximação academia e indústria; capacitação de alunos a partir da Aprendizagem Baseada em Projetos; desenvolvimento de material didático, habilitação técnica-docente para manuseio de tecnologias docentes; difusão do conhecimento construído e desenvolvimento de propostas e de produtos mínimos viáveis (MVPs).

A base metodológica que sustenta o curso é a Aprendizagem Baseada em Projetos, em inglês *Problem-Based Learning* – PBL. Tal abordagem metodológica considera o processo de aprendizagem centrado na resolução de problemas que emergem de contexto empírico, onde o repasse da informação/conhecimento é transmutado pela aplicação prática a partir de determinadas problemáticas (Queiroz-Neto; Farias; Chagas, 2021). Nesta metodologia, de modo geral, os alunos são organizados em equipes com o intuito de identificar, analisar e desenvolver uma estratégia criativa que solucione determinada problemática tendo como parâmetro interdisciplinariedade das áreas que envolvem o mesmo.

Na PBL o aluno é o gerador do seu próprio conhecimento, podendo ser aplicado no ambiente da própria sala de aula ou em outro ambiente/local fora da sala de aula, realizando a quebra de paradigmas do tradicionalismo e tornando o aluno o ator principal no processo de aprendizagem, sendo o professor aquele que faz o elo entre a teoria e a prática, rompendo coma forma tradicional de transmitir o conteúdo [...] (Queiroz-Neto; Farias; Chagas, 2021, p. 1794).

Com efeito, a execução da proposta apresentada trouxe consigo a necessidade pensar 04 (quatro) elementos fundamentais para o seu desenvolvimento: i) Construção do Laboratório de Robótica Industrial (LRI); ii) Formação docente; iii) Seleção e acompanhamento dos alunos; iv) Capacitação.

1.1. LABORATÓRIO DE ROBÓTICA INDUSTRIAL

Com características inovadoras para o contexto amazônico, o *lôcus* de execução do curso se deu a partir da ampliação e modernização dos laboratórios existentes no Polo de Inovação do IFAM (INOVA), com o intuito de comportar e viabilizar a instalação de sistemas robóticos e de outros equipamentos necessários para realização dos treinamentos e capacitações previstas no projeto, conforme se verifica na Figura 1.1.

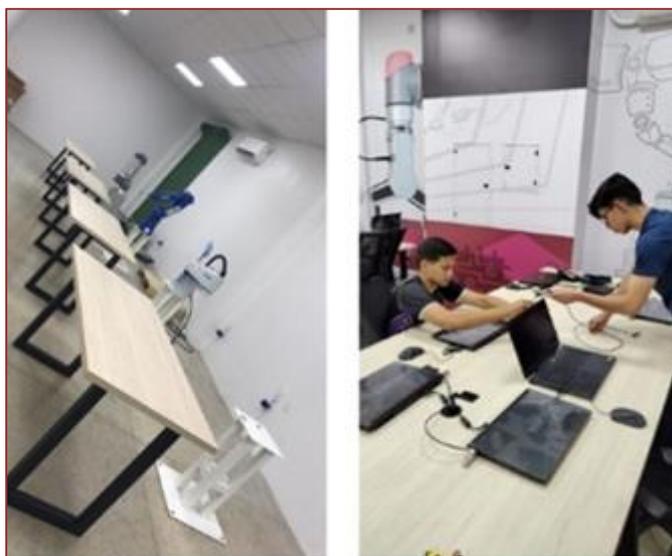
Figura 1.1: Modernização dos laboratórios



Fonte: Os autores, 2023.

No decorrer deste processo foi realizada ainda a aquisição de equipamentos, materiais permanentes e consumos previstos para o desenvolvimento para as aulas e práticas do curso. Destaca-se, principalmente, os equipamentos de multimídias, cabos de rede, telas, acessórios, robôs e recursos digitais, conforme de verifica na Figura 1.2.

Figura 1.2: Modernização dos laboratórios



Fonte: Os autores, 2023.

1.2. FORMAÇÃO DOCENTE

A temática destacada, robótica industrial, possui suas restrições no contexto amazônico. Daí a segunda atenção voltada no processo de desenvolvimento do curso foi a formação docente. Esta etapa se caracteriza pela disseminação do conhecimento e interação dos docentes que iriam atuar no curso com as plataformas robóticas selecionadas para a construção do curso. A principal finalidade é incorporar

conhecimentos avançados de robótica industrial, a partir de encontros, módulos técnicos e *workshops* práticos sobre as plataformas de arte robótica atualmente utilizadas na industrial nacional e internacional, conforme se verifica na Figura 1.3.

Figura 1.3: Encontro com os docentes



Fonte: Os autores, 2023

O fornecimento de informações, o compartilhamento de conhecimentos e a realização de práticas intencioam oportunizar aos docentes uma proficiência na utilização das plataformas bem como construir estratégias para otimizar o processo de aprendizagem com os alunos. Sendo assim, foram selecionadas as seguintes plataformas para fundamentar o desenvolvimento do curso, conforme se observa na Figura 1.4.

Figura 1.4 Plataformas utilizadas.

Robô Epson	Robôs Epson são utilizados em grandes plantas de eletrônicos e semicondutores em todo o mundo. As aplicações específicas incluem manuseio e encaixe de chips, montagem de codificadores, testes de placas e diodo laser e conexão de cabos.
Robô Yaskawa	Adequado para soldagem, manuseio geral, usinagem e também aplicações de uso geral que requerem uma carga útil elevada em uma distância de trabalho relativamente longa.
Robô Staubli	Indicado para aplicações de ciclo curto, como a manipulação ou a carga e descarga de peças de uma máquina de injeção, por exemplo. Devido à sua elevada precisão (0,04 mm), mas também pode ser utilizado para aplicações de maquinação.
Cobot Universal Robots	Um <i>cobot</i> potente permite automatizar processos e tarefas com pesos até 10kg e com um elevado nível de precisão. Graças ao seu alcance de 1.300 mm é perfeito para aplicações de embalagem, paletização e montagem.

Fonte: Os autores, 2023.

1.3. SELEÇÃO E ACOMPANHAMENTO DOS ALUNOS

Como curso de capacitação, as vagas destinadas para o processo formacional se destinou aos alunos de graduação em engenharias e/ou tecnologias do IFAM e instituições parceiras. Para tanto, seu processo de seleção se deu a partir de um edital de chamada pública, tendo como requisitos:

- Aluno matriculado em curso de graduação na área de engenharia ou tecnologia, cursando a partir do 4º. Semestre da graduação;
- Com coeficiente igual ou superior a 7 (sete);
- Histórico composto por disciplinas com alinhamento ao curso de Capacitação em Robótica Industrial.

O processo de seleção fez a organização de duas turmas composta por 15 (quinze) alunos em cada. Para além dos professores de cada disciplina, as turmas contaram com a participação de orientador, apoio pedagógico, laboratoristas para o auxílio no desenvolvimento das atividades, como se observa na Figura 1.4.

Figura 1.4 Orientação de alunos.



Fonte: Os autores, 2023.

1.4. CAPACITAÇÃO

A capacitação em si abrange duas trilhas: i) capacitação em módulos; ii) Jornada Hands On

Trilha 1: Capacitação em Módulos

Realizada a partir de disciplinas que fundamentam e põem em prática os conceitos adquiridos, como Introdução à Robótica Industrial e Múltiplas Competências e *Hands On*, Programação para Robótica, Programação CLP, Programação IHM e Visão Computacional, conforme distribuição apresentada na Tabela 1.1.

Tabela 1.1. Distribuição de carga horária da capacitação

CAPACITAÇÃO	CARGA HORÁRIA
Múltiplas Competências e <i>Hands On</i>	40 h
Introdução a Robótica Industrial	40 h
Programação para aplicações robóticas	60h
Programação em CLP	60h
Programação IHM	60h
Visão computacional e aplicações Industriais	60 h

FONTE: Autores, 2023.

Trilha 2: Jornada *Hands On*

Desenvolvida a partir de ideias e soluções para os “gargalos” industriais. Sua dinâmica será pautada a partir da formação de grupos e a identificação de um problema a ser solucionado e apresentado ao final da jornada. Destaca-se que as ideias implementadas devem ter como orientação a otimização dos processos industriais em diálogo com a empresa investidora (*LG Eletronics*). Sendo assim, o percurso da jornada pode ser identificado, conforme Figura 1.5

Figura 1.5: Jornada *Hands On*



FONTE: Autores, 2023

A construção deste curso viabiliza, por fim, não apenas o desenvolvimento de habilidades técnicas dos alunos (*hard skills*), mas também as socioemocionais (*soft skills*). Sendo assim, os capítulos a seguir apresentam as disciplinas ministradas no curso, com seus professores. O curso também contou com um orientador de alunos, que acompanhou os alunos nos desafios propostos pela empresa patrocinadora (*LG Electronics*).

REFERÊNCIAS

- [1] BORGES, Juarez Camargo. A qualificação profissional do trabalhador para o mercado de trabalho e ambiente organizacional. **Artigo de Mestrado - Faculdades Integradas de Taquara-FACCAT**, 2015.
- [2] NOBRE, Ismael; NOBRE, Carlos. Projeto 'Amazônia 4.0': definindo uma terceira via para a Amazônia. **Futuribles, São Paulo**, v. 2, p. 7-20, 2019.
- [3] OLIVEIRA, Antonio Pereira de. **Polo industrial de Manaus: a biodiversidade como mercadoria**. 2021. 1 recurso online (280 p.) Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Filosofia e Ciências Humanas, Campinas, SP. Disponível em: <https://hdl.handle.net/20.500.12733/1642189>. Acesso em: 25 jan. 2024.
- [4] QUEIROZ-NETO, José Pinheiro; DE FARIAS, Marcella Sarah Filgueiras; CHAGAS, Emanuelle Lorena Teixeira. Project based learning E design thinking Em um projeto de intercâmbio. **Revista Ibero-Americana de Estudos em Educação**, p. 1791-1806, 2021.

2.1. CARACTERIZAÇÃO DA DISCIPLINA

Esta disciplina introduz os conceitos e métodos aplicados durante toda a formação, que serão utilizados nas demais disciplinas e na Jornada *Hands On*. Ancorada na teoria do construtivismo de Jean Piaget e materializada por John Dewey no que chamou de filosofia *Learning by doing* (aprender fazendo), esta formação utiliza o que chamaremos de filosofia *hands on* ao longo de todo o processo formativo (Moran, 2018).

Esta disciplina, portanto, apresenta a integração de duas Metodologias Ativas que, integradas, aproveitam as múltiplas competências dos alunos, utilizando a integração das metodologias de Aprendizagem Baseada em Projetos (PBL) e *Design Thinking* (DT) para desenvolver habilidades cognitivas (*hard skills*) e socioemocionais (*soft skills*) em um ambiente acadêmico/industrial que traz uma educação significativa e potencializa a aprendizagem dos alunos.

2.1.1. EMENTA

Conceitos e aplicações de *Project Based Learning* (PBL); Conceitos e aplicações de *Design Thinking* (DT); Fases do DT: Imersão; Ideação; Prototipação. Aplicação prática.

2.1.2. OBJETIVOS

Preparar os alunos para a compreensão e utilização da filosofia *hands on* da formação, e introduzir a metodologia *Design Thinking* (DT) de desenvolvimento de produtos e serviços focados nas necessidades, desejos e limitações dos usuários, em abordagens de soluções que possam converter dificuldades e limitações em benefícios para o cliente e valor de negócio para a sua empresa.

2.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 24h

Carga Horária Prática: 16h

Total: 40h

¹ Doutor em Ciência da Computação. Orientador Pedagógica do Curso de Capacitação Avançada de Robótica Industrial e professor da Disciplina de Múltiplas Competências e *Hands On*. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

² Especialista em Engenharia de Software com ênfase em Desenvolvimento para Web. Professora do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

2.2. CONTEÚDOS

2.2.1. MÚLTIPLAS COMPETÊNCIAS

Diversas instituições acadêmicas têm repensado suas práticas de ensino para desenvolver não apenas habilidades técnicas (*hard skills*), mas também habilidades socioemocionais (*soft skills*) nos estudantes. As *soft skills*, que incluem liderança, comunicação, resolução de conflitos, empatia, ética, flexibilidade e gestão de equipes, entre outros, são vistas como complementares, mas não menos importantes. A ideia é alcançar uma formação integral, que considera não apenas aspectos cognitivos, mas também físicos, intelectuais, sociais, políticos e científicos. (Queiroz-Neto *et al.*, 2021).

De fato, a formação para o mundo do trabalho não pode ser uma formação meramente tecnicista, superando a dualidade da divisão entre trabalho manual e trabalho intelectual. Segundo o Fórum Econômico Mundial (FÓRUM, 2023), as principais habilidades necessárias para o mundo do trabalho em 2027 podem ser sumarizadas na Figura 2.1. Observe que estas habilidades se dividem entre *soft* e *hard skills*, exigindo do aluno que desenvolva diversificadas habilidades, ou competências, como prefiro chamar.

Figura 2.1 – Habilidades para o mundo do trabalho em 2027.

1.  Analytical thinking	6.  Technological literacy
2.  Creative thinking	7.  Dependability and attention to detail
3.  Resilience, flexibility and agility	8.  Empathy and active listening
4.  Motivation and self-awareness	9.  Leadership and social influence
5.  Curiosity and lifelong learning	10.  Quality control
<p>Type of skill</p> <p>  Cognitive skills  Self-efficacy  Management skills  Technology skills  Working with others </p>	

Fonte: FORUM, 2023.

Portanto, é necessário desenvolver suas Múltiplas Competências, que são habilidades que permitem a uma pessoa desempenhar diferentes tarefas ou funções com eficiência e qualidade. As múltiplas competências podem ser adquiridas por meio de educação, treinamento, experiência ou autoaprendizagem. As múltiplas competências são valorizadas no mercado de trabalho, pois aumentam a flexibilidade, a adaptabilidade e a criatividade dos profissionais. As múltiplas competências também podem contribuir para o desenvolvimento pessoal e social, pois ampliam os horizontes e as possibilidades de atuação.

As múltiplas competências são, na prática, o ato de desenvolver nossas múltiplas inteligências. A teoria das múltiplas inteligências foi proposta pelo psicólogo e educador Howard Gardner, na década de 80, e defende que existem diferentes tipos de inteligência

que se manifestam em diferentes domínios da vida. Segundo Gardner (1995), as inteligências são: linguística, lógico-matemática, musical, espacial, corporal-cinestésica, interpessoal, intrapessoal e naturalista. Cada pessoa possui um perfil de inteligências que se desenvolve ao longo da vida, de acordo com as experiências e oportunidades de aprendizagem. Reconhecer as múltiplas inteligências é importante para valorizar a diversidade humana e promover o desenvolvimento integral dos indivíduos, conforme ilustra a Figura 2.2

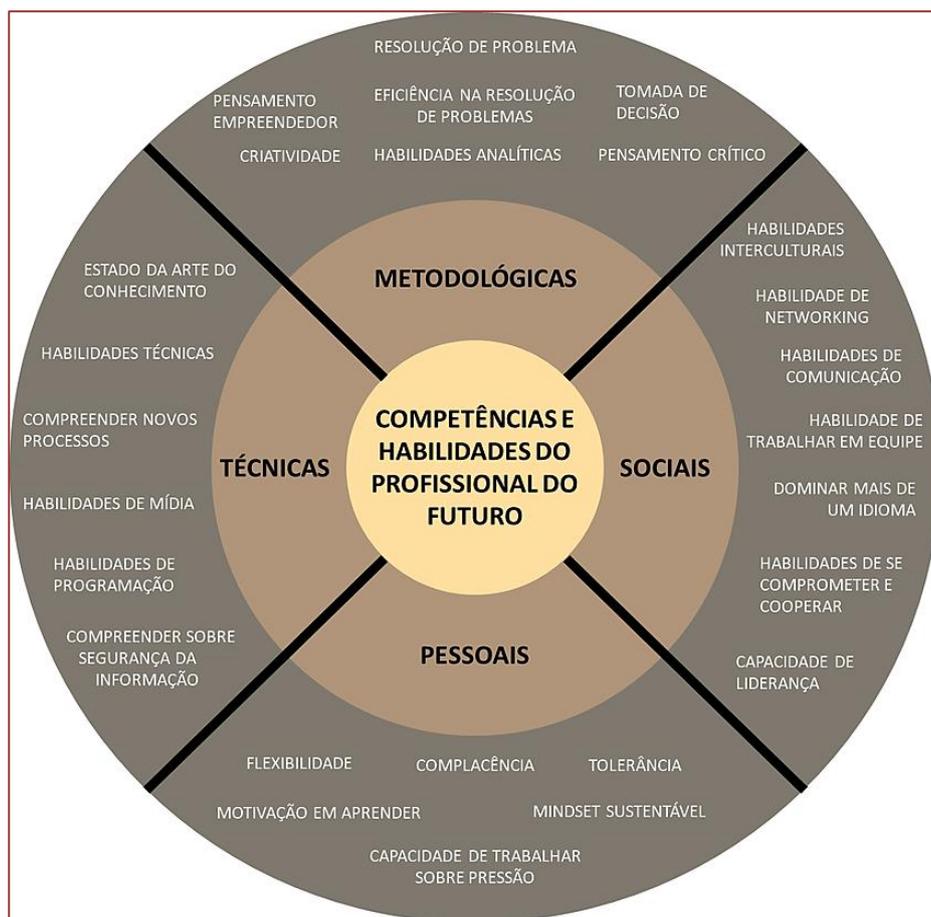
Figura 2.2 – Múltiplas Inteligências do ser humano.



Fonte: Gardner, 1995.

Portanto, utilizar estas múltiplas inteligências para gerar múltiplas competências é um desafio importante quando se trata de formação de alto nível, para a qual se faz necessária uma metodologia que possa entender estes aspectos e explorá-los de maneira proativa. A Figura 2.3 apresenta habilidades e competências que, segundo o autor seriam do futuro, mas que são atuais e cada vez mais necessárias (Hecklau *et al.*, 2017).

Figura 2.3 – Múltiplas Inteligências do ser humano.



Fonte: HECKLAU *et al.*, 2017.

2.2.2. APRENDIZAGEM BASEADA EM PROJETOS INTEGRADA A DESIGN THINKING

A formação empregou uma metodologia ativa para o aprimoramento das *hard* e *soft skills*, combinando a Aprendizagem Baseada em Projeto (*Project Based Learning - PBL*) com o *Design Thinking* (DT), este último uma abordagem prática para a integração de habilidades e mentalidade inovadora na criação de soluções tecnológicas. Esta combinação resulta em um processo de aprendizagem que promove a autonomia dos alunos, exigindo do professor conhecimento e disposição para atuar como mentor. O método coloca o aluno no centro do processo, permitindo-lhe agir de forma autônoma e visando uma formação humana integral (Barbosa; Moura, 2013).

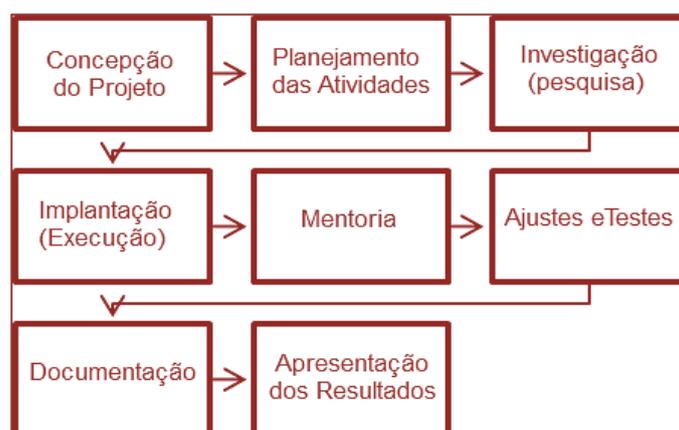
As interações colaborativas com equipes multidisciplinares proporcionaram o ambiente ideal para o desenvolvimento das competências necessárias às aplicações de Robótica Industrial de alto nível. Entre as habilidades aprimoradas, destacam-se a liderança, a comunicação, a resolução de conflitos, a empatia, a ética, a flexibilidade e a gestão de equipes.

A PBL é um método de ensino que envolve os alunos em projetos do mundo real, significativos para eles. Os alunos trabalham em um projeto por um período definido, resolvendo um problema do mundo real ou respondendo a uma pergunta complexa. Eles

demonstram seus conhecimentos e habilidades criando um serviço ou produto para um público real, neste caso, para a empresa parceira do projeto (ALMULLA, 2020).

Importante ressaltar que PBL difere de somente "fazer um projeto", pois exige pensamento crítico, resolução de problemas, colaboração e várias formas de comunicação. Os alunos precisam fazer mais do que apenas lembrar informações. Eles precisam usar habilidades de pensamento para buscar, interpretar, utilizar e transformar essas informações, além de aprender a trabalhar em equipe. (Torre-Neches, 2020). Diversos autores apresentam um fluxo de etapas para aplicação da PBL, e neste projeto utilizamos o desenvolvido por Queiroz-Neto et al. (2015) e dos Santos et al. (2019), apresentando na Figura 2.4.

Figura 2.4 – Etapas da PBL.



Fonte: Traduzido de Queiroz-Neto *et al.*, 2015.

Design Thinking (DT) é um método que visa resolver problemas de maneira criativa, levando em consideração as necessidades dos usuários e seus feedbacks na criação de produtos ou serviços. Ele envolve a observação de contextos da vida real, a análise de dados para entender melhor as necessidades do usuário e a adoção de uma mentalidade criativa para formar ideias e testar soluções. O DT não é um método específico, mas uma abordagem que organiza ideias para estimular a tomada de decisões e a busca pelo conhecimento, que classicamente possui cinco etapas: empatizar, definir, idealizar, prototipar e testar (Panke, 2019).

Nesta formação utilizamos uma versão prática do DT com três etapas que contém as cinco originais (Figura 2.5) que são: imersão (empatia e definições), ideação e prototipação (incluindo testes). A imersão é a forma de estudar o problema e compreendê-lo sobre diversas perspectivas e definir os limites do projeto, criando-se um entendimento das necessidades a serem sanadas. A Ideação é a geração de propostas de soluções direcionadas por critérios norteadores, oriundos da imersão, de forma a atender as necessidades mapeadas dos clientes e responder ao problema definido. Finalmente, a Prototipação tem como objetivo testar de maneira rápida, com baixo custo, a solução encontrada e submetê-la para aprovação do cliente (Queiroz-Neto *et al.*, 2021).

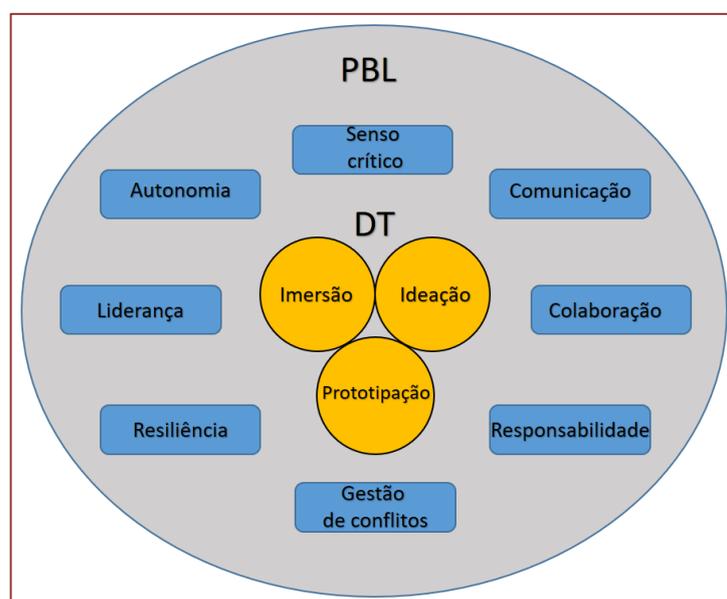
Figura 2.5 – Etapas do Design Thinking



Fonte: Os autores, 2023.

A Figura 2.6 apresenta a integração da PBL e DT. Nesta a DT está atuando como método de desenvolvimento organizado de um produto inovador, dentro do arcabouço da PBL, com suas etapas na Figura 2.4, que se integram às etapas da DT, na Figura 2.5.

Figura 2.6 – PBL-DT integrados para a formação em Robótica Industrial.



Fonte: Queiroz-Neto *et al.*, 2021.

A formação em Robótica Industrial, portanto, foi desenvolvida com base na PBL integrada ao DT. A PBL definiu a organização geral do projeto, incluindo suas etapas macro, o conceito de mentoria, o foco no estudante e sua autonomia, e as avaliações formativas semanais, através de orientações regulares e acompanhamento dos times de alunos. A DT, por sua vez, trouxe a organização interna do projeto, definindo etapas dentro do desenvolvimento do projeto em si, com base na tríade imersão, ideação e prototipação.

Com isto, entendemos que para além da aprendizagem em Robótica Industrial, esta metodologia permitiu ao aluno uma formação humana integral, sendo o entendimento de que esta gira em torno da integração de todas as dimensões e potencialidades do indivíduo no processo educativo. Essas dimensões, segundo Frigotto (2012, p. 267) “envolvem sua vida corpórea material e seu desenvolvimento intelectual, cultural, educacional, psicossocial, afetivo, estético e lúdico”.

2.3 PROCESSO AVALIATIVO

Nesta disciplina, foi utilizada a avaliação formativa, que tem seu foco no processo ensino-aprendizagem e está diretamente interligada a este, constituindo-se parte de um todo e não seu fim em si, e não tem finalidade de medição da aprendizagem em parâmetros quantitativos, mas assume um caráter objetivamente pedagógico sobre o que se aprendeu, como foco maior no processo (Barreira, 2019).

Com isto, a avaliação se deu a partir de experimentação prática da filosofia *hands on* e utilização da metodologia PBL/DT, na qual a turma foi dividida em seis times multidisciplinares, ou seja, diversificando a base de formação dos alunos, e estes desenvolveram as etapas do método para apresentar uma solução a um problema real apresentado no Campus Manaus Distrito Industrial. Os problemas foram de livre escolha dos times, mas delimitados nas seguintes áreas: Portaria/entrada, Limpeza, Ambientes Acadêmicos.

A Figura 2.7 apresenta quatro momentos da disciplina: integração dos alunos para autoconhecimento e conhecimento; discussão em equipe dos dados obtidos na imersão; uso de brainstorming para a fase da ideação e desenvolvimento do protótipo.

A Figura 2.8 apresenta a fase final da avaliação formativa, com a apresentação do desenvolvimento prático das fases do PBL/DT e dos resultados obtidos ao professor da disciplina e convidados, a saber: o diretor do Polo de Inovação e o Professor Orientador da jornada *hands on* do curso. Ao final da disciplina, não houve evasão de nenhum aluno e todos foram aprovados, com eficiência de 100% na disciplina.

Figura 2.7 – Momentos da formação usando PBL/DT na prática.



Fonte: Os autores, 2023.

Figura 2.8 – Apresentação do desenvolvimento da prática e resultados obtidos.



Fonte: Os autores, 2023.

REFERÊNCIAS

- [1] ALMULLA, M. A., “The Effectiveness of the Project-Based Learning (PBL) Approach as a Way to Engage Students in Learning”, SAGE Open, vol. 10. N. 3, July 2020.

- [2] BARBOSA, E.F; MOURA, D.G. Metodologias ativas de aprendizagem na Educação Profissional e Tecnológica. Tec. Senac, Rio de Janeiro, v. 39, n.2, p.48-67, maio/ago. 2013. <Disponível em: <http://www.bts.senac.br/index.php/bts/article/view/349>>. Acesso em: 02 mai. 2019.
- [3] BARREIRA, C. Concepções e práticas de avaliação formativa e sua relação com os processos de ensino e aprendizagem. In book: Avaliar para aprender no Brasil e em Portugal: perspectivas teóricas, práticas e de desenvolvimento (pp.191-218). Editora CRV Ltda. DOI: 10.248224/978854443463.5. Setembro, 2019.
- [4] FORUM Econômico Mundial. The future of job reports 2023. <Disponível em: <https://www.weforum.org/publications/the-future-of-jobs-report-2023/>>. Acesso em 03 nov. 2023.
- [5] FRIGOTTO, G. Educação omnilateral. In: CALDART, R. S. et al. Dicionário da Educação do Campo. Rio de Janeiro, São Paulo: Expressão Popular, 2012.
- [6] GARDNER, H. "Estruturas da mente – A Teoria das Inteligências Múltiplas". 1995
- [7] HECKLAU, F., ORTH, R., KIDSCHUN, F., KOHL, H. Human Resources Management: Meta-Study - Analysis of Future Competences in Industry 4.0., 2017.
- [8] MORAN, J. (org.). Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática. Porto Alegre: Penso, 2018.
- [9] PANKE, Stefanie. "Design Thinking in Education: Perspectives, Opportunities and Challenges" Open Education Studies, vol. 1, no. 1, 2019, pp. 281-306. <https://doi.org/10.1515/edu-2019-0022>.
- [10] PIAGET, Jean. Psicologia e pedagogia. Tradução de Dirceu Accioly Lindoso e Rosa Maria Ribeiro da Silva. São Paulo e Rio de Janeiro: Editora Forense, 1970.
- [11] QUEIROZ-NETO, J. P., SALES, D. C., PINHEIRO, H. S., NETO, B. O., "Using modern pedagogical tools to improve learning in technological contents", 2015 IEEE Frontiers in Education Conference (FIE), El Paso, TX, 2015, pp. 1-8, doi: 10.1109/FIE.2015.7344383.
- [12] QUEIROZ-NETO, J. P.; FARIAS, M. S. F.; CHAGAS, E. L. T., "Project Based Learning e Design Thinking em um projeto de intercâmbio". Revista Ibero-Americana de Estudos em Educação, Araraquara, v. 16, n. 3, p. 1791-1806, jul./set. 2021. e-ISSN: 1982-5587. DOI: <https://doi.org/10.21723/riaee.v16i3.14557>
- [13] QUEIROZ-NETO, J. P.; CHAGAS, E. L. T.; RAMOS, C. . LAPASSION MANAUS UMA EXPERIÊNCIA INTERNACIONAL DE FORMAÇÃO PROFISSIONAL DE ALTO NÍVEL. 1. ed. Curitiba: Appris, 2021. v. 1. 117p .
- [14] SANTOS, J. M.; BREMGARTNER, V.; QUEIROZ-NETO, J. P.; LIMA, H.; PEREIRA, M. "ROBÔ-TI: Educational Robotics and Project-Based Learning Stimulating High School Students in the Information Technology Area," 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028401.
- [15] TORRE-NECHES, B. De La, RUBIA-AVI, M., M., APARICIO-HERGUEDAS, J. L. "Project-based learning: an analysis of cooperation and evaluation as the axes of its dynamic". Humanit Soc Sci Commun 7, 167 (2020). <https://doi.org/10.1057/s41599-020-00663-z>.

03 Introdução à robótica industrial

Ricardo Augusto Medeiros de Oliveira¹, Benjamim Batista de Oliveira Neto²

3.1. CARACTERIZAÇÃO DA DISCIPLINA

O curso de Robótica Industrial tem como objetivo geral promover e desenvolver estratégias de ensino e aprendizado, por meio da capacitação na área de robótica industrial, com enfoque em tecnologias emergentes conectadas às necessidades e potencialidades do Polo Industrial de Manaus.

Já a disciplina de introdução à Robótica Industrial objetiva proporcionar aos estudantes uma compreensão generalista dos conceitos fundamentais e aplicações dos sistemas robóticos industriais; apresentar princípios básicos da robótica, bem como aos diversos tipos de robôs utilizados na indústria; fornece uma visão geral das aplicações práticas dos robôs industriais em diferentes setores; identificar as partes que compõe os robôs industriais e realizar operações com os robôs industriais de manuseio e movimentação.

A disciplina usou como atividade de avaliação prática a criação de um programa em cada um dos robôs estudados. (Epson, Yaskawa e Us-Robotics); Apresentação de atividades práticas Hands On em grupo onde forma avaliado a participação, o interesse e a demonstração de conhecimentos.

Durante o decorrer da disciplina, foram trabalhados os seguintes conteúdos

- Introdução a robótica industrial: Definição; Breve Histórico; Aplicações e Benefícios. Aplicações Robóticas.
- Conceitos e Tecnologias da Robótica Industrial: Quais são as partes de um robô; O manipulador mecânico; Punho do Robô; O efetuador final; Tipos de Acionamentos; Transmissões Mecânicas; Tipos de Algoritmos para Movimentação.
- Dinâmica do Braço Robótico: Precisão de movimentos; Velocidade de movimento; Estabilidade; Repetibilidade; Complacência.
- Servoacionamento Elétrico nos Robôs: Visão geral; Os sensores de movimento; Servomotores e seus drivers; Servomotor de corrente contínua; Drivers para servomotor DC; Motor de Passo; Drivers para motores de passo.
- Programação Básica de Acionamentos, Movimentos e Rotinas nos Robôs Industriais.

¹ Engenheiro Mecânico. Professor da Disciplina de Introdução à Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

² Mestre em Ciência e Engenharia de Materiais. Professor da Disciplina de Introdução à Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM.

3.2. ETAPA 1: INTRODUÇÃO A ROBÓTICA INDUSTRIAL: DEFINIÇÃO; BREVE HISTÓRICO; APLICAÇÕES E BENEFÍCIOS. APLICAÇÕES ROBÓTICAS

Esta etapa tem por objetivo apresentar as origens dos robôs industriais tais como os conhecemos na atualidade. Serão apresentadas algumas definições que esclarecem exatamente o que é um robô industrial, bem como uma visão geral sobre o campo de aplicações e benefícios em relação ao seu uso.

Apresenta ainda um histórico dos principais desenvolvimentos tecnológicos que marcaram os avanços da robótica, bem como o atual estado da técnica e as áreas de competência necessárias ao projeto e à fabricação de robôs.

Robô? Qual o significado dessa palavra?

Segundo o escritor Karel Capek (1890-1938): República Tcheca – Peças teatro: Rossum’s Universal Robots, a sociedade buscava cada vez mais a materialização do conceito do que o significado da palavra robota (do idioma tcheco), naquele espetáculo de um “trabalhador escravo”, feito por criaturas mecânicas de forma semelhante à humana.

Já o engenheiro elétrico Roy J. Wensley apresentou um primeiro mecanismo com aparência humana na Exposição Mundial de Nova York, em 1927 – Robô Televox (um pequeno robô de aspecto humano capaz de cumprir as ações ordenadas por seu operador).

Breve Histórico:

Anos 40: No final da década de 40 e início da de 50 são iniciadas as primeiras pesquisas com robôs reais.

