



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
AMAZONAS - IFAM
CAMPUS MANAUS DISTRITO INDUSTRIAL
ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

WASHINGTON HENRIQUE ALVES JUNIOR

**USO DO DRONE TELLO EDU COMO FERRAMENTA INTEGRADA PARA O
ENSINO DE PROGRAMAÇÃO, VISÃO COMPUTACIONAL E
INTELIGÊNCIA COMPUTACIONAL**

MANAUS-AM

2025

WASHINGTON HENRIQUE ALVES JUNIOR

**USO DO DRONE TELLO EDU COMO FERRAMENTA INTEGRADA PARA O
ENSINO DE PROGRAMAÇÃO, VISÃO COMPUTACIONAL E
INTELIGÊNCIA COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Departamento de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Campus Manaus - Distrito Industrial, como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Vitor Bremgartner da Frota.

**MANAUS-AM
2025**

Dados Internacionais de Catalogação na Publicação (CIP)

| | |
|-------|---|
| A474u | <p>Alves Júnior, Washington Henrique.</p> <p>Uso do drone Tello Edu como ferramenta integrada para o ensino de programação, visão computacional e inteligência computacional / Washington Henrique Alves Júnior. — Manaus, 2025. 62f.: il. color.</p> <p>Monografia (Graduação) — Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, <i>Campus</i> Manaus Distrito Industrial, Curso de Engenharia de Controle e Automação, 2025. Orientador: Prof. Vítor Bremgartner da Frota, Dr.</p> <p>1. Visão computacional. 2. Inteligência computacional. 3. Python. I. Frota, Vítor Bremgartner da. II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.</p> <p style="text-align: right;">CDD 629.89</p> |
|-------|---|

Elaborada por Oziane Romualdo de Souza (CRB11/ nº 734).

WASHINGTON HENRIQUE ALVES JUNIOR

**USO DO DRONE TELLO EDU COMO FERRAMENTA INTEGRADA PARA O
ENSINO DE PROGRAMAÇÃO, VISÃO COMPUTACIONAL E
INTELIGÊNCIA COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado ao Corpo Docente do Departamento de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Campus Manaus - Distrito Industrial, como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação.

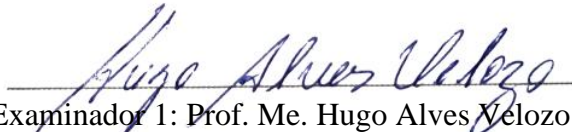
Orientador: Prof. Dr. Vitor Bremgartner da Frota

Manaus, 13 de Março de 2025.

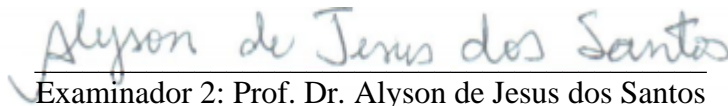
BANCA EXAMINADORA



Orientador: Prof. Dr. Vitor Bremgartner da Frota
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM)



Examinador 1: Prof. Me. Hugo Alves Vellozo
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM)



Examinador 2: Prof. Dr. Alyson de Jesus dos Santos
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM)

MANAUS-AM
2025

AGRADECIMENTOS

Aos meus familiares, expresso minha mais profunda gratidão pelo apoio incondicional durante toda esta jornada. Agradeço especialmente à minha mãe, Rosilda, e à minha companheira, Karen, por acreditarem no meu potencial e por incentivarem meu esforço contínuo.

Aos projetos de Iniciação Científica fomentados pela FAPEAM, que me permitiram construir uma carreira científica como pesquisador. Foi através do meu primeiro projeto que percebi que estava fazendo ciência de verdade. Ao Projeto Aranouá, meu sincero agradecimento, pois me proporcionaram a base para prosseguir na carreira acadêmica, por meio dos cursos e monitorias dos quais tive a oportunidade de participar.

Aos professores que marcaram minha trajetória acadêmica, agradeço de forma especial ao professor Vitor Bremgartner, cuja confiança e orientação me permitiram iniciar na pesquisa científica, quando ainda era apenas um aprendiz. Estendo meus agradecimentos aos demais docentes, em especial aos professores Wagner Nunes, Hillerman Lima, Alyson Santos e Hugo Velozo, e, que, com seus ensinamentos – mesmo nos momentos mais difíceis das disciplinas mais desafiadoras – me motivaram a superar desafios e a continuar fazendo o meu melhor, contribuindo para que eu possa me tornar o pesquisador e profissional que almejo ser.

Aos meus colegas do IFAM, do curso de Engenharia e das demais graduações, foi uma honra lutar ao lado dos senhores; orgulhosamente, combatemos lado a lado e marchamos com bravura rumo à conquista do saber.

“Entre os mais fortes existem os
que nasceram com um dom, e aqueles
que trabalham duro... E eu sou um dos
que trabalhou duro!”

(Rock Lee)

RESUMO

O presente trabalho aborda a utilização do drone Tello EDU como ferramenta integrada ao ensino de programação, visão computacional e inteligência computacional no Instituto Federal do Amazonas (IFAM), Campus Manaus Distrito Industrial. Tem como objetivo geral demonstrar o potencial educacional do drone Tello EDU, evidenciando sua contribuição para o desenvolvimento do pensamento computacional e para a aplicação prática de conteúdos nas respectivas disciplinas. A metodologia utilizada baseia-se em revisão bibliográfica e na análise crítica de dados e experimentos prévios realizados com estudantes dos cursos técnico e superior em Mecatrônica e Eletrônica. Inicialmente, o ensino de programação foi abordado com a linguagem visual Scratch, destacando a facilidade com que os alunos desenvolveram lógica de programação e algoritmos básicos, o que resultou em alto engajamento e motivação nas atividades. Posteriormente, a pesquisa propôs a transição para a linguagem textual Python, permitindo realizar tarefas mais complexas, tais como voos autônomos e sequências avançadas de comandos utilizando a biblioteca DJITelloPy.

Além disso, o estudo explorou a aplicação prática das bibliotecas OpenCV e MediaPipe, possibilitando o desenvolvimento de um sistema de reconhecimento de gestos para controlar o drone. A biblioteca DJITelloPy foi utilizada como intermediária para comunicação e controle do drone, enquanto o OpenCV e o MediaPipe foram empregados para captura, processamento de imagens e reconhecimento de gestos em tempo real. Os resultados dos experimentos prévios mostraram a viabilidade da proposta integrada, oferecendo aos estudantes uma interface intuitiva e dinâmica para o aprendizado prático e significativo de conteúdos relacionados a programação, visão computacional e inteligência computacional.

Conclui-se que o drone Tello EDU é uma ferramenta eficiente para o estímulo ao pensamento computacional, propiciando uma abordagem gradativa do aprendizado de programação, desde linguagens visuais até ambientes mais complexos como o Python, e contribuindo para a formação de habilidades técnicas e cognitivas essenciais no século XXI.

Palavras-chave: Drone Educacional. Pensamento Computacional. Ensino de Programação. Python. Visão Computacional. Inteligência Computacional.

ABSTRACT

This study addresses the utilization of the Tello EDU drone as an integrated educational tool for teaching programming, computer vision, and computational intelligence at the Instituto Federal do Amazonas (IFAM), Manaus Distrito Industrial campus. The general aim is to demonstrate the educational potential of the Tello EDU drone, highlighting its contribution to developing computational thinking and facilitating the practical application of related disciplinary content. The adopted methodology comprises a literature review and a critical analysis of data and previous experiments. Initially, the research focused on visual programming through Scratch, revealing significant student engagement and interest due to the ease and interactivity of the activities. Subsequently, the project introduced Python as a textual programming language, enabling students to execute more complex tasks, such as autonomous flights and advanced command sequences using the DJITelloPy library.

Additionally, the study explored practical applications using OpenCV and MediaPipe libraries to implement gesture-based control systems. Results from prior experiments confirmed the viability of this integrated approach, providing students with an intuitive and dynamic learning interface, promoting practical and meaningful learning experiences in programming, computer vision, and computational intelligence.

In conclusion, the Tello EDU drone proved to be an effective resource for fostering computational thinking and enhancing practical understanding and application in programming, computer vision, and computational intelligence. This approach significantly contributed to interactive and meaningful learning experiences.

Keywords: Educational Drones. Computational Thinking. Programming Education. Python. Computer Vision. Computational Intelligence.

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1 - Drone DJI Tello..... | 28 |
| FIGURA 2 – Protetores de hélice e detector de colisão | 29 |
| FIGURA 3 – Sistema de Posicionamento Visual..... | 30 |
| FIGURA 4 - Tello Boost Combo | 31 |
| FIGURA 5 - Ambiente Virtual de Simulação | 33 |
| FIGURA 6 – Repositório do GitHub: biblioteca DJITelloPy. | 34 |
| FIGURA 7 - Repositório Tello-Python | 35 |
| FIGURA 8 – Aplicações de Visão com MediaPipe | 37 |
| FIGURA 9 - MediaPipe Hands (Landmakrks)..... | 38 |
| FIGURA 10 - Blocos de Programação de Movimento | 40 |
| FIGURA 11 - Blocos de Programação de Controle | 41 |
| FIGURA 12 - Exemplo de Algoritmo com Blocos de Movimento e Controle..... | 41 |
| FIGURA 13 - (a) Engajamento de alunos nas atividades práticas. (b) Aplicação dos conceitos teóricos. | 43 |
| FIGURA 14 - Instalação da biblioteca DJITelloPy pelo Pycharm..... | 44 |
| FIGURA 15 – Estabelecendo Conexão com o Drone | 45 |
| FIGURA 16 - (a) Programação em blocos (b) Programação em Python | 45 |
| FIGURA 17 - (a) Programação em blocos com laço de Repetição (b) Programação em Python com Laço de Repetição | 46 |
| FIGURA 18 - (a) Programação em Blocos com delay (b) Programação em Python com delay | 46 |
| FIGURA 19 - Conversão para Escala de Cinza | 51 |
| FIGURA 20 - Imagem com detecção de bordas Canny | 52 |
| FIGURA 21 – Imagem Segmentada e com Contorno..... | 53 |
| FIGURA 22 - Instalação de biblioteca MediaPipe..... | 54 |
| FIGURA 23 - MediaPipe Hands pela câmera do Drone Tello Edu | 55 |
| FIGURA 24 - Gestos e Comandos | 56 |
| FIGURA 25 – (a) Comando: Back (b) Comando: Stop (c) Comando: Left..... | 57 |

SUMÁRIO

| | | |
|-------|---|----|
| 1. | INTRODUÇÃO | 11 |
| 1.1 | CONTEXTUALIZAÇÃO E DEFINIÇÕES | 11 |
| 1.2 | INOVAÇÃO E RELEVÂNCIA PEDAGÓGICA..... | 11 |
| 1.3 | DESAFIOS NA TRANSIÇÃO E ESTRATÉGIAS PEDAGÓGICAS..... | 12 |
| 1.4 | OBJETIVOS DO TRABALHO | 13 |
| 1.5 | ORGANIZAÇÃO DO TRABALHO | 14 |
| 2. | REFERENCIAL TEÓRICO | 15 |
| 2.1 | PENSAMENTO COMPUTACIONAL | 15 |
| 2.2 | ENSINO DE PROGRAMAÇÃO | 17 |
| 2.3 | VISÃO COMPUTACIONAL E INTELIGÊNCIA COMPUTACIONAL: PRINCÍPIOS E APLICAÇÕES | 19 |
| 2.4 | DRONES NA EDUCAÇÃO | 21 |
| 3. | TRABALHOS RELACIONADOS | 24 |
| 3.1 | DRONES E PROGRAMAÇÃO EM CONTEXTOS EDUCACIONAIS..... | 24 |
| 3.2 | APLICAÇÕES DO TELLO EM CONTROLE, VISÃO COMPUTACIONAL E METODOLOGIAS ATIVAS | 24 |
| 3.3 | CONTROLE POR GESTOS E VISÃO COMPUTACIONAL | 25 |
| 3.4 | OUTRAS PLATAFORMAS E COMPARAÇÕES | 26 |
| 3.5 | CONSIDERAÇÕES SOBRE ESTUDOS RELACIONADOS | 26 |
| 4. | MATERIAIS E METODOS | 28 |
| 4.1 | HARDWARE | 28 |
| 4.1.1 | Drone Tello Edu | 28 |
| 4.2 | SOFTWARE | 31 |
| 4.2.1 | Python | 31 |
| 4.2.2 | Aplicativo Tello Edu | 32 |
| 4.2.3 | Biblioteca DJITelloPy | 33 |
| 4.2.4 | OpenCV | 35 |
| 4.2.5 | MediaPipe (Hands Gesture Control) | 36 |
| 4.3 | METODOLOGIA..... | 38 |
| 5. | ESTUDO DE CASO 1 | 40 |
| 5.1 | PROGRAMAÇÃO DO DRONE COM BLOCOS DE PROGRAMAÇÃO | 40 |
| 5.2 | PROGRAMAÇÃO DO DRONE COM PYTHON | 43 |

| | | |
|-----------|--|----|
| 6. | ESTUDO DE CASO 2 | 50 |
| 6.1 | APLICAÇÃO EM VISÃO COMPUTACIONAL..... | 50 |
| 6.1.1 | Captura de vídeo..... | 50 |
| 6.1.2 | Conversão para escala de cinza | 50 |
| 6.1.3 | Deteccção de bordas com Canny..... | 51 |
| 6.1.4 | Contornos e Segmentação de Objetos..... | 52 |
| 6.2 | APLICAÇÃO EM INTELIGÊNCIA COMPUTACIONAL | 53 |
| 6.2.1 | Configuração do Ambiente | 54 |
| 6.2.2 | Reconhecimento de Gestos..... | 54 |
| 6.2.3 | Implementação Final..... | 56 |
| 7. | CONCLUSÃO | 60 |
| | REFERÊNCIAS | 62 |
| | APÊNDICE | 68 |
| | APÊNDICE A – CAPTURA DE VÍDEO COM DRONE TELLO | 68 |
| | APÊNDICE B - CAPTURA DE VÍDEO EM TONS DE CINZA | 69 |
| | APÊNDICE C - DETECÇÃO DE BORDAS COM CANNY | 70 |
| | APÊNDICE D – CONTORNO E SEGMENTAÇÃO DE OBJETOS | 71 |
| | APÊNDICE E – DETECÇÃO DE MÃOS PELO DRONE | 72 |
| | APÊNDICE F – QR CODE PARA PLAYLIST DA APLICAÇÃO | 74 |
| | ANEXOS | 75 |
| | ANEXO A – ATIVIDADES REALIZADAS PELO ALUNO A | 75 |
| | ANEXO B – ATIVIDADES REALIZADAS PELO ALUNO B | 77 |
| | ANEXO C – ATIVIDADES REALIZADAS PELO ALUNO C | 79 |
| | ANEXO D – ATIVIDADES REALIZADAS PELO ALUNO D | 80 |
| | ANEXO E – ATIVIDADES REALIZADAS PELO ALUNO E | 83 |

1. INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E DEFINIÇÕES

Nos últimos anos, o termo drone tornou-se amplamente conhecido para designar qualquer aeronave não tripulada, comumente empregada em diferentes setores, como entretenimento, agricultura, infraestrutura e segurança (OLIVEIRA, 2022). Tecnicamente, a Agência Nacional de Aviação Civil (ANAC) utiliza a denominação Veículos Aéreos Não Tripulados (VANT) para designar equipamentos que possam operar sem a presença de um piloto a bordo, e distingue duas categorias principais: as aeronaves remotamente pilotadas (RPA), que se destinam a aplicações comerciais ou profissionais, e os aeromodelos, normalmente associados a fins de recreação ou lazer (ANAC, 2024).