1947: Impulsionados pelo programa da bomba atômica, o Laboratório Nacional em Argonne desenvolve o primeiro manipulador robótico tele-operado: – Manipulação de material radioativo

1968 – Shakey

O Stanford Research Institute demonstra o primeiro robô inteligente.

O robô é capaz de interpretar visualmente o meio onde está inserido, localizar objetos, navegar entre os objetos e raciocinar sobre suas ações (Santos e Júnior, 2014).

Foi nomeado “Shakey” por causa de seus movimentos erráticos e “Balançantes”.

1968 A General Electric constrói o “Four Legged Walking Truck” para o exército americano: – Os movimentos são comandados por um operador dentro do robô, mecanicamente.

Cavaleiro mecânico de Leonardo

Robô de Leonardo (ou cavaleiro mecânico de Leonardo) foi um humanóide desenhado e possivelmente construído por Leonardo da Vinci por volta do ano 1495.

Todo o sistema robótico era operado por uma série de roldanas e cabos.

Desde a descoberta do caderno, o robô foi construído fielmente com base no design de Leonardo e descobriu-se que é totalmente funcional.

Para Mataric (2014) o robô é descrito como sendo coberto por uma armadura medieval alemã-italiana e é capaz de fazer vários movimentos parecidos com humanos.

Década de 1990.

Para Carrara (2014) o “baixo custo” e o aumento da capacidade dos microcomputadores possibilita a introdução dos robôs em novas atividades comerciais.

As aplicações de robôs se expandem para a indústria de serviços.

Início de pesquisa séria em robôs humanóides.

Início do uso doméstico. 1993

Honda começa a criar protótipos de robôs humanóides.

Modelo P1: – 195 cm – 175 kg.

Modelos P2 (1996) e P3 (1997) são protótipos para o Robô Asimo

Robótica Educacional: os robôs educativos mais populares

Existem atualmente numerosos robôs educativos para crianças e jovens. Entre os mais populares estão:

Makeblock mBot: este é um robô com rodas projetado para introduzir as crianças à robótica, à programação e à eletrônica. É fácil de montar e é facilmente controlado graças ao software baseado em Scratch, projetado para crianças. Sua compatibilidade com a plataforma Makeblock e suas peças eletrônicas baseadas no ecossistema de código aberto Arduino permitem aos usuários com mais conhecimento criar robôs mais complexos.

Robo Wunderkind: consiste em um conjunto de blocos que as crianças podem conectar como quiserem para construir seu próprio robô. Cada bloco tem uma função identificada com uma cor (câmera, microfone, sensores de movimento...) e, depois de construir seu robô, as crianças podem programá-lo através de um aplicativo para reagir a certos ruídos, evitar obstáculos ou tocar música quando alguém se aproxima, entre outras funções.

OWI 535: é um braço robótico recomendado para jovens a partir dos 13 anos de idade. Ele pode levantar objetos de até 100 gramas e tem uma grande variedade de movimentos que podem ser personalizados graças à programação. Este robô também é recomendado para cursos de treinamento vocacional.

LEGO Mindstorms Robot Inventor: o herdeiro do Lego Mindstorms EV3 Robot apareceu no final de 2020 e conta com 4 motores angulares, vários sensores e 900 peças que permitem a criação de pelo menos 5 tipos diferentes de robôs. Para controlá-lo, é

necessário um tablet ou um computador para realizar sua programação. É recomendado para crianças com mais de 10 anos de idade

NAO: é um dos robôs educacionais mais conhecidos do mundo. Trata-se de robô humanoide de 58 centímetros de altura que está em constante evolução. Além de duas câmeras e quatro microfones, possui uma infinidade de sensores que lhe permitem interagir com o ambiente de uma maneira semelhante à humana. O NAO é capaz de observar, ouvir, manter diálogos e ensinar qualquer assunto. Graças a suas habilidades e diferentes níveis de programação, pode participar do processo de aprendizagem de estudantes de 5 anos de idade até a universidade,

LEGO Mindstorms Robot Inventor: o herdeiro do Lego Mindstorms EV3 Robot apareceu no final de 2020 e conta com 4 motores angulares, vários sensores e 900 peças que permitem a criação de pelo menos 5 tipos diferentes de robôs. Para controlá-lo, é necessário um tablet ou um computador para realizar sua programação. É recomendado para crianças com mais de 10 anos de idade.

Robótica Educacional

O interesse pela mobilidade de robôs teve início com hobistas das áreas de elétrica/eletrônica. A partir do final da década de 70 e início dos anos 80, começaram a aparecer projetos sérios para aplicações como patrulhamento de ambientes, transporte de carga, segurança, exploração, entre outros, quase sempre patrocinados por órgãos militares, governamentais ou grandes empresas (Barbosa e Blikstein, 2020).

A evolução nesta área é surpreendente. O caso mais famoso de projeto bem sucedido é o robô Sojourner. Construído pelo Jet Propulsion Laboratory (JPL) do Instituto de Tecnologia da Califórnia em parceria com a NASA e enviado a Marte1 com o objetivo de explorar as características do terreno deste planeta e realizar experimentos científicos

AGV executa o mesmo trajeto de entrega ao longo de sua vida útil e para navegar precisa ser guiado por fios, tiras magnéticas ou sensores, que normalmente requerem atualizações das instalações.

Já, o AMR navega através de mapas “carregados” por seu software e também consegue se adaptar conforme a necessidade da instalação. Esta capacidade pode ser comparada a um carro com um GPS, por exemplo.

Dessa forma, ele proporciona uma navegação inteligente, que permite detectar seus arredores e escolher a rota mais eficiente até o seu destino.

O AMR funciona de forma completamente autônoma e consegue fazer manobras com segurança, caso obstáculos, pessoas ou empilhadeiras surgirem em sua frente.

Quais as leis da robótica?

O escritor russo-americano **Isaac Asimov** ditou as 3 (três) leis da robótica, as quais servem, inclusive, para os atuais robôs industriais. A figura 3.1 explica cada umas das leis elaboradas pelo escritor.

Figura 3.1: Leis da Robótica



Fonte: Adaptado de Mataric, 2014.

Robô Industrial

Tais máquinas em quase nada se assemelham à aparência humana. Ao contrário, são formadas por articulações mecânicas capazes de realizar os movimentos da tarefa a que se destinam, não importando, para isso, a aparência que venham a ter.

Definições:

Um robô industrial é um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programadas para a realização de uma variedade de tarefas (Robot Institute of America – RIA)

A **ISO 10218** define um **robô industrial** como uma máquina para manipulação, com vários graus de liberdade, controlada automaticamente, reprogramável, multifuncional, que pode ter base fixa ou móvel para utilização em aplicações de automação industrial. (*Organização Internacional de Normalização*).

Projeto e Engenharia

O atual estado da arte de robôs industriais implica preponderantemente tarefas de controle de posicionamento e movimentação, que é ainda o principal objetivo desses mecanismos.

Diversas capacidades de controle lógico de entrada e saída têm sido acrescentadas aos recentes robôs, assim como a possibilidade de comunicação.

Aplicação e benefícios

- Movimentação
- Medição
- Manipulação

3.3. ETAPA 2: CONCEITOS E TECNOLOGIAS DA ROBÓTICA INDUSTRIAL

O objetivo desta aula é descrever os principais fundamentos tecnológicos que compõem os sistemas robóticos atualmente empregados nas indústrias. O robô será descrito por cada uma de suas partes constituintes. Enfatizaremos o braço mecânico manipulador e seu efetuator final, assim como os principais tipos de garras.

Trataremos ainda das questões que envolvem o sistema controle de movimento do robô: tipos de tecnologias de acionamentos e sistemas de controle.

O robô está dividido em 6 (partes) como mostra a figura 3.2.

Figura 3.2: Partes de um robô



Fonte: Os autores, 2023.

Manipulador Mecânico

Elos

São as partes rígidas do manipulador.

Juntas

São as partes do manipulador que realizam uma conexão móvel entre dois elos.

Efetuator Final

Localizado ao final da cadeia cinemática que compõe o manipulador é capaz de aplicar amplo grau de liberdade em termos de orientação espacial.

Juntas

Segundo Carraca (2014) as juntas de um robô são elementos de particular interesse na análise de um manipulador robótico. Afinal, são elas que proporcionam liberdade de movimento. Dentre várias formas mecânicas construtivas possíveis de serem encontradas para juntas, há nos robôs industriais três tipos mais comuns.

Juntas Prismáticas

Também chamadas de juntas lineares ou deslizantes, elas se movem em linha reta sem girar.

São compostas de duas hastes que deslizam entre si de forma telescópica.

Elas se estendem, retraem-se ou se movem para dentro e para fora, como um elevador tipo macaco hidráulico, ou ainda ao longo de um trilho, como o cabeçote de uma impressora.

Juntas Rotacionais

Também chamadas juntas de revolução, giram em torno de uma linha imaginária estacionária, chamada de eixo de rotação.

Elas giram como em uma cadeira giratória, ou abrem e fecham como em uma dobradiça.

Juntas Esféricas

Esta conexão funciona como a combinação de três juntas de revolução, realizando a rotação em torno de três eixos.

No corpo humano existem algumas juntas esféricas, como ombro, punho e quadril.

Grau de Liberdade

O termo pode ser abreviado apenas para GDL ou, como frequentemente se encontra em catálogos de robôs, pela sigla equivalente em inglês, DOF (Degrees of Freedom).

Tipos de Manipuladores

Os robôs são normalmente classificados conforme o tipo de juntas, ou, mais exatamente, pela configuração das juntas que formam seu sistema de articulação. Esta divisão em classes possibilita categorizar certas características comuns, tais como:

Robô Cartesiano

Neste tipo de estrutura, os três elos deslocam-se de forma translacional, aqui denominada TTT.

Devido à ortogonalidade entre os elos mecânicos obtém-se boa rigidez mecânica do conjunto, porém há baixa habilidade de posicionamento devido à limitada orientação angular com que efetuam as tarefas

Robô Cilíndrico

A estrutura cilíndrica difere da estrutura cartesiana pelo fato de que a primeira junta mecânica provê movimento de rotação, em vez de translação, e por isso esta configuração é denominada RTT.

Esta estrutura ainda apresenta características de rigidez mecânica satisfatória, apesar de ter menor resolução de posicionamento na direção horizontal

Robô Esférico

A estrutura esférica é composta por dois elos mecânicos com movimento rotacional, seguidos por um terceiro, com movimento translacional, e por isso é denominada RRT.

Em função desta configuração, os movimentos apresentam menor resolução de posicionamento ao longo do eixo radial do robô, além de comportarem-se com menor grau de rigidez mecânica.

Robô SCARA

SCARA - Selective Compliance Assembly Robot Arm.

Nesta configuração, apresentam-se dois elos mecânicos com movimento rotacional, seguidos por um terceiro, com movimento translacional, e também é denominada RRT, como no caso do robô esférico.

Esta estrutura apresenta alta rigidez dinâmica para movimentos verticais. Ao mesmo tempo, para movimentos horizontais, a rigidez é reduzida.

Robô Articulado

Neste tipo de estrutura, os três elos mecânicos de posicionamento deslocam-se de forma rotativa - RRR.

Tal estrutura, apesar de ter menor rigidez mecânica do que as anteriores, assim como resolução de posicionamento variável no espaço de trabalho, é a que apresenta maior grau de habilidade para posicionamento e orientação espacial.

Robôs Colaborativos (Cobots)

Segundo a Associação Brasileira de Normas Técnicas (2018) um cobot ou robô colaborativo é um robô criado para interagir com humanos em ambientes de trabalho. O

termo foi usado pela primeira vez em 1999 e resulta da junção das palavras "colaboração" e "robot". Quais são as suas principais características?

Colaboração: Os cobots foram criados para ajudar as pessoas, portanto a interação com os operários é fundamental.

Exige menos adequações físicas da área de trabalho para adequação as normas de segurança.

Normas complementares foram desenvolvidas para lidar especificamente com a robótica colaborativa. Como é o caso da TS 15066 e ISO 10218, que no Brasil foram incorporadas como ABNT NBR ISO 10218-1, ABNT NBR ISO 10218-2 e ISO/TS 15066.

A figura 3.3 mostra alguns alunos entendendo sobre os manipuladores e suas especificidades.

Figura 3.3: Alunos em aprendizagem dos manipuladores.



Fonte: os autores, 2023.

Punho do Robô

Punho é o nome normalmente dado às três últimas juntas do braço do robô, o número de graus de liberdade para orientação depende do punho que em geral, é formado por juntas rotacionais (Santos e Júnior, 2015).

Idealmente seus eixos de rotação são mutuamente perpendiculares, configurando o que se conhece como punho esférico, ou seja, punhos cujos eixos das juntas se cruzam em um único ponto.

Uma das vantagens do punho esférico é que seu uso simplifica consideravelmente a cinemática do manipulador, permitindo o desacoplamento entre a posição e a orientação do efetuador.

O braço e o punho são utilizados basicamente para o posicionamento e a orientação do efetuador final, seja na forma de garra ou de qualquer ferramenta que ele possa carregar.

Efetuator Final

O último componente na sequência cinemática de um robô industrial é o seu efetuator, localizado imediatamente após o punho. Os efetutores finais podem ser divididos em dois principais grupos: Garras e ferramentas.

Tipos de Acionamentos

Como você deve se recordar da Física, para alterar o movimento de um corpo é necessário aplicar uma força externa a ele. Tecnicamente falando, para gerar força é necessário haver algum dispositivo motor. Em robótica industrial há basicamente três tecnologias principais para isto, ou seja, três tipos de fontes de força motriz para acionamentos, vide figura 3.4.

Figura 3.4: Tipos de Acionamentos



Fonte: os autores, 2023.

Acionamento hidráulico:

Vantagens

- Apresenta maior disponibilidade de força/torque.
- Mantém esta força/torque constante para uma grande faixa de variação de velocidade.
- Possui média precisão de posicionamento (menor que o elétrico e maior que o pneumático).
- Pode manter uma elevada força (ou torque) por um longo período de tempo, mesmo quando parado

Desvantagens

- Requer uma fonte de energia mais cara.
- Depende de constante manutenção.
- Utiliza válvulas de precisão que ainda apresentam custo elevado.
- Está sujeito a vazamento de óleo e, portanto, é potencialmente poluente.

3.4. ETAPA 3: DINÂMICA DO BRAÇO ROBÓTICO

O objetivo desta etapa é proporcionar uma compreensão abrangente dos princípios fundamentais que regem o movimento e o controle de braços robóticos. Ao final da aula, os alunos devem estar preparados para compreender os princípios fundamentais da dinâmica de braços robóticos, bem como sua aplicação em situações do mundo real, e devem ser capazes de analisar e resolver problemas relacionados à dinâmica e controle de robôs.

Para Carraca (2014) a dinâmica do robô refere-se como ele realiza sua Locomoção, ou seja, à maneira como um corpo (no nosso caso, um robô) se desloca de um lugar para outro. O termo vem do latim *locus*, que significa “lugar”, e *movere*, que significa “mover”. Neste contexto, **o desempenho dinâmico** do braço robótico está associado à alguns fatores, tais como:

- Precisão
- Velocidade de resposta
- Estabilidade

Precisão do Movimento

Relaciona-se com a velocidade e estabilidade, pois é uma medida de erro na posição do órgão terminal.

A precisão de movimento está intrinsecamente correlacionada com três características dos braços:

Resolução Espacial

A resolução espacial é influenciada pelas características do sistema de controle e pelas imprecisões. O sistema de controle é encarregado de gerenciar todos os incrementos individuais das articulações. As imprecisões são determinadas pela qualidade dos componentes que compõem as conexões entre as articulações, tais como folgas nas engrenagens, tensões nas polias e histereses mecânicas e magnéticas, entre outros fatores.

Precisão

Refere-se à habilidade de um braço em posicionar seu pulso em um ponto específico dentro do volume de trabalho. Está intrinsecamente ligada à resolução espacial, já que a precisão depende dos incrementos que as articulações podem realizar para movimentar-se e alcançar um ponto determinado. A precisão é impactada por vários fatores, e é crucial definir as condições nas quais ela será avaliada: no final do punho; na pior condição (geralmente com o braço totalmente estendido); no modo ponto-a-ponto de controle.

Repetibilidade

A repetibilidade está relacionada com a capacidade do braço robótico de posicionar repetidamente seu pulso num ponto determinado.

Estes movimentos podem sofrer influências de folgas mecânicas, da flexibilidade e das limitações do sistema de controle.

Erro de Posicionamento

Devido a várias razões, o braço não para precisamente no ponto indicado, apresentando um erro. O ponto-alvo é, de fato, o ponto desejado; na pior situação, o ponto-alvo está localizado exatamente no meio de dois pontos endereçáveis.

Precisão do Movimento

A precisão de um braço também depende da sua geometria.

Robôs pórticos, ou cartesianos, possuem uma grade de endereçamento perfeitamente retangular, os pontos endereçáveis formam um padrão cúbico.

Nos braços que utilizam juntas rotativas a grade de pontos endereçáveis não é regular.

Estabilidade

A estabilidade está relacionada com a maneira pela qual o braço alcança sua posição final com velocidade zero. Se a estabilidade for baixa, é possível aplicar elementos de amortecimento no braço para melhorar a estabilidade, mas isso afeta a velocidade de resposta. A maioria dos robôs precisa ser estável, o que significa que, ao realizar suas tarefas, não devem balançar, inclinar-se ou cair facilmente.

Existem diferentes formas de alcançar a estabilidade, e em particular, destacam-se dois tipos: estática e dinâmica. Um robô estaticamente estável pode permanecer parado sem cair, sendo, portanto, estático e estável. Isso é um recurso útil, mas requer que o corpo do robô tenha pernas ou rodas suficientes para proporcionar pontos de apoio estáticos capazes de mantê-lo em equilíbrio.

Na estabilidade dinâmica, o corpo precisa equilibrar-se ativamente ou mover-se para permanecer estável, sendo assim chamado dinamicamente estável. Por exemplo, robôs que saltam com uma perna são dinamicamente estáveis: podem pular sem se deslocar ou saltar para diferentes destinos sem cair.

Velocidade

Refere-se à agilidade do braço robótico ao deslocar-se de um ponto a outro em um curto período de tempo. Durante o projeto do manipulador, é crucial analisar tanto o torque presente em cada junta do braço quanto a aceleração de cada elo.

A velocidade das juntas exerce uma forte influência na duração do ciclo de trabalho e na quantidade produzida. Por outro lado, características como precisão no posicionamento do braço, carga (peso) do objeto manipulado e alcance do braço impactam diretamente na velocidade.

Não há almoço grátis: para um mesmo tipo de acionamento, quanto maior a velocidade, menor será a precisão e a capacidade de manipular cargas pesadas.

Complacência

Complacência é a capacidade do braço robótico de resistir ao movimento causado por uma força ou torque na extremidade do punho (Santos e Júnior, 2014).

A complacência é direcional e pode ser diferente em direções distintas.

Um exemplo disto é o robô SCARA, que possui complacência seletiva nas juntas, isto é, a rigidez apresentada pelas juntas é maior na direção que suporta a carga, o que o torna mais preciso no posicionamento.

Em contraste, um braço com alta complacência pode deformar-se ao suportar uma carga, comprometendo assim a precisão.

Cinemática de manipuladores

Sistema de Coordenadas

Qualquer ponto no espaço pode ser visto de diferentes formas consoante o referencial usado; trata-se sempre do mesmo ponto fisicamente, mas é a sua descrição que se adequa com o ponto vista (referencial)

Transformação geométrica de translação

Movimentar um ponto no espaço significa alterar as coordenadas, isto é, passar de uma posição q para uma posição q visto do mesmo referencial; este processo designa-se por translação no espaço

Transformação geométrica de rotação

Mudança na posição de um ponto a partir de um movimento de rotação ao redor de um eixo.

A figura 3.5 mostra alguns momentos de atividades práticas para compreender os conteúdos abordados e esclarecimentos.

Figura 3.5: Aula prática de Cinemática dos Manipuladores



Fonte: os autores, 2023.

Cinemática Direta

A cinemática de um manipulador: é o estudo do conjunto de relações entre as posições, velocidades e acelerações dos seus elos (Carrara, 2014).

A relação entre o referencial de origem e o referencial da extremidade (mão) do manipulador é dada por uma transformação T.

Para fazer um estudo da cinemática, e outros, é necessário, antes de mais, definir sistemas de coordenadas associados a cada elo, ou seja, atribuir-lhe referenciais

A cinemática direta: consiste em achar a posição do manipulador a partir da posição das juntas do braço robótico.

A cinemática inversa: consiste em achar a das juntas do braço robótico a partir do manipulador.

Singularidade

É uma situação (configuração do robô) na qual seria necessário impor velocidades infinitamente altas numa ou mais juntas para manter determinadas velocidades no espaço operacional.

Haverá um atuador (ou mais) cujo controle será sempre insuficiente para cumprir as velocidades desejadas no espaço operacional.

Numa situação de singularidade há a perda de um ou mais graus de liberdade.

3.5. ETAPA 4: SERVOACIONAMENTO ELÉTRICO NOS ROBÔS:

O objetivo desta etapa é apresentar os elementos que compõem um sistema elétrico de servoacionamentos empregados nos robôs. O tema principal da aula será o controle de movimento, tratando mais especificamente dos servossistemas de posicionamento. Será analisado aspectos construtivos, métodos de operação e drivers para acionamento.

Para Silva e Blikstein (2014) o servossistema é um conceito destinado à realização de controle de qualquer grandeza física, e pode ser classificado como sendo de malha aberta ou de malha fechada.

Um sistema de malha aberta é aquele em que a saída do sistema não é diretamente controlada com base no feedback. Em vez disso, a entrada é definida para um valor desejado, e o sistema tenta produzir a saída desejada sem monitorar ou corrigir erros.

Um sistema de malha fechada é aquele em que a saída do sistema é continuamente monitorada e comparada com uma referência (valor desejado). Qualquer diferença (erro) entre a saída real e desejada é usada para ajustar a entrada e, assim, corrigir o sistema.

Em se tratando de robôs, as grandezas físicas de interesse são movimentos, e por isso é importante que o profissional da área conheça os componentes e os princípios tecnológicos que regem os servossistemas de acionamentos.

Atualmente, há uma tendência tecnológica cada vez maior ao uso de motores do tipo elétrico.

Para sistemas de posicionamento, o motor elétrico usado em um robô deve ser: motor de corrente contínua, motor de passo, ou motor síncrono.

Como um robô possui diversos elos móveis, o sistema completo de controle de movimento nada mais é do que o conjunto de diversos subsistemas de servoacionamentos.

Sensores de Movimento

Qualquer tipo de servossistema opera com base no conceito de realimentação negativa.

No caso dos robôs industriais, o interesse fundamental é o de controlar a posição do braço.

Em princípio, existe uma grande variedade de dispositivos disponíveis para detectar posição angular ou linear.

Potenciômetros

Os potenciômetros são dispositivos analógicos cuja tensão de saída é diretamente proporcional à posição do cursor. Ao contrário dos potenciômetros convencionais utilizados em circuitos eletrônicos, muitas vezes esses dispositivos não têm restrições no movimento circular, permitindo assim realizar múltiplas voltas.

Resolvers

O resolver é outra categoria de dispositivo analógico, cuja saída varia proporcionalmente ao ângulo de um elemento rotativo em relação a um elemento fixo. Na sua configuração mais básica, um resolver possui um único enrolamento em seu rotor e um par de enrolamentos no estator.

Encoder

O encoder é um dispositivo que registra e gera movimentos rotacionais. Esses dispositivos são empregados para medir tanto movimentos rotativos quanto deslocamentos lineares, convertendo-os em impulsos elétricos, sendo assim classificados como geradores de pulsos. Existem dois tipos principais de encoders: os incrementais e os absolutos.

Encoder incremental

O *encoder* incremental fornece o número de pulsos a partir de cada rotação inteira do eixo. Ou seja, a medida da distância angular ou linear movimentada é representada por essa quantidade. Cada pulso surge por meio do trabalho de um emissor luminoso e um receptor.

Equipamento que possui este tipo de encoder necessita referenciamento toda vez que que energizado.

Encoders absolutos

Os *encoders* absolutos fornecem informações sobre a posição real, permitindo identificar a posição exata mesmo após uma interrupção de energia, sem a necessidade de reinicialização. Essa característica é especialmente útil em aplicações onde a precisão e a continuidade são essenciais, como sistemas de controle, robótica e máquinas CNC.

Tacômetro

Um dos dispositivos mais comumente usados para a realimentação de informações de velocidade é o *tacômetro* de corrente contínua. Um tacômetro é essencialmente um gerador de corrente contínua, que fornece uma tensão de saída proporcional à velocidade angular do induzido

Servoacionamentos e seus Drivers

Os movimentos executados pelo motor são definidos pelo componente chamado controle, que estabelece parâmetros como torque, velocidade, distância, direção ou aceleração. Independentemente do tipo de motor utilizado (passo, DC ou síncrono), a função de controle é sempre existirá.

Servomotor de Corrente Contínua

Os motores DC foram a primeira alternativa adotada pelos projetistas de sistemas robóticos em substituição aos acionamentos hidráulicos.

Diferentes tipos construtivos de motores de corrente contínua são atualmente conhecidos, e os mais comuns são aqueles denominados núcleo de ferro (Iron Cored), Bobina Móvel (Moving Coil), e Sem Escovas (Brushless).

Como esses tipos de motores não apresentam núcleo de ferro girante imerso em campo magnético, também não apresentam um fenômeno denominado perdas no ferro. Consequentemente, maiores velocidades podem ser obtidas com baixa potência de entrada.

O maior fator limitante no desempenho dos motores de núcleo de ferro ou bobina móvel é o excessivo calor gerado no rotor devido à corrente que circula nos enrolamentos.

Atualmente, esta categoria construtiva de motor DC é a mais difundida para uso em sistemas robóticos. Para se obter maiores torques com menores inércias, os fabricantes normalmente utilizam ímãs permanentes feitos com um material chamado terras-raras.

Por fim, indiferente ao tipo construtivo, outro aspecto importante nos servomotores DC diz respeito à sua especificação.

Drivers de para servomotor DC

O driver, também conhecido como amplificador de potência, em um sistema de servoacionamento elétrico, é o componente responsável por fornecer ao servomotor a energia elétrica necessária (tensão, corrente) para que ele se mova de acordo com os sinais de comando recebidos na sua interface de entrada.

Em geral, os servoacionamentos são do tipo reversível, ou, em uma expressão tecnicamente mais precisa: para funcionamento em quatro quadrantes. Isso significa que o driver pode fornecer energia tanto para acionar o motor (em ambos os sentidos de giro) como para receber energia durante a frenagem do mesmo (igualmente em ambas as direções de movimento).

Três métodos básicos são utilizados para controlar a tensão em amplificadores chaveados: modulação por largura de pulso (PWM – pulse width modulation), modulação da frequência de pulso (PFM – pulse frequency modulation), utilizando transistores, e conformação de onda (WS – wave shaping), utilizando tiristores (SCR – silicon controlled rectifier).

Para entender cada um desses métodos, é importante ter em mente que um motor DC responde mecanicamente da seguinte maneira: sua velocidade está diretamente relacionada à tensão elétrica a ele aplicada, enquanto o torque (ou “força” no eixo) está diretamente relacionado à corrente elétrica consumida pelo motor.

O motor de passo

Para Carrara (2014) esses motores transformam pulsos elétricos em movimentos mecânicos de incremento fixo.

Informa ainda que devido à natureza digital, são eficientes para operar em malha aberta, ou seja, não é necessário utilizar elemento sensor no eixo de saída.

Segundo Santos e Júnior (2014) os motores de passo apresentam boa estabilidade quando parados, não apresentando erros acumulativos.

Geralmente têm baixa necessidade de manutenção e suportam sobrecargas mecânicas sem causar prejuízos ao motor.

Talvez a principal vantagem advinda da aplicação com motores de passo seja o menor custo, pois se uma aplicação pode ser realizada com motor de passo, então esta deve ser a escolha mais adequada.

Os motores de passo também apresentam estabilidade em paradas, pois sem a comutação na corrente de fase o rotor permanece completamente parado, e o rotor não tende a vibrar em torno de uma posição do encoder ou do resolver (como pode ocorrer nos demais tipos de servomotores).

Fornecem altos torques de saída em baixas velocidades.

A perda de suavidade em baixas velocidades é o maior problema. Já para altas velocidades (~2000 rpm) começam a surgir ressonâncias mecânicas, o que os torna inadequados para operação em determinada faixa de velocidade.

O motor síncrono (*Brushless AC*)

Sem sombra de dúvidas, a tecnologia de ponta atual em termos de servoacionamento elétrico é definida pelo uso dos motores síncronos.

Popularmente, este tipo de motor é chamado de brushless (do inglês, brush quer dizer escova; portanto, brushless significa sem escovas).

Motores síncronos brushless (ou BLDC - Brushless Direct Current) são uma variante dos motores síncronos AC que não possuem escovas mecânicas. Em vez disso, eles usam ímãs permanentes e eletrônica de controle para gerar um campo magnético rotativo. Isso resulta em menos desgaste, maior eficiência e vida útil mais longa em comparação com os motores síncronos tradicionais.

Esses motores apresentam razão potência/volume superior à de servomotores de corrente contínua.

Drivers de para servomotor Brushless AC

O driver para um motor brushless utiliza a tecnologia de inversores, que são circuitos capazes de converter corrente contínua em corrente alternada, de maneira semelhante aos sistemas empregados em motores síncronos CA e motores de indução convencionais.

O princípio fundamental de operação de um inversor reside no controle da amplitude e da frequência da tensão na saída, por meio do chaveamento dos transistores que compõem o driver.

3.6. ETAPA 5: PROGRAMAÇÃO BÁSICA DE ACIONAMENTOS, MOVIMENTOS E ROTINAS NOS ROBÔS INDUSTRIAIS.

Esta etapa do treinamento deve ser desenvolvida paralelamente às atividades teóricas e consiste em fazer as atividades práticas de movimentação e programação nos robôs dos diferentes fabricantes, Yaskawa, Us-Robotics, Epson e Staubli. Durante esta etapa os devem ter oportunidade de utilizar os robôs na prática, desde as movimentações mais simples, até a elaboração de programas de movimentação e simulação básica dos movimentos necessários para cumprir atividades práticas previamente definidas. A Metodologia utilizada nesta etapa deve ser a realização de atividades de forma individual e também em equipes com a utilização dos próprios equipamentos bem como dos simuladores específicos de cada fabricante disponíveis no laboratório de robótica industrial.

REFERÊNCIAS

- [1] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO 10128-1: Robôs e dispositivos robóticos - Requisitos de segurança para robôs industriais Parte 1: Robôs**. Rio de Janeiro: ABNT, 2018.
- [2] _____. **ABNT NBR ISO 10128-1: Robôs e dispositivos robóticos - Requisitos de segurança para robôs industriais Parte 1: Sistemas robotizados e integração**. Rio de Janeiro: ABNT, 2018.
- [3] Carrara, V. **Apostila de Robótica**. Mogi das Cruzes: Universidade Braz Cubas, 2014.
- [4] Mataric, Maja J. **Introdução á robótica**. Editora Blucher, 2014.
- [5] Santos, Winderson Eugênio dos.; Júnior, José Hamilton Chaves. **Robótica Industrial - Fundamentos, tecnologias, programação e simulação**. 1ª ed. Érica, 2015.
- [6] Silva, Rodrigo B.; Blikstein, Paulo. **Robótica educacional: experiências inovadoras na educação brasileira**. Penso, 2019.

4.1. CARACTERIZAÇÃO DA DISCIPLINA

O curso de Robótica Industrial tem como objetivo geral promover e desenvolver estratégias de ensino e aprendizado, por meio da capacitação na área de robótica industrial, com enfoque em tecnologias emergentes conectadas às necessidades e potencialidades do Polo Industrial de Manaus. Já a disciplina de Programação para aplicações Robóticas objetiva:

- apresentar os princípios Funcionamento dos Módulos de programação dos robôs industriais abordados durante o curso;
- proporcionar aos estudantes uma capacitação em programação e aplicações dos sistemas industriais;
- desenvolver programas nas plataformas dos fabricantes utilizados durante o curso resolvendo possíveis problemas de programação que possam vir a existir em seus projetos

A disciplina usou como atividade de avaliação prática a criação de programas em cada um dos robôs estudados. (Epson, Yaskawa e Us-Robotics), bem como a apresentação de atividades práticas *Hands On* em grupo onde forma avaliado a participação, o interesse e a demonstração de conhecimentos. Durante o decorrer da disciplina, foram trabalhados os seguintes conteúdos:

- Partes do robô Controlador, Manipulador e interface de Programação.
- Coordenadas de Movimentação Manual.
- Coordenadas de usuário. (User Frames).
- Calibração de Ferramenta.
- Tipos de movimentos a serem programados.
- Instruções de controle (CALL,JUMP,LABEL, FOR etc.).
- Instruções de I/O entradas e saídas. Variáveis de controle.
- Identificação Binária de I/O. Instruções aritméticas.
- Funções de deslocamento. Criação de programas.
- Exemplos de aplicações.
- Programação Hand Free. Aplicações com Manipulação.

¹ Engenheiro de Automação e Elétrica Industrial. Professor da Disciplina de Programação para Práticas Robóticas.

- Integração de Sistemas de I/O entradas e saídas.
- Programação Aplicada com robôs.

4.2. ROBÔ MANIPULADOR YASKAWA

O Robô Manipulador Yaskawa, também chamado de braço robótico, é um robô industrial muito utilizado no Polo Industrial de Manaus, e apresenta soluções modernas e totalmente aderente aos padrões propostos para a indústria 4.0. Devido à sua grande utilização na região, este foi um dos escolhidos para dar as bases conceituais e práticas da programação de robôs industriais. Os outros são o Robô Manipulador Universal COBOT, O Robô Epson SCARA e o Robô Manipulador Stäubli.

4.2.1. PARTES DO ROBÔ YASKAWA

O Robô Yaskawa é formado por controlador/unidade de potência, manipulador (com os sensores internos e externos) e interface de programação. O controlador é unidade central de controle do robô, e contém os modelos de software do manipulador e ambiente de programação, assim como o algoritmo de controle do robô, que usa as informações dos sensores e do programa para comandar a unidade de potência.

O manipulador é estrutura mecânica constituída por diversos corpos rígidos (elos) ligados em cadeia, movimentados através dos atuadores que se encontram ligados aos mesmos através das respectivas transmissões. Neste caso, considerando o número de articulações e rotores, ele é um manipulador com 5 graus de liberdade. A Figura 4.1 apresenta a controlador/unidade de potência e o manipulador.

Figura 4.1 Robô Yaskawa: Controlador (esquerda) e manipulador (direita).



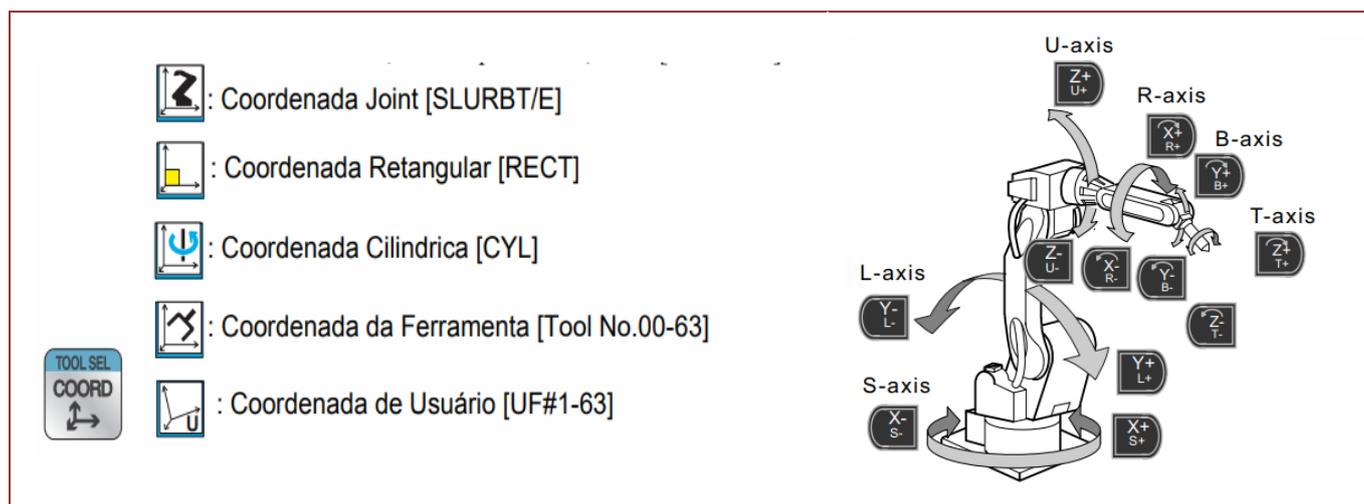
Fonte: <https://catalogo.yaskawa.com.br/>, 2023.

4.2.2. SISTEMA DE COORDENADAS

O sistema de coordenadas é aquele que determina como o robô se movimentará, quando fica ativo se encontra identificado na área de status, e atua ao pressionar as teclas

de movimentação. Quatro dos cinco tipos disponíveis estão visíveis na tecla COORD (há uma configuração disponível no menu TEACHING CONDITION para variar entre a coordenada RECT [Retangular] e CYL [Cilíndrica]). As teclas X, Y, Z, Rx, Ry e Rz ficarão ativas em todos os sistemas de coordenadas, exceto para Joint, onde [SLURBT] estarão ativas. A Figura 4.2 ilustra as teclas e as rotações em XYZ do sistema de coordenadas.

Figura 4.2 Teclas e as rotações em XYZ do sistema de coordenadas.



Fonte: <https://catalogo.yaskawa.com.br/>, 2023.

Estas coordenadas são usadas para movimentação Manual do robô não sendo ainda inseridas no programa. A Coordenada Joint Robô de 6 eixos [SLURBT], quando selecionada, as letras S, L, U, R, B e T nos 06 pares de teclas dos eixos ficam ativas, permitindo o operador mover cada motor do robô individualmente para qualquer direção. Essa é a coordenada ativa padrão quando o sistema é iniciado.

A coordenada TOOL movimenta o Ponto Central da Ferramenta (TCP), que fica “na ponta” do robô manipulador, baseando-se na *TOOL FRAME* (TF). Esse sistema de coordenada não é estacionário, ou seja, ele se movimenta junto com o robô. As teclas Z normalmente movimentam o robô para aproximação ou afastamento do ponto desejado, mas isso pode variar com a aplicação.

A coordenada de usuário (USER) movimenta o TCP baseando-se na coordenada ativa, onde os pontos e planos de origem são criados pelo usuário. A coordenada RECT movimenta o TCP em uma linha reta baseada no *ROBOT FRAME* (coordenada do robô), onde os eixos XYZ estão dispostos obedecendo a regra da mão direita, com o X movimentando o TCP para frente o para trás, o Y para a esquerda ou para a direita e o Z para cima e para baixo. A coordenada CYL movimenta o TCP em um movimento cilíndrico quando usamos as teclas XY; as teclas Z funcionam da mesma maneira.

4.2.3. INTERFACE DE PROGRAMAÇÃO.

O programa é feito através de uma de uma interface homem-máquina (IHM) chamada de programador pendente (P.P), mais conhecido pelo termo em inglês *Teach Pendant* (TP), que é o dispositivo mais comum na programação de robôs industriais, bastante utilizado para programar e controlar os movimentos dos robôs industriais e seus

braços mecânicos passo a passo. Neste, são gravados os dados de posicionamento, velocidade e cálculos com variáveis dos movimentos feitos via o TP (Figura 4.3).

Figura 4.3 Teach Pendant do Robô Yaskawa.

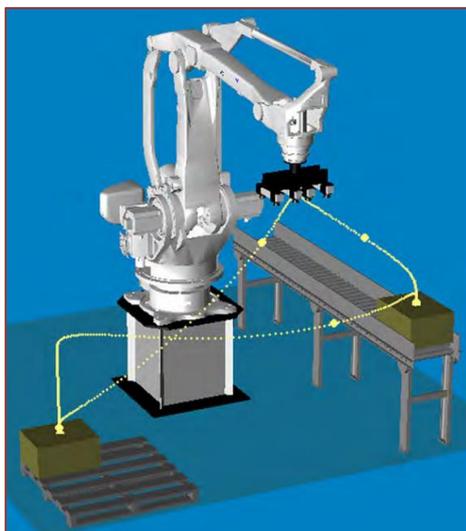


Fonte: <https://catalogo.yaskawa.com.br/>, 2023.

4.2.4. TIPOS DE MOVIMENTOS

O *Joint Motion Type* (MOVJ) é o movimento que controla apenas o destino do robô. O *Control Group* se movimenta até alcançar os valores de encoder gravados na posição. A trajetória do robô não pode ser controlada, pois o controlador não interpola sua trajetória. Por isso, aplicações que precisem de precisão e trajetória conhecida (solda, pintura, etc.) não utilizam movimentos em Joint. Essa movimentação só é utilizada para movimentos aéreos e de aproximação. A velocidade $V_j = \text{Joint_speed}$ representa a velocidade máxima de cada motor do robô durante a movimentação para a posição gravada. (Em alguns movimentos alguns motores podem não se mover, de acordo com a necessidade). A Figura 5.4 ilustra este tipo de movimento. (YASKAWA, 2014).

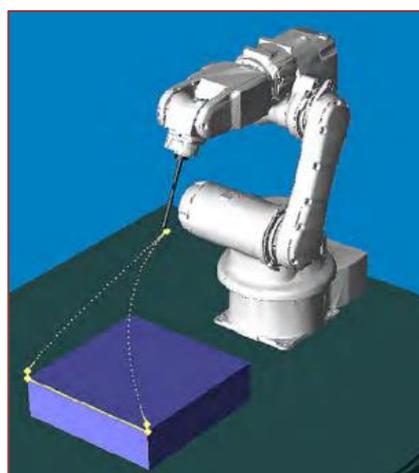
Figura 4.4 Ilustração do movimento *Joint Motion Type* (MOVJ).



Fonte: <https://catalogo.yaskawa.com.br/>, 2023.

O *Linear Motion Type* (MOVL) é um movimento linear interpolado do TCP. Esse tipo de movimentação faz com que o TCP do robô se movimente em uma linha reta até chegar à posição gravada. Se houver uma mudança de orientação da ferramenta entre a posição atual para o destino, esta irá mudar gradativamente durante a trajetória linear até o destino. A velocidade do movimento MOVL é a velocidade com que o TCP se move ($V=Control_point_speed$) ou a velocidade de rotação (ou velocidade angular) da ferramenta ($VR=Angle_speed$), quando o robô rotaciona a ferramenta com o TCP fixo. A Figura 4.5 ilustra este tipo de movimento.

Figura 4.5 Ilustração do movimento *Linear Motion Type* (MOVL).

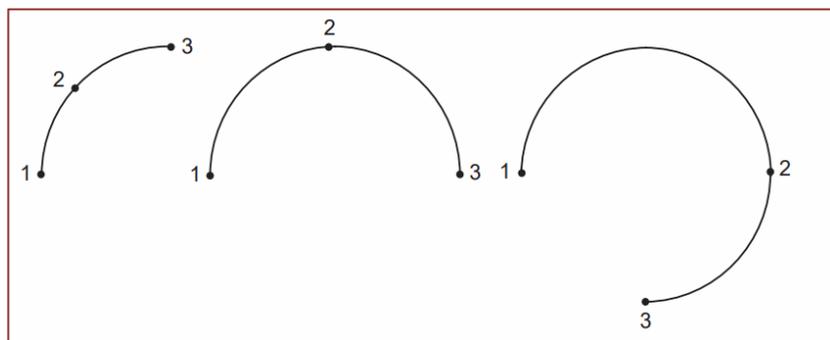


Fonte: <https://catalogo.yaskawa.com.br/>, 2023.

O *Circular Motion Type* (MOV C) é um movimento no qual o robô pode mover o seu TCP em uma trajetória circular simétrica (seguindo uma trajetória parabólica ou

descrevendo um polinômio de segundo grau). Para isso, basta gravar 3 pontos consecutivos com o movimento MOV. Os 3 pontos permitem ao controlador determinar o plano, raio e centro do círculo interpolado. Cada ponto MOV a mais que for gravado desnecessariamente fará com que o controlador recalcule todos os dados anteriormente mencionados. A Figura 4.6 ilustra este tipo de movimento.

Figura 4.6 Ilustração do movimento *Circular Motion Type* (MOVC).



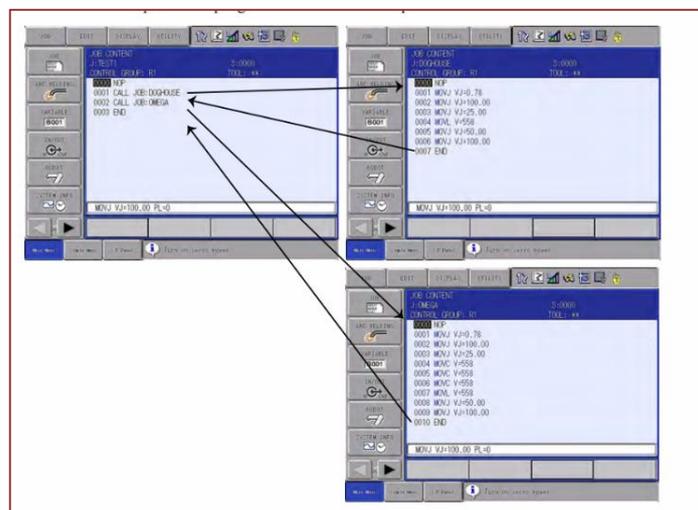
Fonte: YASKAWA, 2014.

4.2.5. INSTRUÇÕES DE CONTROLE

- CALL e RET

A instrução CALL permite que um programa chame outro como uma sub-rotina. O programa que chama o outro é chamado de “Pai”. O programa chamado é o “filho”, que pode ser finalizado com a instrução RET ou pelo fim do programa (linha END). O controlador continuará executando o programa pai com a próxima linha após a instrução CALL ser executada. Programas podem ser conectados criando um programa Pai com instruções CALL para cada programa Filho na ordem que eles devem ser executados. A Figura 4.7 demonstra o uso das instruções CALL e RET.

Figura 4.7 Exemplo do uso das instruções CALL e RET.



Fonte: YASKAWA, 2014.

- JUMP*LABEL e *LABEL

A instrução JUMP*LABEL funciona somente com a instrução *LABEL. Juntas, elas instruem o controlador a ir de uma linha com a instrução JUMP*LABEL para outra com a mesma função *LABEL. Essas instruções podem ser utilizadas para pular uma parte do programa ou criar um loop infinito. Um label automaticamente inclui um asterisco (*).

A instrução JUMP é utilizada para pular/omitir linhas ou para ignorar uma parte de um código. A instrução JUMP*LABEL precisa preceder a instrução *LABEL. No exemplo do trecho de código a seguir, o controlador pulará as linhas 6, 7 e 8, encontrando o label correspondente na linha 9, e continuará a execução na linha 10 (se a condição for verdadeira); caso contrário, o controlador executará todas as linhas.

```
0000 NOP
0001 MOVJ VJ=0.78
0002 MOVJ VJ=100.00
0003 MOVJ VJ=25.00 PL=0
0004 MOVL V=675
0005 JUMP *LABEL1 IF IN#(4) = ON
0006 MOVL V=675
0007 MOVL V=675
0008 MOVL V=675
0009 *LABEL1
0010 MOVL V=675
0011 MOVJ VJ=25.00
0012 MOVJ VJ=100.00
0013 END
```

A condição IF é uma condição que diz se a instrução de controle a ela atrelada será executada. Ela pode ser adicionada na Linha *de Buffer* (linha temporária) para uma nova instrução no lado dos endereços, ou para uma instrução existente no lado das instruções. A condição IF só pode ser acessada pela tela de detalhes das seguintes instruções de controle: • JUMP • CALL • RET • PAUSE • ABORT As condições disponíveis para a função IF são: • IN# - verifica o status de uma Input (ON ou OFF); • IG# - verifica o status de um input group (= ou <>; 0 a 255).

No exemplo do trecho de código a seguir, temos um exemplo para cada uso da instrução IF com várias instruções de controle:

```
0000 NOP
0001 CALL JOB:START
0002 CALL JOB:A IF IN#(1) = ON
0003 CALL JOB:B IF IN#(2) = ON
0004 JUMP*LABEL1 IF IG#(1) = 0
0005 SET B004 2
0006 *LABEL2
0007 MOVL V=558
0008 MOVL V=558
0009 MOVL V=558
```

```

0010 DEC B004
0011 JUMP *LABEL2 IF B004 > 0
0012 MOVL V=558
0013 *LABEL1 0014
MOVJ VJ=25.00
0015 MOVJ VJ=100.00
0016 PAUSE IF IN#(7) = OFF
0017 END
    
```

4.2.6. VARIÁVEIS E INSTRUÇÕES ARITMÉTICAS

Uma variável aritmética é um elemento de codificação capaz de armazenar um valor numérico, geralmente inteiro ou real. Estas variáveis são normalmente utilizadas para armazenar valores de controle do robô. O tipo da variável determina os valores que a variável pode receber. A Tabela 4.1 apresenta os tipos, endereços, faixa de utilização e aplicações destas variáveis.

Tabela 4.1 Tipos, endereços, faixa de utilização e aplicações das variáveis aritméticas.

Type	Addresses	Range	Application
Integer	I000-I099	-32768 to +32767	Counting
Double Precision	D000-D099	-2147483649 to +2147483647	Distance (Microns)
Real	R000-R099	±3.400000 E+38 (minimum/maximum) ±9.999999E-38 (values between -1 and +1)	Higher Arithmetic Calculations (DIV, SQRT, SIN, COS, ATAN, etc.)
Byte	B000-B099	0-255	I/O Communication

Fonte: YASKAWA, 2014.

Uma Instrução Aritmética é aquela utilizada com variáveis aritméticas, ou seja, para cálculos ou atribuições de valores de maneira geral. O menu ARITH do INFORM LIST contém uma vasta seleção de instruções aritméticas comumente utilizadas. A Tabela 4.2 apresenta as instruções usadas para contadores, começando por um valor inicial, e limpando o conteúdo de uma sequência de variáveis para o padrão, zero.

Tabela 4.2 Instruções para contadores: funcionalidade, formato, exemplo e resultado obtido.

	<i>Function</i>	<i>Format</i>	<i>Example</i>	<i>Result</i>
INC	Adds one to the contents of specified address.	INC B<address> INC I<address> INC D<address>	INC B023 INC I023 INC D023	If contents = 6 increments to 7.
DEC	Subtracts one from the contents of specified address.	DEC B<address> DEC I<address> DEC D<address>	DEC B023 DEC I023 DEC D023	If contents = 6 decrements to 5.
SET	Sets Data 1 equal to Data 2.	SET <Data 1> <Data 2>	SET I012 3	I012 becomes a constant of 3.
CLEAR	Clears range from address to specific value.	CLEAR <address> CLEAR <address>	CLEAR I007 4 CLEAR I007 ALL	Clears I007, 8, 9, 10 Clears from I007 on. (I007-I099)

Fonte: YASKAWA, 2014.

A instrução INC incrementa o valor armazenado em uma variável (+1). Por exemplo, se I012 tem 6 no seu valor, após a execução da instrução INC012, ele terá 7 em seu valor. A instrução INC só pode ser utilizada com as variáveis tipo B, I e D. O trecho de código a seguir exemplifica isto.

```
0013 MOVJ VJ=25.00
0014 MOVJ VJ=100.00
0015 INC I012
0016 END
```

A instrução DEC decrementa o valor armazenado em uma variável (-1). Por exemplo, se I022 tem 10 no seu valor, após a execução da instrução DEC I022, ele terá 9 em seu valor. A instrução DEC só pode ser utilizada com as variáveis do tipo B, I e D. A instrução DEC deve ser usada para alcançar uma meta de produção. Uma variável é selecionada para armazenar o valor necessário de peças a ser produzidas. Ela será decrementada cada vez que uma peça for produzida. Para melhores resultados, faça o decremento somente no final do programa de produção. O trecho de código a seguir exemplifica isto.

```
0004 MOVL V=725
0005 MOVL V=725
0006 MOVL V=725
0007 MOVL V=725
0008 DEC I022
0009 PAUSE IF I022 <= 0
0010 END
```

A instrução SET armazena qualquer valor em uma variável. Isso permite que o programa determine valores iniciais para variáveis sem depender da entrada manual de valores. Pode também ser usada para zerar uma variável individual; se for zerar mais de

uma variável consecutiva, utilize a função CLEAR. O trecho de código a seguir exemplifica isto.

```
0004 MOVJ VJ=25.00
0005 MOVJ VJ=50.00
0006 SET B020 15
0007 SET I099 0
0008 END
```

A instrução CLEAR permite que variáveis consecutivas sejam zeradas. A primeira variável da sequência procede a instrução CLEAR. No trecho de código a seguir, CLEAR I007 4 zera o conteúdo de 4 variáveis, I007, I008, I009 e I010.

```
0004 MOVJ VJ=25.00
0005 MOVJ VJ=100.00
0006 CLEAR I007 4
0007 CLEAR B000 ALL
0008 END
```

4.2.7. INSTRUÇÕES DE ENTRADA E SAÍDA - I/O

Um sistema robótico raramente trabalha sem interação com outros dispositivos. O controlador deve se comunicar com equipamentos externos, como fontes de solda, posicionadores, dispositivos, sensores. A comunicação pode ser feita através do uso de Entradas e Saídas Universais (I/O). A linguagem INFORM III possui instruções de I/O digitais e analógicas, tais como a DOUT, WAIT e DIN.

A instrução DOUT com a tag OT#() opera apenas uma saída universal. É utilizado a qualquer hora que um dispositivo, como uma lâmpada, grampo, garra, deve ser ligado ou desligado. O trecho de código a seguir exemplifica o uso da função DOUT sendo utilizada com uma única saída universal

```
0003 DOUT OT#(3) ON
0004 MOVL V=2000
0005 MOVL V=2000
0006 DOUT OT#(3) OFF
```

A instrução WAIT é usada quando o programa do robô depende de um sinal de uma chave switch, sensor, PLC, etc. Isso abrange IN#, OT#, IG#, OG#, IGH#, OGH# (universais) e SIN#, SOUT# (específicos), assim como outras variáveis *Byte*. Um *TIMER* pode ser usado com *WAIT*. Quando o timer é utilizado, o robô espera que a condição seja verdadeira ou até que o tempo se esgote, o que vier primeiro. A seguir, todas as possibilidades da instrução WAIT:

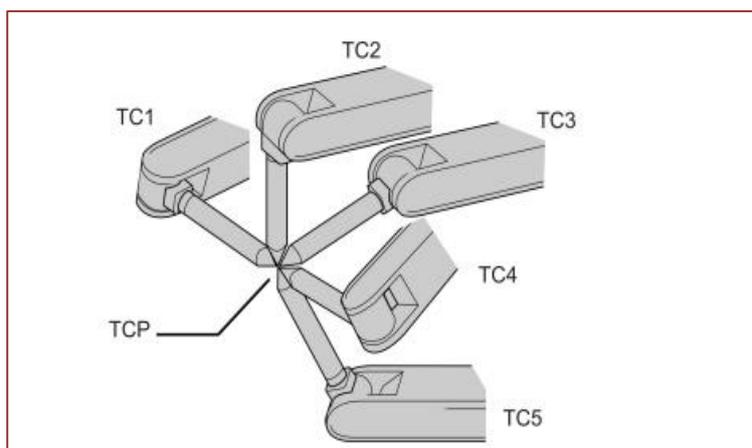
- WAIT IN#(3)=ON [espera até que a IN#3 fique ON]
- WAIT IG#(1)=255 [espera até que o IG#(1) = 255]
- WAIT IN#(5)=OFF T=3.25 [espera que a IN#5 seja desligada por 3.25s continuando para a próxima linha].

A instrução DIN (Digital Input) é usada para armazenar o estado de I/O em uma variável Byte. As informações de I/O são recebidas de um CLP, chave *thumbwheel* (chave hexadecimal) ou qualquer outro dispositivo de entrada. Isso inclui I/O universais: IN#, OT#, IG#, OG#, IGH# e OGH# ou específicos: SIN# e SOUT#, assim como qualquer outra variável Byte. Quando a instrução DIN é executada, o controlador verifica o estado de uma entrada ou saída e armazena o valor binário na variável Byte. Esse valor será entre 0 e 255, conteúdo este que deverá ser utilizado posteriormente no programa: 0001 DIN B005 IG#(1)

4.2.8. CALIBRAÇÃO DE FERRAMENTA

A calibração da ferramenta de robôs é fundamental para a melhoria de desempenho de posicionamento da ferramenta, ou atuador final, garantindo precisão necessária para a realização de determinadas tarefas. A calibração consiste na estimativa de parâmetros cinemáticos do mecanismo através de diversas medições da posição da ferramenta que pode fornecer resultados de posição e/ou orientação. Existem dois métodos de calibrar a ferramenta para o robô Yaskawa o método manual e automático. A Figura 4.8 demonstra as posições para a maneira automática da calibração.

Figura 4.8 Posições para calibração automática.



Fonte: YASKAWA, 2014.

No método manual basta digitar o tamanho da ferramenta utilizada nas coordenadas X,Y,Z. Geralmente o fabricante da ferramenta informa seu dimensional em seu manual.

4.3. ROBÔ MANIPULADOR UNIVERSAL (UR)

O Robô Manipulador Universal, ou Universal Robots (UR), é do tipo conhecido como Robô Colaborativo (Collaborative Robot - COBOT), que é um robô que permite a interação entre humanos e robôs em um espaço compartilhado ou onde estes estão próximos um ao outro, portanto possui todo um sistema de controle e segurança prevendo esta interação com seres humanos.

4.3.1. PARTES E LIGAÇÃO DO ROBÔ UR-COBOT.

O robô UR-COBOT é formado pelo controlador, o manipulador e a interface de programação. A Figura 4.9 apresenta uma imagem do robô.

Figura 4.9 Robô Manipulador Universal Colaborativo – UR COBOT.

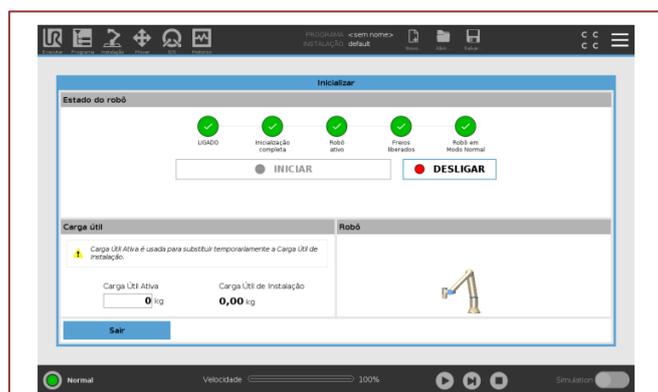


Fonte: UNIVERSAL, 2023.

Na parte inferior esquerda da tela do controlador (Figura 4.10), o ícone Inicializar indica o status do braço do robô utilizando cores:

- Vermelho: O Braço do robô está em um estado parado.
- Amarelo: O braço do robô está ligado, mas não está preparado para funcionar normalmente.
- Verde: O braço do robô está ligado e preparado para funcionar normalmente.

Figura 4.10 Tela do controlador de inicialização do Robô.



Fonte: UNIVERSAL, 2023.

4.3.2. MODOS DE UTILIZAÇÃO E PROGRAMAÇÃO.

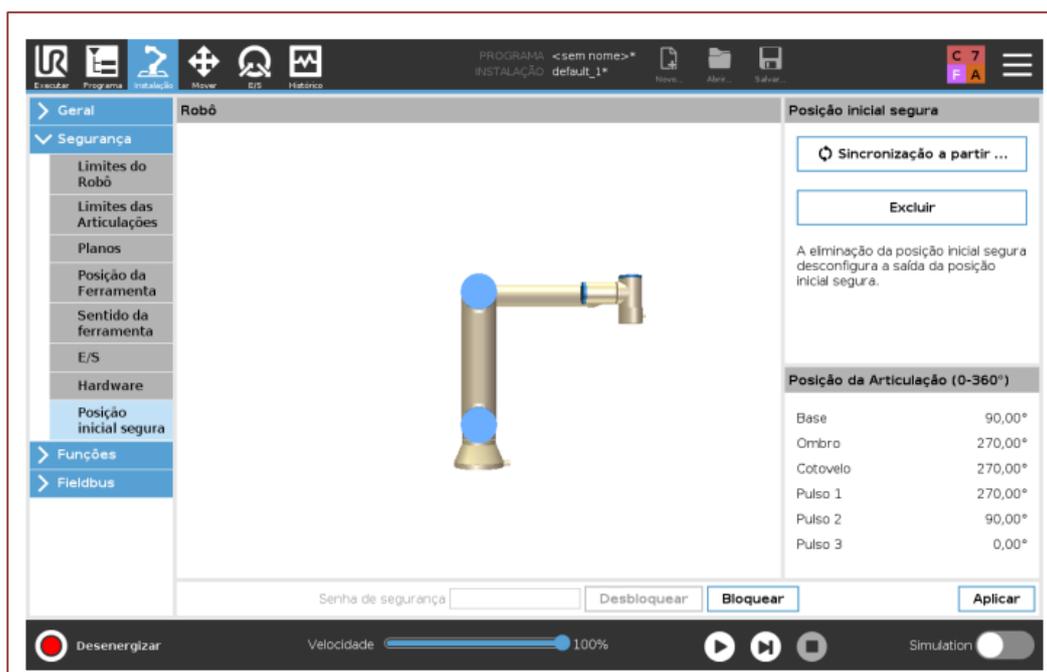
- Modos Operacionais

Os Modos Operacionais são ativadas quando você configurar um Dispositivo de Ativação de Posição 3 ou definir uma senha. Modo Automático uma vez ativado, o robô só pode executar as tarefas pré-definidas. A guia Mover e o modo de Condução Livre estão indisponíveis. Você não pode modificar ou salvar programas e instalações. No Modo Manual, depois de ativado, você programar que o robô está usando a guia Mover, o Modo de Condução Livre e o Botão Deslizante de Velocidade. Você pode modificar ou salvar programa e instalações.

- Posição Inicial Segura

A Posição inicial segura (Figura 4.11) é uma posição de retorno definida usando a posição inicial definida pelo usuário. As E/S iniciais seguras estão ativas quando o braço do robô está na posição inicial segura e uma E/S inicial segura é definida. O braço do robô está na posição inicial segura se as posições da articulação estão nos ângulos da articulação especificados ou em um múltiplo de 360 graus.

Figura 4.11 Definindo a posição inicial segura do Robô.



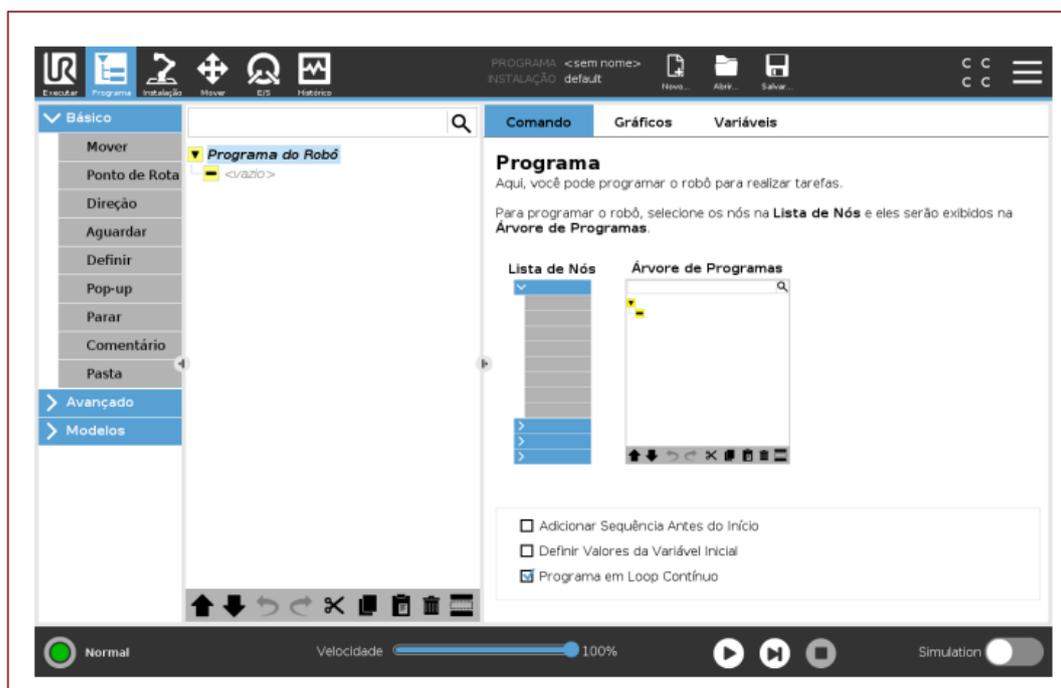
Fonte: UNIVERSAL, 2023.

- Árvore de Programas

Ao tocar em Comando, você adiciona nós do programa à Árvore do programa (Figura 4.12), e configura a funcionalidade dos nós do programa adicionados no lado direito da tela. Uma Árvore do programa vazia não pode ser executada. Os programas contendo nós do programa configurados incorretamente também não podem ser

executados. Os nós do programa inválidos são destacados em amarelo para indicar o que deve ser consertado antes que programa possa ser executado.

Figura 4.12 Árvores de programas.



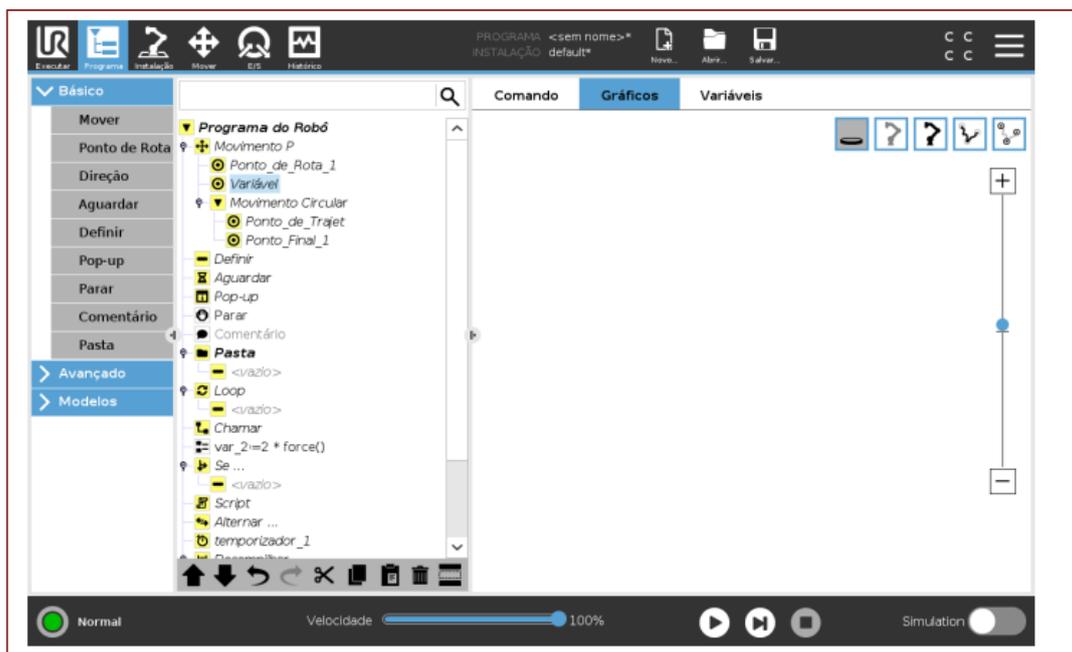
Fonte: UNIVERSAL, 2023.

Quando o programa é executado, o nó do programa que está sendo executado atualmente é indicado por um ícone pequeno ao lado do nó. Além disso, o caminho de execução é destacado com uma cor azul. Pressionar o ícone no canto do programa fará com que ele rastreie o comando que está sendo executado.

- Representação gráfica do programa

A representação gráfica do programa apresentar a posição e orientação atual do robô. O caminho do TCP é mostrado na visualização 3D, com segmentos de movimentação em preto e segmentos de união (transições entre segmentos de movimentação) mostrados em verde. Os pontos verdes especificam as posições do TCP em cada um dos pontos de rota no programa. O desenho 3D do braço do robô mostra a posição atual do braço, e sua sombra mostra como o braço do robô pretende chegar ao ponto de rota selecionado do lado esquerdo da tela. Se a posição atual do TCP do robô se aproximar de um plano de segurança ou de acionamento, ou a orientação da ferramenta do robô estiver perto do limite de orientação da ferramenta. (Figura 4.13).