O mercado de drones vive um crescimento exponencial, alavancado pela redução de custos de produção e pela diversidade de aplicações. De acordo com Oliveira (2022), o cenário brasileiro é particularmente promissor, pois, embora ainda seja menor quando comparado a países como Estados Unidos e Japão, mostra forte expansão em áreas como mapeamento, inspeções industriais, segurança e agricultura. Ao mesmo tempo, empresas especializadas desenvolvem modelos cada vez mais avançados, dotados de sistemas de GPS, câmeras de alta resolução e sensores de desvio de obstáculos, tornando as operações mais precisas e seguras (ANAC, 2024).

Vale ressaltar que, segundo a regulamentação da ANAC (2024), os drones com peso máximo de decolagem de até 250 gramas, quando utilizados exclusivamente para recreação, são classificados como aeromodelos. O drone Tello, cujo peso está bem abaixo desse limite, enquadra-se, portanto, na categoria de aeromodelo caso seja utilizado para fins recreativos. Tal definição é importante, pois orienta o tipo de cadastro ou registro necessário, bem como os cuidados a serem observados nas operações.

1.2 INOVAÇÃO E RELEVÂNCIA PEDAGÓGICA

O presente trabalho investiga o uso do drone Tello EDU como uma ferramenta inovadora para o ensino de programação e para o desenvolvimento do pensamento computacional. Em um cenário onde a educação precisa se adaptar às demandas da sociedade digital, o Tello EDU surge não apenas como um dispositivo de voo, mas como um recurso

pedagógico capaz de transformar conceitos abstratos em experiências tangíveis. Por meio de sua interface visual – semelhante à utilizada no Scratch – os alunos podem experimentar a construção de algoritmos e a lógica da programação de maneira lúdica e interativa, tornando o aprendizado mais significativo e estimulante.

Essa proposta está alinhada com os princípios defendidos por Seymour Papert, que, já na década de 1980, destacou a importância de utilizar tecnologias interativas para estimular a experimentação e a construção de conhecimento. Em “Mindstorms: Children, Computers, and Powerful Ideas”, Papert argumenta que, quando as crianças interagem com ferramentas tecnológicas, elas constroem modelos mentais que facilitam a compreensão de conceitos complexos (PAPERT, 1980). Ao aplicar essa ideia, o Tello EDU pode ser visto como um “objeto para pensar”, que permite aos estudantes não só aprender a programar por meio de blocos, mas também desenvolver uma mentalidade que os prepara para desafios mais avançados, como a transição para linguagens textuais como Python.

Além disso, a relevância do tema ganha ainda mais força quando consideramos a perspectiva do pensamento computacional. Jeannette Wing (2006) argumenta que o pensamento computacional envolve resolver problemas, projetar sistemas e compreender o comportamento humano a partir de conceitos fundamentais da ciência da computação – habilidades que vão muito além do simples ato de programar (WING, 2006). Assim, a utilização do Tello EDU não se resume à execução de comandos, mas se configura como uma estratégia para desenvolver competências analíticas e criativas essenciais para o século XXI.

Em paralelo, o avanço das políticas educacionais, como a Base Nacional Comum Curricular, reforça a necessidade de incorporar tecnologias que estimulem não apenas o aprendizado técnico, mas também o pensamento crítico e a capacidade de resolver problemas de forma colaborativa. Nesse contexto, o uso do drone Tello EDU representa uma proposta que integra o ensino de programação com a prática do pensamento computacional, ampliando as oportunidades de aprendizagem e preparando os alunos para um ambiente profissional cada vez mais orientado pela inovação digital (GUARDA e PINTO, 2020).

1.3 DESAFIOS NA TRANSIÇÃO E ESTRATÉGIAS PEDAGÓGICAS

Apesar do sucesso de tais ambientes no estágio inicial de aprendizagem, a transição para linguagens textuais, como Python, representa um desafio significativo para os estudantes. Estudos (por exemplo, Mladenović, Karpan e Mladenović, 2016; Moraes et al., 2016) indicam que, embora a abordagem visual seja motivadora, muitos alunos enfrentam uma lacuna de

conhecimento quando precisam lidar com a complexidade das linguagens textuais. A transição entre ambientes visuais e linguagens textuais apresenta desafios significativos para os alunos, especialmente relacionados à memorização de sintaxe e interpretação de erros (MAJEED et al., 2022).

Esses desafios fundamentam a importância de adotar estratégias pedagógicas e ferramentas intermediárias que facilitem essa passagem e que possam contribuir diretamente para um aprendizado mais consistente e eficaz. Nosso trabalho propõe ampliar as pesquisas anteriores, integrando uma estratégia progressiva que inicia com o ensino de programação via Scratch e evolui para o Python. Essa progressão visa não apenas consolidar os fundamentos de lógica e algoritmos, mas também preparar os alunos para a codificação em ambientes textuais, os quais demandam maior precisão e compreensão da sintaxe.

1.4 OBJETIVOS DO TRABALHO

O presente trabalho tem como foco a ampliação da pesquisa realizada no artigo “Uso de drones para ensino de Lógica de Programação e estímulo do Pensamento Computacional”. Enquanto aquele estudo demonstrou o potencial motivacional dos drones no estímulo do pensamento computacional, aqui propomos integrar, de forma progressiva, o ensino de programação: iniciando com a abordagem lúdica e visual oferecida pelo Scratch e evoluindo para a linguagem textual Python. Essa estratégia visa não apenas manter o engajamento dos alunos, mas também preparar os estudantes para o ambiente de programação mais complexo que encontrarão em contextos avançados (MENEZES, 2010).

Além disso, foi incorporada a aplicação de visão computacional – por meio das bibliotecas OpenCV e Mediapipe – para implementar o controle por gestos do drone Tello EDU. Essa integração tem o potencial de transformar a experiência de aprendizagem ao oferecer uma interface mais natural e interativa, onde os alunos podem, por exemplo, manipular fisicamente o drone para verificar os efeitos de seus comandos. Dessa forma, o uso do Tello EDU não se limita à execução de voos, mas passa a ser um recurso que interliga a programação com a percepção visual e motora, contribuindo para o desenvolvimento de uma compreensão mais holística dos conceitos de programação e do pensamento computacional.

A delimitação deste estudo concentra-se na análise dos benefícios e dos desafios decorrentes da integração dessas tecnologias no ensino de programação. Em vez de realizar novas aplicações práticas com turmas, a pesquisa baseia-se na análise crítica dos dados e experimentos previamente realizados durante minha experiência em projetos finais em

disciplinas e pesquisas de Iniciação Científica do curso de Engenharia de Controle e Automação do Instituto Federal do Amazonas - IFAM, complementados por uma revisão teórica que evidencia as potencialidades e as limitações da abordagem proposta. Assim, o foco recai sobre como a progressão do ensino – partindo do ambiente visual do Scratch, passando pela evolução para Python, e culminando na aplicação de técnicas de visão computacional para controle por gestos – pode oferecer uma experiência educativa mais integrada e motivadora para os alunos.

Para atingir esses objetivos, a metodologia adotada combina uma revisão bibliográfica com a análise crítica de dados e experimentos prévios. Inicialmente, serão examinadas as principais contribuições teóricas e estudos de caso que fundamentam a integração entre ambientes de programação visual e textuais, bem como a aplicação de visão computacional no controle de drones. Em paralelo, será realizada uma análise dos dados coletados durante os projetos finais realizados no IFAM, os quais oferecem evidências práticas dos impactos e desafios dessa abordagem. Essa análise mista permite a identificação de padrões, benefícios e limitações, preparando o terreno para uma discussão aprofundada nos capítulos seguintes, onde serão detalhados os procedimentos metodológicos.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em sete capítulos principais. Após esta Introdução (Capítulo 1), o Capítulo 2 apresenta o referencial teórico, discutindo os fundamentos, conceitos e métodos que sustentam a proposta, com ênfase nas abordagens de ensino de programação, no desenvolvimento do pensamento computacional e na aplicação de tecnologias inovadoras no ambiente educacional. No Capítulo 3, são revisados os trabalhos relacionados, evidenciando pesquisas e experiências que dialogam com o uso do Tello EDU e outras tecnologias interativas. O Capítulo 4 descreve os materiais e métodos empregados, detalhando os equipamentos, softwares e a metodologia adotada para a realização dos experimentos. Em seguida, o Capítulo 5 apresenta o primeiro estudo de caso, analisando a aplicação do Tello EDU no ensino de programação – desde a abordagem visual com o Scratch até a proposta de transição para a linguagem Python – e os impactos dessa estratégia no estímulo ao pensamento computacional. O Capítulo 6 foca no segundo estudo de caso, detalhando a implementação de aplicações práticas do Tello EDU em disciplinas de visão computacional e inteligência computacional. Por fim, o Capítulo 7 reúne as considerações finais, nas quais são discutidos os principais resultados, as contribuições do trabalho, os desafios enfrentados e as perspectivas para futuras pesquisas.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento teórico que sustenta o desenvolvimento deste trabalho, organizado em quatro grandes eixos. Primeiramente, discorre-se sobre o Pensamento Computacional, ressaltando conceitos, relevância e estratégias de aplicação em contexto educacional. Em seguida, analisa-se o Ensino de Programação, abordando desde metodologias específicas até os principais desafios na transição entre linguagens visuais (como o Scratch) e textuais (como o Python). Posteriormente, são explorados os fundamentos de Visão Computacional e Inteligência Computacional, com ênfase nas técnicas de processamento de imagem suportadas por ferramentas como OpenCV e Mediapipe, incluindo exemplos de aplicação em controle por gestos. Por fim, discute-se o papel dos Drones na Educação, evidenciando suas potencialidades como recurso interativo de ensino, com destaque para o drone Tello EDU.

2.1 PENSAMENTO COMPUTACIONAL

O termo Pensamento Computacional (PC) ganhou maior visibilidade a partir dos trabalhos de Jeannette Wing, que o descreve como a habilidade de “resolver problemas, projetar sistemas e compreender o comportamento humano baseando-se em conceitos fundamentais da computação” (WING, 2006, p. 33). Esse tipo de pensamento não se restringe à capacidade de programar ou ao uso de computadores; envolve, sobretudo, um conjunto de práticas cognitivas que englobam abstração, decomposição de problemas, identificação de padrões, elaboração de algoritmos e avaliação de soluções (WING, 2008). Dessa forma, o PC tem sido considerado uma habilidade essencial no século XXI, pois fortalece competências de resolução de problemas e de criatividade, aplicáveis em diversas áreas.

O interesse pelo Pensamento Computacional não surgiu de maneira isolada. Seus antecedentes remontam aos estudos de Seymour Papert, que explorou como o aprendizado pode ser apoiado pela tecnologia. Papert (1980), ao criar a linguagem LOGO e introduzir “turtles” para crianças, já apontava que, ao “conversar” com o computador, o aprendiz desenvolvia novas formas de raciocínio e de construção de conhecimento. Essa perspectiva se alinha ao construcionismo, em que Papert adapta e expande conceitos de Piaget (1964), destacando o papel ativo da criança na formação de estruturas mentais. Em *Mindstorms*, Papert (1980) mostra como o computador pode ser um “objeto para pensar” (object-to-think-with), combinando o

aspecto lúdico ao desafio intelectual, de modo que crianças possam formular e testar hipóteses de forma autônoma.

Seguindo essa trilha, autores como Resnick e Brennan aprofundaram o debate sobre os elementos constitutivos do Pensamento Computacional. Brennan e Resnick (2012) propuseram uma estrutura baseada em três dimensões principais: (1) Conceitos Computacionais, envolvendo sequências, laços (loops), eventos, paralelismo, operadores, variáveis e condições; (2) Práticas Computacionais, relativas a estratégias de projeto como experimentação, teste de hipóteses, depuração (debugging) e remix de soluções; e (3) Perspectivas Computacionais, relacionadas à forma como o indivíduo se enxerga diante da tecnologia, desenvolvendo autoconfiança e identidade criadora na resolução de problemas.

De modo similar, em uma revisão sistemática, Guarda e Pinto (2020) mostram que, embora as definições e enfoques acerca do Pensamento Computacional sejam variados, todas convergem na importância de habilidades como abstração, decomposição e identificação de padrões para a resolução de problemas. Além disso, os autores discutem a viabilidade de integrar tais dimensões em um modelo teórico unificado, voltado a sustentar práticas pedagógicas que promovam o desenvolvimento do PC desde os anos iniciais. O objetivo, dessa forma, não se limita à formação de competências tecnológicas, mas também ao fortalecimento de estratégias cognitivas essenciais em uma sociedade em constante transformação.

Atualmente, discute-se a importância de que esse modo de pensar, mais do que uma competência técnica, seja visto como algo transversal. No contexto brasileiro, por exemplo, a Base Nacional Comum Curricular (BNCC) enfatiza a compreensão, uso e criação de tecnologias digitais como parte das competências gerais, destacando a resolução de problemas e o protagonismo na produção de conhecimento (BRASIL, 2018). Nesse sentido, autores como Yadav et al. (2014) argumentam que o Pensamento Computacional deve fazer parte da formação de professores, a fim de que práticas pedagógicas sejam capazes de integrar de forma crítica e reflexiva esses conceitos ao currículo. Assim como fazer parte da formação de professores, favorecendo a criação de estratégias didáticas capazes de despertar o interesse do aluno por processos investigativos e práticas científicas (ABRANTES, 2023).

Outro ponto essencial é que o PC não se resume ao coding. Embora a programação seja uma poderosa ferramenta para exercitar o raciocínio computacional, é possível desenvolver esse pensamento sem necessariamente escrever código, como mostram iniciativas de atividades desconectadas. Contudo, a programação costuma potencializar o aprendizado de conceitos fundamentais (condicionais, laços de repetição, modularização) e estimula práticas como

depurar o código ou refatorar algoritmos, o que torna o processo de aprendizagem tangível e interativo (WING, 2006; BRENNAN e RESNICK, 2012).

Em síntese, o PC pode ser compreendido como um conjunto de práticas (BRENNAN e RESNICK, 2012) que envolve tanto conceitos computacionais (estruturas de dados, algoritmos, abstração, paralelismo) quanto atitudes (curiosidade, persistência, colaboração), integrando saberes das ciências exatas, das ciências humanas e de contextos socioculturais diversos. Seu potencial educativo está em promover a apropriação de estratégias de pensamento que extrapolam o contexto da computação, contribuindo para o desenvolvimento de “hábitos de mente” essenciais na sociedade contemporânea (WING, 2006).

2.2 ENSINO DE PROGRAMAÇÃO

O ensino de Programação de Computadores tem passado por uma evolução significativa ao longo do tempo, em resposta às mudanças tecnológicas e às demandas formativas de diferentes faixas etárias e perfis de curso (desde o nível técnico até a graduação) (JESUS; SANTANA; BITTENCOURT, 2019). A promulgação de diretrizes como a Base Nacional Comum Curricular (BNCC) no Brasil, que enfatiza a cultura digital e as práticas de Pensamento Computacional, reforça a importância de abordagens pedagógicas condizentes com as necessidades contemporâneas de formação (JESUS; SANTANA; BITTENCOURT, 2019).

Nesse cenário, o ensino de programação deixa de se restringir à abordagem estritamente teórica ou a linguagens complexas para iniciantes, e passa a incorporar metodologias ativas (aprendizagem baseada em projetos, aprendizagem colaborativa, gamificação etc.). Tais metodologias buscam ampliar a motivação e o engajamento dos estudantes, o que é particularmente relevante considerando-se os desafios históricos de evasão e reprovação em disciplinas de programação (MORAES et al., 2016).

A adoção de ferramentas lúdicas e visualmente interativas, como o Scratch, tem sido apontada como uma estratégia eficaz para introduzir conceitos de programação. O Scratch se destaca por ser uma linguagem visual baseada em blocos, na qual o estudante pode desenvolver algoritmos por meio da montagem de peças que se encaixam (MARJI, 2014). Essa característica possibilita focar mais na lógica e no raciocínio computacional do que em detalhes de sintaxe, reduzindo dificuldades iniciais (JESUS; SANTANA; BITTENCOURT, 2019). Diversas atividades, como criação de jogos, animações e narrativas interativas, funcionam como

exemplos de aprendizagem ativa, pois permitem ao discente descobrir soluções e depurar problemas em um contexto lúdico (MLADENOVIC; KRPAN; MLADENOVIC, 2016).

Conforme o aluno avança, torna-se importante a transição para linguagens textuais de maior propósito geral, de modo que projetos mais robustos possam ser desenvolvidos (DORLING; WHITE, 2015). Em relação à transição de linguagens visuais para textuais, Majeed et al. (2022) destacam que a utilização de ambientes intermediários, como o CodeStruct, pode reduzir significativamente barreiras de aprendizagem, como a dificuldade em memorizar comandos textuais e interpretar erros de sintaxe. Ao disponibilizar mecanismos contextuais inteligentes e sugestões automáticas de correção de código, tais ferramentas aumentam a autonomia do estudante e reduzem a frequência de pedidos de ajuda ao professor, facilitando uma transição mais suave e eficaz para linguagens textuais. Nesse ponto, a linguagem Python tem se mostrado uma opção viável, pois combina sintaxe relativamente simples e clara com recursos que possibilitam a criação de sistemas mais complexos (MENEZES, 2014). O Python também favorece o aprendizado progressivo de estruturas como laços de repetição, condicionais, funções e manipulação de variáveis, que podem ser correlacionadas aos blocos do Scratch, porém agora em um ambiente textual (MARJI, 2014; JESUS; SANTANA; BITTENCOURT, 2019).

Desse modo, há um percurso formativo sugerido na literatura em que se inicia a formação do Pensamento Computacional com uma linguagem de blocos (p.ex. Scratch) e, gradativamente, evolui-se para uma linguagem textual (p.ex. Python). Esse percurso facilita a compreensão de conceitos fundamentais – como sequência, decisão, repetição e abstração de problemas – antes de se lidar com possíveis frustrações típicas do paradigma textual (JESUS; SANTANA; BITTENCOURT, 2019; MENEZES, 2014). Além disso, o Pensamento Computacional na prática inclui não apenas a programação em si, mas também a compreensão e análise de problemas, a decomposição em partes menores, a identificação de padrões e a criação de algoritmos para solucioná-los, bem como a avaliação e a correção de erros (MARJI, 2014).

A utilização de projetos como estratégia de aprendizagem (aprendizagem baseada em projetos) e a gamificação são exemplos de metodologias que colaboram na manutenção do interesse e na interação ativa do estudante. O estímulo à criatividade e à criação de produtos palpáveis (como jogos ou ferramentas de uso real) colabora para que os discentes desenvolvam confiança em sua capacidade de programar, reduzindo índices de evasão (JESUS; SANTANA; BITTENCOURT, 2019). Apesar disso, alguns desafios persistem: muitos estudantes encontram barreiras na transição de blocos para código textual, lidando com novas questões de sintaxe e

depuração de erros (MLADENović, KRPAÑ e MLADENović, 2016). A abordagem docente, portanto, precisa prover suporte e atividades guiadas que auxiliem o discente a compreender e a superar tais lacunas conceituais (JESUS; SANTANA; BITTENCOURT, 2019; MORAES et al., 2016).

Em síntese, o ensino de programação necessita articular recursos didáticos adequados, metodologias ativas e ferramentas de desenvolvimento que sejam acessíveis e motivadoras. O Scratch, com seu viés lúdico e intuitivo, permite a introdução progressiva de conceitos e estimula a criatividade do aprendiz. Em seguida, Python, por ser uma linguagem de uso geral e sintaxe relativamente amigável, complementa o processo, possibilitando projetos mais complexos e reforçando a prática do Pensamento Computacional. Assim, viabiliza-se uma formação coerente com os objetivos formativos da BNCC e com as realidades de diferentes perfis e faixas etárias nos cursos técnicos ou de graduação (MENEZES, 2014; JESUS; SANTANA; BITTENCOURT, 2019).

2.3 VISÃO COMPUTACIONAL E INTELIGÊNCIA COMPUTACIONAL: PRINCÍPIOS E APLICAÇÕES

A área de Visão Computacional (VC) tem como objetivo desenvolver métodos e algoritmos capazes de extrair, analisar e interpretar informações contidas em imagens ou sequências de imagens (SZELISKI, 2010). Em essência, busca-se reproduzir computacionalmente a capacidade humana de perceber e compreender cenas visuais, permitindo a criação de sistemas que vão desde simples detectores de bordas até soluções complexas de reconhecimento de objetos e ambientes. Gonzalez e Woods (2008) destacam que o processamento digital de imagens – parte fundamental da Visão Computacional – abrange etapas como filtragem, realce, restauração, segmentação e descrição de características, sendo aplicado em domínios tão variados quanto análise médica, segurança, inspeção industrial e entretenimento.

No contexto de interação humano-computador, o reconhecimento de gestos é uma das aplicações mais relevantes da Visão Computacional, pois permite o controle de dispositivos sem a necessidade de contato físico ou interfaces tradicionais, como teclados e mouses (MITRA e ACHARYA, 2007). Para isso, técnicas de processamento de imagens são empregadas a fim de identificar e acompanhar características de interesse – por exemplo, mãos e articulações – ao longo do tempo. Essa identificação envolve estágios de captura, segmentação, rastreamento

e, por fim, classificação de gestos ou posturas (SZELISKI, 2010; GONZALEZ e WOODS, 2008).

Uma das ferramentas de maior destaque para experimentos práticos na área de Visão Computacional é a biblioteca OpenCV (Open Source Computer Vision Library), amplamente utilizada devido ao seu conjunto de rotinas otimizadas para processamento de imagens e vídeos. Com ela, é possível efetuar operações como conversão de espaço de cor, detecção de bordas, extração de descritores, segmentação e, sobretudo, rastreamento de objetos ou pontos de interesse. Sua robustez e comunidade ativa de desenvolvimento a tornaram um padrão de fato em inúmeros projetos acadêmicos e industriais (SZELISKI, 2010).

No que tange à Inteligência Computacional (IC), seu foco reside em métodos inspirados em processos naturais ou heurísticos, como redes neurais, lógica fuzzy, algoritmos genéticos e sistemas híbridos (ENGELBRECHT, 2007). Uma das características primordiais da IC está na habilidade de lidar com incertezas, dados incompletos e cenários dinâmicos – algo essencial em sistemas que operam no mundo real, como os que envolvem Visão Computacional (SZELISKI, 2010). Em conjunto com algoritmos de aprendizado de máquina (machine learning), métodos de IC possibilitam o desenvolvimento de soluções autoadaptativas, capazes de melhorar o desempenho com o tempo, conforme são expostas a novos dados (ENGELBRECHT, 2007).

Paralelamente, surgem soluções mais especializadas, como o MediaPipe, um framework desenvolvido pelo Google Research com foco em criação de pipelines de percepção em tempo real (LUGARESI et al., 2019). O MediaPipe fornece modelos pré-treinados e componentes (“calculators”) que facilitam a detecção e o rastreamento de landmarks – por exemplo, posições-chave da mão ou do corpo – diretamente em vídeo. Em especial, o módulo MediaPipe Hands (ZHANG et al., 2020) realiza a localização de até 21 pontos na mão e antebraço em tempo real, o que possibilita a criação de sistemas interativos baseados em gestos. Em cenários educacionais, essa tecnologia viabiliza projetos de “controle por gestos”, onde a detecção da posição dos dedos ou da palma da mão é convertida em comandos para, por exemplo, mover robôs ou drones (HAN et al., 2022).

Assim, o uso de OpenCV e MediaPipe em conjunto permite a construção de aplicações que fazem todo o fluxo: (i) captura de frames de uma câmera; (ii) pré-processamento (ajuste de cor, redução de ruído etc.); (iii) detecção e rastreamento de mãos ou objetos de interesse; (iv) interpretação de eventos ou gestos; (v) envio de comandos a um sistema externo. Essa abordagem tem se mostrado promissora em aulas de Robótica, Inteligência Computacional e Sistemas de Visão, pois fornece recursos para experimentos práticos de maneira relativamente

simples, dispensando hardware muito complexo além de uma câmera convencional (LUGARESI et al., 2019; ZHANG et al., 2020).

2.4 DRONES NA EDUCAÇÃO

A robótica educacional, historicamente, abriu caminho para que dispositivos físicos se tornassem elementos centrais no processo de ensino-aprendizagem em áreas como Matemática, Física, Programação e Engenharias (BARONE e YEPES, 2018). Inicialmente, as atividades de robótica escolar se concentravam em kits de montagem terrestre e no uso de linguagens de programação que permitissem controlar pequenos robôs em cenários limitados. Contudo, a crescente disponibilidade de drones acessíveis e programáveis trouxe novas possibilidades, ampliando o potencial de motivação e experimentação prática nas salas de aula (YEPES, 2020).

Os drones se destacam como um recurso didático singular por despertarem forte engajamento nos estudantes, sobretudo pela combinação entre o desafio tecnológico e a possibilidade de executar tarefas de forma autônoma ou semiautônoma. Essa motivação inicial serve como porta de entrada para o desenvolvimento de habilidades de pensamento computacional, resolução de problemas e trabalho em equipe, alinhando-se aos objetivos de uma educação mais prática e interdisciplinar (ALVES e BREMGARTNER, 2022).

Dentro desse contexto, o Tello EDU surge como uma evolução notável: trata-se de um dispositivo voador compacto, de baixo custo e com câmera embutida, o que permite conectar a prática de voo com aplicações de visão computacional. Como afirmam Ghazi e Voyer (2024), “o DJI Tello [...] inclui um kit de desenvolvimento de software (SDK) que possibilita o controle do drone em diversas linguagens de programação, incluindo Python, além de contar com sensores que reforçam sua estabilidade em voo”. A facilidade de uso é outro ponto-chave, pois o Tello EDU pode ser controlado a partir de ambientes de programação em blocos (como o Scratch), além de oferecer suporte para linguagens textuais, fator que viabiliza tanto atividades introdutórias quanto projetos avançados (ALVES e BREMGARTNER, 2022; YEPES, 2020).

Essa adaptabilidade faz do Tello EDU uma plataforma que se destaca em relação a outras tecnologias já presentes em laboratórios de robótica ou de informática. Por exemplo, ao compará-lo a plataformas sobre rodas ou a kits mais tradicionais, percebe-se que um drone tem a vantagem de explorar o espaço em três dimensões, enriquecendo as possibilidades de experimentação em disciplinas como Física (estudo de vetores e cinemática), Matemática (funções espaciais e trigonometria) e Computação (desenvolvimento de algoritmos de

navegação). Em termos de custo-benefício, o Tello EDU oferece um equilíbrio interessante: seu preço é geralmente menor do que drones profissionais de maior porte, ao mesmo tempo em que mantém a robustez e a versatilidade necessárias para aplicações educacionais (GHASI e VOYER, 2024).

No que se refere ao ensino de programação, o Tello EDU dialoga com diferentes estágios de aprendizado. Em um primeiro momento, é possível empregar linguagens de programação em blocos, como Scratch, para criar sequências simples de voo, desenvolver noções de laços de repetição e condicionais, além de estimular o pensamento lógico (ALVES e BREMGARTNER, 2022). Conforme os estudantes avançam, o drone permite a transição para Python, habilitando projetos mais complexos, como voos autônomos que utilizam dados de sensores (acelerômetro, giroscópio) para estabilização, ou até a integração com bibliotecas de visão computacional (GHAZI e VOYER, 2024; YEPES, 2020). Essa progressão, de um ambiente visual para uma linguagem de texto, favorece o aprofundamento gradual dos conceitos de lógica e estrutura de dados, ao mesmo tempo em que preserva o caráter lúdico e interativo.

A presença de uma câmera embutida e a compatibilidade com bibliotecas como OpenCV ampliam ainda mais o alcance pedagógico, pois permitem o desenvolvimento de atividades que envolvem visão computacional (BARONE e YEPES, 2018). Os estudantes podem ser incentivados a criar algoritmos de detecção de cores, rastreamento de objetos ou reconhecimento de gestos, explorando na prática conceitos de inteligência computacional. Projetos desse tipo podem também promover discussões sobre ética e segurança no uso de drones, além de estimular a criatividade na busca de soluções inovadoras para problemas reais, como mapeamento de ambientes, inspeção de áreas e monitoramento de variáveis ambientais (VENTURA et. al., 2022).

Em termos de aprendizagem prática e experimentação, pesquisas mostram que o uso de drones em sala de aula gera engajamento em todos os níveis de ensino, desde a educação básica até a graduação e projetos de pesquisa (ALVES e BREMGARTNER, 2022; YEPES, 2020). Na educação superior, por exemplo, docentes de cursos de Engenharia e Computação têm adotado drones como plataformas de prototipagem e investigação científica, possibilitando desde estudos de modelagem e controle até aplicações de inteligência computacional em sistemas de navegação autônoma (CAÑAS et al., 2020). No ensino médio, conforme relatado por Alves e Bremgartner (2022), a introdução de um drone nas aulas de Lógica de Programação despertou interesse imediato, “pois a combinação entre o desafio da programação e a expectativa de ver o drone voar estimulou os estudantes a solucionar problemas de forma colaborativa”.

Com isso, percebe-se que a adoção do Tello EDU no contexto educacional pode contribuir significativamente para a formação de competências do século XXI, integrando teoria e prática, motivação e aprendizagem significativa. A facilidade de uso do drone, aliada à sua integração com linguagens de programação, custo acessível e flexibilidade de aplicações – que vão de voos autônomos até projetos de visão computacional – faz com que essa tecnologia represente uma “porta de entrada” efetiva para a robótica educativa e para a formação de futuros profissionais em áreas STEAM (Ciência, Tecnologia, Engenharia, Artes e Matemática). Em suma, o Tello EDU consolida-se como uma evolução natural no cenário da robótica pedagógica, abrindo caminho para experiências de aprendizagem mais criativas, colaborativas e alinhadas às demandas tecnológicas do presente e do futuro.

3. TRABALHOS RELACIONADOS

Nos últimos anos, o uso de drones em ambientes educacionais vem se consolidando como uma tendência promissora e vem sendo explorado em diversas iniciativas e pesquisas. Em particular, o modelo Tello (ou Tello EDU) tem-se destacado por apresentar boa estabilidade de voo, baixo custo e relativa facilidade de programação, o que possibilita sua adoção em diferentes níveis de ensino e áreas do conhecimento.

3.1 DRONES E PROGRAMAÇÃO EM CONTEXTOS EDUCACIONAIS

Petrovič e Verčimák (2023) descrevem a incorporação de drones programáveis em projetos e competições educacionais, enfatizando como esses dispositivos podem expandir o repertório de situações de aprendizagem em robótica. Os autores relatam o sucesso do uso de drones em desafios estudantis, em que participantes devem planejar rotas de voo, efetuar leituras de sensores e resolver pequenos problemas de navegação de forma autônoma. A experiência relatada destaca o Tello como um marco, pois sua estabilidade e simplicidade de conexão via Wi-Fi facilitam a adoção nas escolas.