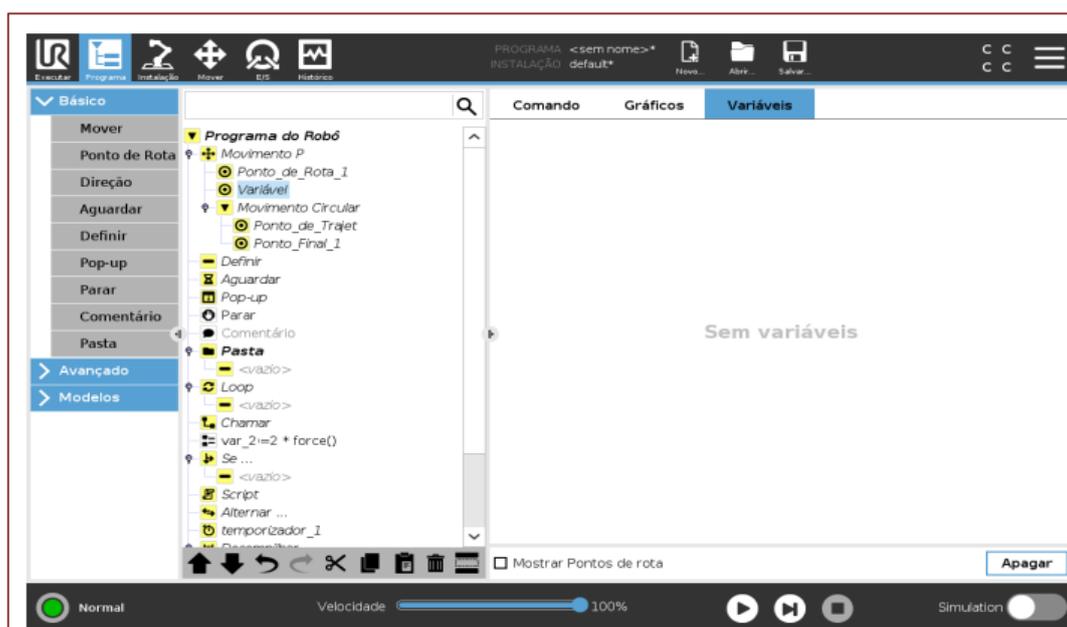
Figura 4.13 Representação gráfica: Posição do TCP.



Fonte: UNIVERSAL, 2023.

A guia Variáveis mostra os valores ativos das variáveis no programa em execução e mantém uma lista de variáveis entre execuções do programa. Ela aparece somente quando tem informações para exibir. Variáveis de pontos de rota serão mostradas na lista, se Mostrar Pontos de rota estiver ativado. e-Series II-40 Versão 5 (Figura 4.14).

Figura 4.14 Tela das variáveis ativas.

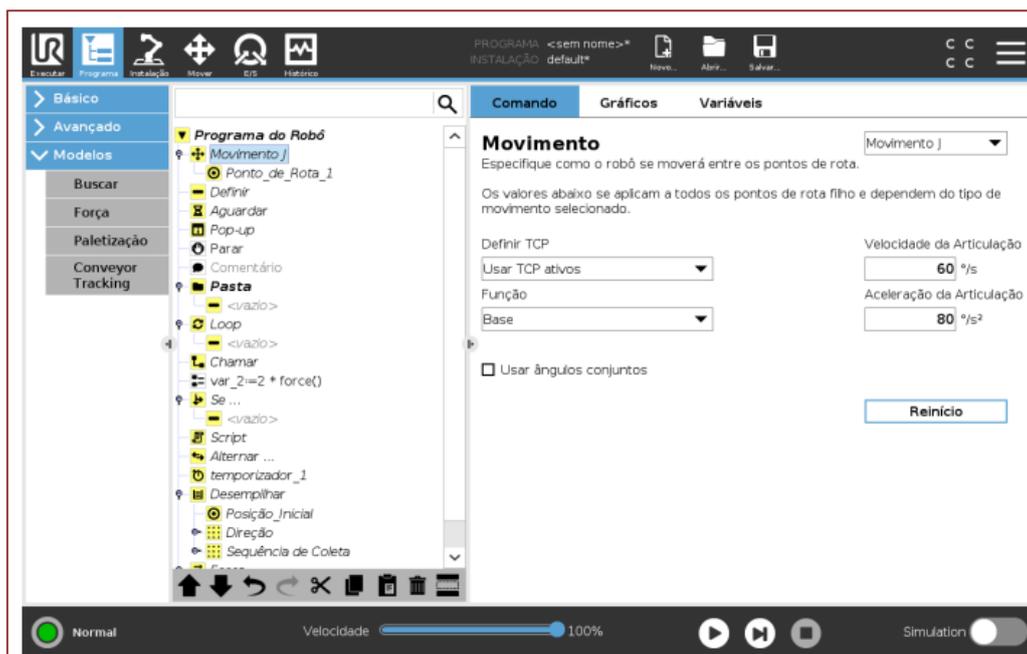


Fonte: UNIVERSAL, 2023.

- Comando Mover

O comando Mover controla a movimentação do robô por meio de pontos de rota subjacentes. Os Pontos de Rota precisam estar sob o comando Mover. O comando Mover define a aceleração e a velocidade em que o braço do robô se moverá entre esses pontos de rota (Figura 4.15).

Figura 4.15 Comando MOVER de movimentação do robô.



Fonte: UNIVERSAL, 2023.

Em relação aos tipos de movimentos, é possível selecionar um de três tipos de movimentos: Movimento J, Movimento L e Movimento P. Cada tipo será explicado a seguir.

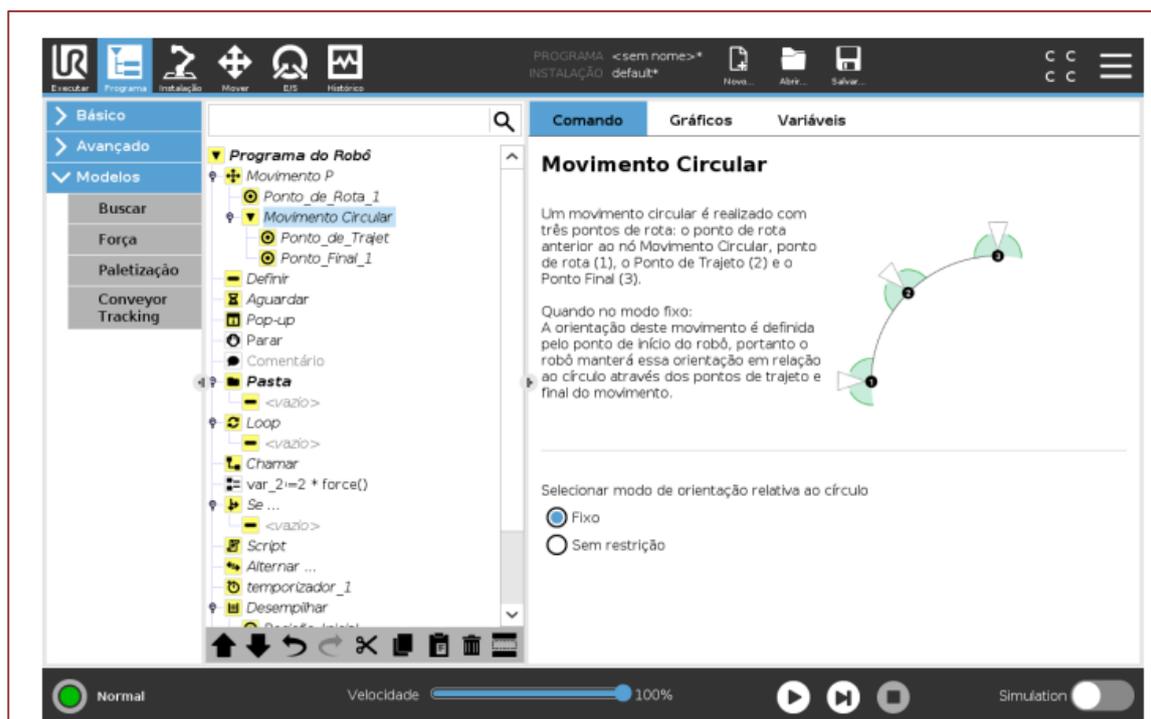
O Movimento J faz movimentos que são calculados no espaço de articulação do braço do robô. As articulações são controladas para terminar os seus movimentos ao mesmo tempo. Esse tipo de movimento resulta em um caminho curvo para a ferramenta. Os parâmetros compartilhados que se aplicam a esse tipo de movimento são a velocidade máxima da articulação e a aceleração da articulação, especificados em graus/s e graus/s², respectivamente. Se for desejado que o braço do robô se mova rapidamente entre pontos de rota, desconsiderando o caminho da ferramenta entre esses pontos de rota, esse tipo de movimento será a opção preferencial.

O Movimento L move o Ponto Central da Ferramenta (TCP) linearmente entre pontos de rota. Isso significa que cada articulação executa uma movimentação mais complicada para manter a ferramenta em um caminho de linha reta. Os parâmetros compartilhados que podem ser definidos para esse tipo de movimento são a velocidade desejada da ferramenta e a aceleração da ferramenta especificadas em mm/s e mm/s², respectivamente, além de um recurso.

O Movimento P move a ferramenta linearmente com velocidade constante com uniões circulares e é destinado a algumas operações do processo, como colar e descartar. O tamanho do raio de união é por padrão um valor compartilhado entre todos os pontos de rota. Um valor menor fará o caminho se tornar mais angular, enquanto um valor mais alto fará o caminho mais suave. Enquanto o braço do robô estiver se movendo pelos pontos de rota com velocidade constante, a caixa de controle do robô não poderá esperar por uma operação de E/S ou uma ação do operador. Fazer isso poderá parar a movimentação do braço do robô ou causar uma parada de proteção.

O Movimento circular (Figura 4.16) pode ser adicionado a um Movimento P para fazer um movimento circular. O robô inicia o movimento a partir de sua posição atual ou ponto de início, movimenta-se através de um Ponto de Trajeto especificado no arco circular, e um Ponto Final, que completa o movimento circular. Um modo é usado para calcular a orientação da ferramenta, através do arco circular. O modo pode ser: – Fixo: apenas o ponto de início é usado para definir a orientação da ferramenta – Sem restrição: o ponto de início se transforma em Ponto Final para definir a orientação da ferramenta. Parâmetros compartilhados.

Figura 4.16 Movimento circular do robô.



Fonte: UNIVERSAL, 2023.

4.4. ROBÔ EPSON SCARA

O robô do tipo SCARA foi desenvolvido pela primeira vez nos anos 80 no Japão, e o nome SCARA significa *Selective Compliance Assembly Robot Arm*. Apesar do nome extenso, na prática ele possui um braço articulado de 2 eixos que, em certa medida, imita o braço humano, embora ele opere em um único plano numa movimentação do tipo XYZ.

4.4.1. PARTES DO ROBÔ EPSON SCARA.

O Robô Epson SCARA é formado por controlador e manipulador e a programação é feita através do computador com software específico do fabricante. Sua construção permite a ele se estender ou se retrair em áreas confinadas.

Quando se trata de operações de *pick-and-place* ou montagem, muito comuns no Polo Industrial de Manaus, os robôs SCARA são a opção perfeita, já que estas aplicações demandam alta velocidade e alta precisão. Neste cenário, eles são capazes de operar com altas acelerações e alta velocidade, tanto em ambientes agressivos quanto limpos.

A Figura 4.17 apresenta o Robô Epson SCARA, o manipulador e o controlador.

Figura 4.17 Manipulador e controlador do Robô Epson SCARA



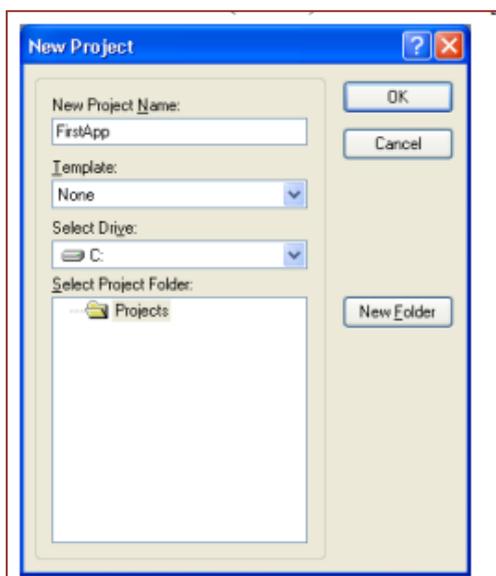
Fonte: EPSON, 2023.

4.4.2. PROGRAMAÇÃO DO ROBÔ EPSON SCARA.

- Criando um programa

Depois de instalar o controlador, o robô e o software EPSON RC+ 7.0 no Controlador de Robô RC700, siga estas instruções para criar um programa aplicativo simples para se familiarizar com o ambiente de desenvolvimento do EPSON RC+ 7.0. 1. Inicie o EPSON RC+ 7.0 Dê um clique duplo no ícone do robô do EPSON RC+ 7.0 na área de trabalho. (Figura 4.18).

Figura 4.18 Criando um novo programa (projeto).

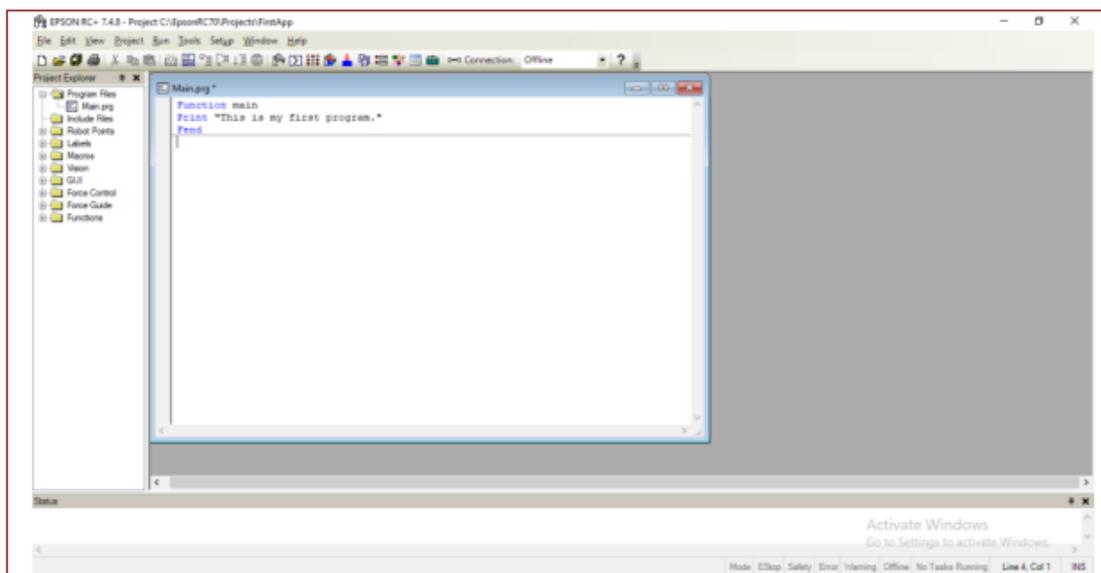


Fonte: EPSON, 2023.

Digite um nome para o projeto na caixa [New Project Name] (Nome do novo projeto). P.ex. FirstApp (3) Clique em OK para criar o novo projeto. Quando um novo projeto é criado, um programa chamado “Main.prg” é criado. Você verá a janela “Main.prg” aberta com um cursor piscando no canto superior esquerdo. Agora você está pronto para começar a entrar em seu primeiro programa.

Digite as seguintes linhas de programa na janela de edição “Main.prg”. Function main Print "This is my first program." Fend 4. Execute o programa, para isto, pressione F5 para executar o mesmo. A Figura 4.19 apresenta o resultado deste pequeno exemplo.

Figura 4.19 Exemplo de primeiro programa.



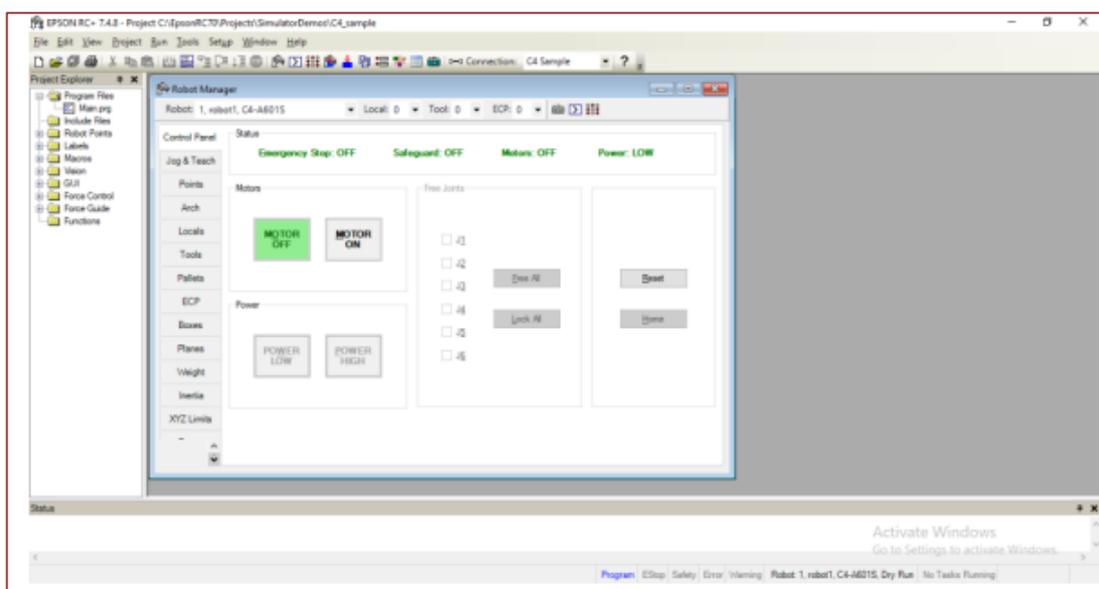
Fonte: EPSON, 2023.

Durante a construção do projeto, seu programa é compilado e vinculado. A comunicação é estabelecida com o controlador e os arquivos do projeto são enviados para ele. Se não houver erros durante a construção, a janela de execução aparecerá. A partir daí, clique no botão [Iniciar] na janela [Run] para executar o programa. Você deverá ver um texto similar ao seguinte exibido na janela [Status]: Task main started - All tasks stopped. Na janela [Run], você verá a saída da instrução de impressão.

- Ensinando os pontos ao robô

- 1) Certifique-se de que é seguro operar o robô, com o mesmo com o motor no estado off. Clique no botão (Gerenciador do robô) na barra de tarefas. Você verá a janela [Robot Manager] com a página [Control Panel] (Painel de controle) exibida (Cf. Figura 4.20).

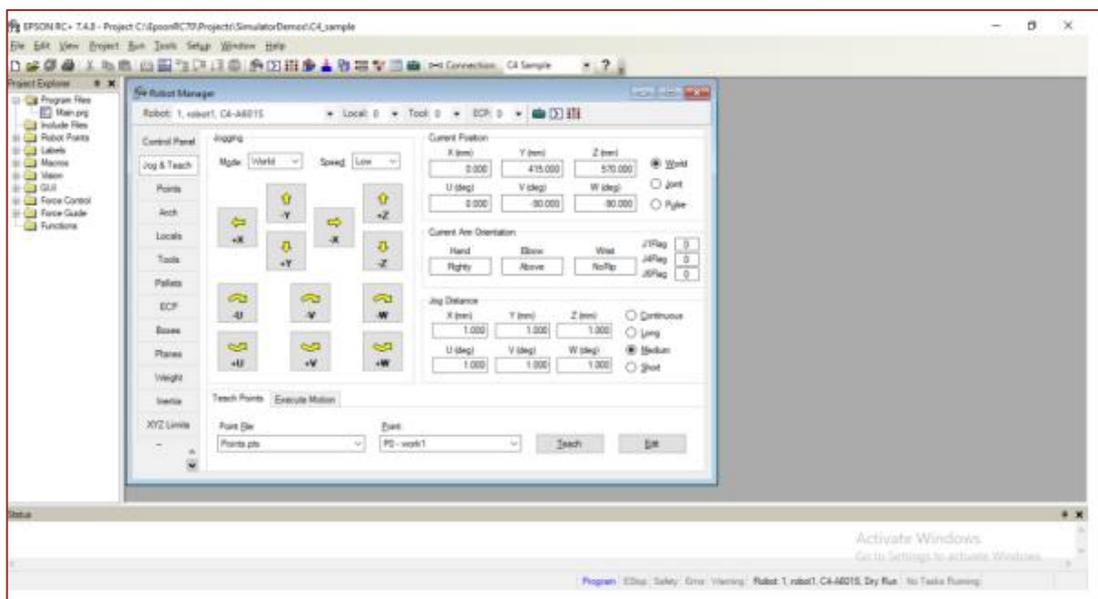
Figura 4.20 Robô com motor off seguro para programar.



Fonte: EPSON, 2023.

- 2) Clique no botão para ligar os motores do robô. Você será solicitado a confirmar a operação.
- 3) Responda (Sim) para continuar.
- 4) Clique na aba [Jog & Teach] (Deslocar e ensinar) (Cf. Figura 4.21)

Figura 4.21 Tela Jog & Tech para definir os pontos de deslocamento do Robô.



Fonte: EPSON, 2023.

- 5) Clique no botão (Ensinar) no canto inferior direito para ensinar o ponto P0. Você será solicitado a inserir um rótulo e a descrição do ponto. (6) Desloque o robô clicando no botão de deslocamento. Mantenha o botão pressionado para continuar deslocando. Solte quando o robô estiver aproximadamente na metade do caminho em direção à área de trabalho. (7) Desloque o robô para baixo clicando no botão <-Z>. (8) Agora mude o ponto atual para P1 selecionando P1 na lista suspensa Point (Ponto) ao lado do botão . (9) Clique no botão . Você verá uma mensagem de confirmação para ensinar o ponto. (10) Responda . (11) Clique no botão para deslocar o robô na direção +X. (12) Mude o ponto atual para P2 selecionando P2 na lista suspensa Point.

Como exemplo, insira três novas instruções Go no programa Main.prg como é mostrado no trecho de código a seguir:

```
Function main Print "This is my first program."
```

```
Motor On
```

```
Go P1
```

```
Go P2
```

```
Go P0
```

```
Fend
```

Depois, insira os comandos Power (Potência), Speed (Velocidade) e Accel (Aceleração) como é mostrado no trecho de código a seguir:

Function main Print "*This is my first program.*"

Motor On

Power High

Speed 50

Accel 50, 50

Go P1

Go P2

Go P0

Fend

Em seguida, clique no botão na janela Run. O robô deverá ir para cada um dos pontos que você ensinou à metade da velocidade, aceleração e desaceleração. A instrução Power High (Potência alta) permite que seu programa execute o robô em alta potência (normal), o que, por sua vez permite que a velocidade e a aceleração sejam aumentadas.

4.5. ROBÔ MANIPULADOR STÄUBLI

O robô manipulador Stäubli é similar ao robô manipulador Yaskawa, sendo mais robusto e com maior carga de atuação, devido sua composição estrutural. Por conta disto, é um robô industrial muito utilizado para o processamento de peças metálicas, desde a usinagem até a carga e descarga de máquinas-ferramenta.

4.5.1. PARTES DO ROBÔ STÄUBLI.

Como os demais, é formado por controlador e manipulador e a programação é feita através do computador com software específico do fabricante. O robô Stäubli foi projetado com braços robóticos compactos e fechados com várias opções de fixação (Figura 4.22).

Figura 4.22 Manipuladores e Controlador do Robô Stäubli.



Fonte: <https://www.staubli.com>, 2023.

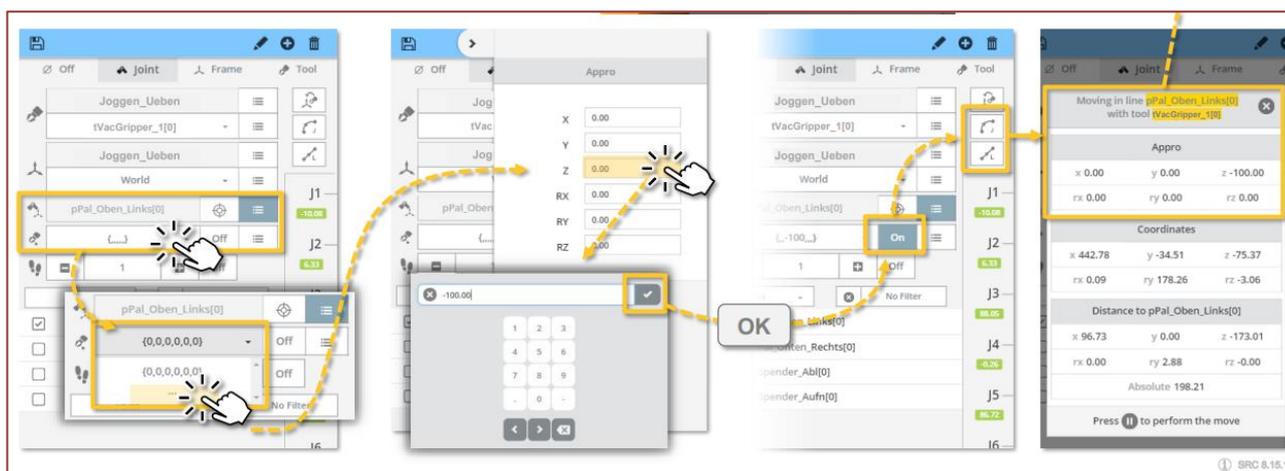
4.5.2. PROGRAMAÇÃO DO ROBÔ STÄUBLI.

Devido a um atraso na entrega do equipamento, importado, este robô foi o menos utilizado na capacitação em robótica industrial avançada. Portanto, daremos um breve resumo de sua programação. Toda a programação é feita na plataforma *Stäubli Robotics Suite*, que pode se conectar à controladora para obter automaticamente os parâmetros do modelo de robô utilizado, evitando que tais parâmetros sejam colocados manualmente. A programação deverá sempre ser no modo *offline*, e assim como outros robôs similares, pode ser utilizado o TP (*Teach Pendant*).

- Função Approach (aproximação)

Quando da definição dos pontos e velocidades para a trajetória do robô, pode ser gravado um ponto de aproximação gravando um valor próximo do ponto, utilizando a função *approach*. Nesta etapa é utilizado o TP para gravar o ponto. A Figura 4.23 apresenta um exemplo do uso desta função.

Figura 4.23 Uso da função Approach.



Fonte: STAUBLI, 2023.

- Salvando alterações.

A cada ponto de posição, tempo ou velocidade que for modificado, as alterações devem ser salvos para que sejam retidos após o aplicativo ser fechado. O modo Jog é indicado com um asterisco próximo ao ícone “Salvar”, e isto indica que foram feitas mudanças que ainda não foram salvas. Para salvar as alterações, toque no ícone Salvar – o asterisco desaparece. (vide Figura 4.24).

Figura 4.24 Salvando as alterações.

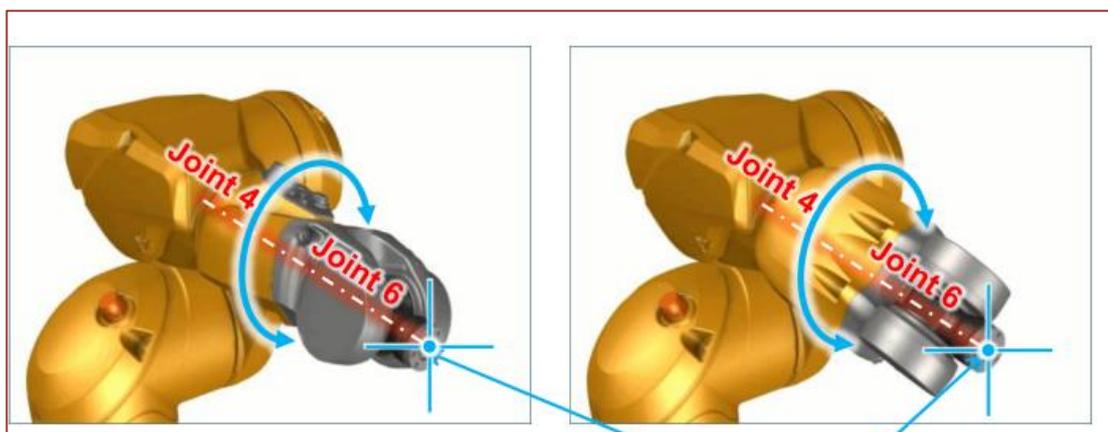


Fonte: STAUBLI, 2023.

- Singularidades

Singularidades são pontos no espaço que podem ser alcançados por uma distância infinita, e com isto geram um número de posição conjunta. A Figura 4.25 exemplifica um ponto singular.

Figura 4.25 Pontos singulares.



Fonte: STAUBLI, 2023.

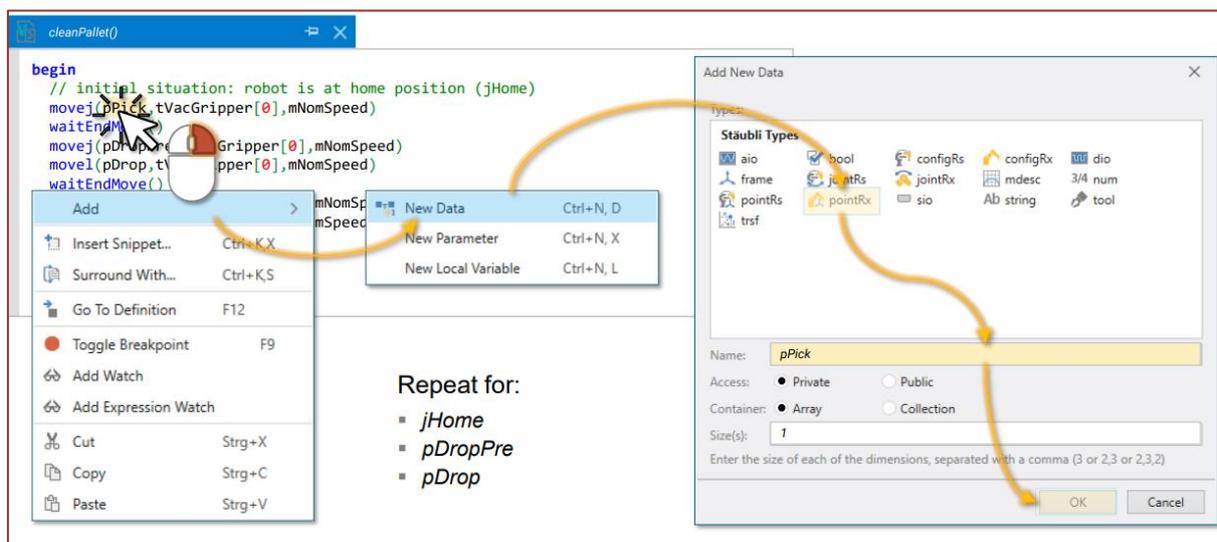
- Exemplo de programa.

A seguir apresentamos um trecho de Código como exemplo de programação do robô stäubli. Basicamente, são um conjunto de instruções de movimentação do robô. O comando *movej* movimenta o ponto da ferramenta, tendo como parâmetros a variável ponto, tipo de movimento e velocidade, e o comando *waitEndMove* impõe uma espera do final do movimento anterior para que o próximo comando seja efetuado.

```
begin // initial situation: robot is at home position (jHome)
movej(pPick,tVacGripper[0],mNomSpeed)
waitEndMove()
movej(pDropPre,tVacGripper[0],mNomSpeed)
movel(pDrop,tVacGripper[0],mNomSpeed)
waitEndMove()
movel(pDropPre,tVacGripper[0],mNomSpeed)
movej(jHome,tVacGripper[0],mNomSpeed)
```

A variável ponto pode ser adicionada ou definida seu tipo simplesmente colocando o curso sobre ela e seguindo o fluxo das telas conforme a Figura 4.26.

Figura 4.26 Adicionar ou redefinir variável ponto.



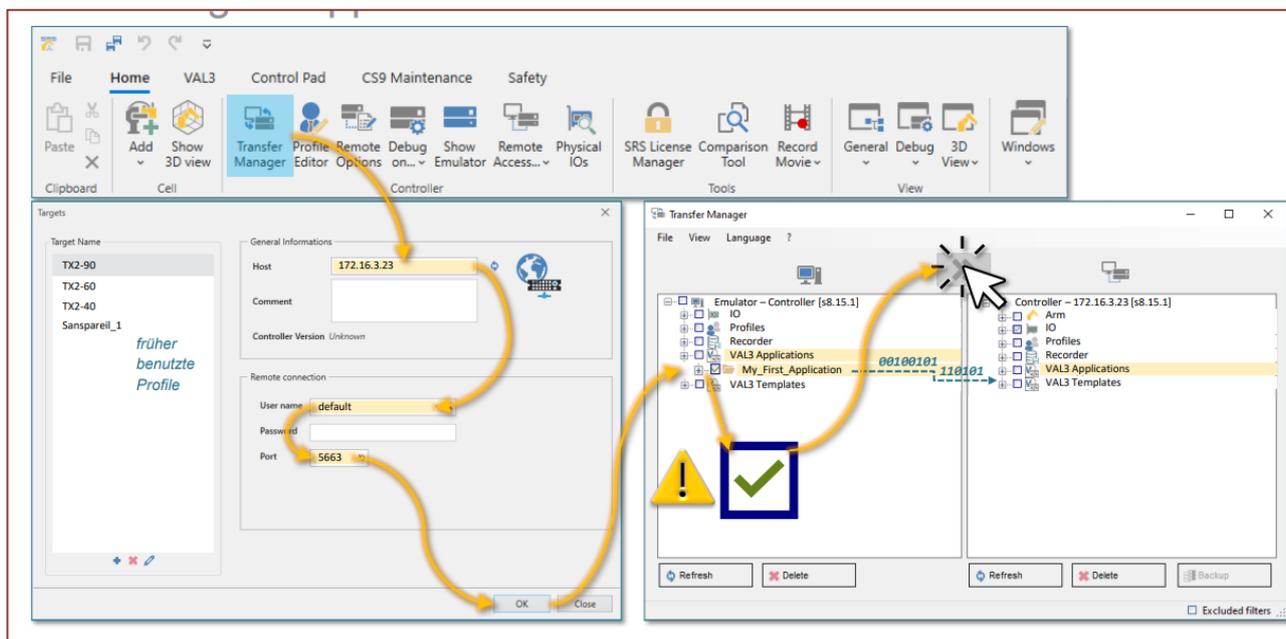
Fonte: STAUBLI, 2023.

- Transferindo a programação para o controlador.

Uma vez concluído o programa na plataforma, de modo offline, ele deve ser transferido para o controlador. Esta etapa necessita configurar a rede (endereço IP) para a controladora e o equipamento utilizado para a programação (desktop ou notebook).

O programa pode ser transferido seguindo o fluxo das telas conforme a Figura 4.27.

Figura 4.27 Transferindo a programação para o controlador.



Fonte: STAUBLI, 2023.