No contexto do ensino a distância, Cañas e colaboradores (2020) propõem um curso aberto de programação de drones, voltado a estudantes de engenharia, ressaltando a relevância de uma abordagem prática mesmo em ambientes remotos. Embora eles façam uso de um ecossistema com simuladores e drones distintos (como Parrot e outras plataformas), diversas conclusões podem ser estendidas ao Tello EDU, haja vista a adoção de linguagens acessíveis (por exemplo, Python) e recursos de controle simples. Essa experiência reforça a importância de ambientes de prática, mesmo quando não se tem contato presencial constante com o drone.

3.2 APLICAÇÕES DO TELLO EM CONTROLE, VISÃO COMPUTACIONAL E METODOLOGIAS ATIVAS

Diversos trabalhos recentes têm explorado o Tello como uma plataforma de demonstração e experimentação no campo do controle e da robótica. Ghazi e Voyer (2024), por exemplo, relatam o uso do DJI Tello para reforçar conteúdos de engenharia de controle, integrando técnicas de identificação de sistemas e de projeto de controladores. Nesse projeto, os alunos conduzem experimentos de modelagem do drone e, em seguida, desenvolvem

controladores (por exemplo, proporcional-integrativo, PI) para estabilizar a aeronave em uma dada altura. Essa abordagem mostra como o Tello pode ser usado na prática, sem grandes riscos de danos ou elevados custos de manutenção.

García et al. (2022) investigaram o uso do DJI Tello como uma plataforma acessível para detecção, rastreamento e execução autônoma de trajetórias, utilizando técnicas avançadas de inteligência artificial para reconhecer e seguir objetos em tempo real. Os autores demonstraram que, mesmo com limitações intrínsecas ao hardware, é possível realizar atividades complexas ao integrar tecnologias externas como o sistema de posicionamento OptiTrack. Esses resultados validam e enriquecem a proposta apresentada neste trabalho, especialmente no que se refere ao potencial do drone Tello EDU como ferramenta educacional para o ensino de visão computacional, oferecendo suporte adicional às argumentações apresentadas sobre sua utilização prática em contextos acadêmicos e reforçando sua relevância pedagógica.

Uma vertente igualmente importante relaciona-se à preparação de estudantes para carreiras em um mercado em plena expansão. Lobo et al. (2021) desenvolveram uma abordagem baseada em metodologias de aprendizagem ativa (active learning) para familiarizar alunos com os fundamentos de programação, aerodinâmica básica e regulamentação de voos comerciais. Embora o estudo não seja restrito ao Tello, seus achados evidenciam que plataformas de voo de pequena escala, como o Tello EDU, podem servir de introdução efetiva antes que os estudantes lidem com aeronaves mais complexas.

3.3 CONTROLE POR GESTOS E VISÃO COMPUTACIONAL

Outro campo de crescente interesse é o controle de drones a partir de visão computacional, no qual técnicas de processamento de imagens e aprendizado de máquina permitem interações mais naturais — por exemplo, comandos gestuais. Saini (2021) descreve um sistema para manobrar drones Tello e Tello EDU usando poses corporais, empregando uma combinação de bibliotecas de visão (como MediaPipe) para detecção de pontos-chave e de algoritmos de decisão. De forma semelhante, Khoza, Owolabi e Malele (2022) apresentam uma solução para controlar o drone Tello via detecção de gestos com OpenCV, com ênfase em manter a estabilidade do voo ao mesmo tempo em que o sistema interpreta sinais da mão do operador.

Esses esforços também dialogam com trabalhos focados no uso de bibliotecas genéricas de visão computacional, como no caso de Latif, Buckley e Secco (2022), que exploram a

interação humano-drone por meio de reconhecimento de gestos, trazendo reflexões sobre aspectos de segurança, responsividade do sistema e possíveis aplicações práticas na área de entretenimento, vigilância ou mesmo suporte a usuários com restrições de mobilidade. Em paralelo, Rocha (2023) implementou técnicas de posicionamento autônomo de drones empregando visão computacional para identificar e rastrear alvos, demonstrando como o Tello consegue manter uma distância constante em relação a um objeto detectado ou a uma pessoa em movimento.

3.4 OUTRAS PLATAFORMAS E COMPARAÇÕES

Para além do Tello, Giernacki et al. (2017) propuseram o uso do Crazyflie 2.0 como plataforma aberta para ensino e pesquisa em engenharia de controle. Apesar das diferenças em termos de hardware e bibliotecas de controle em comparação ao Tello (o Crazyflie tem peso menor e uma arquitetura eletrônica modular), as conclusões acerca da eficácia de drones de pequeno porte no engajamento e na compreensão prática de técnicas de controle são bastante semelhantes. Em ambos os casos, a facilidade de reparo e as ferramentas de programação simples favorecem a adoção em disciplinas de robótica, controle e áreas afins.

Por fim, estudos focados em ensino de Lógica de Programação com drones também surgem como destaque. Em uma aplicação realizada em ambiente educacional, Alves e Bremgartner (2022) relatam que a adoção do Tello no contexto de ensino de programação em blocos (Scratch) contribuiu para o desenvolvimento do Pensamento Computacional e para o estímulo do engajamento dos alunos, pois eles conseguem observar de forma imediata os resultados de seus códigos no voo do drone. A proposta, de maneira geral, corrobora experiências anteriores que apontam para a importância de atividades práticas no aprendizado de conteúdos ligados à computação.

3.5 CONSIDERAÇÕES SOBRE ESTUDOS RELACIONADOS

A literatura revisada demonstra a ampla gama de aplicações que o Tello (e drones similares) pode desempenhar no âmbito educacional, abrangendo desde introduções básicas à programação até estudos mais avançados em controle e visão computacional. Em síntese, há um consenso de que essas plataformas promovem motivação e aprendizado prático, mas também existe a constatação de desafios: alguns trabalhos destacam questões relacionadas à

autonomia da bateria, à precisão dos sensores para aplicações em visão computacional e à eventual necessidade de espaço físico adequado para voos em segurança. Ainda assim, o conjunto de estudos analisados reforça a pertinência de integrar o Tello EDU no ensino de programação, lógica, automação e visão computacional, especialmente quando se busca aproximar os estudantes de problemas concretos, alinhados às demandas de um mercado cada vez mais tecnológico.

4. MATERIAIS E METODOS

Na elaboração deste trabalho, foram definidos os materiais e métodos que possibilitam a integração das tecnologias utilizadas no ensino de programação e no desenvolvimento do pensamento computacional. Foram empregados recursos de hardware, como o drone Tello ED, que oferece suporte para a realização de simulações e para a implementação prática de comandos. No âmbito do software, a programação foi realizada utilizando a linguagem Python em ambiente de desenvolvimento como o PyCharm, e foram adotadas bibliotecas específicas, como a DJITelloPy para o controle do drone, a OpenCV para o processamento de imagens e a MediaPipe para o reconhecimento de gestos. Essa abordagem metodológica permite a análise dos algoritmos implementados e a verificação da transição entre a programação visual e a programação textual, contribuindo para a avaliação das práticas pedagógicas aplicadas.

4.1 HARDWARE

4.1.1 Drone Tello Edu

Segundo o site da Ryze Robotics (2025), o drone Tello, desenvolvido pela Ryze Robotics em parceria com a DJI, é um quadricóptero compacto que integra a avançada tecnologia de voo da DJI com um sistema de posicionamento visual, operando via conexão Wi-Fi. Projetado para ser acessível e fácil de usar, o Tello é ideal tanto para entusiastas iniciantes quanto para projetos educacionais, proporcionando uma experiência prática para o ensino de programação e o desenvolvimento do pensamento computacional. Existem duas versões deste drone: o Tello tradicional e o Tello EDU, que compartilham o mesmo hardware, mas se diferenciam pelo firmware e pelas funcionalidades adicionais voltadas para o ambiente educacional. A Figura 1 apresenta o modelo do drone utilizado.

FIGURA 1 - Drone DJI Tello



Fonte: Ryze Robotics, 2025

A segurança do Tello para uso em sala de aula é assegurada por um conjunto de mecanismos que minimizam riscos durante o voo. O drone conta com um sistema de pouso automático que é acionado se não receber comandos em um intervalo de 15 segundos, evitando quedas descontroladas. Além disso, os protetores de hélice reduzem significativamente a possibilidade de ferimentos ou danos a objetos, o que é fundamental em ambientes internos, conforme ilustra a Figura 2. Observa-se que na Figura 3a que o uso do sistema de posicionamento visual, que auxilia na estabilidade do voo mesmo em espaços confinados, juntamente com o monitoramento contínuo do nível de bateria e a possibilidade de interromper os motores em situações de emergência, garantem que o Tello opere de forma controlada e segura.

FIGURA 2 – Protetores de hélice e detector de colisão



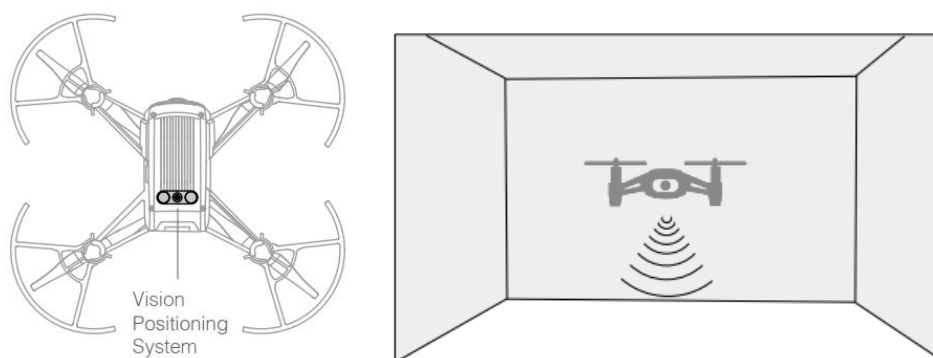
Fonte: Ryze Robotics, 2025

A câmera integrada ao drone é outro destaque importante, possibilitando a captura de fotos com resolução de 5 megapixels e a transmissão de vídeo em 720p com estabilização eletrônica de imagem. Essa combinação de recursos permite que os usuários não apenas experimentem os fundamentos do voo, mas também se envolvam em atividades que envolvem a captura e análise de imagens, tornando o aprendizado mais interativo e visual.

Os sensores integrados no drone Tello são essenciais para garantir um voo estável e preciso. O equipamento utiliza um sistema de posicionamento visual, ilustrada pela Figura 3, que combina uma câmera com sensores infravermelhos, permitindo que o drone mantenha sua

posição em ambientes internos e execute manobras com segurança dentro de uma faixa de altitude ideal. Além desse sistema, o Tello conta com acelerômetros e giroscópios que monitoram continuamente sua orientação e movimentos, possibilitando ajustes automáticos que corrigem desvios e asseguram uma trajetória de voo controlada. Medições provenientes do barômetro também contribuem para a determinação precisa da altitude, tornando o drone apto a operar de maneira confiável (RYZE ROBOTICS, 2025).

FIGURA 3 – Sistema de Posicionamento Visual



Fonte: Ryze Robotics, 2025

O controle e a programação do Tello são viabilizados através do Tello SDK, que estabelece comunicação via Wi-Fi utilizando o protocolo UDP, permitindo o envio de comandos em texto para o drone. Bibliotecas como a DJITelloPy possibilitam que tanto o Tello tradicional quanto o Tello EDU sejam programados em Python, oferecendo a oportunidade de desenvolver desde comandos simples de voo até trajetórias complexas e aplicações avançadas. Essa capacidade de programação torna o drone uma ferramenta poderosa para integrar conceitos de robótica e automação em projetos educacionais.

Embora ambos os modelos possam ser programados em Python, o Tello EDU se diferencia por oferecer um firmware atualizado com acesso ao SDK 2.0, o que inclui funcionalidades adicionais voltadas para o ensino (DJI-SDK, 2025). Entre essas funcionalidades, destacam-se os Mission Pads – pequenos cartões QR que o drone é capaz de reconhecer para executar comandos programados, permitindo a realização de missões interativas e voo em enxame – que ampliam as possibilidades de uso educacional e proporcionam um ambiente mais dinâmico para o aprendizado. Dessa forma, enquanto o Tello tradicional atende às necessidades básicas do consumidor, o Tello EDU oferece recursos avançados que enriquecem a experiência de ensino e aprendizado em áreas de programação e robótica.

No mercado, estão disponíveis dois kits para o drone Tello, sendo um deles o Tello Boost Combo, apresentado na Figura 4, que se destaca por incluir baterias adicionais e um hub de carregamento, proporcionando maior tempo de voo e menos interrupções durante as atividades (DJI STORE, 2025). Essa vantagem é particularmente relevante em contextos acadêmicos e educacionais, pois permite a continuidade das experiências práticas sem a necessidade de recargas frequentes. O Instituto Federal do Amazonas – IFAM CMDI chegou a dispor de quatro kits do Tello Boost Combo, o que facilita o emprego simultâneo do drone em diversas turmas e a realização de atividades estendidas, ampliando as possibilidades de ensino e aprendizagem.

FIGURA 4 - Tello Boost Combo



Fonte: DJI Store, 2025.

4.2 SOFTWARE

4.2.1 Python

Python é uma linguagem de programação de alto nível, interpretada e de tipagem dinâmica, destacando-se pela sua sintaxe clara e legível. Essa característica facilita o aprendizado e a implementação de algoritmos, permitindo que tanto iniciantes quanto desenvolvedores experientes se concentrem na lógica e na resolução de problemas sem se

perder em detalhes complexos de sintaxe. Conforme destacado na documentação oficial da Python Software Foundation (2025), a linguagem foi desenvolvida para ser intuitiva e poderosa, proporcionando uma experiência de programação eficiente e “batteries included” – ou seja, com uma ampla biblioteca padrão que atende a diversas necessidades.

A importância de Python para o ensino de programação também é evidenciada na obra “Introdução à Programação com Python”, de Nilo Ney Coutinho Menezes. O autor apresenta a linguagem como uma ferramenta acessível para o aprendizado dos fundamentos da programação, enfatizando que sua simplicidade permite que os conceitos de algoritmos, estruturas de controle e depuração sejam assimilados de maneira gradual e prática. Essa abordagem didática facilita a transição para projetos mais complexos, servindo como uma base sólida para o desenvolvimento do pensamento computacional.

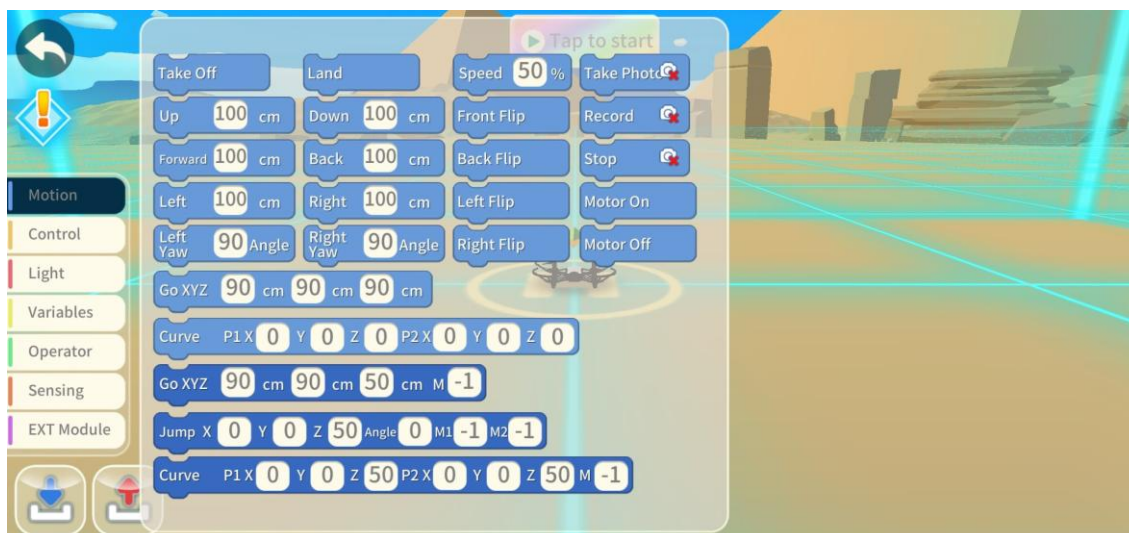
No contexto deste TCC, Python assume um papel central ao possibilitar a criação de protótipos e a implementação de experimentos que integram diversas tecnologias, como bibliotecas de visão computacional e controle de dispositivos (por exemplo, através da biblioteca DJITelloPy para o controle de drones). Além disso, optou-se pela utilização da versão 3.9 devido à sua melhor compatibilidade com as bibliotecas necessárias, garantindo maior estabilidade e desempenho durante o desenvolvimento dos experimentos.

4.2.2 Aplicativo Tello Edu

O Tello Edu App é uma ferramenta educacional oficialmente autorizada pela Ryze Technology, projetada para ser totalmente compatível com os drones Tello e Tello Edu. Desenvolvido com foco em ambientes de ensino para alunos do jardim de infância até o ensino médio, o aplicativo utiliza a abordagem da programação por blocos para conectar os estudantes ao mundo dos drones e da programação por meio de atividades lúdicas e interativas (RYZE ROBOTICS, 2025).

A aplicação oferece duas experiências principais. Por um lado, possibilita a demonstração do voo real dos drones, permitindo que os alunos controlem os dispositivos utilizando comandos que englobam funções como decolagem, pouso, movimentos laterais, avanço, recuo, subida, descida, além de manobras acrobáticas como o flip de 360 graus. Por outro lado, quando não há acesso a um drone físico, o Tello Edu App dispõe de um simulador virtual de programação de voo, no qual os estudantes podem experimentar a pilotagem em um ambiente virtual, desenvolvendo habilidades de lógica e criatividade. O ambiente pode ser visto na Figura 5.

FIGURA 5 - Ambiente Virtual de Simulação



Fonte: Tello Edu App, 2025.

Um dos destaques do aplicativo é a sua estrutura de jogos de programação em blocos, organizada em 12 níveis temáticos distribuídos em missões planetárias. Cada nível propõe desafios específicos que demandam a aplicação de conceitos fundamentais, tais como estruturas condicionais, laços de repetição e manipulação de variáveis, promovendo o desenvolvimento progressivo do pensamento computacional. Embora a versão completa, que disponibiliza todos os níveis, esteja restrita à modalidade paga, as funcionalidades oferecidas na versão gratuita se mostraram suficientes para atender aos objetivos do artigo de Alves e Bremgartner (2022) e para a realização deste Trabalho de Conclusão de Curso.

4.2.3 Biblioteca DJITelloPy

A biblioteca DJITelloPy oferece uma interface simplificada para controlar os drones DJI Tello por meio de comandos em Python. Conforme a documentação disponível no repositório do GitHub, apresentada na Figura 6, o projeto desenvolvido por Damià Fuentes Escoté e em contribuições da comunidade, detalha as funções básicas e avançadas que possibilitam a comunicação com o drone via sockets UDP, além de gerenciar o envio e recebimento de comandos e dados de telemetria (ESCOTÉ, 2025). Essa implementação inclui mecanismos para lidar com atrasos, reenvios e a captura do stream de vídeo, o que é fundamental para a execução de comandos precisos e seguros.

FIGURA 6 – Repositório do GitHub: biblioteca DJITelloPy.

The screenshot shows the GitHub repository page for DJITelloPy. At the top, the repository name 'DJITelloPy' is displayed along with the user 'damiafuentes'. Below this, there are statistics: 63 watchers, 506 forks, and 1.3k stars. The main content area shows a list of files and their commit history. The right sidebar contains 'About' information, 'Releases' (Version 2.5), and 'Packages'.

| File | Commit Message | Time Ago |
|----------------------|---|--------------|
| djitellopy | Fixed sequential typo in docs (in d31te11opy/swarm) | 2 months ago |
| docs | bump to 2.3.1 which will be available in the original pypi pac... | 4 years ago |
| examples | Merge pull request #116 from MichalKormanik/panorama-fe... | 4 years ago |
| .gitignore | Initial commit | 7 years ago |
| .readthedocs.yaml | add a .readthedocs.yaml file | 2 years ago |
| LICENSE.txt | Before uploading to PyPi | 7 years ago |
| README.md | update DJITelloPy in the wild information (more schools, ... | 6 months ago |
| README_CN.md | translate commits and examples into Chinese | 4 years ago |
| doc-requirements.txt | update requirements to make this library work with pyth... | 2 years ago |
| mkdocs.yml | add site_url, repo_url and repo_name to mkdocs.yml | 5 years ago |
| requirements.txt | update requirements to make this library work with pyth... | 2 years ago |

Fonte: Escoté, 2025.

Conforme a documentação oficial da DJI para a API Tello, encontrada no repositório em seu repositório no GitHub, como pode ser visto na Figura 7, complementa essas informações ao apresentar os comandos básicos e avançados, os parâmetros e as limitações impostas pelo firmware do drone (DJI-SDK, 2025). Essa referência serve como base para a implementação dos algoritmos de controle e para a integração com outras bibliotecas, como OpenCV e MediaPipe, que expandem as possibilidades de aplicação para visão computacional e controle por gestos.

FIGURA 7 - Repositório Tello-Python

The screenshot shows the GitHub interface for the 'Tello-Python' repository. At the top, it displays the repository name and owner 'dji-sdk'. Below that, there are navigation links for 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. The repository is marked as 'Public' and has statistics for 'Watch' (96), 'Fork' (648), and 'Star' (1.4k). The main content area shows a list of files and folders with their commit history. The 'About' section on the right provides a brief description: 'This is a collection of python modules that interact with the Ryze Tello drone.' It also lists features like 'Readme', 'View license', 'Activity', 'Custom properties', '1.4k stars', '96 watching', and '648 forks'.

Fonte: DJI-SDK, 2025.

Dessa forma, a utilização da DJITelloPy no contexto deste trabalho não só facilita o desenvolvimento dos experimentos como também contribui para uma abordagem progressiva no ensino de programação – permitindo a transição do ambiente visual do Scratch para a linguagem textual Python, com o suporte de técnicas avançadas de processamento de imagem e de machine learning.

4.2.4 OpenCV

O OpenCV (Open Source Computer Vision Library), é uma biblioteca de código aberto, multiplataforma e extensível, projetada para aplicações de visão computacional e processamento de imagens. Seu principal objetivo é oferecer uma ampla variedade de recursos, desde operações básicas de manipulação de imagens até algoritmos mais avançados de detecção, segmentação e reconhecimento de padrões (BRADSKI e KAEHLER, 2008). A biblioteca pode ser utilizada em diversas linguagens de programação, incluindo Python, o que favorece a criação de soluções robustas e personalizadas nos mais variados contextos de pesquisa e desenvolvimento.

Conforme enfatizado pelos mantenedores do projeto, o OpenCV conta com uma comunidade ativa, que contribui constantemente para o aprimoramento de suas funcionalidades e para a correção de eventuais falhas de implementação, além de prover documentação de apoio em seu site oficial e repositório público (OPENCV, 2023a; OPENCV, 2023b). Esse ecossistema

de suporte facilita o acesso a exemplos de uso, tutoriais e recursos adicionais, impulsionando a adoção da biblioteca tanto em projetos acadêmicos quanto em aplicações industriais.

Em um projeto voltado para análise de imagens ou desenvolvimento de sistemas inteligentes, o uso do OpenCV pode simplificar a etapa de implementação de rotinas de filtragem, detecção e rastreamento de objetos, permitindo ao pesquisador concentrar esforços no refinamento dos algoritmos e na interpretação dos resultados experimentais. Assim, a biblioteca se configura como uma ferramenta fundamental no âmbito de projetos que demandam soluções confiáveis para visão computacional.

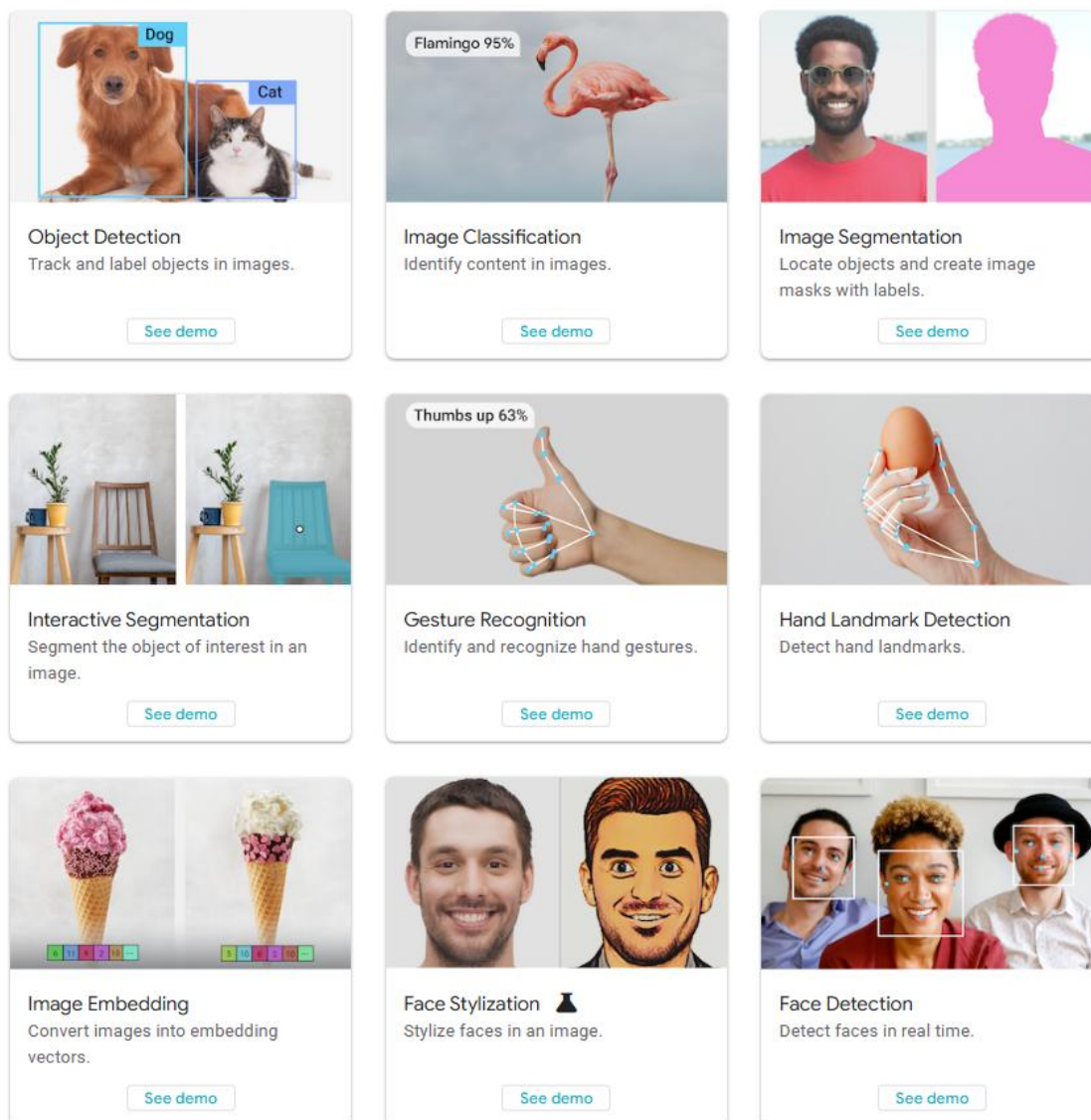
4.2.5 MediaPipe (Hands Gesture Control)

O MediaPipe é uma estrutura (framework) desenvolvida pelo Google para a criação de pipelines de processamento de dados de forma escalável, voltada especialmente para aplicações de visão computacional e aprendizado de máquina (GOOGLE AI EDGE, 2023). Em linhas gerais, trata-se de um conjunto de soluções que facilitam o desenvolvimento de aplicativos capazes de realizar tarefas como detecção de objetos, rastreamento de poses humanas, reconhecimento facial, entre outras. A principal vantagem do MediaPipe está em sua arquitetura modular e multiplataforma, o que permite a execução de seus modelos tanto em dispositivos móveis quanto em desktops de maneira eficiente (GOOGLE DEV, 2023). Além disso, a biblioteca oferece ferramentas otimizadas para processamento em tempo real, viabilizando a integração em projetos que exigem respostas rápidas e precisas.

Dentro do ecossistema do MediaPipe, há diversos “solutions” (soluções prontas) para diferentes aplicações, como por exemplo, face detection, holistic body tracking e object detection, algumas com aplicações de visão são apresentadas na Figura 8. Essas soluções utilizam modelos de aprendizado de máquina pré-treinados e pipelines otimizadas para garantir alto desempenho, simplificando o processo de implementação para desenvolvedores e pesquisadores (GOOGLE DEV, 2023). O MediaPipe, portanto, atua como um intermediário entre os modelos de ML e as etapas de pré e pós-processamento de dados, possibilitando a criação de aplicações complexas com relativa facilidade e bom desempenho computacional.

FIGURA 8 – Aplicações de Visão com MediaPipe

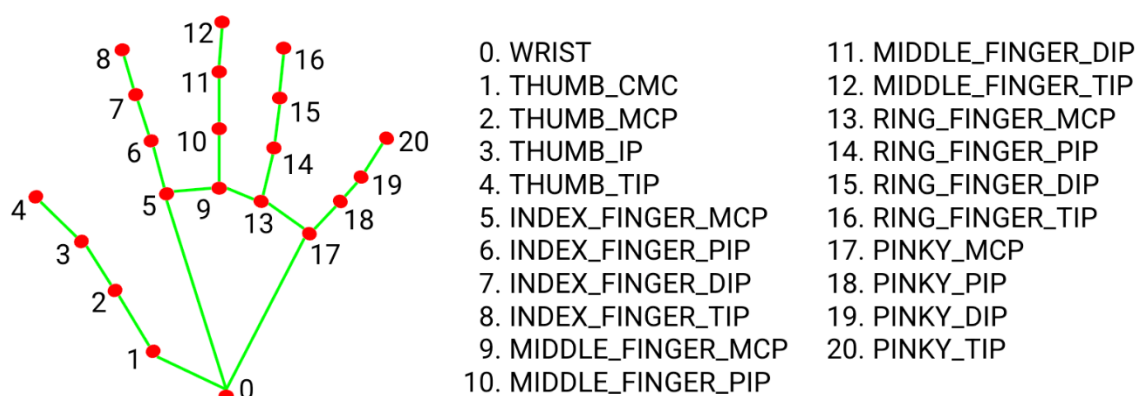
VISION



Fonte: MediaPipe, 2025.