REFERÊNCIAS

- [1] EPSON Robot. Manual Epson LS10-B Robot (Revision 6) Portuguese. Disponível em https://files.support.epson.com/far/docs/epson_ls10b_robot_manual_r6_portuguese_cpd60411.pdf, acesso em dez 2023.
- [2] EPSON RC+ 7.0 Guia do usuário Gestão e desenvolvimento de projetos Ver.7.5 REV1. 606 páginas.
- [3] UNIVERSAL Robots Academy,2023.Disponível em : <https://www.universal-robots.com/br/academy/>. Acesso em 10/08/2023.
- [4] STAUBLI CS9 Programming Course - (16.03.2023) - © Staubli. Documentação privada e não publicada, com acesso somente mediante participação no treinamento.
- [5] YASKAWA MOTOMAN. **Manual de Programação Básica DX200**. São Paulo-SP: Yaskawa do Brasil, 1ª Edição ,17 de junho de 2014, 183 páginas.

05

Disciplina de programação em CLP

Hugo Alves Velozo¹

5.1. CARACTERIZAÇÃO DA DISCIPLINA

Esta disciplina oferece uma abordagem abrangente à Programação em Controladores Lógicos Programáveis (CLP), destacando-se a utilização de linguagem *ladder*, blocos de função e texto estruturado. Os estudantes serão guiados na aquisição de habilidades técnicas fundamentais de programação de CLPs, explorando desde a representação gráfica intuitiva da lógica de controle até a abordagem modular para sistemas complexos. Uma ênfase especial será dada à integração de sistemas robotizados com CLPs e sistemas de visão, proporcionando aos alunos uma visão holística das tecnologias modernas de automação.

A metodologia adotada para ministrar desta disciplina é a Aprendizagem Baseada em Projetos (PBL), onde os estudantes serão desafiados a aplicar seus conhecimentos teóricos na resolução de problemas práticos seguindo a metodologia desenvolvida por Queiroz-Neto et al. (2015). Isso proporcionará uma experiência de aprendizado mais envolvente e aplicável, preparando os alunos para enfrentar os desafios reais encontrados na implementação de sistemas de controle e automação industrial. Ao final do curso, os alunos não apenas terão domínio das linguagens de programação em CLP, mas também estarão aptos a integrar eficientemente sistemas robotizados com CLPs e tecnologias de visão em projetos de automação avançada.

5.1.1. EMENTA

Nesta disciplina, será adotada a metodologia de ensino centrada na Aprendizagem Baseada em Projetos (PBL), proporcionando uma abordagem prática e aplicada aos estudantes. O curso iniciará com a conceituação de Controladores Lógicos Programáveis (CLPs), incluindo uma análise dos diferentes tipos disponíveis no mercado e suas respectivas aplicações industriais. Serão explorados casos reais e exemplos práticos para ilustrar a diversidade de implementações possíveis. Em seguida, serão discutidos os componentes modulares de um CLP, detalhando a arquitetura e as funcionalidades essenciais. Aprofundar-se-á no conhecimento da programação por meio do software GXWorks 2, com foco nas linguagens Ladder, Blocos de Função e Texto Estruturado. Além disso, será abordada a integração deste software com Factory IO para simulação realista, bem como a interação com sistemas robóticos e sistemas de visão, proporcionando aos alunos uma visão abrangente e prática da automação industrial moderna.

¹ Mestre em Automação e Controle. Professor da disciplina de Programação em CLP. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM).

5.1.2. OBJETIVOS

O objetivo da disciplina é capacitar os alunos na programação de Controladores Lógicos Programáveis (CLPs) para automação industrial de sistemas robotizados. Utilizando a metodologia de Aprendizagem Baseada em Projetos (PBL), o curso abrange conceitos de CLPs, tipos disponíveis, aplicações práticas e programação, com enfoque em linguagens como Ladder, Blocos de Função e Texto Estruturado. A formação inclui simulações realistas e exploração da interação com sistemas robóticos e de sistemas de visão, proporcionando uma preparação abrangente para enfrentar desafios na automação e robotização de ambientes industriais.

5.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 30 h

Carga Horária Prática: 30 h

Total: 60 h

5.2. CONTEÚDOS

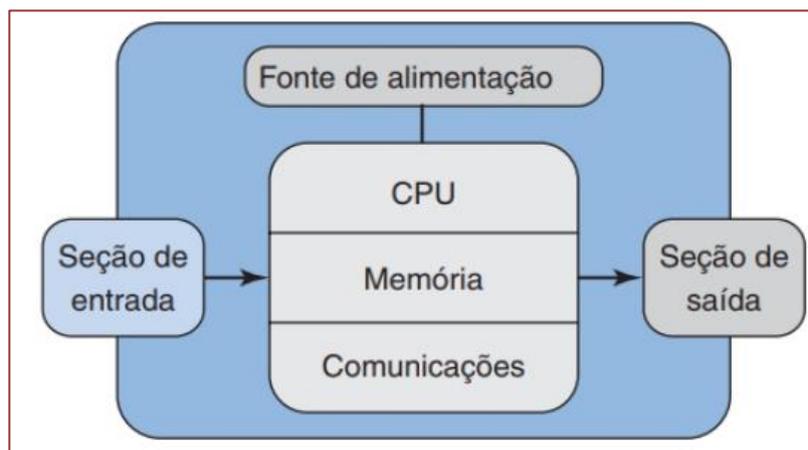
A seguir serão apresentadas breves introduções aos conteúdos ministrados durante a disciplina.

5.2.1. FUNDAMENTOS DE CONTROLADORES LÓGICOS PROGRAMÁVEIS - HARDWARE

Um Controlador Lógico Programável (CLP) é um computador usado para realizar funções de controle e automação em Ambiente Industrial tais como, operações condicionadas (operações lógicas), selos, intertravamentos, operações temporizadas, contagem, operações matemáticas, operações de controle, dentre outras (DA SILVA, 2016).

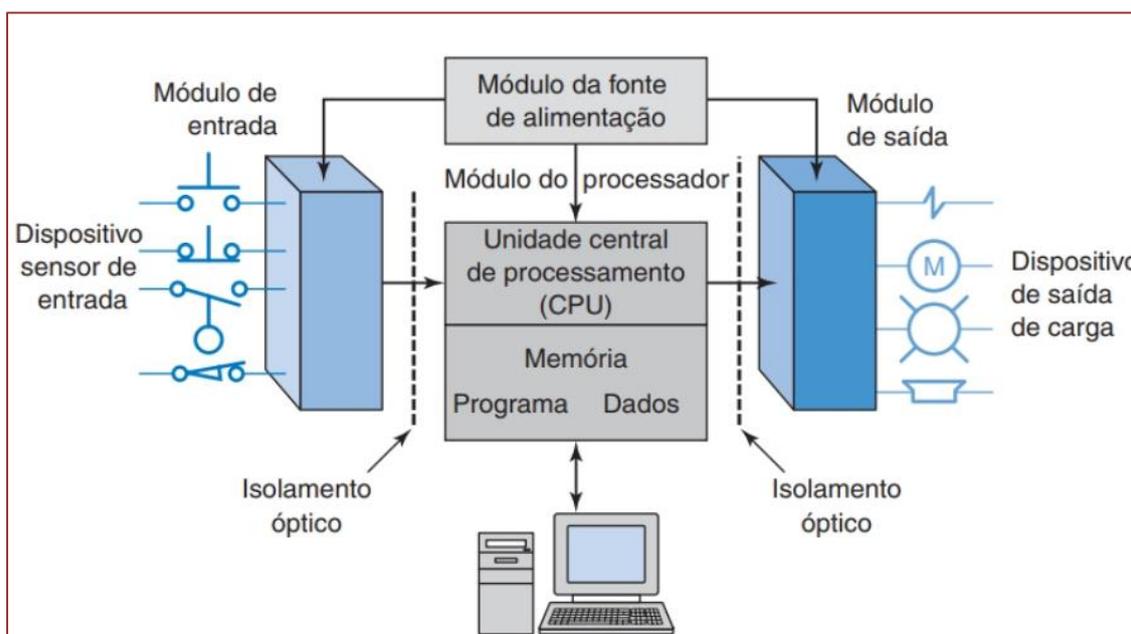
Os CLPs podem ser subdivididos em duas categorias principais quanto a forma da distribuição das entradas e saídas (no inglês, *I/O inputs e outputs*): os fixos e os modulares (PRUDENTE, 2015). Nos CLPs de I/O fixas, as entradas e saídas são incorporadas a estrutura do CLP sem possuir unidades removíveis, conforme ilustrado na Figura 5.1. Por outro lado, nos CLPs modulares, o CLP é dividido em módulos, ou seja, partes removíveis que formam a estrutura completa do CLP: base, fonte, unidade central de processamento (CPU, *central processing unit*), módulo de comunicação e módulos de entradas e saídas (WHITE, 2023). A Figura 5.2 ilustra genericamente a ideia de um CLP modular.

Figura 5.1. Diagrama de um CLP com entradas/saídas fixas



Fonte: Petruzella, 2014.

Figura 5.2. Diagrama de CLP com entradas/saídas modulares.



Fonte: Petruzella, 2014.

Para ministração da disciplina foram utilizados 3 CLPs modulares da Série Q da fabricante Mitsubishi. Contendo os seguintes componentes:

- Unidade base: codificação Q38B com 8 slots, ilustrado na Figura 3;
- Fonte de alimentação: codificação Q61P 100 – 240 Vac
- CPU: codificação Q03UDV
- Módulo de entrada: codificação QX41
- Módulo de saída: codificação QY41P

Figura 5.3. Unidade base com 8 slots.



Fonte: Autor, 2023.

A unidade base Q38B é caracterizada por três conexões iniciais sequenciais: a de extensão, alimentação e CPU. Após essa etapa, disponibiliza oito slots codificados como I/00, I/01, I/02, I/03, I/04, I/05, I/06 e I/07, destinados à conexão de CPUs, módulos de comunicação e módulos de entrada/saída, conforme destacado na Figura 5.3.

Quanto à fonte de alimentação Q61P, ela opera com uma faixa de tensão de entrada de 100 a 240 VAC, uma frequência de 50/60 Hz e fornece uma tensão de saída de 5 VDC. A correta conexão desta fonte à rede elétrica de 127/220 VAC é apresentada de maneira detalhada na Figura 5.4, fornecendo uma referência visual para a instalação adequada.

Figura 5.4. Fonte de alimentação Q61P

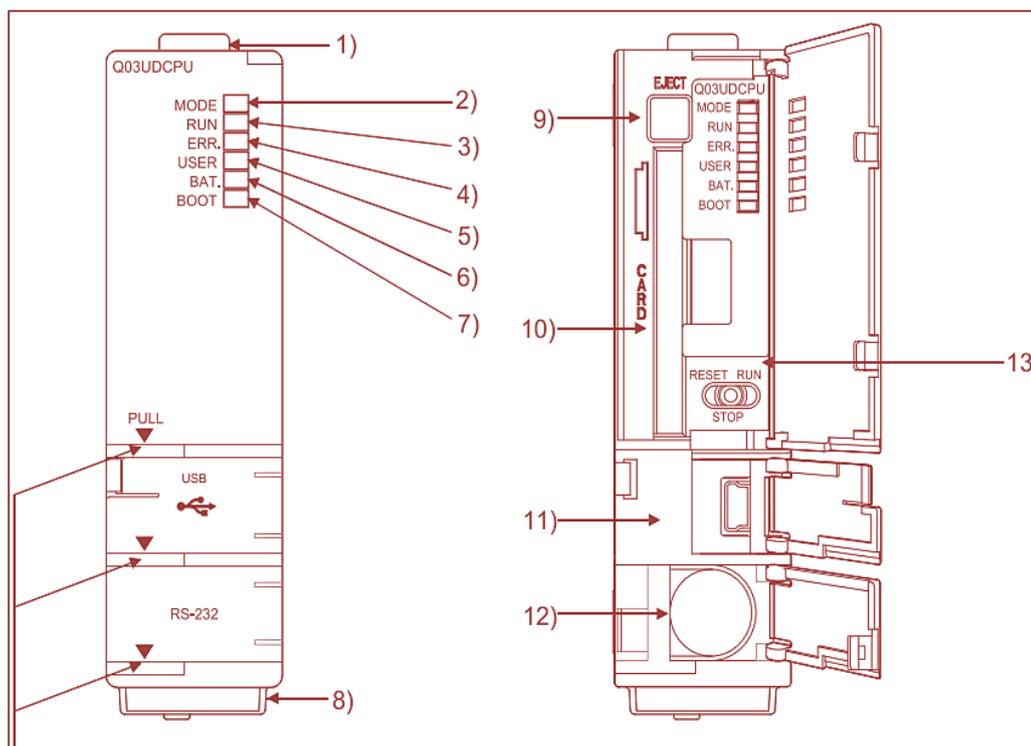


Fonte: Autor, 2023.

A CPU Q03UDV apresenta capacidade de programação de 30.000 passos e capacidade de gerenciamento de 4096/8192 pontos de entradas/saídas. Além disso esta

CPU possui período do ciclo de programa de 20 ns/*log. Instruction*. Dentro da CPU existe uma bateria (item 12 na Figura 5.5) com vida útil de 43.800 horas (5 anos).

Figura 5.5. Unidade central de processamento Q03UD.



Fonte: Mitsubishi Electric Corporation, 2004.

Na Figura 5.5 são ilustradas as principais partes de um módulo CPU. No item 1 é identificado o gancho utilizado para fixar o módulo à unidade base. No item 2 identifica-se o Led responsável por indicar o modo da CPU, ou seja, ligado ou piscando (a função de ativação/desativação forçada de entrada/saída externa está sendo executada). Por sua vez, o led RUN (item 3) pode estar em três estados: ligado (quando o interruptor RUN/STOP/RESET (item 13) estiver em RUN), desligado (quando o interruptor RUN/STOP/RESET estiver em STOP) e piscando (quando o interruptor RUN/STOP/RESET mudar de STOP para RUN). O item 4 identifica o led ERR responsável por indicar a detecção de um erro que não parou a operação (ligado), estado normal (desligado) e piscando quando identificado um erro que interrompe a operação. O led USER (item 5) ligado quando annunciator (F) ligado e desligado em estado normal. Já o led BAT no item 6, quando ligado, indica erro na bateria devido à redução nas tensões da bateria do módulo CPU ou cartão de memória, caso contrário, estado normal.

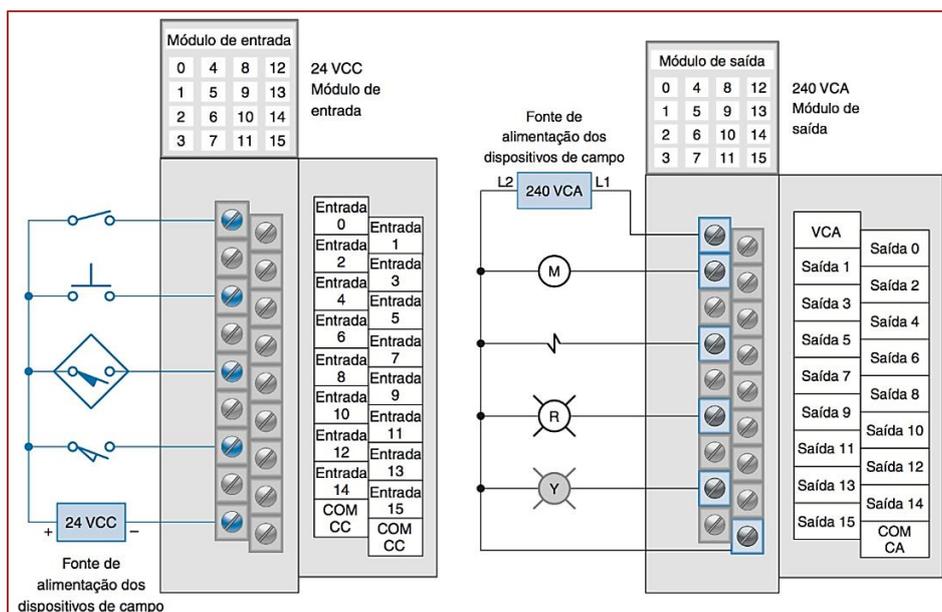
Figura 5.6. Unidade central de processamento Q03UDV utilizadas nas aulas.



Fonte: Autor, 2023.

Os módulos de entradas/saídas desempenham um papel fundamental na automação industrial, conectando dispositivos de campo ao controlador. Os módulos de entrada recebem sinais de sensores, botões e chaves-limite, fornecendo informações em tempo real. Já os módulos de saída enviam sinais para dispositivos como sinaleiras, válvulas solenoides e motores, traduzindo os comandos do controlador em ações físicas. Essa interação eficiente é essencial para otimizar processos, garantindo uma resposta ágil e precisa do sistema a mudanças nas condições de operação. Em resumo, esses módulos são cruciais para a eficiência, confiabilidade e adaptação de sistemas de automação industrial.

Figura 5.7. Ilustração dos módulos de entrada e saída.



Fonte: Petruzella, 2014.

Os módulos de entradas e saídas podem ser classificados quando ao sentido do fluxo de corrente convencional (do menor potencial ao maior potencial) entre dois dispositivos como *sinking* (dreno) e *sourcing* (fonte).

Considere dois dispositivos genéricos: dispositivo 1 e dispositivo 2. Se o sentido do fluxo de corrente é do dispositivo 1 ao dispositivo 2, então denomina-se ao dispositivo 1 como do tipo *sourcing* e o dispositivo 2 como do tipo *sinking*. Por outro lado, se o sentido do fluxo de corrente for do dispositivo 2 ao dispositivo 1, chamar-se-á o dispositivo 2 de *sourcing* e o dispositivo 1 de *sinking*. A Tabela 5.1 resume esta classificação.

Tabela 5.1. Classificação de dispositivos em relação ao sentido do fluxo de corrente.

Sentido do Fluxo de Corrente	Dispositivo 1	Dispositivo 2
Do Dispositivo 1 ao Dispositivo 2	<i>Sourcing</i> (+ comum)	<i>Sinking</i> (- comum)
Do Dispositivo 2 ao Dispositivo 1	<i>Sinking</i> (- comum)	<i>Sourcing</i> (+ comum)

Fonte: Autor, 2023.

Para um módulo funcionar como *sourcing*, o comum deverá estar conectado a um potencial positivo, de outra forma, para funcionar como *sinking*, o comum deverá estar conectado ao negativo do módulo.

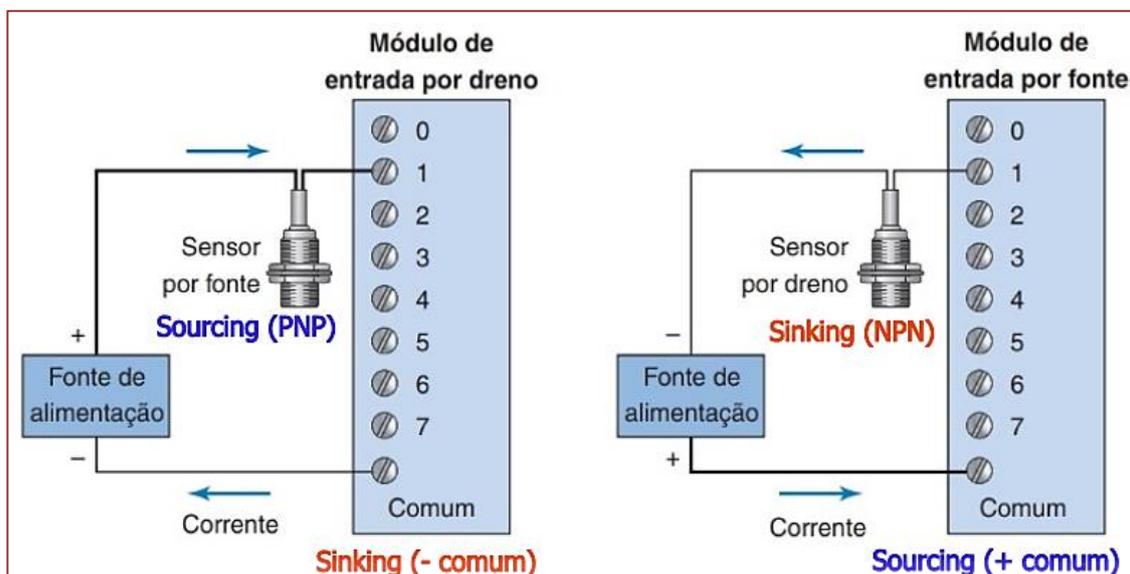
Considerando estes fatos, se um módulo de entrada é do tipo *sourcing* então o sensor conectado neste módulo é do tipo NPN. Caso contrário, módulo de entrada do tipo *sinking*, então o sensor conectado a este módulo será PNP. A Tabela 5.2 resume esta classificação e a Figura 5.8 ilustra a mesma.

Tabela 5.2. Classificação de módulos e sensores.

Sentido do Fluxo de Corrente	Módulo	Sensor
Do Módulo ao Sensor	<i>Sourcing</i> (+ comum)	<i>Sinking</i> (NPN)
Do Sensor ao Módulo	<i>Sinking</i> (- comum)	<i>Sourcing</i> (PNP)

Fonte: Autor, 2023.

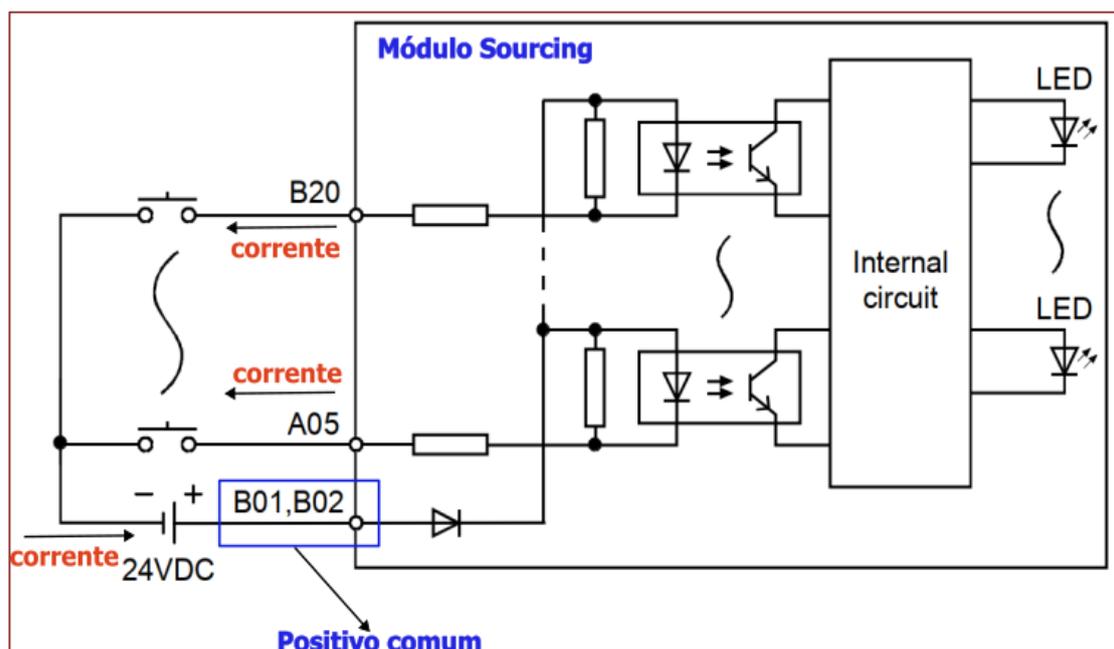
Figura 5.8. Ilustração da classificação de módulos e sensores.



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

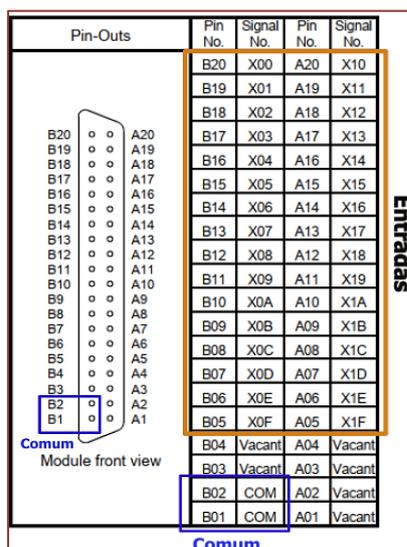
O módulo QX41 possui 32 pontos, método de isolamento fotoacoplador, tensão de entrada de 24 VDC e corrente de entrada nominal de 4 mA. O circuito de conexão externa do módulo é ilustrado na Figura 5.9, que apresenta um módulo positivo comum, portanto do tipo *sourcing*.

Figura 5.9. Circuito de conexão externa QX41



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

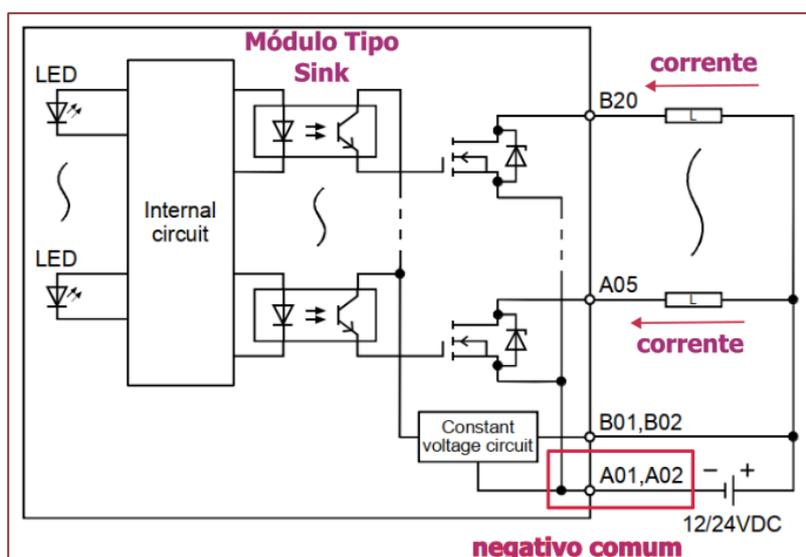
Figura 5.10. Pinagem do módulo QX41



Fonte: Mitsubishi Electric Corporation, 2004).

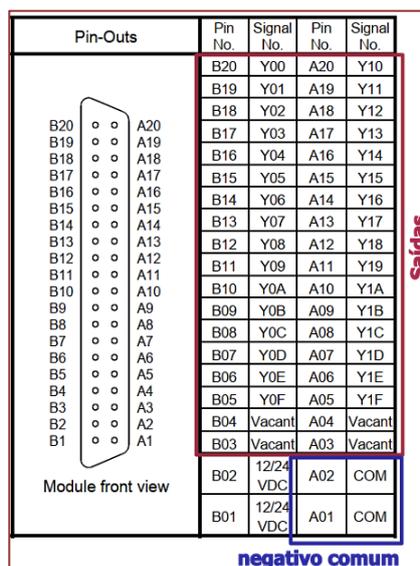
O módulo de saída digital QY41P apresenta 32 pontos de saída e utiliza o método de isolamento fotoacoplador para assegurar confiabilidade operacional. Com uma faixa de tensão nominal de 12 a 24 V DC, o dispositivo suporta uma máxima corrente de carga de 0.1 A por ponto e 2 A por comum, proporcionando versatilidade em aplicações industriais. O tempo de resposta é de 1 ms, garantindo uma atuação rápida e precisa. O módulo incorpora um supressor de surto com diodo Zener para proteção contra picos de tensão. Possui resistência de isolamento de 10 MΩ e utiliza o conector A6CON2. Como pode ser observado na Figura 5.11, o módulo QY41P possui negativo comum, portanto é um módulo do tipo sink.

Figura 5.11. Circuito de conexão externa QY41P



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

Figura 5.12. Pinagem módulo QY41P



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

Figura 5.13. Módulo de entrada digital QX41 e módulo de saída digital QY41P.



Fonte: Autor, 2023.

Como pode ser observado na Figura 5.11, o módulo QY41P tem o menor potencial da fonte de 12/24 VDC conectado nos terminais comuns A01 e A02, caracterizando assim um módulo negativo comum, ou seja, um módulo do tipo *sink*.

A codificação das entradas e saídas dos módulos na unidade base obedece a uma codificação em representação hexadecimal, como mostrada na Figura 5.14.

Figura 5.14. Codificação das entradas/saídas em hexadecimal

		0	1	2	3	4	5	6	7	← Slot No.
HCO	Power supply module	00	20	40	60	80	A0	C0	E0	
	PCU	to 1F	to 3F	to 5F	to 7F	to 9F	to BF	to DF	to FF	

Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

5.3. PROCESSO AVALIATIVO

Nesta disciplina, a avaliação teve como foco a análise qualitativa do progresso na assimilação de novos conceitos e no aprimoramento de habilidades por parte dos alunos, ao longo do ciclo de criação de soluções utilizando o *Design Thinking*. Além disso, incluiu-se a avaliação da execução de planos de solução fundamentados na abordagem de aprendizagem baseada em projeto.

Ao longo da disciplina, três grupos de cinco alunos foram expostos a situações-problema (projetos parciais) de programação em CLP que tinham como objetivo adquirir conhecimentos teóricos e habilidades práticas de programação em CLP. Estes alunos foram avaliados quanto ao engajamento; pesquisa e ideação; desenvolvimento e comunicação. Cada um destes indicadores foi qualificado como: superou as expectativas; atendeu às expectativas; precisa melhorar e inadequado.

Os seguintes projetos parciais foram desenvolvidos durante a disciplina:

- **Projeto parcial 1:** aquisição de habilidades de programação nas linguagens *ladder*, diagrama de blocos e texto estruturado utilizando o software GXWorks 2, a Figura 5.15 retrata este momento.

Figura 5.15. Aluno programando um CLP utilizando bloco de função (Projeto Parcial 1).



Fonte: Autor, 2023.

- **Projeto parcial 2:** simulação de projetos de automação por meio da integração do software de simulação FactoryIO e o GXWorks 2, a Figura 5.16 retrata este momento.

Figura 5.16. Alunos automatizando um ambiente simulado no software FactoryIO integrado ao GXWorks 2 (Projeto Parcial 2).



Fonte: Autor, 2023.

- **Projeto parcial 3:** integração do CLP da Mitsubishi Série Q ao sistema de visão Cognex, a Figura 5.17 retrata este momento.

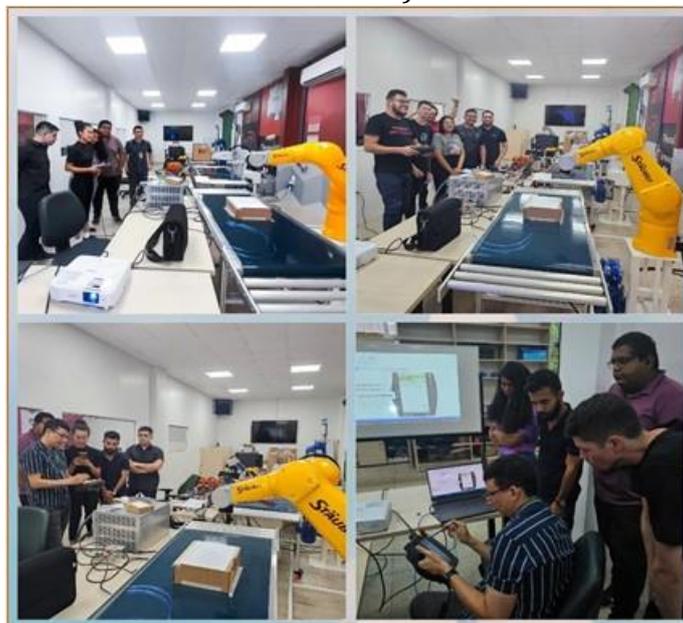
Figura 5.17. Alunos integrando sistema de visão Cognex ao CLP Mitsubishi utilizando os softwares In-Sight Explore e GXWorks 2 (Projeto Parcial 3).



Fonte: Autor, 2023.

- **Projeto parcial 4:** integração do CLP da Mitsubishi Série Q à controladora CS9 do robô Staubli TX90, a Figura 5.18 retrata este momento.

Figura 5.18. Alunos participando da integração do robô Staubli TX90 ao CLP (Projeto Parcial 4).



Fonte: Autor, 2023.

5.4. RESULTADOS DA DISCIPLINA

Os resultados alcançados pela disciplina de Programação de CLP foram:

- **Compreensão dos Conceitos Fundamentais:** Os alunos demonstraram uma compreensão sólida dos conceitos fundamentais relacionados à programação de Controladores Lógicos Programáveis (CLPs), abrangendo desde a teoria até a aplicação prática em ambientes industriais.
- **Solução de Problemas em Ambientes Simulados:** Os alunos foram capazes de aplicar os conhecimentos adquiridos na solução de problemas em ambientes simulados, simulando situações práticas com o software *Factory IO* e validando suas lógicas de controle antes da implementação em ambientes industriais reais.
- **Capacidade de Integração de Sistemas:** A disciplina culminou na capacidade dos alunos de integrar diferentes sistemas, como CLPs, sistemas de visão, e robôs industriais, evidenciando uma compreensão holística da automação industrial. Essa habilidade é crucial para a implementação bem-sucedida de soluções abrangentes em contextos industriais diversificados.
- **Habilidade Avançada em Programação de CLP:** Os alunos desenvolveram habilidades avançadas em programação de Controladores Lógicos Programáveis (CLPs), abordando linguagens como *ladder*, *Functional Block Diagram (FBD)*, e Texto Estruturado. Essa competência inclui a capacidade de criar, modificar e depurar programas complexos, garantindo uma aplicação eficaz dos princípios de automação aprendidos durante o curso.

- **Habilidade de Aprendizagem Ativa:** Os alunos demonstraram habilidades significativas em aprendizagem ativa, envolvendo-se proativamente em atividades práticas, discussões em grupo e resolução de problemas. A capacidade de aplicar os conceitos teóricos em situações práticas, bem como a participação ativa em simulações e integração de tecnologias, reflete uma abordagem de aprendizagem ativa que prepara os estudantes para desafios dinâmicos e em constante evolução na área de automação industrial.

A disciplina de Programação em CLP proporcionou uma experiência de aprendizado abrangente e altamente eficaz para os alunos. A estrutura do curso, que incluiu uma combinação equilibrada de teoria e prática, permitiu que os estudantes desenvolvessem uma compreensão profunda dos conceitos fundamentais da programação de Controladores Lógicos Programáveis. As atividades práticas, desde a programação em linguagem ladder até a integração de tecnologias avançadas como visão computacional e robótica industrial, desempenharam um papel crucial no fortalecimento das habilidades técnicas dos alunos. A utilização de metodologias ativas de aprendizagem proporcionou uma dinâmica envolvente, incentivando a participação ativa dos alunos e promovendo uma aplicação prática dos conhecimentos adquiridos.

Além disso, a diversidade de tópicos, que abrangeu desde programação de CLPs até a programação de robôs industriais e sistemas de visão, refletiu a natureza abrangente da automação industrial. A habilidade dos alunos em integrar esses conceitos e tecnologias diversos demonstra não apenas uma compreensão pontual, mas uma visão integrada e holística da automação. A incorporação de habilidades de aprendizagem ativa, juntamente com a aplicação prática dos conhecimentos, solidificou a preparação dos alunos para enfrentar desafios complexos e se destacar em ambientes industriais modernos.

REFERÊNCIAS

- [1] DA SILVA, Edilson Alfredo. Introdução às linguagens de programação para CLP. Blucher, 2016.
- [2] MALEKAR, Avinash. Learn everything about PLC programming: Practical lessons on Allen-Bradley, Siemens, and Mitsubishi PLC with real world examples. Independently published, 2021.
- [3] Mitsubishi Electric Corporation. QCPU User's Manual (Hardware Design, Maintenance and Inspection). Manual number SH(NA)-080483ENG-AZ(2309)MEE, 2004.
- [4] Mitsubishi Electric Corporation. I/O Module Type Building Block – User's Manual. Manual number SH(NA)-080042-T(0810) MEE, 2000.
- [5] PETRUZELLA, Frank D. Controladores lógicos programáveis. AMGH Editora, 2014.
- [6] PRUDENTE, Francesco. Automação Industrial PLC - Teoria e Aplicações - Curso Básico. Grupo Gen-LTC, 2015.
- [7] QUEIROZ-NETO, J. P., SALES, D. C., PINHEIRO, H. S., NETO, B. O. Using modern pedagogical tools to improve learning in technological contents. 2015 IEEE Frontiers in Education Conference (FIE), El Paso, TX, 2015, pp. 1-8, doi: 10.1109/FIE.2015.7344383.
- [8] WHITE, M. T. Mastering PLC Programming: The software engineering survival guide to automation programming. Packt Publishing, 2023.

6.1. CARACTERIZAÇÃO DA DISCIPLINA

Na disciplina de Programação de IHM, antes de começar a parte mais prática e de programação, é feita uma introdução dos conceitos, definições e fluxos de projeto que podem ser utilizados na Programação das IHMs. Contudo é na execução das práticas e dos projetos que muitos desses conceitos vão sendo mais bem entendidos e absorvidos pelos alunos, lhe dando um sentimento maior das opções de como um determinado projeto pode ser feito, que em língua inglesa usamos o termo *feeling* para tal situação.

Nesta disciplina, como nas demais, foi feita com a integração das metodologias de Aprendizagem Baseada em Projetos (PBL) e *Design Thinking* (DT), criando métricas ou indicadores para verificar como foi o rendimento desses alunos com essa integração.

6.1.1. EMENTA

Conceitos, funcionamento e histórico das IHMs. Apresentação geral do *hardware* e *Softwares* do fabricante, e seus principais manuais de criação de projeto no *software* do fabricante e suas configurações básicas, como: edição, compilação, simulação, programação e monitoramento da IHM. Simulação da IHM em conjunto com a simulação do CLP (esse último usando programação em Ladder e em Texto Estruturado). Programar um projeto na IHM real, comunicação, manuseio e ensaios. Construção das Telas da IHM. Inserindo e configurando elementos de Imagens, Lâmpadas e Botões *etc.* Inserindo e configurando elementos de display numérico e texto, data e hora, receita, alarmes e segurança. Animação e Movimento de Objetos na IHM. Comunicação da IHM com CLP ou outros periféricos via padrão Ethernet dentre outros protocolos de comunicação industrial.

6.1.2. OBJETIVOS

Desenvolver no discente os conceitos básicos do que é uma Interface Homem Máquina (IHM), suas aplicações e limitações.

Desenvolver sua habilidade para fazer projetos em IHMs, mostrando como o projeto da IHM pode está vinculado diretamente com um Controlador Lógico Programável (CLP) dentre outras opções.

¹ Doutor em Engenharia Elétrica. Professor da disciplina de Programação em IHM. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM).

6.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 20h

Carga Horária Prática: 40h

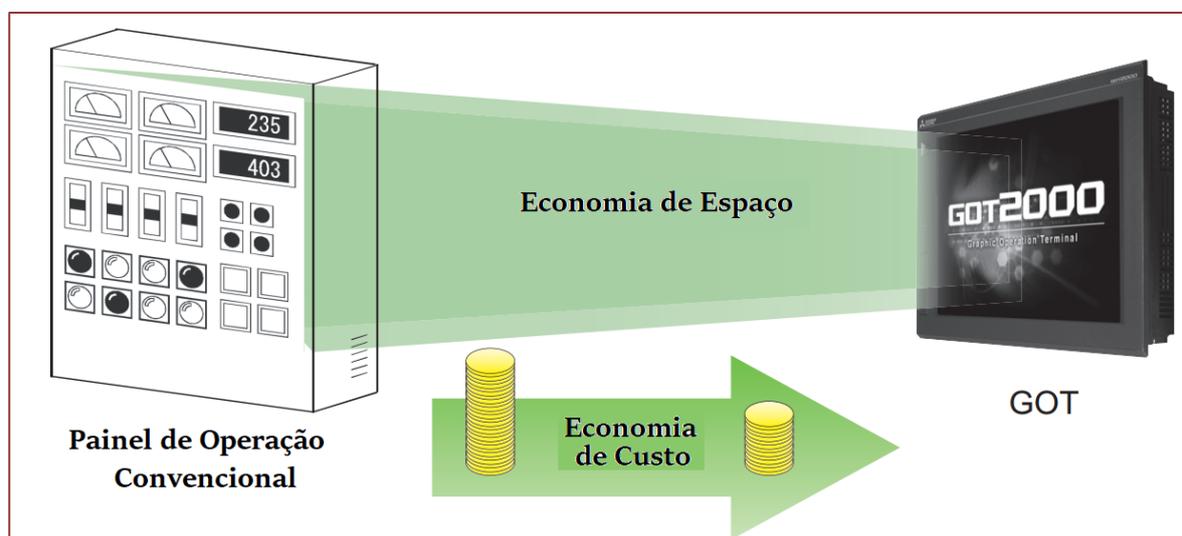
Total: 60h

6.2. CONTEÚDOS

6.2.1. O QUE É UMA IHM

A rigor poderíamos dizer que qualquer mecanismo que faz uma integração entre um operador (“Homem”) com uma máquina é uma Interface Homem Máquina, como exemplo um painel com botões, chaves, sinaleiras, *displays* e *etc.* Contudo no ambiente industrial contemporâneo uma IHM é muitas vezes conhecida como um *display* gráfico com *touch*, capaz de ser programado para criar de forma gráfica todos os dispositivos contidos em um painel de operação convencional e outros mais, como gráficos e animações. Talvez o termo IHM não seja o mais correto para definir essas IHMs modernas, alguns fabricantes costumam a dar seus próprios nomes para suas IHMs (mas claro que isso é jargão de empresa), como exemplo a MITSUBISHI usa o termo GOT para suas IHMs. GOT é um acrônimo para *Graphic Operation Terminal*, que fazendo uma tradução para língua portuguesa, ficaria algo como Terminal de Operação Gráfico (TOG). Na Figura 6.1 é possível comparar visualmente um painel de operação convencional com uma IHM da MITSUBISHI, uma GOT da sua linha GOT2000.

Figura 6.1 – Comparativo entre uma IHM antiga e uma moderna.



Fonte: Editado de MITSUBISHI 04, 2018.

Na própria figura 6.1 já é possível observar alguns benefícios ao usar uma IHM moderna, no caso uma GOT. Contudo vamos listar em seguida algumas dessas vantagens (MITSUBISHI 04, 2018):

- 1) **Redução do painel de operação:** O GOT substitui chaves e lâmpadas físicas por gráficas, economizando pontos de E/S no PLC e reduzindo o tamanho do painel de operação.
- 2) **Redução dos custos de fiação:** A interface gráfica do usuário do GOT elimina o tempo e o custo de cabeamento de componentes dentro da operação painel.
- 3) **Padronização de painéis de operação:** Os GOTs são flexíveis, permitindo construir ou modificar projetos sem alterar o programa PLC ou a fiação do sistema, o que facilita a padronização dos painéis de operação.
- 4) **IHM (Interface Homem-Máquina) de valor agregado:** GOT também pode exibir gráficos, textos e alarmes, além de interruptores e lâmpadas, contribuindo com valor além de todo o equipamento.

Como nesse curso as práticas e os projetos foram realizados utilizando a IHM da MITSUBISHI, vamos focar nossas explicações para IHM desse fabricante, ficando claro que um caminho muito parecido poderia ser feito para outros fabricantes.

Embora seja possível utilizar uma IHM sem o uso de um CLP (Controlador Lógico Programável), na grande maioria dos casos as IHM trabalham em conjunto com um CLP. Vale destacar que muitos fabricantes de IHMs são também fabricantes de CLPs, de tal forma que a integração entre a IHM e o CLP do mesmo fabricante costuma ser mais simples, do que se os fabricantes forem diferentes, neste nosso curso usamos tanto a IHM quando o CLP do mesmo fabricante, a MITSUBISHI, o curso como um todo foi idealizado de forma que a disciplina de IHM viesse *a posteriori* ao curso de CLP. Por tanto os alunos já sabem programação LADDER e Texto Estruturado.

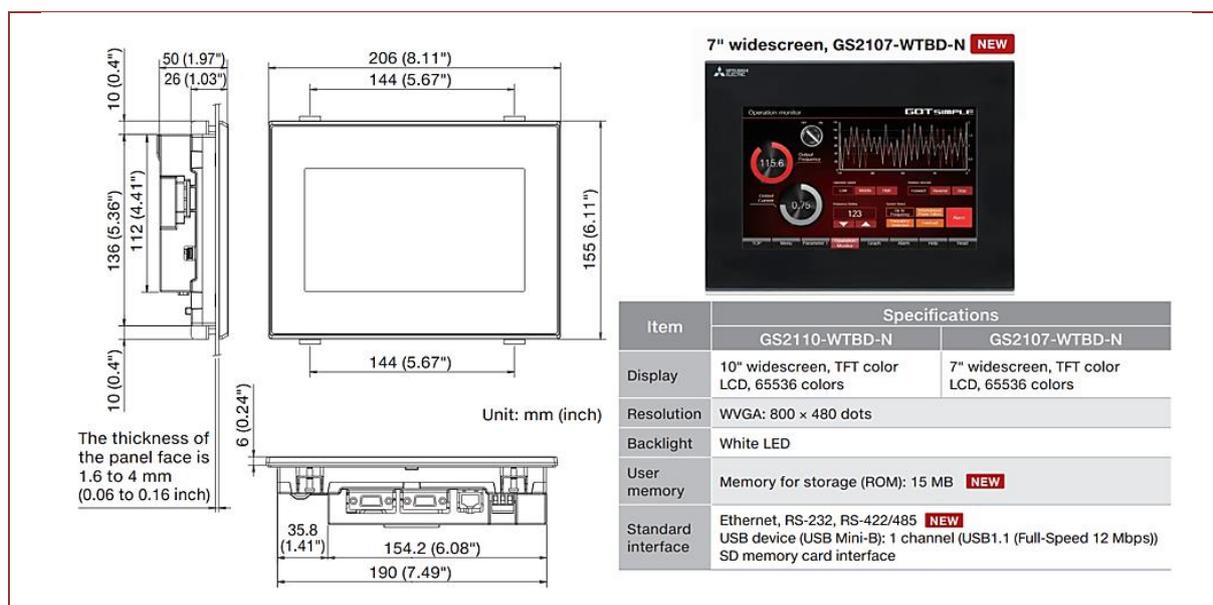
Uma pergunta frequente que é feita pelos alunos é sobre a possibilidade da troca de uma IHM por um monitor de computador ou algo do tipo, a resposta é que se o ambiente não for hostil dentro de determinadas situações pode ser possível, contudo o ambiente fabril é sempre considerado hostil, assim as IHMs são projetadas para trabalharem várias horas seguidas, com diversos níveis de poeira, humidades, ruídos variações de tensão *etc*, esses são mais alguns diferenciais para usamos essas IHMs projetadas para esse ambiente fabril, é mais ou menos a diferença de um computador pessoal comum e um computador industrial.

6.2.2. HISTÓRICO DAS IHM

Definir quando começou e quem foi o primeiro fabricante das ditas IHMs modernas, é complicado, pois deveríamos definir exatamente o que é uma IHM moderna, pois as primeiras IHM não tinham *touch* nas suas telas, os meios de comunicações eram mais simples, ou seja, uma série de tecnologias foram e vem sendo incorporadas ano a ano. Dessa forma, como estamos trabalhando com as IHMs da MITSUBISHI (Os GOTs), vamos falar brevemente do histórico desse fabricante.

Uma das primeiras IHMs da MITSUBISHI foi à da série A64 pelos anos de 1990. Era bem rudimentar sobre vários aspectos tecnológicos atuais. Na Figura 6.2 é possível observar um gráfico dos modelos da MITSUBISHI ao longo dos anos.

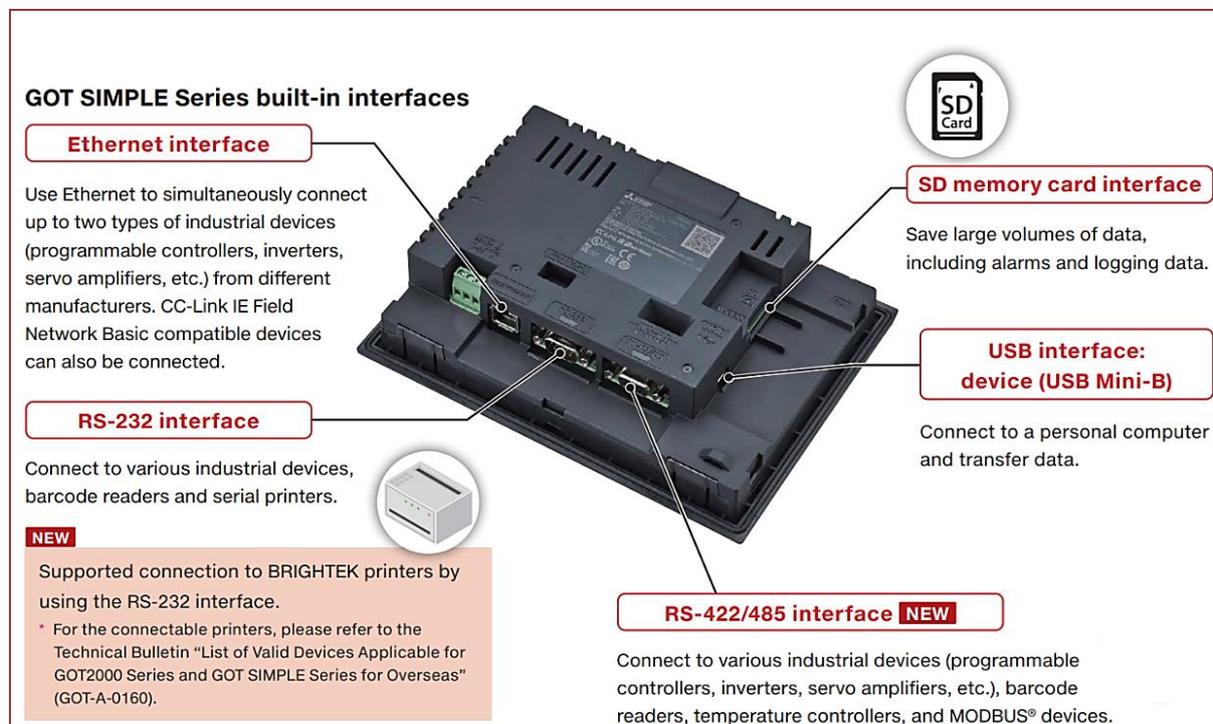
Figura 6.3 – Detalhes da IHM GOT2107-WTBD-N.



Fonte: Editado de MITSUBISHI 03, 2021.

Na Figura 6.4 tem-se a imagem da parte traseira da IHM GOT2107-WTBD-N, onde são localizadas suas interfaces.

Figura 6.4 – Detalhes da IHM GOT2107-WTBD-N, parte traseira.



Fonte: MITSUBISHI 03, 2021.

6.2.4. TIPOS DE CONEXÃO – IHM (GOT) E CLP

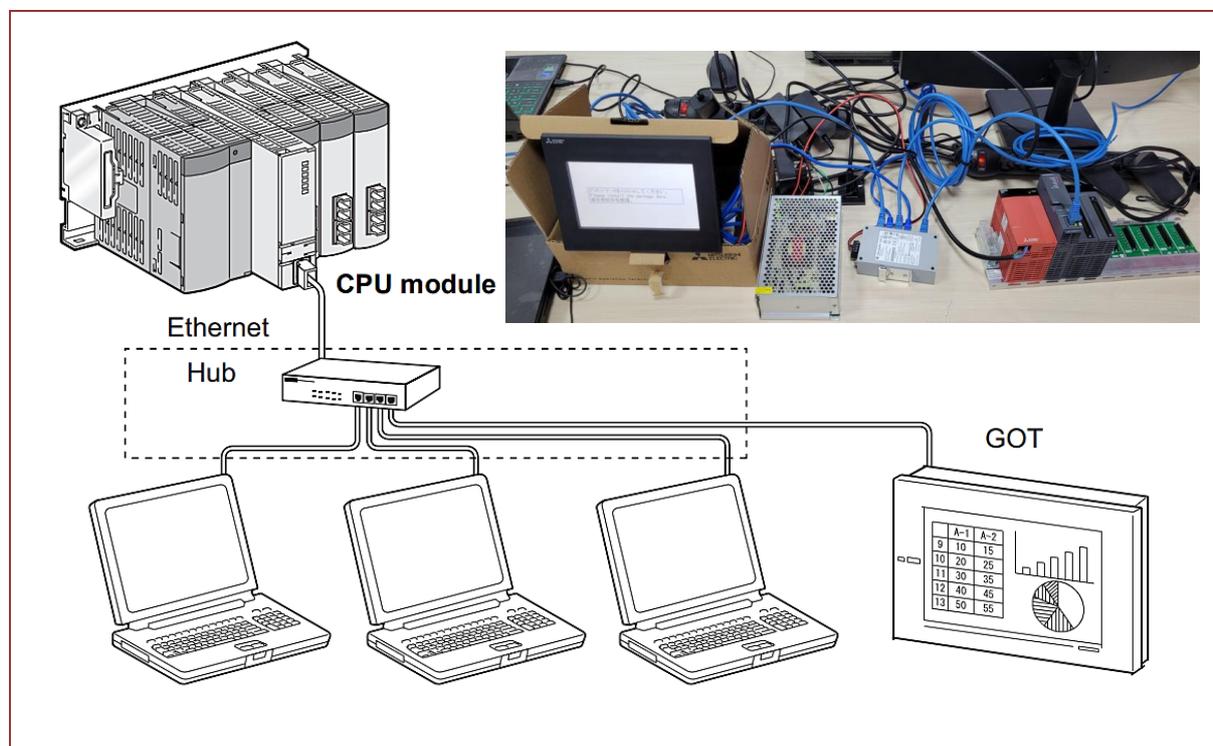
Existe uma grande variedade de tipos diferentes de conexões (topologias) entre uma IHM e um CLP, uma grande variedade dessas topologias é apresentada aos alunos, contudo aqui vamos destacar apenas a que foi mais usada nas práticas dessa disciplina.

Na figura 6.5 é mostrada essa conexão na topologia estrela. Essas conexões podem usar diversos tipos de cabos e conectores, bem como vários tipos de protocolos de comunicação industriais, contudo foi usado preferencialmente o padrão Ethernet e uma variação do protocolo CC-Link que usa o padrão Ethernet.

O desenvolvimento do projeto é feito em duas etapas, a primeira na bancada do laboratório LRI e a segunda etapa na célula robótica.

Primeira etapa: Inicialmente é ligado ao HUB (ou Switch Não Gerenciável) que fica nas bancadas do laboratório uma IHM e um CLP, depois os computadores para o desenvolvimento do projeto, programação da IHM e do CLP, a quantidade de computadores depende da quantidade de alunos por bancada. Conforme pode ser visto na figura 3.5. Assim todos os alunos daquela bancada podem compartilhar o uso da IHM e do CLP, podendo fazer a edição, compilação, simulação, programação, monitoramento e ensaios dos projetos desenvolvidos. É nessa etapa que feita boa parte das explicações do professor usando vários recursos didáticos como projetor de imagem (*86atashow*), lousa branca dentre outros. Vale destacar que os alunos da mesma bancada fazem parte do mesmo projeto final a ser desenvolvido, fomentando assim a interação entre os mesmos.

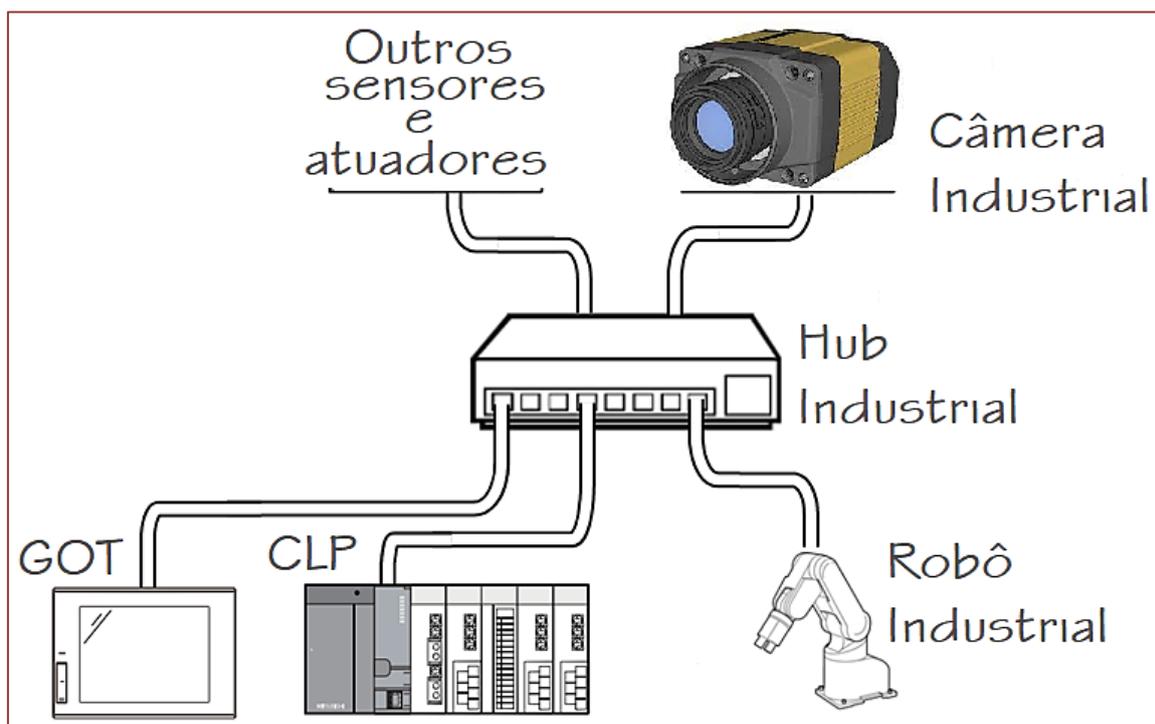
Figura 6.5 – Primeira Etapa: Desenvolvimento nas bancadas.



Fonte: Editado de MITSUBISHI 04, 2018.

Segunda etapa: Essa já na célula robótica os alunos vão fazer a integração do CLP e da IHM (GOT), com outros dispositivos como: o robô industrial, câmera industrial, e outros sensores e atuadores, conforme pode ser visto na figura 6.6.

Figura 6.6 – Segunda Etapa: Desenvolvimento nas Células Robóticas.



Fonte: Editado de MITSUBISHI 04, 2018.

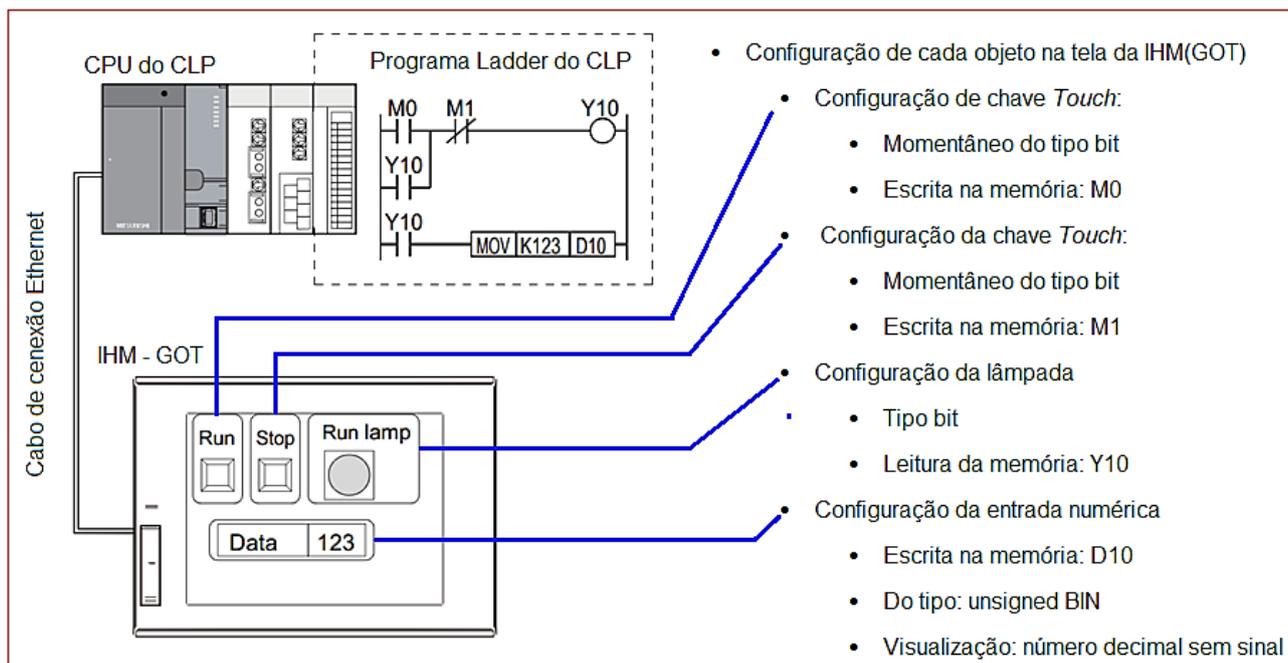
É claro que muitos sensores e atuadores podem ser ligados diretamente nos cartões (módulos) entradas e saídas do CLP e comandados pela IHM, inclusive os próprios robôs por meio das suas controladoras podem fazer essa comunicação direta mais simples. Como exemplo de sensores ligados ao conjunto CLP/IHM podemos destacar: câmera industrial da COGNEX, sensor de presença capacitivo e indutivo (tanto analógico e digital), sensor de pressão *etc.* Os principais atuadores na célula robótica além do robô são as ferramentas ligadas a ele como ventosas e garras, essas ferramentas são acionadas eletricamente ou por meio de válvulas eletropneumáticas, e o motor da esteira controlado diretamente por um inversor de frequência ligado ao conjunto CLP/IHM.

6.2.5. PROGRAMAÇÃO E DESENVOLVIMENTO DA IHM (GOT)

Com o *software* de desenvolvimento GT Designer3, é feita a construção de telas na IHM da empresa MITSUBISHI, bem como as figuras do painel de exibição chamadas objetos, como interruptor de toque, lâmpada e display numérico, *etc.* A IHM (GOT) permite a leitura e gravação em dispositivos de memória internos do CLP (dos tipos *bit* e *word*) e são associados com os vários objetos criados nas telas da IHM(GOT) como interruptores, lâmpadas, entrada/saídas numérica e em texto, exibição de mensagens, *etc.*

A seguir é feita uma pequena demonstração de um projeto do CLP/IHM. São criados os objetos em uma tela da IHM, chamados Run, Stop, Run lamp e Data. Esses objetos são associados a elementos de memória do CLP e configurados conforme pode ser visualizado na Figura 6.7. Observe que alguns desses objetos são de escrita outros de leitura dessas memórias internas do CLP (cada tipo desses de memória do CLP a MITSUBISHI chama de *device*).

Figura 6.7 – Demonstração de Funcionamento de um Projeto CLP/IHM.



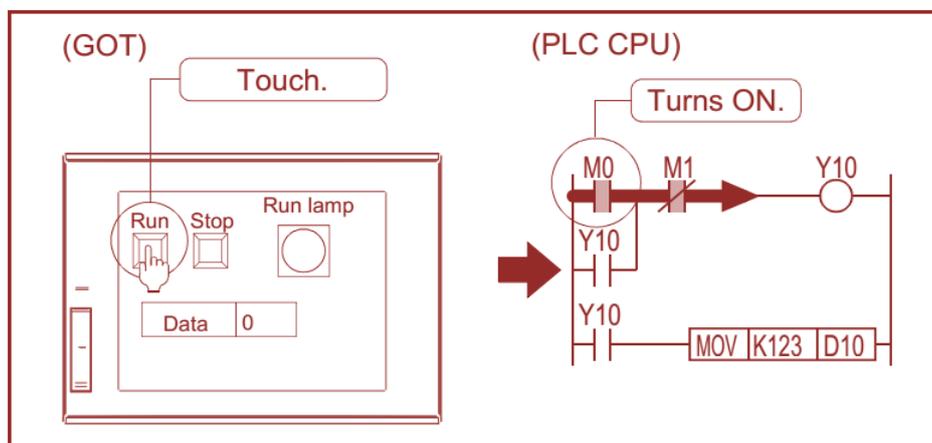
Fonte: Editado de MITSUBISHI 04, 2018.

É possível visualizar também na figura 6.7 um pequeno código em Ladder sendo executado na CPU do CLP.

Descrição das ações:

- 1) Acionamento da chave Run: Enquanto o botão (chave) de toque "Run" no GOT estiver sendo tocado, o contato do tipo bit "M0" no programa Ladder do CLP estará fechado, conforme é possível visualizar na Figura 6.8.

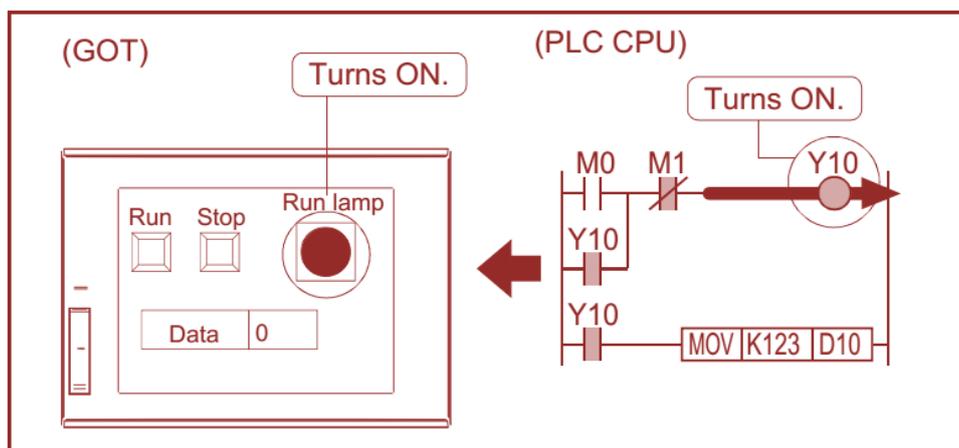
Figura 6.8 – Acionamento da chave Run.



Fonte: MITSUBISHI 04, 2018.

- 2) Saída Y10 ligada: A consequência imediata após o contato "M0" ser fechado, é o acionamento do dispositivo de saída "Y10", conforme é possível visualizar na Figura 6.9. Além disso, a figura Run Lamp na IHM(GOT) é modificada, ilustrando uma lâmpada ligada (em ON), pois ela foi associada anteriormente ao dispositivo de saída "Y10". Ou seja, se a saída Y10 do CLP é ativada a figura Run Lamp na IHM é modificada, ilustrando uma lâmpada ligada.

Figura 6.9 – Saída Y10 do CLP é ligada, Lâmpada da IHM é acesa e o Selo é Fechado.

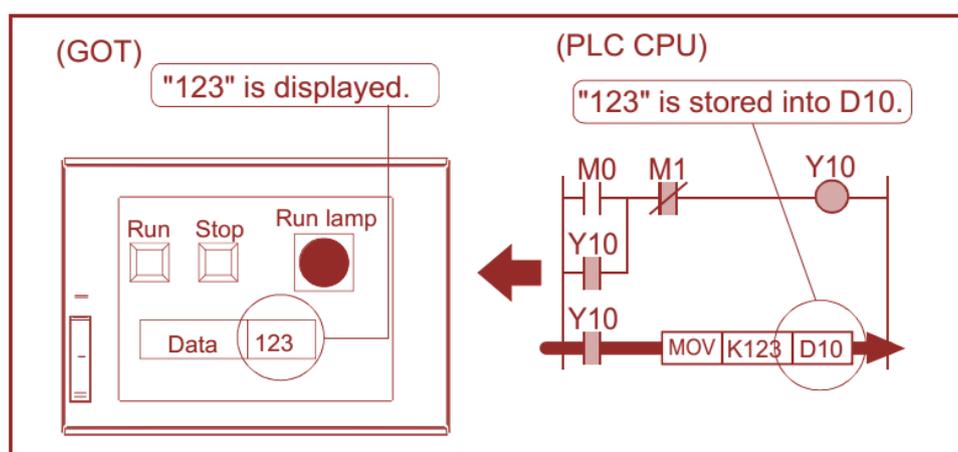


Fonte: MITSUBISHI 04, 2018.

- 3) Selo Acionado: Quando a chave Run (na IHM(GOT)) não é mais acionada a saída Y10 continua sendo ativo pelo contato de selo Y10, a saída Y10 só pode ser desligada agora abrindo o contato M1(que é normalmente fechado). Não confundir a saída Y10 com o contato Y10, embora os dois estejam associados são coisas distintas. Conforme é possível visualizar também na figura 6.9

- 4) Ativação da instrução MOV: Outra consequência imediata do acionamento da saída Y10 é a ativação da instrução MOV por meio do contato Y10 na última linha do programa em Ladder. Essa instrução movimenta (Escreve, Grava) no dispositivo de memória interna do CLP D10 (que é do tipo word) o valor em decimal cento e vinte e três (123). É claro que 123 também é exibido no display numérico Data da IHM(GOT) no qual o dispositivo monitor foi definido para o dispositivo de palavra "D10". Conforme é possível visualizar também na figura 6.10

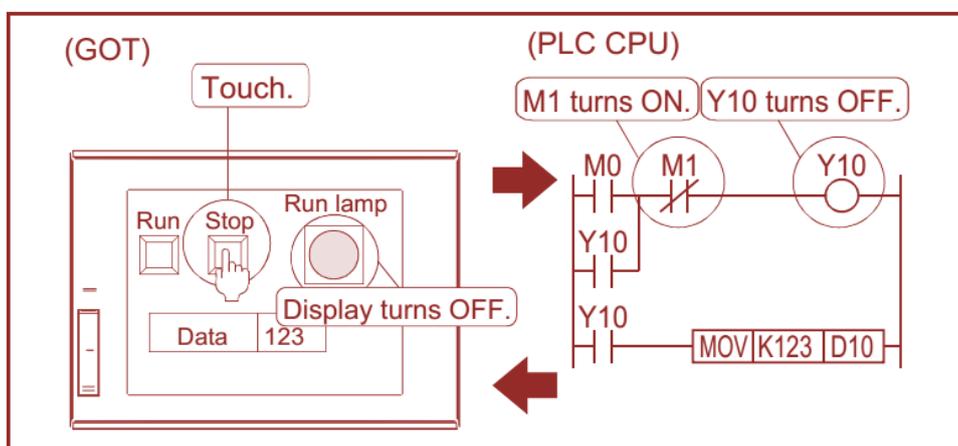
Figura 6.10 – Ativação da Instrução MOV.