Entre as soluções oferecidas, o MediaPipe Hands é uma das mais populares quando se trata de aplicações de detecção e reconhecimento de gestos manuais (CHUOLING, 2023). Essa solução realiza a localização (landmarks) das mãos em imagens ou vídeos, identificando pontos-chave que permitem avaliar a posição e a forma de cada dedo em tempo real, conforme observado na Figura 9. Com base nessa análise, é possível fazer o reconhecimento de gestos específicos, como indicar se a mão está fechada, aberta ou se determinado dedo está apontando para uma direção, entre outras possibilidades (KINIVI, 2023a).

FIGURA 9 - MediaPipe Hands (Landmarks)



Fonte: MediaPipe, 2025.

O reconhecimento de gestos manuais tem aplicações amplas: sistemas de interação homem-computador, controle de drones, auxílio na comunicação de pessoas com deficiência auditiva, e até mesmo a criação de interfaces mais intuitivas em jogos e aplicações de realidade aumentada (KINIVI, 2023b). Graças ao MediaPipe, desenvolvedores podem combinar esses recursos com outras bibliotecas e linguagens de programação, incluindo Python e OpenCV, agilizando o desenvolvimento de protótipos funcionais sem a necessidade de criar algoritmos de baixo nível para detecção de bordas ou segmentação de imagens.

4.3 METODOLOGIA

Este trabalho visa explorar as potencialidades do drone Tello EDU no ensino de programação, Visão Computacional e Inteligência Computacional, por meio da implementação de comandos em Python integrados às bibliotecas OpenCV e MediaPipe. Para atingir esse objetivo, foram conduzidos experimentos que visaram testar e validar funcionalidades do drone, utilizando a programação textual para controlar os voos, processar imagens e interpretar gestos.

Inicialmente, os experimentos consistiram na implementação de comandos básicos de voo através da biblioteca DJITelloPy, permitindo o controle do drone via Python. Em seguida, integrou-se o OpenCV para capturar e processar os frames de vídeo, possibilitando a análise das imagens e a extração de informações relevantes. Complementarmente, a utilização do MediaPipe permitiu o desenvolvimento de um sistema de reconhecimento de gestos, no qual os movimentos das mãos eram convertidos em comandos para o drone.

Adicionalmente, durante a disciplina de Inteligência Computacional, foi desenvolvido um projeto final que incorporou esses experimentos. Nesse projeto, os conhecimentos

adquiridos foram aplicados para criar um sistema integrado, demonstrando a viabilidade do controle do drone através de comandos textuais e de gestos, o que reforçou a proposta de utilizar o Tello EDU como ferramenta didática inovadora.

De forma resumida, a metodologia adotada combinou a experimentação prática e a análise dos resultados, servindo como base para os Estudos de Caso, onde os detalhes e os impactos dos experimentos serão aprofundados.

5. ESTUDO DE CASO 1

Este estudo de caso tem como ponto de partida o artigo anterior intitulado "Uso de drones para ensino de Lógica de Programação e estímulo do Pensamento Computacional" (ALVES e BREMGARTNER, 2022) e propõe uma abordagem complementar, introduzindo a biblioteca DJITelloPy para facilitar a transição para o ensino de Python.

5.1 PROGRAMAÇÃO DO DRONE COM BLOCOS DE PROGRAMAÇÃO

Nesse artigo, foram realizados experimentos iniciais que tinham como objetivo principal introduzir conceitos fundamentais de programação por meio de uma abordagem lúdica e interativa utilizando drones, especificamente o DJI Tello EDU, em conjunto com a linguagem visual baseada em blocos Scratch. A principal motivação dessa iniciativa foi estimular o pensamento computacional e facilitar o aprendizado da Lógica de Programação, proporcionando aos estudantes uma forma prática e tangível de explorar conceitos que, frequentemente, são considerados abstratos.

O ambiente virtual do aplicativo Tello Edu, permite a simulação antes da prática real. Nas Figuras 10 e 11 é apresentado o ambiente onde eram realizadas as simulações da programação, junto com os blocos de comandos normalmente mais utilizados durante esta fase inicial.

FIGURA 10 - Blocos de Programação de Movimento



Fonte: Tello Edu App, 2025.

FIGURA 11 - Blocos de Programação de Controle



Fonte: Tello Edu App, 2025.

Como ilustrado na Figura 12, apresenta-se um exemplo prático de programação que destaca a sequência lógica dos comandos empregados utilizando blocos de movimento e de controle, evidenciando a clareza e a eficiência dos algoritmos utilizados.

FIGURA 12 - Exemplo de Algoritmo com Blocos de Movimento e Controle



Fonte: Autor, 2025.

Esses experimentos iniciais, utilizando programação em blocos, já foram realizados com alunos de cursos técnicos e superiores em Mecatrônica e Eletrônica do Instituto Federal do Amazonas (IFAM), tanto em formato remoto quanto presencial, devido às restrições impostas pela pandemia de Covid-19. Conforme descrito no artigo anterior, os resultados obtidos demonstraram uma resposta altamente positiva por parte dos estudantes, refletida em maior

engajamento e interesse nas atividades propostas. Além disso, observou-se que a utilização do drone DJI Tello, através do aplicativo Tello Edu com interface baseada em blocos visuais semelhantes ao Scratch, facilitou significativamente a compreensão dos fundamentos de lógica de programação e promoveu uma maior motivação entre os alunos (ALVES e BREMGARTNER, 2022).

Seguindo a proposta de Jesus, Santana e Bittencourt (2019) de utilizar projetos como estratégia de aprendizagem, evidenciada na aplicação da aprendizagem baseada em projetos e na gamificação, essas metodologias colaboram para manter o interesse e promover a interação ativa dos estudantes. O estímulo à criatividade e à produção de produtos tangíveis, como jogos ou ferramentas de uso real, favorece o desenvolvimento da confiança dos alunos em sua capacidade de programar, contribuindo para a redução dos índices de evasão. Nesse contexto, os estudantes foram desafiados a completar as fases do jogo, otimizando seus algoritmos para alcançar pontuações mais elevadas, pois uma programação mais enxuta, com menos blocos de comandos, resultava em uma pontuação maior. Dessa forma, na seção de Anexos, é possível visualizar algumas das soluções desenvolvidas para completar as fases disponíveis de forma gratuita no Aplicativo Tello Edu. As soluções estão dispostas conforme foram entregues pelos alunos.

As atividades desenvolvidas com o drone Tello Edu proporcionaram aos estudantes uma experiência prática e interativa, permitindo-lhes testar imediatamente os algoritmos criados e compreender de forma concreta os conceitos teóricos abordados. Os resultados demonstraram que o uso de drones teve um impacto positivo, auxiliando não apenas na compreensão dos conteúdos, mas também no desenvolvimento de habilidades relacionadas ao pensamento computacional, como a decomposição de problemas e raciocínio lógico (ALVES e BREMGARTNER, 2022). As Figuras 13 (a) e 13 (b), demonstram o engajamento dos alunos e a aplicação dos conceitos teóricos durante as atividades práticas.

FIGURA 13 - (a) Engajamento de alunos nas atividades práticas. (b) Aplicação dos conceitos teóricos.



(a)

(b)

Fonte: Autores, 2022.

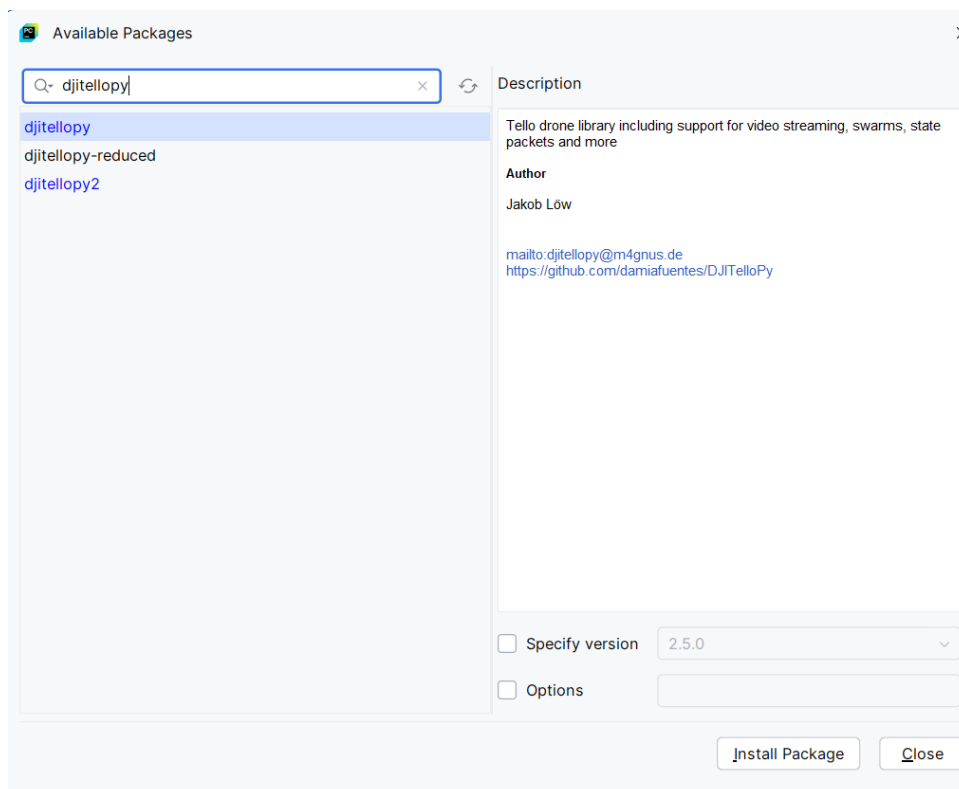
Diante dos resultados promissores obtidos com a abordagem visual (Scratch), decidiu-se expandir essa pesquisa para investigar a viabilidade de utilizar o drone Tello com uma linguagem de programação textual amplamente adotada na educação superior e em contextos profissionais, como é o caso do Python. Essa decisão foi fundamentada pela constatação de que o drone DJI Tello oferece suporte a comandos em Python por meio da biblioteca DJITelloPy, possibilitando desenvolver programas mais elaborados e complexos, tais como voos autônomos e rotinas específicas (GHAZI; VOYER, 2024; PETROVIČ; VERČIMÁK, 2023). A transição para Python é crucial, pois, embora a programação visual seja eficaz no estímulo inicial, a literatura sugere que linguagens textuais são essenciais para desenvolver habilidades mais profundas e preparar os estudantes para desafios tecnológicos mais avançados (DORLING; WHITE, 2015; JESUS; SANTANA; BITTENCOURT, 2019).

5.2 PROGRAMAÇÃO DO DRONE COM PYTHON

Para explorar as possibilidades do drone Tello Edu com Python, foi realizada a instalação da versão 3.9 do Python, devido sua melhor compatibilidade com as bibliotecas necessárias, e o Pycharm para realização da programação.

Conforme apresentado na Figura 14, dentro do Pycharm foi realizada a instalação da biblioteca DJITelloPy. Consultando a biblioteca e o repositório do GitHub de Escoté (2025), foi possível analisar e compreender as suas aplicações.

FIGURA 14 - Instalação da biblioteca DJITelloPy pelo Pycharm



Fonte: Autor, 2025.

No início do código, são importadas as bibliotecas essenciais para o funcionamento do sistema: a biblioteca DJITelloPy para a comunicação e controle do drone Tello, cv2 do OpenCV para o processamento de imagens (explorada mais a frente), e time para o gerenciamento de intervalos temporais. Conforme a Figura 15, após essa etapa inicial, estabelece-se a conexão com o drone por meio da criação de um objeto da classe Tello, atribuído à variável drone. Em seguida, o método connect() é invocado para iniciar a comunicação com o dispositivo e, logo após, o método get_battery() é utilizado para obter e exibir o status da bateria, garantindo que o drone esteja com energia suficiente para a realização das operações.

FIGURA 15 – Estabelecendo Conexão com o Drone



```

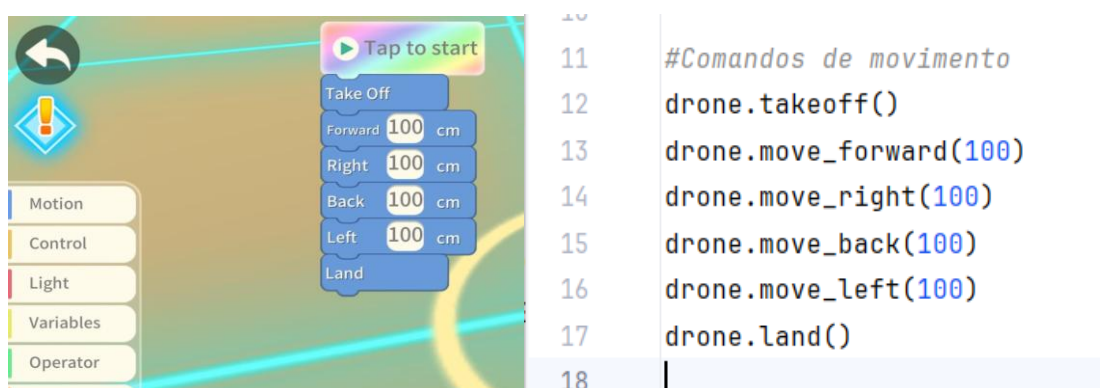
Tello_edu_1.py x  tello.py
1  from djitellopy import Tello
2  import cv2
3  import time
4
5  #Estabelecendo conexão com o drone e status da bateria
6  drone = Tello()
7  drone.connect()
8  bateria_status = drone.get_battery()
9  print('Bateria_status:', bateria_status)
10

```

Fonte: Autor, 2025.

Uma das atividades iniciais realizadas pela turma, utilizando a programação com blocos, foi a tarefa de programar o drone para executar um movimento em forma de quadrado, conforme demonstrado no artigo de Alves e Bremgartner (2022). Nesse contexto, o exercício será usado como ponte para a transição da linguagem visual em blocos para a linguagem textual (Python), possibilitando a comparação entre as abordagens. A Figura 16 (a) e 16 (b) apresentam a resolução do exercício inicial, onde foram utilizados apenas blocos básicos de movimento; enquanto a Figura 17 (a) e 17 (b) ilustram a resolução do mesmo problema por meio de um laço de repetição. Dessa maneira, a análise lado a lado permite identificar as correspondências e evoluções conceituais entre as duas linguagens.

FIGURA 16 - (a) Programação em blocos (b) Programação em Python



(a) Programação em blocos

Visual representation of a block-based program for drone movement. The sequence of blocks is: Tap to start, Take Off, Forward 100 cm, Right 100 cm, Back 100 cm, Left 100 cm, and Land.

(b) Programação em Python

```

11  #Comandos de movimento
12  drone.takeoff()
13  drone.move_forward(100)
14  drone.move_right(100)
15  drone.move_back(100)
16  drone.move_left(100)
17  drone.land()
18

```

(a)

(b)

Fonte: Autor, 2025.