Fonte: MITSUBISHI 04, 2018.

- 5) Acionamento do Stop: Enquanto a chave de toque "Stop" no GOT não for tocada, o dispositivo de bit do PLC "M1" fica ligado. Quando é tocado na IHM o botão Stop o dispositivo do PLC "Y10" é DESLIGADO e a Lâmpada de funcionamento (Run Lamp) é DESLIGADA. Conforme é possível visualizar também na figura 6.11.

Figura 6.11 – Acionamento do Stop.



Fonte: MITSUBISHI 04, 2018.

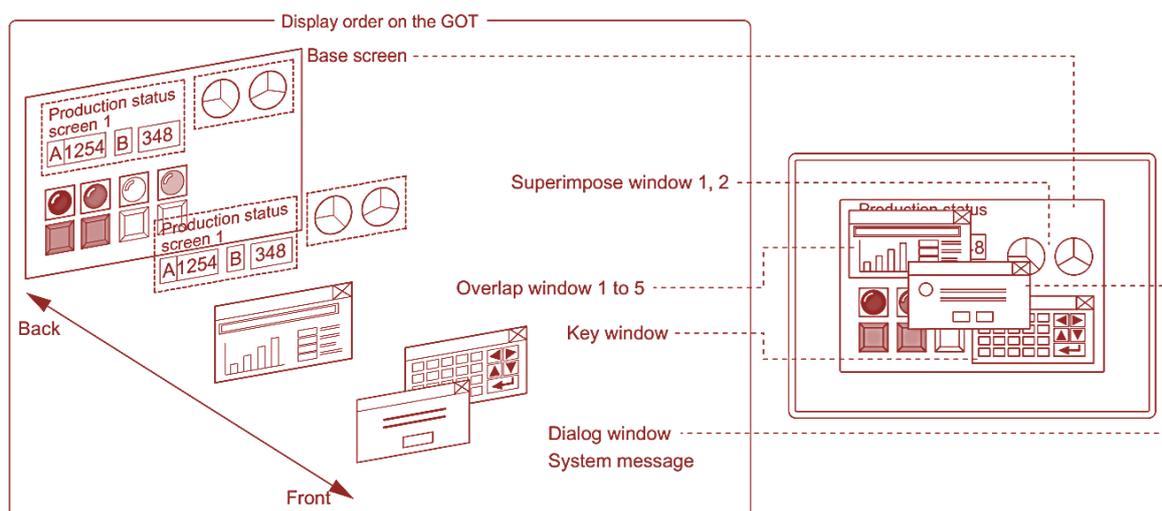
Antes de falarmos sobre alguns objetos como *Switch*(chaves) e *Lamp*(lâmpadas), vamos explicar sobre as Telas (Screens) da IHM.

As telas da IHM podem ser do tipo Base Screen e Window Screen. As Base Screen são caracterizadas por obrigatoriamente ocupar toda a área do Display da IHM(GOT), e é possível e bem comum criar várias dessas telas, com botões (Switch) criados e dedicados para navegação entre essas telas. As Window Screen são caracterizadas por não obrigatoriamente ocupar toda a área do Display da IHM, podem ser de 3 tipos, a saber:

- **Overlap Window:** São pop-up's que possui um X no canto superior direito, usado para fechar a janela. Podendo criar objetos como botões e lâmpadas como em feito em qualquer tela.
- **Superimpose Window:** São pop-up's (semelhante ao Overlap Window) sem o X, devendo ser criados pelo programador para ser possível fechar essa janela, como não possui o X, esses mecanismos podem ser feito no CLP ou mesmo na IHM. Assim geralmente é criado alguns botões nessa janela impondo o operador a escolher uma das opções.
- **Dialog Window:** São pop-up's (semelhante ao Overlap Window) que serve apenas para dar um aviso (informação).

Essas telas e Janelas tem uma ordem de apresentação no Display da IHM conforme pode ser observado na Figura 6.12

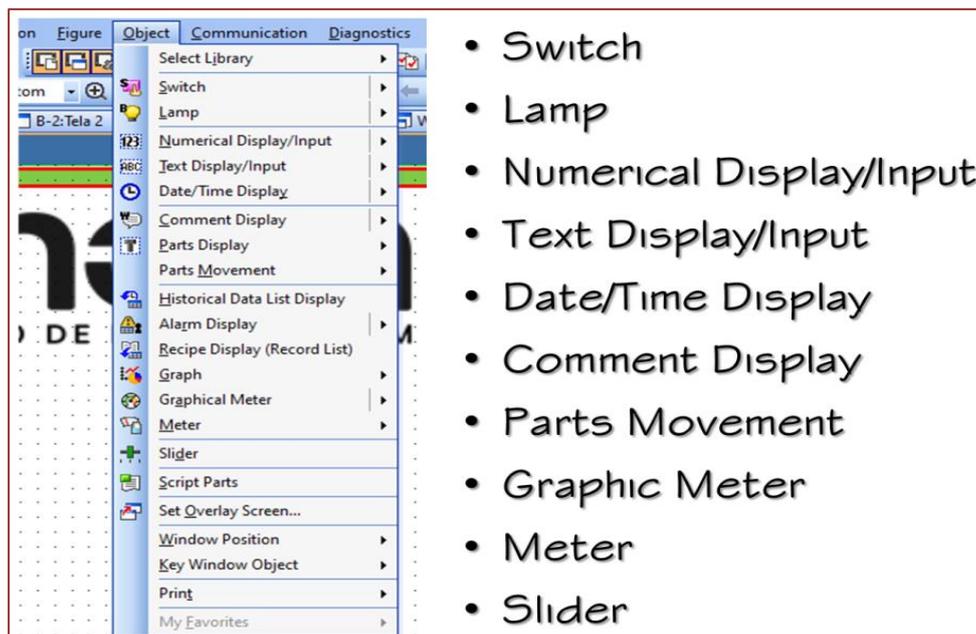
Figura 6.12 – Ordem de Apresentação das Telas e Janelas na IHM GOT2000.



Fonte: MITSUBISHI 02, 2015.

Em seguida é mostrado na Figura 6.13, uma lista dos nomes dos principais objetos para criação das telas da IHM (GOT) e onde eles podem ser localizados no *Software GT Designer3* da Empresa MITSUBISHI.

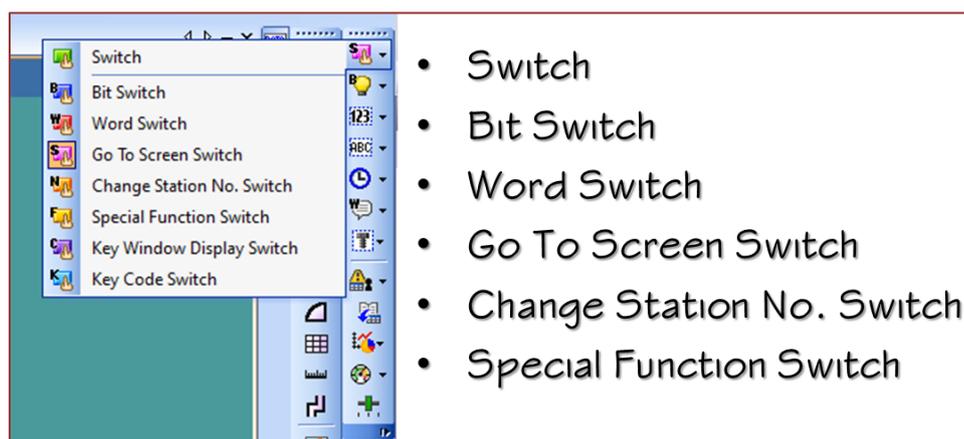
Figura 6.13 – Principais objetos usados na criação das telas da IHM.



Fonte: O autor, 2023.

Cada um desses objetos possui uma série de variações, como exemplo é mostrado na figura 6.14 as variações do objeto Switch(chave).

Figura 6.14 – Variações, ou tipos distintos do objeto Switch.



Fonte: O autor, 2023.

É claro que existe uma grande variedade de outras configurações (funções) fora os objetos, Telas e Janelas que podem ser utilizadas em uma programação de uma IHM, tais como: Alarmes, Senhas, Receitas, Arquivos de *logs*, Animações *etc.*

6.3. PROCESSO AVALIATIVO

As avaliações foram realizadas observando o cumprimento das atividades (pequenos projetos de fixação) durante as aulas e ao final com apresentação do projeto da IHM de cada equipe, muitas dessas atividades são feitas em equipe fomentando a integração dos membros de cada grupo.

Um dos propósitos dessas atividades avaliativas é a construção em etapas do seu projeto final do curso. Assim, o critério para correção é baseado principalmente em aspectos qualitativos e de forma secundária os quantitativos, valorizando a criatividade e pontualidade.

REFERÊNCIAS

- [1] MITSUBISHI 01, A. Apostila: Treinamento Básico GOT2000. [S. l.]: MITSUBISHI, 2020.
- [2] MITSUBISHI 02. GT Designer3 (GOT2000) Screen Design Manual. [S. l.: s. n.], 2015.
- [3] MITSUBISHI 03. Manual MITSUBISHI ELECTRIC CORPORATION: Graphic Operation Terminal GOT SIMPLE Series. [S. l.: s. n.], 2021.
- [4] MITSUBISHI 04. Graphic Operation Terminal (GOT2000 Series) Basic Training Textbook (for GT Works3). [S. l.]: MITSUBISHI ELECTRIC CORPORATION, 2018.

7.1. CARACTERIZAÇÃO DA DISCIPLINA

A disciplina de Visão Computacional foi elaborada de forma a possibilitar que todos os alunos, mesmo os que não tinham conhecimento prévio dos conceitos, técnicas, ou embasamento necessário na linguagem de programação utilizada, pudessem compreender, aplicar e codificar cada assunto. O método *hands-on* foi utilizado de forma que, a cada tópico do conteúdo apresentado, os conceitos mais importantes eram detalhados e em seguida era feita a codificação, na qual criávamos passo a passo os programas na linguagem Python, explicando cada comando e funcionalidade abordada com um ou mais exemplos práticos, e cada aluno replicava 100% dos códigos. Após essa etapa, algumas atividades práticas eram indicadas, e os alunos elaboravam os programas e conversavam para refletir onde cada técnica poderia ser aplicada nas possíveis soluções para os projetos. Devido à quantidade de assuntos necessários, a elaboração do conteúdo da disciplina visou a aplicação prática de alguns dos principais conceitos, sem aprofundamento dos fundamentos matemáticos e de álgebra.

7.1.1. EMENTA

Introdução à visão computacional; Fundamentos de processamento de imagens e vídeo e técnicas tradicionais de visão computacional; Aprendizado de máquina e aplicações em visão computacional; Aplicação de bibliotecas e ferramentas para visão computacional; Aplicações industriais; Elaboração de projetos práticos no contexto industrial.

7.1.2. OBJETIVOS

Capacitar os alunos no entendimento dos principais conceitos e na utilização e aplicação de métodos, técnicas e estratégias correlatas com visão computacional e processamento de imagens em ambientes industriais.

Objetivos específicos:

- Apresentar e implementar os principais algoritmos de processamento digital de imagem, de visão computacional e de aprendizagem de máquina utilizados com visão computacional;
- Avaliar os desafios associados à operação em tempo real, considerando fatores de latência, processamento e envio de sinal aos robôs.

¹ Mestre em Ciência da Computação. Professor da disciplina de Visão Computacional. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM).

7.1.3. CARGA HORÁRIA

Carga Horária Teórica: 16 h

Carga Horária Prática: 24 h

7.2. CONTEÚDOS

Essa seção descreve os conteúdos aplicados no decorrer da disciplina.

7.2.1. INTRODUÇÃO À VISÃO COMPUTACIONAL

Visão Computacional é um campo interdisciplinar e refere-se basicamente à criação de programas capazes de interpretar, ou dar sentido ao que imagens representam. A diferença entre visão computacional e processamento de imagens é que as técnicas de processamento de imagens podem ser utilizadas para realizar adequações e melhorias, e a partir disso, extrair características que são importantes para dar significado ao que existe em imagens.

Para que seja possível aplicação da visão computacional, é necessário combinar conhecimentos de várias áreas tais como:

- Processamento de imagem e vídeo: aplicação de técnicas fundamentais e essenciais para que um sinal seja processado e melhorado, possibilitando a extração de informações relevantes para o problema em questão;
- Física: entendimento de conceitos básicos de aquisição de imagens utilizando câmeras, tipos de lente, tipos de iluminação e características importantes e influência da luz na geração de imagens;
- Ciência da Computação: conhecimento e utilização de algoritmos e estruturas de dados adequadas para possibilitar que a aplicação das técnicas seja realizada com um bom desempenho;
- Aprendizado de Máquina: aplicação de técnicas envolvendo modelos que consigam aprender, classificar e auxiliar na tomada de decisão;
- Robótica: utilização do resultado final da interpretação da imagem (tomada de decisão) no envio de comandos para robôs executarem tarefas para os quais foram previamente programados.

Existem algumas divisões na Visão Computacional, relacionadas diretamente com a aplicação para qual uma solução está sendo elaborada. Alguns exemplos que podem ser obtidos a partir da combinação de conhecimentos citados anteriormente são:

- Medições: comparações de tamanho de objetos tendo um modelo prévio como referência, ou confirmações de medidas de objetos, utilizando proporções pré-determinadas e uma tolerância que também pode ser especificada previamente. Exemplos de medição são: identificar distância entre determinados componentes presentes em determinado circuito, ou o tamanho / espessura de algum componente ou produto produzido;
- Contagem: identificação do número (quantidade) de partes, ou a quantidade de determinada característica de determinadas partes, com por exemplo: confirmar se a quantidade de furos identificados em determinado componente está conforme

o esperado; contar o número de determinados itens em uma peça, e dessa forma, pode-se identificar quantos itens estão ausentes;

- **Localização:** indicação de posição ou orientação de uma peça ou componente. Como exemplo, pode ser citado: detecção de componentes desalinhados ou mal encaixados; possibilidade de reconhecimento de determinadas características, mesmo que os objetos estejam desalinhados, como por exemplo, realizar o reconhecimento de um padrão;
- **Leitura de padrões:** leitura de códigos numéricos ou alfanuméricos através de reconhecimento ótico de caracteres (OCR) e leitura de dados através de códigos de barras e *QR-Codes*. A partir da identificação do código, pode-se definir os tipos de padrão esperados do produto em produção.
- **Detecção de anomalias:** a partir de um ou mais modelos pré-determinados, pode-se detectar se o objeto ou produto contém alguma não conformidade, como por exemplo, arranhões, cores diferentes do padrão esperado, partes quebradas, e inúmeras outras situações.

É normal que determinadas aplicações de visão computacional necessitem da combinação de vários desses itens relacionados, dependendo da complexidade do problema e quantidades de etapas necessárias para automatizar processos de identificação em imagens ou vídeo.

Esses processos são muito presentes em indústrias de vários tipos, e o objetivo da aplicação de técnicas automatizadas é basicamente a otimização de etapas ineficientes, redução de defeitos e rastreamento, muitas vezes realizadas por seres humanos através de inspeção visual. Por mais eficientes e eficazes que sejam em atividades repetitivas e padronizadas, não podem ser comparados com as máquinas e algoritmos. Estes possuem diversas vantagens, tais como: trabalho de forma ininterrupta sem paralização por cansaço, férias, doenças; maior velocidade, que pode ser adaptada com a aplicação de equipamentos tecnologicamente cada vez mais avançados; e consistência, sem subjetividade para a tomada de decisão.

7.2.2. FUNDAMENTOS DE PROCESSAMENTO DE IMAGENS E VÍDEO E TÉCNICAS TRADICIONAIS DE VISÃO COMPUTACIONAL

O passo inicialmente necessário para abordar tópicos de visão computacional é entender o processo de aquisição de uma imagem, quais características são importantes considerar, e por fim, quais são as técnicas que podem ser aplicadas e suas finalidades.

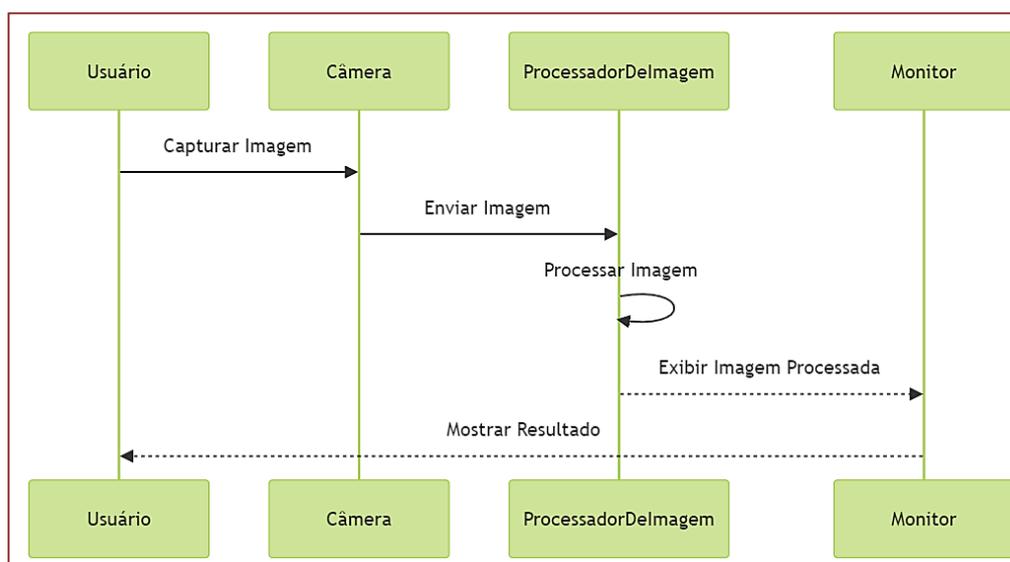
Uma imagem é uma visão bidimensional (2D) gerada a partir de uma cena real em um mundo tridimensional. É formada a partir da captura (aquisição), processada por uma câmera. Uma vez que é feita a aquisição da imagem de um objeto ou cena, é gerado um conjunto de pixels com intensidades variadas. A quantidade de pixels é relacionada com a resolução da captura (quantidade de linhas x quantidade de colunas), e a intensidade de cada pixel é resultante da intensidade de luz refletida nos objetos capturados.

Objetos existentes têm como resultados imagens diferentes, dependendo da perspectiva de captura. Além disso, existem fatores não relacionados com a construção do dispositivo de captura que afetam o processo de formação de imagens e vídeos, e a qualidade dos arquivos resultantes. Estes fatores são chamados de **extrínsecos**, e incluem:

- Iluminação, tais como tipo, intensidade, direção, cor e uniformidade da luz;
- Posição e orientação da câmera: distância da câmera para o objeto, assim como o ângulo da lente para o alvo afetam a perspectiva, o enquadramento e a composição da imagem;
- Movimentação: qualquer movimentação da câmera ou do objeto pode causar perda de foco e distorções;
- Condições ambientais: temperatura, umidade e poluição afetam a qualidade da imagem; e
- Interferências externas: refrações, reflexos ou obstruções ou sobreposições alteram a aparência dos objetos.

O processamento de imagens consiste normalmente das seguintes etapas: a captura da imagem pelo usuário, o processamento da imagem (que pode ser repetido ou combinado com diversas técnicas) e geração do resultado, para ser mostrado para o usuário. A Figura 7.1 apresenta um diagrama que ilustra esse processo.

Figura 7.1 - Etapas para aplicação do processamento de imagens.



Fonte: Autor, 2023.

A partir dessa conceituação, foi possível iniciar a codificação utilizando a metodologia *hands-on*. Inicialmente, os alunos geraram diversas imagens e vídeos de objetos e de si mesmos, para aplicar algumas técnicas explicadas e codificadas. Foram utilizados os próprios aparelhos celulares, pois a intenção era realizar variações de luminosidade e deixar que cada aluno tivesse a liberdade para capturar qualquer tipo de objeto presente no laboratório, considerando luzes ligadas e desligadas, assim como uso ou não de iluminação adicional de flash. Inicialmente todas as imagens foram lidas na configuração RGB (*Red, Green & Blue*).

A partir disso, foram aplicadas técnicas de análise de histograma, como ilustra a Figura 7.2, para que os alunos tivessem a percepção da influência da luz na intensidade dos pixels das imagens. Para tal, foi utilizado o método Ravel da biblioteca NumPy (Harris,

2020). Para que essa análise seja possível, é necessário a aplicação deste método, que converte uma matriz multidimensional em uma única dimensão, utilizando a otimização da biblioteca para processar as intensidades e computar as frequências de pixels em cada nível de intensidade, de 0 a 255.

Todas as atividades práticas foram realizadas com a utilização da linguagem de programação Python 3, van Rossum & Drake (2009), pela sua rápida curva de aprendizado, garantindo dessa forma que diversos alunos que não tinham formação em desenvolvimento de *software* conseguissem participar e assimilar sem preocupações com complexidades inerentes à linguagem com maior rigidez sintática. Como alguns alunos ainda não tinham conhecimento da linguagem, foi necessário utilizar cerca de 4 horas do conteúdo para explicar características e realizar codificações utilizando estruturas de dados específicas (listas, tuplas, dicionários), assim como operações de entrada e saída de dados, utilização de estruturas de controle de fluxo e de repetição, sempre reforçando com diversas atividades de fixação. Após esse nivelamento, foi dado início ao conteúdo de processamento de imagens e vídeo, utilizando a biblioteca OpenCV, conforme implementações em Bradski e Kaehler (2008), sendo totalmente de código aberto.

Figura 7.2 - Análise de histograma para intensidade dos pixels.



Fonte: Autor, 2023.

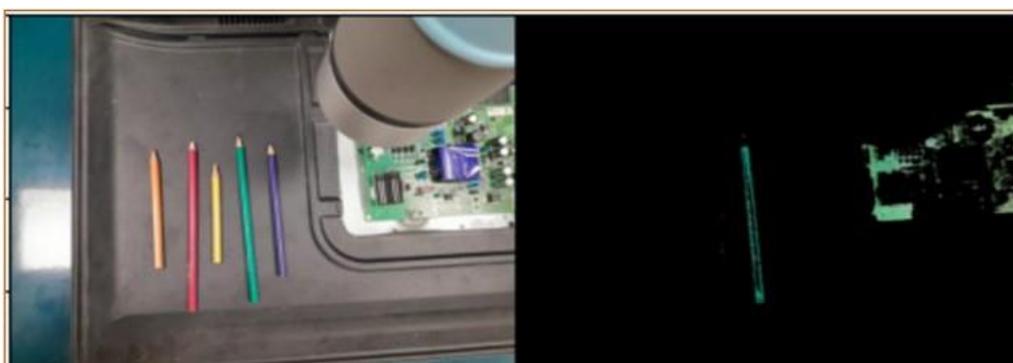
Para entender as demais técnicas de processamento de imagens, foi feita a codificação e explicação de cada uma, com aumento gradual de complexidade e variação de parâmetros, e os alunos replicavam os códigos, aplicando primeiramente em seu conjunto específico de imagens geradas de forma individual, e posteriormente praticando com atividades individuais ou em grupo.

Após o entendimento das questões referente a aquisição e impacto da luz na geração das imagens, iniciamos o estudo de técnicas de transformações geométricas importantes para possibilitar ajustes nas imagens, mostrando que se torna desnecessário, em muitos casos, alterar a posição de câmera ou objetos e realizar novas aquisições, a partir de aplicação de transformações como: rotação, redimensionamento – explicando o relevante conceito de interpolação de Jähne (2005). Além disso, operações de mudança de eixos, transposição e espelhamento também foram apresentadas e implementadas. (Szeliski, 2022).

Para a detecção de cores, foi utilizado inicialmente as imagens em RGB, e posteriormente demonstrado que o espaço de cores HSV (*hue, saturation and value*), possibilita maior flexibilidade e precisão no processo de filtro de faixas de cores (Smith,1978).

Com a aplicação dessa técnica, foi possível que os alunos realizassem atividades visando detecção de objetos de cores específicas, conforme exemplo na Figura 7.3, na qual o lado esquerdo mostra a imagem original em RGB, e o lado direito a imagem resultante após definição dos limiares inferior e superior, e aplicação da operação de comparação bit a bit *bitwise_and* conforme Gonzalez & Woods (2010), que possibilita filtrar somente a cor alvo (verde).

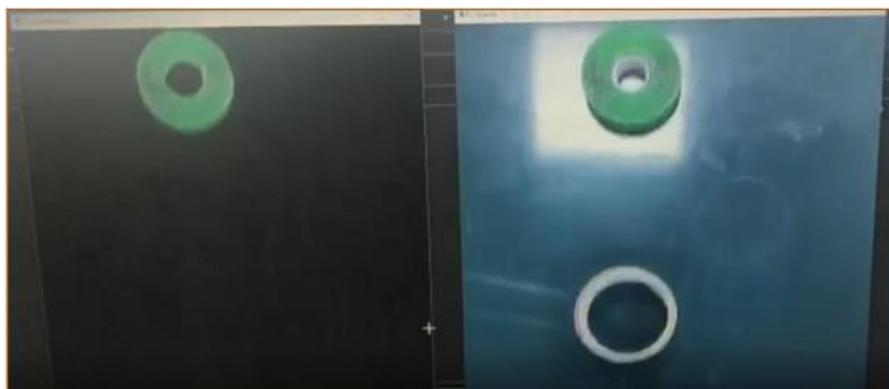
Figura 7.3 – Aplicação de detecção de cores em arquivos de imagem.



Fonte: Autor, 2023.

Posteriormente à detecção de cores em imagens, foi feita a explicação de como capturar vídeos em tempo real e realizar a aplicação do filtro de cores, assim como outros filtros. Com utilização da esteira existente, foi então realizada a detecção de objetos em tempo real, conforme a Figura 7.4, que mostra a aplicação da mesma técnica, dessa vez em um vídeo. Do lado esquerdo, a detecção e do lado direito o vídeo original.

Figura 7.4 – Detecção de objetos verdes em tempo real (vídeo).

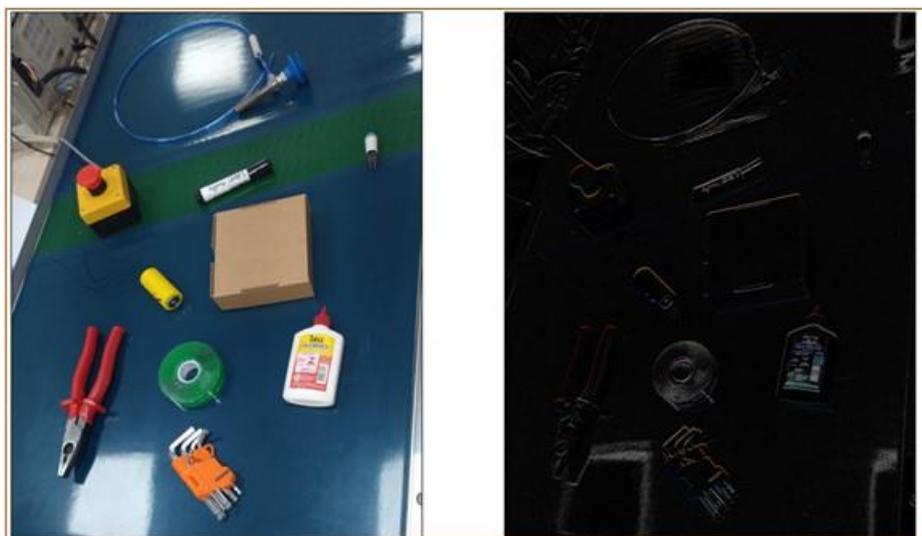


Fonte: Autor, 2023.

Ainda referenciando importantes conceitos e técnicas necessárias, foi necessário demonstrar a função *filter2D*, que realiza o processo de convolução em imagens a partir de um *kernel* específico para cada finalidade (Jäjäne, 2005). Os *kernels* podem ser entendidos como filtros, que são matrizes numéricas pequenas, utilizadas para realizar

determinadas operações na imagem (matriz) original. Alguns exemplos dessas operações são: suavização; desfoque; nitidez; detecção de bordas, tais como Sobel e LaPlace, ambos demonstrados em Gonzalez & Woods (2010), e Canny (Canny, 1986). Um exemplo da aplicação realizada como atividade prática pode ser visualizada na Figura 7.5, onde do lado esquerdo está a imagem original, e do lado direito a imagem processada com a detecção das bordas.

Figura 7.5 – Aplicação de detecção de borda.



Fonte: Autor, 2023.

Com o objetivo de possibilitar o entendimento a respeito da necessidade de avaliar a qualidade das imagens, assim como determinados ruídos afetam na aplicação dos filtros, foi utilizada uma imagem de Chowdhury (2017), ilustrada na Figura 7.6, com um nível alto de ruído do tipo *Salt & Pepper*, caracterizado pela presença de pontos (ruídos) brancos (*salt*) e pretos (*pepper*), e a partir disso, foram aplicados filtros de média para observar o resultado. (GONZALEZ & WOODS, 2010). Após a aplicação do filtro, foi possível realizar a diferença entre as imagens e verificar visualmente a quantidade de ruído eliminado da imagem original.

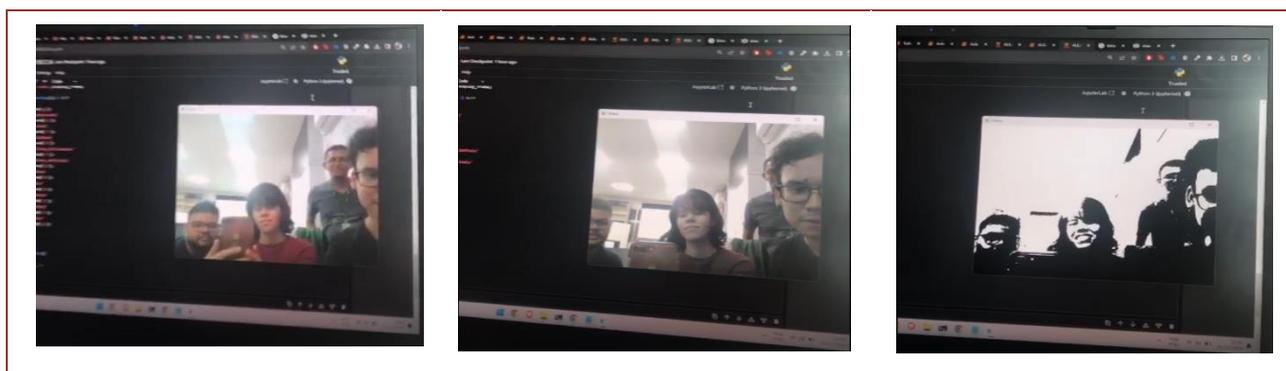
Figura 7.6 – Aplicação e visualização da quantidade de ruído existente na imagem.



Fonte: Adaptado a partir de Chowdhury, 2017.

Para finalizar os tópicos relativos a esta seção, foi realizada uma atividade na qual seria necessário que os alunos criassem uma função para gerar vídeos em tempo real pela câmera, e a partir de teclas pré-definidas, aplicassem determinados filtros pré-definidos, conforme a Figura 7.7, que mostra a aplicação de conversão de formato dos vídeos de RGB para tons de cinza e binário.

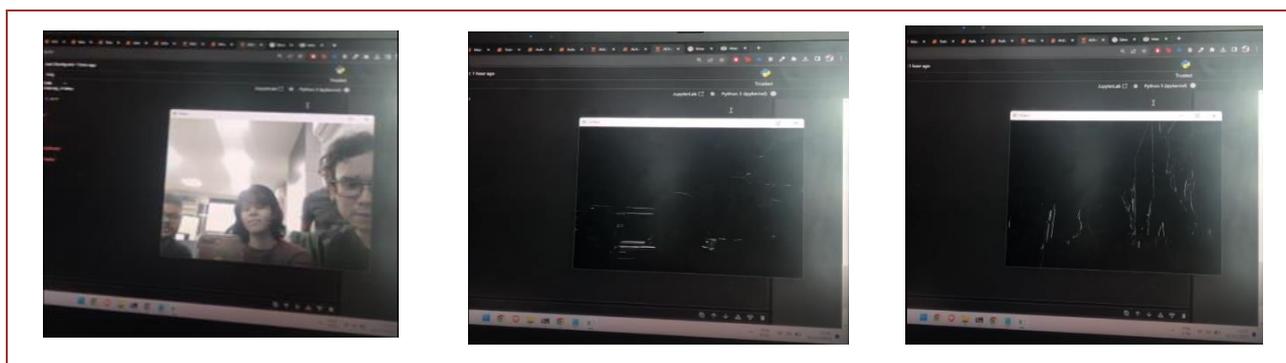
Figura 7.7 – Conversão de vídeos RGB em tons de cinza e binário, em tempo real.



Fonte: Autor, 2023.

Além dessas conversões de imagens, outros filtros aplicados também em tempo real foram os de detecção de linhas horizontais e de linhas verticais em imagens em tons de cinza, conforme a Figura 7.8.