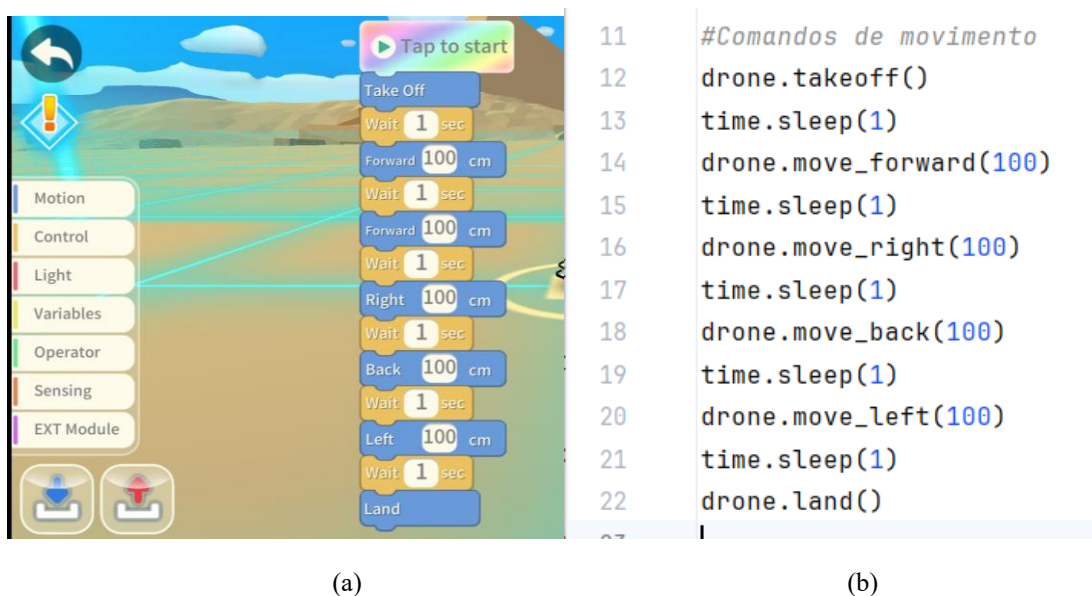
FIGURA 17 - (a) Programação em blocos com laço de Repetição (b) Programação em Python com Laço de Repetição



Fonte: Autor, 2025.

Embora o algoritmo esteja correto, ele precisa ser otimizado para condições reais de operação, onde é essencial considerar o tempo de estabilização do drone após cada comando. Dessa forma, inserir intervalos entre os comandos de movimento permite que o drone se acomode, garantindo maior segurança e precisão na execução. A Figura 18 (a) e 18 (b) ilustra essas alterações, evidenciando a importância dos delays para uma operação mais estável e segura no mundo real.

FIGURA 18 - (a) Programação em Blocos com delay (b) Programação em Python com delay



Fonte: Autor, 2025.

Essa fase da pesquisa ainda não foi implementada em sala de aula, representando uma exploração conceitual e prática sobre a viabilidade do uso da programação textual com o drone

Tello. Espera-se que essa transição contribua para o ensino progressivo de programação, permitindo que os alunos aprofundem suas habilidades técnicas ao desenvolverem projetos mais complexos, como voos autônomos e sequências específicas de comandos.

Dando continuidade a essa perspectiva de ensino progressivo, é possível imaginar roteiros de voo que façam uso dos recursos disponíveis na biblioteca DJITelloPy para tornar a experiência de programação ainda mais dinâmica. Por exemplo, além de comandos básicos como “takeoff” e “land”, a biblioteca oferece a possibilidade de consultar dados de estado do drone e controlar velocidades e direções de forma mais refinada. Dessa forma, roteiros semiautônomos ou mesmo autônomos podem ser elaborados a partir dos valores de altura, bateria e velocidade fornecidos pela própria Tello.

Um primeiro passo nesse sentido pode envolver o uso de métodos que consultam dados de sensores e de telemetria. Por exemplo, para obter a altura do drone (em centímetros), podemos empregar:

```
height = drone.get_height()
print(f"Altura atual: {height} cm")
```

Com essa informação, o programa pode tomar decisões automáticas. Imagine, por exemplo, que o drone deva subir a uma altura mínima antes de iniciar um trajeto horizontal; se `get_height()` retornar um valor abaixo de 50 cm, um bloco condicional poderia acionar o comando `move_up(20)` até que a altura desejada fosse atingida:

```
if height < 50:
    drone.move_up(20)
```

Além de consultar a altura, a biblioteca também permite verificar o nível de bateria, a temperatura e a velocidade do drone. Caso o programa detecte que a bateria está abaixo de um determinado limiar — por exemplo, 20% — é possível forçar a aterrissagem imediatamente, garantindo maior segurança:

```
battery_level = drone.get_battery()
if battery_level < 20:
    drone.land()
    print("Bateria baixa! O drone foi pousado automaticamente.")
```

Essas leituras tornam o roteiro de voo mais flexível e robusto, pois o drone consegue “perceber” o ambiente (ou pelo menos seu próprio estado interno) e responder de maneira quase autônoma. Para deslocamentos, a função `send_rc_control()` possibilita ajustes finos de velocidade em cada um dos eixos, o que permite criar movimentações horizontais ou verticais graduais. Por exemplo:

```

# Ajustando velocidades nos eixos X, Y e Z, e yaw (rotação)
# move-se para frente
drone.send_rc_control(0, 30, 0, 0)
# voa para frente por 2 segundos
time.sleep(2)
# para o movimento
drone.send_rc_control(0, 0, 0, 0)

```

Combinando esse tipo de controle a loops e condicionais, o drone pode executar um circuito de voo sem a intervenção constante do usuário. Ele pode, por exemplo, verificar a própria altura a cada iteração de um loop, subindo ligeiramente se estiver abaixo de um valor predefinido ou deslocando-se para frente até atingir um certo tempo de voo.

Em cenários mais avançados, a biblioteca DJITelloPy oferece comandos como `curve_xyz_speed()`, que possibilitam a realização de trajetórias curvas definidas por dois pontos intermediários, tornando o voo mais elaborado e permitindo, por exemplo, contornar obstáculos dispostos em sala de aula. Tudo isso sem depender estritamente da intervenção manual do operador, mas sim do código que analisa as leituras de sensores e executa manobras pré-programadas.

Essa capacidade de controlar o drone por meio de algoritmos sofisticados não só amplia as possibilidades de aplicações práticas, como também abre caminho para uma abordagem educacional mais robusta. De fato, a introdução do drone na programação textual é respaldada por estudos que destacam o potencial dos drones na educação, especialmente em contextos STEAM (Ciências, Tecnologia, Engenharia, Artes e Matemática). Segundo Yepes (2020), o uso de drones pode estimular um aprendizado significativo através de metodologias ativas, tornando o ensino mais dinâmico e engajando alunos em disciplinas frequentemente vistas como abstratas.

No entanto, alguns desafios precisam ser considerados na implementação prática dessa proposta. Aspectos como segurança, custos envolvidos e complexidade técnica do equipamento são mencionados na literatura como possíveis limitadores para a ampla adoção dessa tecnologia em escolas (PETROVIČ; VERČIMÁK, 2023). Portanto, planejamento detalhado e adequado treinamento de docentes são essenciais para garantir que a aplicação dos drones na educação transcenda o aspecto meramente lúdico e tenha impacto positivo efetivo na aprendizagem dos alunos.

O presente Estudo de Caso 1 contribui para a continuidade e expansão das pesquisas sobre o uso educacional do drone Tello EDU, destacando a importância de uma transição

gradual e fundamentada do ambiente visual Scratch para a programação textual em Python. Essa abordagem progressiva visa fortalecer o pensamento computacional dos estudantes e preparar os mesmos para ambientes de programação mais avançados, potencializando sua capacidade para o desenvolvimento de projetos tecnológicos complexos.

Além disso, a adoção do Python como linguagem de programação textual abre portas para um leque amplo de aplicações em disciplinas avançadas da graduação, como visão computacional e inteligência computacional. Estas áreas poderão aproveitar as capacidades do drone Tello EDU para aplicar técnicas sofisticadas de processamento de imagem e machine learning, potencializando ainda mais o aprendizado prático e significativo. Essa abordagem será detalhadamente explorada no Estudo de Caso 2, destacando as potencialidades didáticas e tecnológicas dessas ferramentas na formação acadêmica dos estudantes.

6. ESTUDO DE CASO 2

6.1 APLICAÇÃO EM VISÃO COMPUTACIONAL

O drone Tello EDU, aliado à biblioteca OpenCV, oferece um ambiente prático e acessível para o ensino de Visão Computacional no nível de graduação. A possibilidade de capturar, processar e analisar imagens em tempo real com um dispositivo aéreo permite aos estudantes explorarem conceitos fundamentais da área, como filtragem de imagens, segmentação, detecção de bordas e rastreamento de objetos (GONZALEZ e WOODS, 2008).

A integração do Tello EDU com a biblioteca DJITelloPy facilita o desenvolvimento de aplicações que envolvem a manipulação e análise de imagens adquiridas pelo drone. Essa biblioteca permite que comandos sejam enviados ao drone de maneira programática, possibilitando a captura de imagens enquanto o dispositivo se desloca, o que é essencial para experimentos em visão computacional. O OpenCV possibilita o processamento dessas imagens, permitindo a implementação de algoritmos clássicos, como o detector de bordas de Canny e a segmentação de cores, que ajudam na extração de informações relevantes para diferentes aplicações (BRADSKI e KAEHLER, 2008).

6.1.1 Captura de vídeo

A primeira etapa para qualquer aplicação de Visão Computacional com o Tello EDU é a captura de vídeo em tempo real durante o voo do drone. A biblioteca DJITelloPy permite iniciar o fluxo de vídeo diretamente da câmera embarcada, possibilitando o processamento das imagens com OpenCV. Isso fornece uma base para a aplicação de filtros e algoritmos de segmentação em tempo real. A partir dessas imagens capturadas em voo, foram realizados testes aplicando diversas técnicas de processamento de imagem utilizando funções do OpenCV.

Esse código, presente no Apêndice A, estabelece a conexão com o drone, inicia a transmissão de vídeo e exibe os frames em tempo real usando OpenCV, permitindo que os estudantes testem diferentes algoritmos diretamente sobre as imagens capturadas (SAINI, 2021).

6.1.2 Conversão para escala de cinza

A conversão de imagens coloridas para escala de cinza é um dos primeiros passos no processamento de imagens, pois reduz a complexidade dos cálculos necessários para análises

subsequentes (SZELISKI, 2010). O código correspondente, implementada a partir da câmera do drone, está disponível no Apêndice B, enquanto a Figura 19 apresenta um frame capturado pela câmera do drone durante o voo, ilustrando essa aplicação.

Essa conversão é útil para aplicações como detecção de bordas e segmentação de objetos.

FIGURA 19 - Conversão para Escala de Cinza



Fonte: Autor, 2025.

6.1.3 Detecção de bordas com Canny

A detecção de bordas é uma técnica fundamental para realçar os contornos de objetos e facilitar o reconhecimento de padrões visuais. O detector de Canny realça os contornos presentes na imagem, tornando-se uma ferramenta valiosa para a navegação e o rastreamento visual do drone (KHOZA; OWOLABI; MALELE, 2024). O algoritmo correspondente está disponível no Apêndice C, enquanto a Figura 20 apresenta o resultado da detecção de bordas com o método de Canny.

FIGURA 20 - Imagem com detecção de bordas Canny

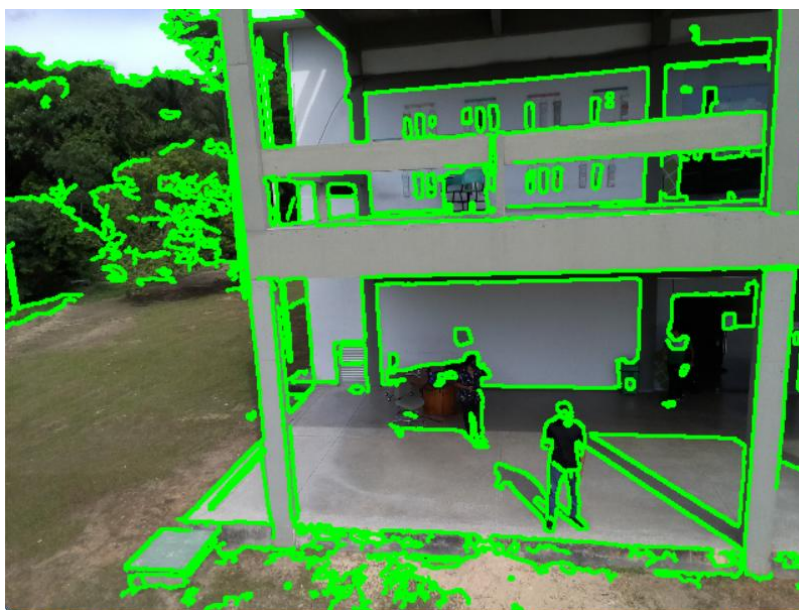


Fonte: Autor, 2025.

6.1.4 Contornos e Segmentação de Objetos

A identificação de contornos em uma imagem é um processo essencial para a segmentação de objetos de interesse, facilitando a detecção e o reconhecimento de elementos específicos na cena. A segmentação de objetos é amplamente utilizada em aplicações de visão computacional, permitindo isolar e analisar componentes individuais em um ambiente complexo (GONZALEZ; WOODS, 2008). No contexto da navegação e rastreamento por drones, essa técnica é fundamental para o reconhecimento de alvos e a tomada de decisões autônomas. A implementação desse processo está detalhada no Apêndice D, enquanto a Figura 21 apresenta o resultado da segmentação e do contorno dos objetos capturados pela câmera do drone.

FIGURA 21 – Imagem Segmentada e com Contorno



Fonte: Autor, 2025.

Aplicando essa técnica, o drone poderia ser utilizado para detectar e seguir objetos específicos, como marcadores visuais em um ambiente de laboratório ou identificar áreas específicas para navegação autônoma. Em um cenário educacional, os alunos poderiam programar o drone para reconhecer diferentes formas geométricas e responder com ações específicas, como pousar ao identificar um círculo ou alterar sua trajetória ao detectar uma linha. Essas aplicações tornam a aprendizagem de Visão Computacional mais interativa e próxima de aplicações reais no mercado.

Os exemplos apresentados demonstram como a combinação do drone Tello EDU com OpenCV pode ser utilizada para o ensino de Visão Computacional. Cada uma dessas aplicações oferece uma abordagem prática para conceitos como filtragem de imagens, segmentação e rastreamento, permitindo que os alunos experimentem algoritmos de forma interativa e aplicada a cenários do mundo real. Dessa forma, essa abordagem contribui significativamente para a formação de estudantes em engenharia, ciência da computação e áreas correlatas, preparando-os para desafios práticos no campo de Visão Computacional.