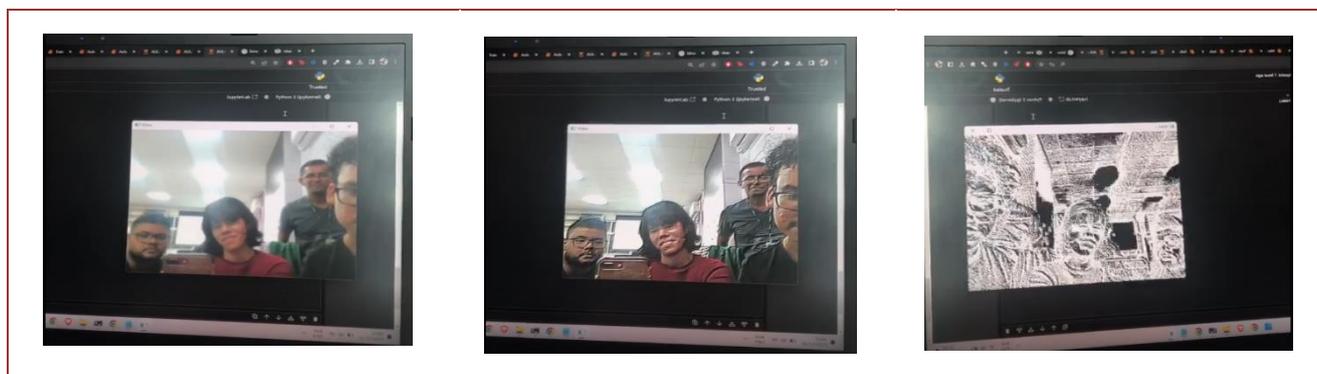
Figura 7.8 – Detecção de linhas horizontais e verticais em tempo real.



Fonte: Autor, 2023.

Por fim, na Figura 7.9 é mostrado o resultado da aplicação dos filtros *sharp*, que realça as diferenças de intensidades entre os pixels adjacentes; de relevo, responsável por destacar as bordas e criar um efeito que imita uma superfície elevada; e de Sobel, que realça as bordas detectadas, em tons de cinza.

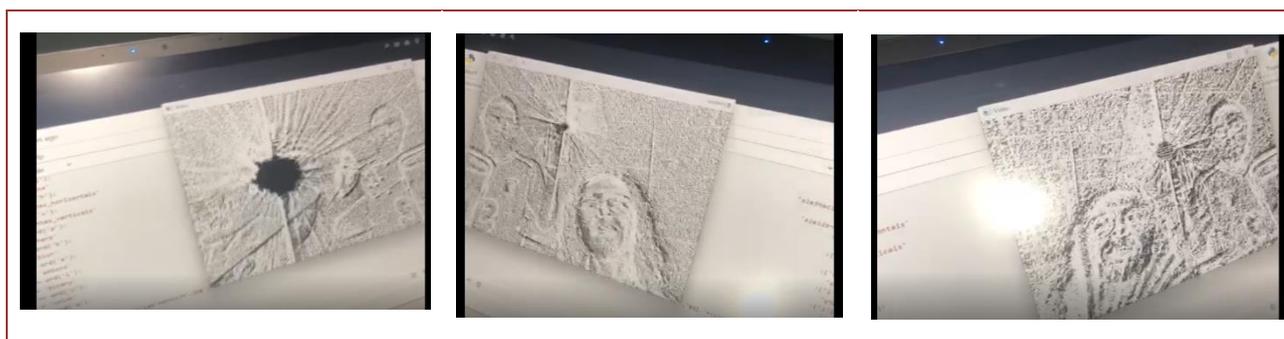
Figura 7.9 – Aplicação dos filtros sharp, relevo e Sobel em tempo real.



Fonte: Autor, 2023.

Esta atividade despertou a curiosidade de vários alunos, possibilitando que se aprofundassem com a realização de diversos outros testes, já que as capturas estavam sendo feitas em tempo real. Isso permitiu que por exemplo, fossem feitas análises de como determinados filtros, como o Sobel, se comportam quando é utilizado luzes do tipo *flash*. Esses resultados são ilustrados na Figura 7.10, mostrando os feixes da frequência luminosa e com o foco principal da luz invertido.

Figura 7.10 – Resultado de uso de flashes em tempo real com o método Sobel.



Fonte: Autor, 2023.

7.2.3. APRENDIZADO DE MÁQUINA E APLICAÇÕES EM VISÃO COMPUTACIONAL

O aprendizado de máquina constitui o estado da arte em problemas de visão computacional com objetivo de detecção, classificação e reconhecimento de objetos. Por este motivo, conhecer seus fundamentos, assim como quais são as etapas necessárias para utilizar modelos existentes é crucial para qualquer projeto que envolva visão computacional. A codificação de tais modelos é realizada de forma bem diferente da programação tradicional, onde são apresentados os dados de entrada e elaborados os programas com determinada lógica para que seja gerada a saída (resultado). Em

aprendizado de máquina, são definidos os dados de entrada e a saída esperada (resultado), e o modelo realiza as inferências necessárias para estabelecer funções de aproximação, “aprendendo” características dos dados. Estes modelos podem ser usados para dados de texto ou imagens.

Em tarefas envolvendo imagens, os modelos mais robustos e precisos utilizam várias camadas dispostas sequencialmente entre a camada de entrada e de saída, e são conhecidos como modelos de *deep learning*. Os modelos utilizados durante as aulas foram basicamente modelos supervisionados de classificação, conforme James *et al.* (2013), e esta seção trata de conceitos sobre esse tipo de modelo, e como ele foi aplicado durante as aulas.

O objetivo de um modelo de classificação é prever a categoria a qual uma instância (amostra) pertence, com base em suas características. O resultado de uma classificação é um conjunto de valores percentuais, indicando probabilidades para as classes para as quais o modelo foi treinado.

Treinar um modelo significa realizar extração das características mais significativas dos dados, e a partir disso, ao submeter uma imagem qualquer, o modelo será capaz de dizer qual classe a imagem pertence. Modelos de classificação podem ser projetados para identificar objetos de n classes distintas. (Alpaydin, 2020).

No caso de modelos para tarefas em imagens, as atividades realizadas durante o treinamento são normalmente baseadas na aplicação de filtros de convolução, ou Redes Neurais Convolucionais (CNNs) (Lecun, Bengio, Hinton, 2015).

O processo para utilização de um modelo de classificação pode ser entendido em duas etapas: a primeira é o treinamento do modelo, e a segunda ocorre quando o modelo já foi treinado e pode ser utilizado para inferências, com a submissão de entradas para a classificação. Para realizar treinamento de modelos de *deep learning* é necessário utilizar uma quantidade extremamente grande de imagens para que as características relevantes dos dados sejam extraídas e associadas. Uma outra questão a ser observada é a disponibilidade de grandes conjuntos de dados públicos para possibilitar o treinamento. O processo de construção desses conjuntos de dados requer um investimento de tempo e recurso computacionais nem sempre disponíveis. Por essas questões, opta-se normalmente pela utilização de modelos já treinados em grandes bases de dados.

Um dos primeiros conjuntos de dados disponibilizado foi o ImageNet, por Deng *et al.* (2009), e a partir dele, diversos modelos foram treinados com as milhões de imagens que o conjunto contém, com 1000 classes disponíveis.

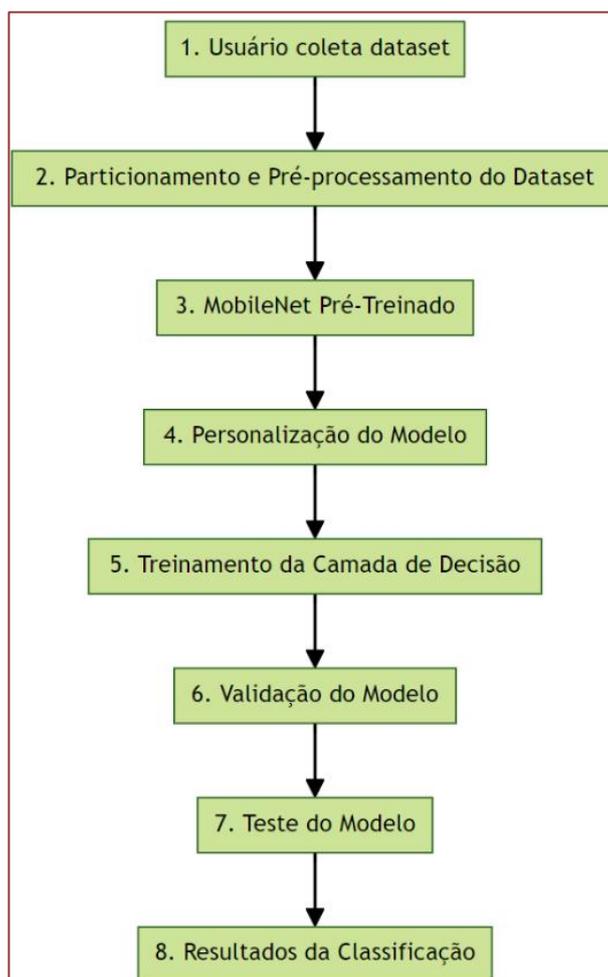
Shin *et al.* (2016) utilizaram modelos “pré-treinados”, ou seja, substituíram a camada que realiza a classificação das imagens (camada de decisão), pois todas as demais camadas são capazes de identificar as características básicas utilizadas no processo de identificação das imagens, pois os neurônios existentes já contém os pesos (valores) ajustados para o reconhecimento. Essa técnica é conhecida como *Transfer Learning*, utilizada em grande parte dos projetos que utilizam modelos de *deep learning* e por isso, optamos por utilizar nas aulas alguns modelos pré-treinados (Kornblith, Shlens, Le. 2019).

As primeiras atividades foram utilizando o conjunto de dados CIFAR-10, elaborada por Krizhevsky, Nair e Hinton (2009), para que os alunos entendessem como estruturar e configurar parâmetros de um modelo e realizar o carregamento do modelo pré-treinado. Em seguida, foram realizadas atividades práticas com mais 3 modelos: ResNet, uma rede

mais profunda, com mais camadas entre as camadas de entrada e de saída, He et al. (2016); MobileNet, um modelo mais otimizado para hardware com menor poder de processamento, Howard et al. (2017) e EfficientNet, um modelo que exige menos recursos e que obtém melhor desempenho. (TAN & LE, 2019).

A Figura 7.11 ilustra resumidamente as etapas necessárias para utilização de um modelo pré-treinado na MobileNet: aquisição de imagens-alvo; definição e particionamento do modelo; pré-processamento das imagens coletadas; carregamento dos pesos das camadas; personalização, com a troca da camada de decisão pela camada que representa as classes-alvo do problema em questão; treinamento da camada de decisão; validação e teste do modelo; e finalmente, avaliação dos resultados (desempenho) do modelo.

Figura 7.11 – Etapas para utilização do modelo pré-treinado MobileNet.



Fonte: Autor, 2023.

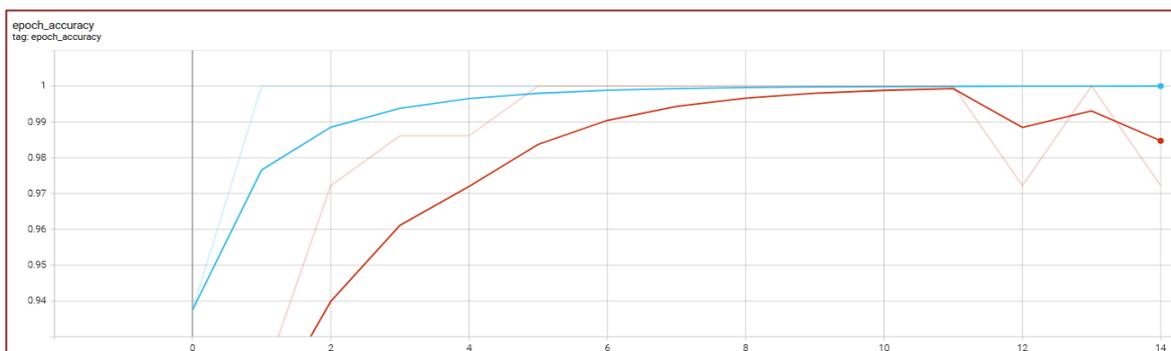
O funcionamento básico pode ser entendido conforme a sequência de cada numeração da legenda, e todos os passos foram seguidos nas atividades práticas:

- 1) O usuário realiza a aquisição ou coleta das imagens e/ou vídeos necessários, de acordo com o tipo de problema que precisa ser aplicado. No caso de problemas de

visão computacional, os dados de entrada são as imagens ou vídeos, que são agrupados por classe em diretórios diferentes, e as saídas são as classes dos objetos (nomes dos diretórios). Como exemplo de aplicação de treinamento, podemos utilizar imagens de produtos que precisam ser analisados em relação à ocorrência ou não de defeitos de fabricação. Para que isso seja possível, foi necessário fazer a geração de imagens de produtos normais (não defeituosos) e outro conjunto de imagens dos produtos defeituosos. Como analogia, podemos exemplificar que em determinada linha de produção, uma peça precisa ter exatamente 4 furos. Ao gerar imagens desse tipo, podemos utilizar a estratégia de nomear o diretório das imagens com 4 furos como “Aprovado”, e todas as outras imagens de produtos defeituosos, com 1, 2 ou 3 furos serão agrupadas em outro diretório, com o nome “Defeito”. É importante que o número de imagens seja equivalente entre as classes, e com isso, criamos um dataset balanceado e evitamos que o modelo gere alguma tendência por “enxergar” mais características de determinada classe. Se quisermos somente que o modelo classifique imagens como produto aprovado ou defeituoso, podemos entender que esse é um problema de classificação binária. (Konrad, Lohmann, Abel, 2019);

- 2) O próximo passo é o particionamento do dataset, ou seja, divisão de um número determinado de imagens, tanto para a classe “Aprovado” quanto para a classe “Defeito”. Esse processo é feito configurando parâmetros entre “treino”, “validação” e “teste”, e optamos pela proporção percentual de 70/10/20, respectivamente. Após o particionamento, é feito o pré-processamento do dataset, ou seja, são aplicadas técnicas de acordo com as características do modelo escolhido. No mínimo é feito o redimensionamento das imagens para que possam ser entendidas pelo modelo, pois no caso do MobileNet, a resolução das imagens para entrada no modelo deve ser (224, 224);
- 3) É feita então a definição do modelo a ser treinado, com o carregamento do modelo pré-treinado, utilizando todas as camadas a não ser a de decisão “congeladas”, evitando dessa forma que os seus pesos sejam atualizados;
- 4) A personalização do modelo pode ser realizada com acréscimos de camadas adicionais e aplicação de várias técnicas que ajudem em melhores resultados. Além disso, podemos considerar realização de ajustes de hiper parâmetros, tais como número de épocas e tamanho do lote;
- 5) A partir das etapas de definição e personalização, inicia-se o treinamento do modelo. Em do MobileNet tentar reconhecer as 1000 classes para as quais o modelo ImageNet treinado inicialmente, irá utilizar nossa classificação binária de acordo com a estrutura dos diretórios.
- 6) Para modelos de classificação, normalmente é utilizado como indicador de desempenho a métrica acurácia, e também é feita a análise da função de perda, ou erro do modelo, tanto nas etapas de treinamento, quanto na validação e teste. A acurácia é definida pela proporção (%) de previsões corretas resultante do treinamento e validação, ou seja, qual a taxa de acerto da classificação. A figura 7.12 ilustra o comportamento da métrica “acurácia”, para 15 épocas que o modelo foi treinado.

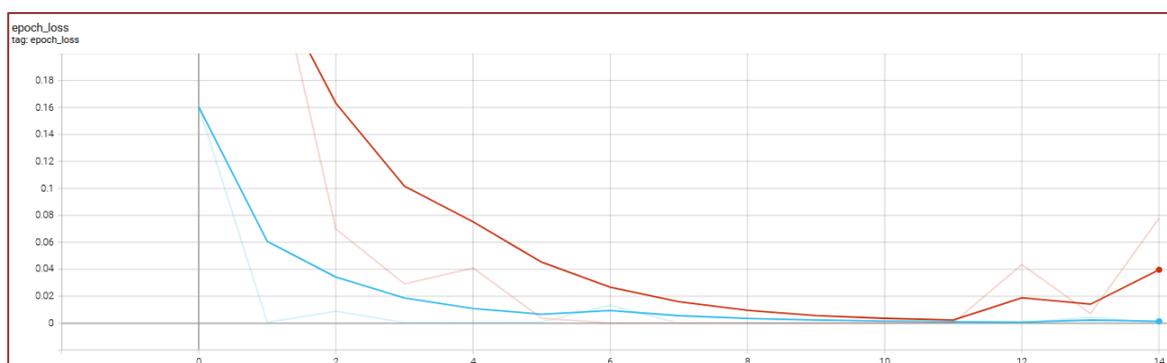
Figura 7.12 - Evolução da acurácia por época no treinamento (vermelho) e validação (azul) do modelo pré-treinado da MobileNet.



Fonte: Autor, 2023.

Outra variável que precisa ser analisada é o erro (loss). A Figura 7.13 mostra como se comportou essa variável durante o treinamento das 15 épocas.

Figura 7.13 - Evolução do erro por época no treinamento (vermelho) e validação (azul) do modelo pré-treinado da MobileNet.



Fonte: Autor, 2023).

- 7) Para realizar os testes no modelo, são feitas várias inferências com imagens que o modelo não conhece, ou seja, não foram utilizadas no treino ou na validação. Conforme a proporção inicialmente definida, 20% das imagens foram destinadas para o teste.
- 8) Por fim, é aferida a acurácia e erro no teste.

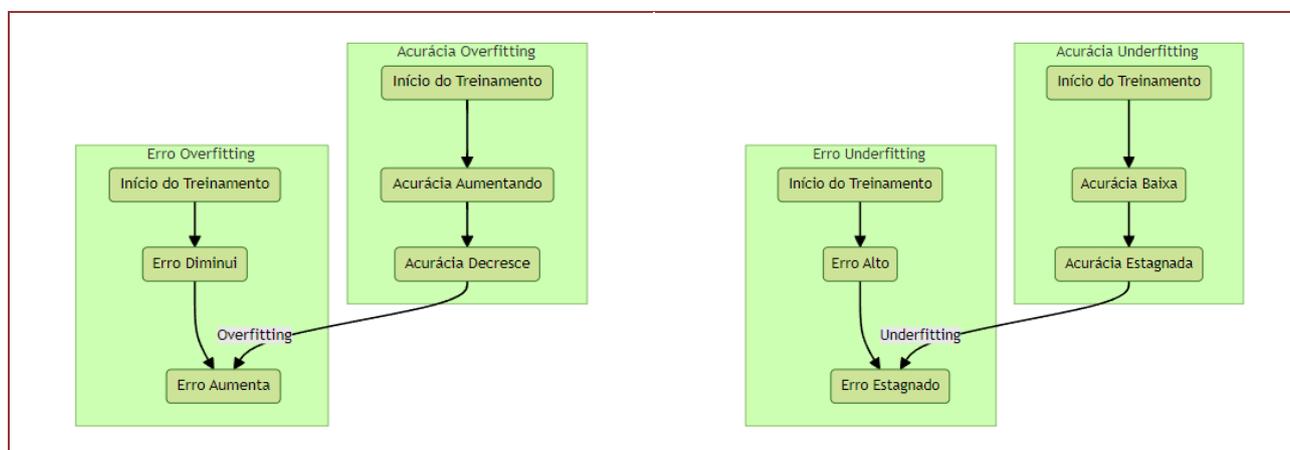
Todas as etapas descritas foram codificadas e testadas utilizando o framework Keras, criada por Chollet (2015), com aplicação de diversas variações de otimizadores, conforme Goodfellow, Bengio e Courville (2016) e Gulli e Pal (2017), para que os alunos tivessem a noção do impacto dos otimizadores no treinamento e resultados no desempenho do modelo.

Além disso, dependendo da quantidade de imagens que os alunos tinham gerado, foi necessário explicar questões relativas a *overfitting* e *underfitting*, pois com

quantidades insuficientes de imagens coletadas, o modelo não conseguiu se adaptar ao novo conjunto de dados. A Figura 7.14 ilustra o tipo de *insight* demonstrado para os alunos, para que pudessem perceber a ocorrência de *overfitting* (lado esquerdo) e *underfitting* (lado direito), analisando valores de acurácia e erro durante o processo de treinamento.

Para contornar esses problemas, foi demonstrado como realizar o ajuste do modelo, descongelando mais camadas além da camada de decisão para o treino, e também, explicação e demonstração da técnica de aumento de dados (*data augmentation*), conforme Nusrat e Jang (2018), com operações de rotação com limite de até 30 graus, espelhamento, zoom, deslocamento na altura e deslocamento na largura.

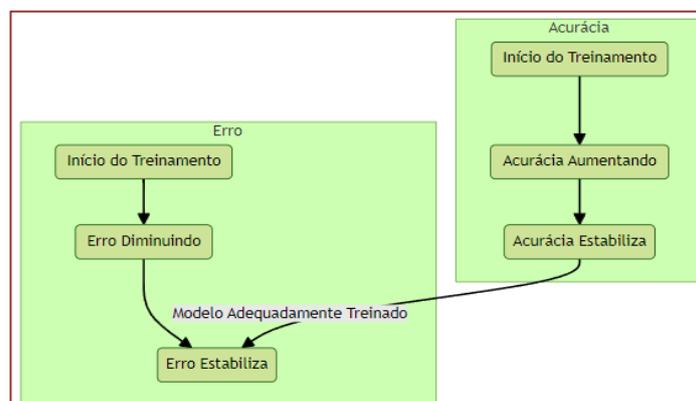
Figura 7.14 – Indicação de uma técnica de análise de valores de erro e de acurácia, para detectar *overfitting* e *underfitting* durante o treinamento de modelos.



Fonte: Autor, 2023.

Foi explicado também como seria o comportamento ideal de um modelo, sem a ocorrência dos problemas mencionados. A Figura 7.15 ilustra esse comportamento, ainda com a verificação da acurácia e do erro durante as épocas do treinamento.

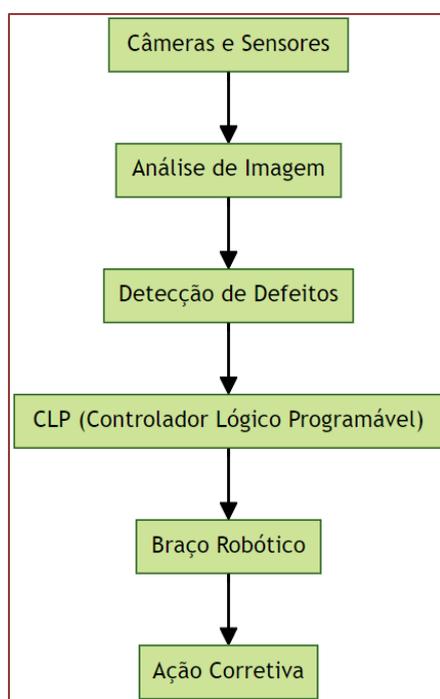
Figura 7.15 – Comportamento ideal da acurácia e erro durante o treinamento de modelos.



Fonte: Autor, 2023.

Ao final da disciplina, foi possível que os alunos entendessem a importância da Visão Computacional e diversas aplicações na área da indústria. Integrada à robótica industrial, e conforme Mane (2021), existem inúmeros avanços tecnológicos, principalmente com a aplicação de modelos de *deep learning*, sendo fundamental para a detecção de defeitos e otimização de processos produtivos. A Figura 7.16 mostra, de forma resumida, como é interação desde a aquisição feita por câmeras ou sensores, análise das imagens, detecção de defeitos, comunicação com braço robótico via CLP, e a ação corretiva, que pode ser um programa previamente configurado, com por exemplo, retirar a peça defeituosa da esteira para uma área específica.

Figura 7.16 – Etapas integradas da aplicação de Visão Computacional integrada à robótica.



Fonte: Autor, 2023.

REFERÊNCIAS

- [1] ALPAYDIN, Ethem. **Introduction to machine learning**. MIT press, 2020.
- [2] BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer vision with the OpenCV library**. " O'Reilly Media, Inc.", 2008.
- [3] CANNY, John. A computational approach to edge detection. **IEEE Transactions on pattern analysis and machine intelligence**, n. 6, p. 679-698, 1986.
- [4] CHOLLET, F. (2015). Keras. [Software]. Disponível em <https://keras.io>.
- [5] CHOWDHURY, Debkumar et al. A quantum study on digital image noises and their in-depth clusterization. In: **2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix)**. IEEE, 2017. p. 1-7.
- [6] DENG, Jia et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE conference on computer vision and pattern recognition**. Ieee, 2009. p. 248-255.

- [7] GONZALEZ, Rafael C.; WOODS, Richard E. Processamento digital de imagem. **Pearson, ISBN-10: 8576054019**, v. 10, 2010.
- [8] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT press, 2016.
- [9] GULLI, Antonio; PAL, Sujit. Deep learning with Keras. Packt Publishing Ltd, 2017.
- [10] HARRIS, Charles R. et al. Array programming with NumPy. **Nature**, v. 585, n. 7825, p. 357-362, 2020.
- [11] HE, Kaiming et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016. p. 770-778.
- [12] HOWARD, Andrew G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.
- [13] JÄHNE, Bernd. **Digital image processing**. Springer Science & Business Media, 2005.
- [14] JAMES, Gareth et al. **An introduction to statistical learning**. New York: springer, 2013.
- [15] KRIZHEVSKY, A., NAIR, V., & HINTON, G. (2009). CIFAR-10 (Canadian Institute for Advanced Research) [Dataset]. Disponível em: <https://www.cs.toronto.edu/~kriz/cifar.html>
- [16] KORNBLITH, Simon; SHLENS, Jonathon; LE, Quoc V. Do better imagenet models transfer better?. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. 2019. p. 2661-2671.
- [17] KONRAD, Thomas; LOHMANN, Lutz; ABEL, Dirk. Surface defect detection for automated inspection systems using convolutional neural networks. In: **2019 27th Mediterranean Conference on Control and Automation (MED)**. IEEE, 2019. p. 75-80.
- [18] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, v. 521, n. 7553, p. 436-444, 2015.
- [19] MANE, Tanmay et al. Identification and Classification of Industrial Elements using Artificial Intelligence and Image Processing Techniques. In: **2021 International Conference on Emerging Smart Computing and Informatics (ESCI)**. IEEE, 2021. p. 165-169.
- [20] NUSRAT, Ismoilov; JANG, Sung-Bong. A comparison of regularization techniques in deep neural networks. *Symmetry*, v. 10, n. 11, p. 648, 2018.
- [21] SHIN, Hoo-Chang et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, v. 35, n. 5, p. 1285-1298, 2016.
- [22] SMITH, Alvy Ray. Color gamut transform pairs. *ACM Siggraph Computer Graphics*, v. 12, n. 3, p. 12-19, 1978.
- [23] SZELISKI, Richard. **Computer vision: algorithms and applications**. Springer Nature, 2022.
- [24] VAN ROSSUM, Guido; DRAKE, Fred L. **Introduction to python 3: python documentation manual part 1**. CreateSpace, 2009.

O processo de formação profissional, marcado pelas rápidas modificações tecnológicas e mudanças do mercado de trabalho, configura-se como um desafio complexo e contínuo. De modo geral, não existem parâmetros fixos e acabados que indiquem um percurso seguro e eficaz no decorrer de tal processo. No entanto, uma perspectiva tem se tornado cada vez mais relevante: as parcerias desempenham um papel crucial (Martino, 2006).

As interações e o diálogo entre as instituições de ensino e empresas incidem na superação entre o dualismo teoria e prática. A cooperação entre ambas as partes, cada vez mais tem se intensificado, e anunciado um contexto em que os estudantes vislumbrem a aplicação de informações e conhecimentos em situações que emergem da realidade vivida, desenvolvendo modos de aprender essenciais para as nuances que atravessam a visão de trabalho contemporâneo bem como para a própria vida.

A construção de parcerias reverbera, fundamentalmente, a existência de um ambiente que se constitui pelo diálogo. A intersecção de ideias e suas concretizações são emolduradas pelos interesses, mudanças e adaptações que forjam não somente a individualidade dos sujeitos (no caso desse projeto dos estudantes), mas também incide na estrutura das organizações, seja da empresa seja da instituição de ensino. As parcerias impactam, ainda, o entorno (a comunidade, a cidade, o Estado) destacando a importância de práticas de desenvolvimento e sustentabilidade.

Sendo assim, destaca-se que as parcerias entre empresas e instituições de ensino é um imperativo para:

- *Atualização contínua e acesso ao mercado de trabalho*: a realização de projetos e programas promovem e garantem uma formação profissional alinhada às tendências tecnológicas e amplia as oportunidades de estágio e emprego aos participantes (Rego; Rosas; Prado, 2021).
- *Incentivo à inovação e à pesquisa*: a colaboração entre os setores orienta esforços no desenvolvimento de soluções inovadoras. A prática científica e o progresso tecnológico basilar os processos de aprendizagem que nascem com a finalidade de compreender e dar uma resposta as intercorrências do cotidiano. Vale destacar que esse tipo de interação há tempos foi nominada de habitat de inovação (Spolidoro, 1997).
- *Construção de habilidades socioemocionais*: para além das competências técnicas, a formação profissional forjada a partir das parcerias exercita vivências e

¹ Doutor em Ciências do Ambiente e Sustentabilidade na Amazônia. Apoio Pedagógico do Curso de Capacitação Avançada de Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM).

experiências que mobilizam o desenvolvimento de habilidades, como comunicação eficiente, trabalho em equipe, adaptação e flexibilidade, etc (BERLINGERI, 2018).

Sendo assim, o Curso Capacitação Avançada em Robótica Industrial se constituiu a partir de uma rede de parcerias retroalimentada pelos seguintes atores: Instituto Federal de Educação, Ciência e Tecnologia (IFAM); LG *Eletronics*; Fundação de Apoio ao Ensino, Pesquisa, Extensão e Interiorização do Ifam (FAEPI) e Polo de Inovação de Manaus (INOVA)

8.1. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA (IFAM)

Figura 8.1 – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM.



Fonte: www.ifam.edu.br, 2024.

O IFAM faz parte da Rede Federal de Educação Profissional e Tecnológica, a qual possui como origem comum o Decreto Lei Nº 11.892 de 29 de dezembro de 2008, assinado pelo Presidente da República, Luís Inácio Lula da Silva. O decreto criou trinta e oito Institutos Federais (IF's), entre eles o IFAM.

Como instituição tem como missão promover com excelência a Educação, Ciência e Tecnologia para o desenvolvimento sustentável da Amazônia. Como visão, busca consolidar o IFAM como referência nacional em Educação, Ciência e Tecnologia (IFAM, 2024a). Sendo assim, os valores que sustentam a organização são:

- Acessibilidade e inclusão social;
- Valorização das pessoas;
- Cidadania e justiça social;
- Ética e transparência
- Excelência na gestão educacional;
- Gestão democrática participativa;
- Inovação e empreendedorismo;

- Respeito à diversidade;
- Responsabilidade socioambiental;
- Solidariedade.

8.2. LG ELETRONICS DO BRASIL.

A empresa LG Eletronics é um conglomerado multinacional, que possui uma diversidade de produtos e serviços eletrônicos, a qual tem como missão fornecer produtos e serviços digitais inovadores e topo de gama e garantir a satisfação dos consumidores (LG Eletronics, 2024), conforme se observa na Figura 8.1. No Brasil, possui fábrica no Polo Industrial de Manaus, atuando com a produção de eletroeletrônicos e linha branca, com destaque para a produção de Televisores e Condicionadores de Ar.

Figura 8.2: Visão



Fonte: LG Eletronics, 2024.

8.3. FUNDAÇÃO DE APOIO AO ENSINO, PESQUISA, EXTENSÃO E INTERIORIZAÇÃO DO IFAM (FAEPI)

Figura 8.3 – FAEPI: Fundação de Apoio do IFAM.



Fonte: <https://faepi-ifam.org.br/>, 2024.

A FAEPI se constitui como uma instituição de direito privado, a qual está vinculada (credenciada) para apoiar as ações do IFAM a partir de projetos de pesquisa, ensino, extensão e desenvolvimento institucional (IFAM, 2024b).

8.4. POLO DE INOVAÇÃO DE MANAUS – INOVA

Figura 8.4 – INOVA: Polo de Inovação de Manaus / IFAM.



Fonte: <https://inova.ifam.edu.br/>, 2024.

A criação do Polo de Inovação de Manaus vincula-se aos objetivos do com o intuito de fomentar um ambiente para o desenvolvimento de pesquisa aplicada e inovação tecnológica. Desta forma, possui como finalidade “*promover a pesquisa aplicada, o desenvolvimento científico e tecnológico e a formação de recursos humanos na Amazônia, buscando a excelência na área de competência intitulada Controle e Processos Industriais com abrangência de atuação no Polo Industrial de Manaus (PIM)*” (Polo de Inovação Ifam, 2024).

Por fim, cabe destacar que construir parcerias no âmbito formacional amazônica é crucial para enfrentar os desafios únicos desta vasta região. Essa perspectiva oportuniza considerar a complexidade do ambiente amazônico, sua rica biodiversidade e a pujança das diversidades étnicas existente. A formação de parceria oferta um olhar acerca da formação profissional adaptada às necessidades locais e o urgente compromisso para com a sustentabilidade.

REFERÊNCIAS

- [1] BERLINGERI, Matheus Mascioli. Competências socioemocionais e mercado de trabalho: um estudo para o caso brasileiro. Dissertação de mestrado. Universidade de São Paulo: Ribeirão Preto, 2018.
- [2] Instituto Federal do Amazonas – IFAM. Fundação de Apoio. Disponível em: <http://www2.ifam.edu.br/acesso-a-informacao/fundacao-de-apoio>. Acesso no dia 24 de janeiro de 2024b.
- [3] Instituto Federal do Amazonas - IFAM. Missão e Visão. Disponível em: <http://www2.ifam.edu.br/instituicao/missao-e-visao>. Acesso no dia 24 de janeiro de 2024a.
- [4] LG Electronics. A nossa visão. Disponível em: <https://www.lg.com/pt/descubra-lg/a-nossa-visao/>. Acesso no dia 24 de janeiro de 2024.
- [5] Martino, Mariluci Alves. A importância das parcerias na educação profissional. 2006. 221 f. Dissertação (Mestrado em Educação: Currículo) - Programa de Estudos Pós-Graduados em Educação: Currículo, Pontifícia Universidade Católica de São Paulo, São Paulo, 2006.
- [6] Polo de Inovação IFAM. Finalidade. Disponível em: <https://inova.ifam.edu.br/quem-somos/>. Acesso no dia 24 de janeiro de 2024.
- [7] REGO, Fátima Aparecida; ROSAS, Iris Renata de Carvalho; PRADOS, Rosália Maria Netto. Educação Profissional e Tecnológica como alternativa de acesso ao mercado de trabalho. *Brazilian Journal of Development*, 7(2), 14585–14596, 2021.
- [8] SPOLIDORO, Roberto. A Sociedade do Conhecimento e Seus Impactos no Meio Urbano. In: PALADINO, Gina ; MEDEIROS, Lucília Atas (org). *Parques Tecnológicos e Meio Urbano*. Artigos e Debates. Brasília: ANPROTEC, 1997.

www.poisson.com.br
contato@poisson.com.br

LRI

LABORATÓRIO DE
ROBÓTICA INDUSTRIAL

LG Academy