6.2 APLICAÇÃO EM INTELIGÊNCIA COMPUTACIONAL

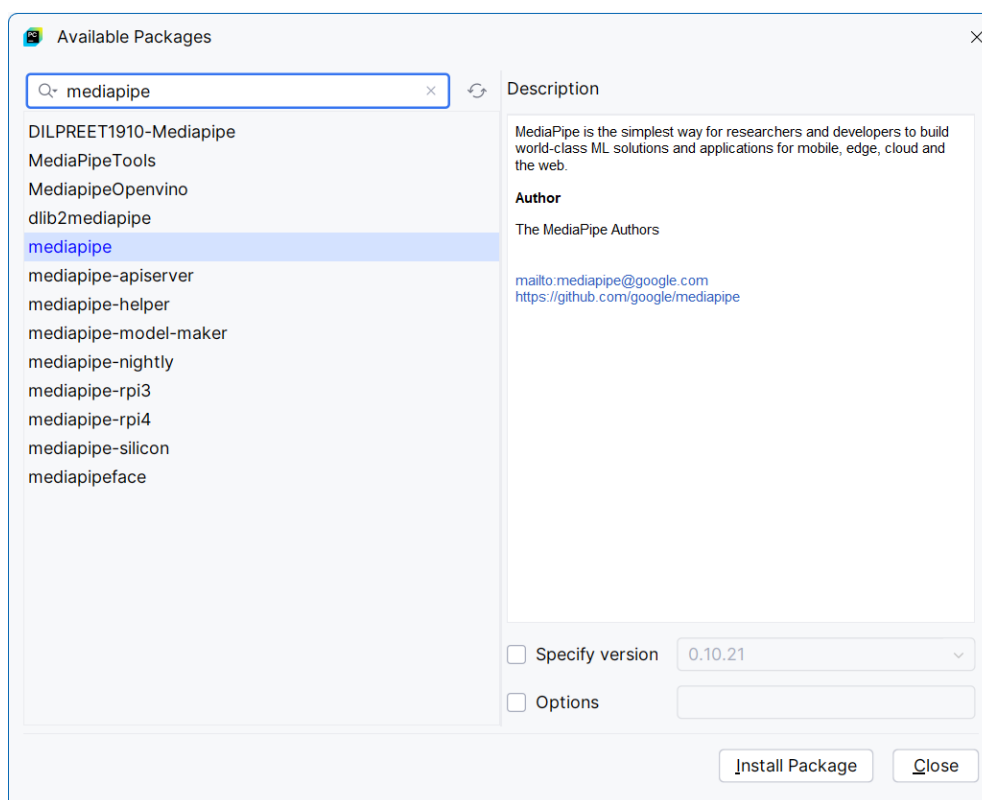
As etapas apresentadas até o momento deste trabalho culminaram em um projeto final realizado na disciplina de Inteligência Computacional. Nesta seção, será apresentada a

aplicação do drone Tello EDU no contexto da disciplina, com foco na implementação do controle por gestos de mãos utilizando a biblioteca MediaPipe. O objetivo do projeto foi desenvolver um sistema que permitisse controlar o drone apenas com gestos manuais, eliminando a necessidade de um controle físico ou comandos digitados. Esse projeto foi desenvolvido como parte de uma avaliação da disciplina de Inteligência Computacional, visando demonstrar a aplicação da inteligência artificial em um sistema autônomo.

6.2.1 Configuração do Ambiente

Para a implementação do projeto, além das instalações das bibliotecas DJITelloPy e OpenCV, foi também realizada a instalação da biblioteca MediaPipe, apresentada na Figura 22. Após a configuração do ambiente, foi implementado um script para capturar os frames do drone, conforme apresentado anteriormente, cujo código se encontra no Apêndice A.

FIGURA 22 - Instalação de biblioteca MediaPipe



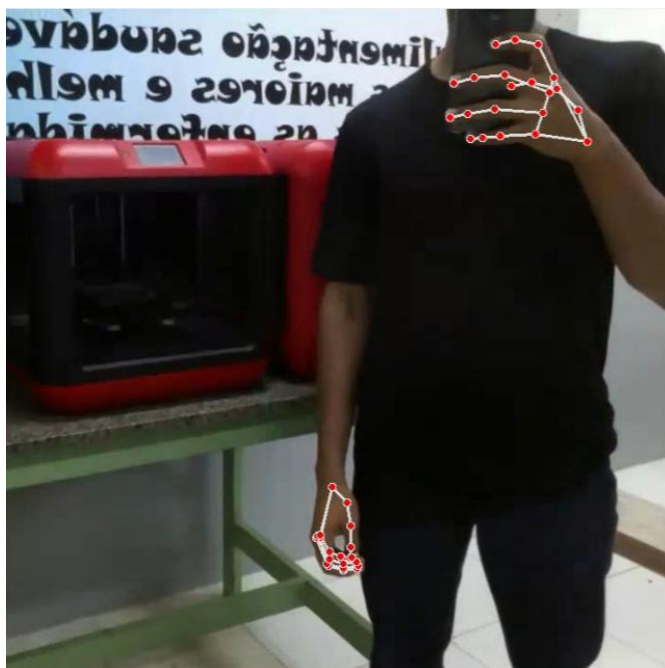
Fonte: Autor, 2025.

6.2.2 Reconhecimento de Gestos

O passo seguinte consistiu na implementação do reconhecimento de gestos. Para isso, foi configurado o MediaPipe Hands (CHUOLING, 2023) para detectar pontos-chave da mão e

interpretar gestos. Foi utilizada a API de Solução Python do MediaPipe (GOOGLE DEV, 2023), que fornece um código de exemplo para implementação rápida e testes através da webcam do computador. No entanto, este código foi adaptado para funcionar a partir da câmera do drone. O código final pode ser encontrado no Apêndice E, e a Figura 23 mostra este resultado.

FIGURA 23 - MediaPipe Hands pela câmera do Drone Tello Edu

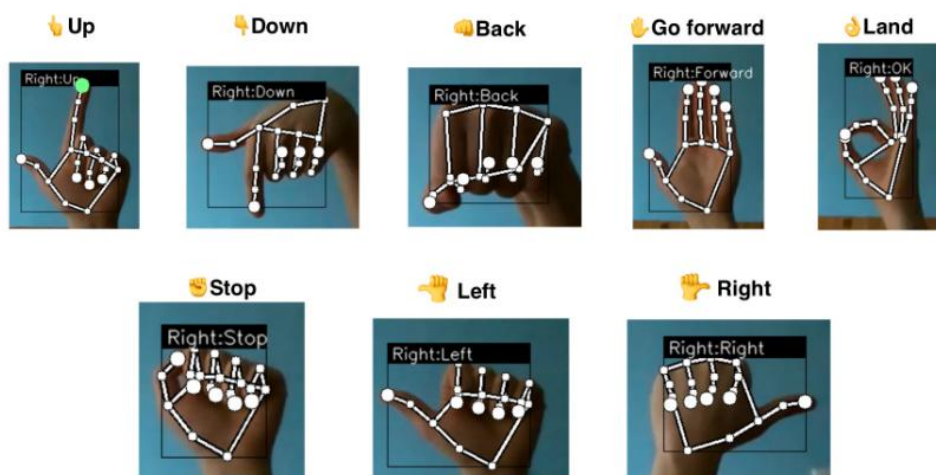


Fonte: Autor, 2025.

O MediaPipe Hands identifica e rastreia a posição das mãos e dos dedos em tempo real, mas, por si só, não realiza o reconhecimento de gestos. Ele fornece um conjunto de pontos-chave (landmarks) das mãos, conforme foi ilustrado na Figura 9, que podem ser usados para interpretar gestos. Para que o reconhecimento de gestos fosse viabilizado, foi necessário um processamento adicional, como a criação de regras baseadas nas posições dos pontos ou o uso de modelos de aprendizado de máquina.

Neste projeto, foi utilizado um modelo de aprendizado de máquina para o reconhecimento dos gestos e sua conversão em comandos para o controle do drone. O modelo KeyPointClassifier, treinado previamente (KINIVI, 2023b), foi utilizado para classificar os gestos e mapeá-los para os comandos do drone. Os gestos e seus respectivos comandos são ilustrados na Figura 24.

FIGURA 24 - Gestos e Comandos



Fonte: Kinivi, 2025.

6.2.3 Implementação Final

A implementação final consistiu na automação do movimento do drone com base nos gestos das mãos capturados pela câmera do drone. Para isso, foi realizada uma integração eficiente entre os módulos de reconhecimento de gestos e os comandos enviados ao drone através da biblioteca DJITelloPy. Durante os testes, foi possível observar que o drone respondia de maneira ágil aos gestos pré-definidos, como decolagem, pouso, movimentação lateral e ajustes de altitude. A precisão do reconhecimento de gestos foi elevada, garantindo um controle intuitivo e responsivo.

A Figura 25 demonstra o resultado final do projeto, evidenciando a eficácia da solução implementada. Além disso, ela ilustra os comandos, através dos gestos, utilizados para controlar o drone, conforme descrito a seguir:

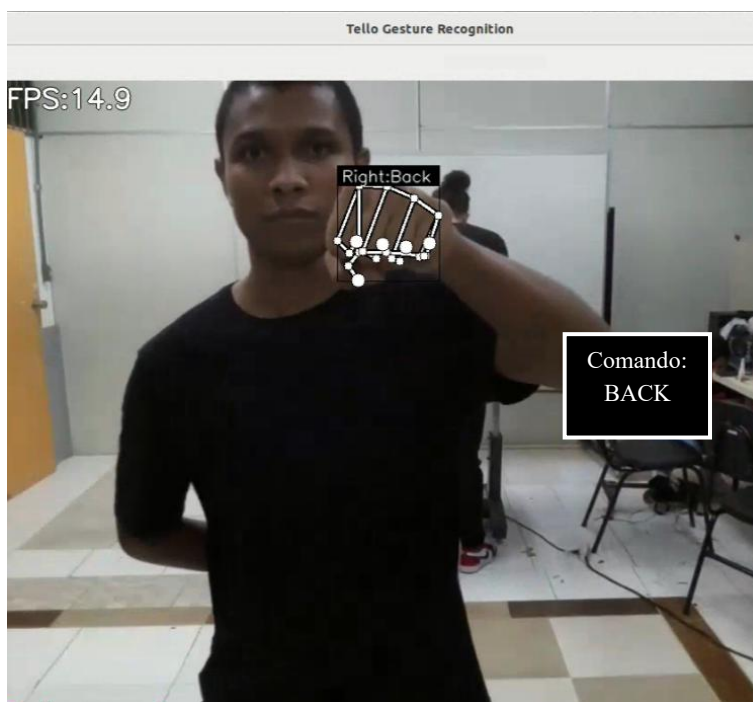
Figura 25 (a): ilustra o gesto que aciona o movimento para trás;

Figura 25 (b): exibe o comando para a parada;

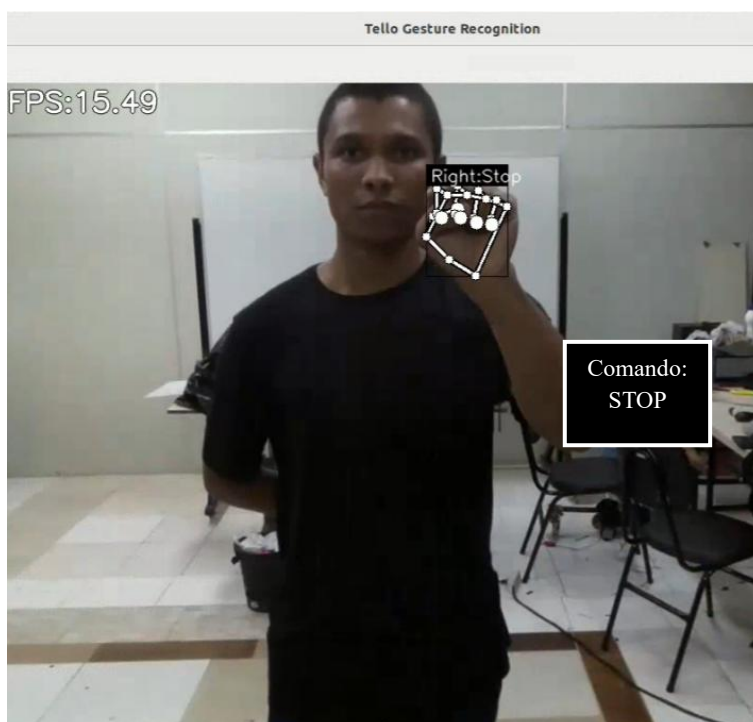
Figura 25 (c): apresenta o gesto para deslocamento à esquerda.

Com o intuito de documentar e validar o desempenho do sistema, foi produzido um vídeo demonstrativo da aplicação, acessível por meio do QR Code apresentado no Apêndice F. Esse código direciona para uma playlist no meu canal no YouTube, onde estão disponíveis os testes e a demonstração do funcionamento do sistema em tempo real. Alternativamente, a playlist pode ser acessada pelo link: <https://shorturl.at/YkBn0>.

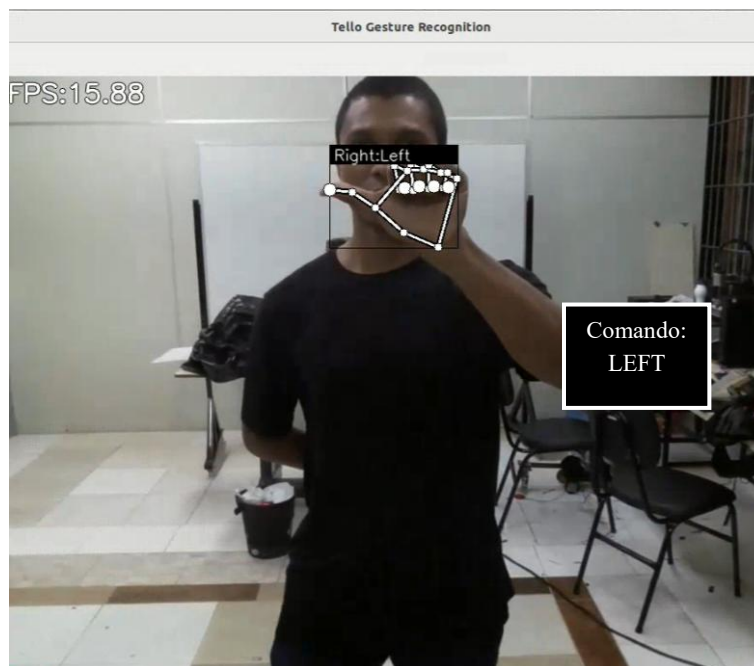
FIGURA 25 – (a) Comando: Back (b) Comando: Stop (c) Comando: Left



(a)



(b)



(c)

Fonte: Autor, 2025.

Algumas limitações foram observadas durante os testes, como dificuldades em ambientes com iluminação inadequada e a necessidade de uma câmera de boa qualidade para uma detecção confiável. As condições do ambiente devem ser favoráveis para a realização dos testes. Apesar disso, o projeto demonstrou a viabilidade do uso da Inteligência Computacional para controle de drones via gestos manuais. A integração entre OpenCV, MediaPipe e DJITelloPy permitiu a criação de um sistema eficiente e de fácil operação. Como trabalhos futuros, sugere-se a exploração de redes neurais mais avançadas para melhorar a robustez da detecção de gestos em diferentes condições ambientais.

Para a demonstração da aplicação do controle por gestos no drone Tello EDU, foi utilizado o repositório Tello Gesture Control (KINIVI, 2023b), que implementa um sistema baseado no MediaPipe Hands e aprendizado de máquina. O foco deste trabalho não foi o desenvolvimento de um novo sistema de reconhecimento de gestos, mas sim a exploração e validação da aplicabilidade dessa tecnologia para o controle autônomo de drones. Dessa forma, o código disponível no repositório foi empregado para validar a viabilidade da interação humano-drone por meio de gestos, sem modificações significativas no seu funcionamento.

Este trabalho demonstrou a viabilidade do controle gestual do drone Tello EDU utilizando um sistema baseado no MediaPipe Hands. A implementação utilizada foi baseada no repositório Tello Gesture Control (KINIVI, 2023b), o que permitiu validar a eficácia desse método sem a necessidade de desenvolver um sistema do zero. Como trabalhos futuros, sugere-

se a personalização do reconhecimento de gestos para atender a aplicações específicas, bem como a otimização dos comandos para diferentes cenários de operação.

7. CONCLUSÃO

A presente pesquisa evidenciou como os drones Tello EDU podem desempenhar um papel central na renovação de práticas pedagógicas no IFAM, especialmente quando se busca integrar o ensino de programação, visão computacional e inteligência computacional em diferentes níveis de ensino. A partir da análise dos estudos de caso desenvolvidos neste TCC, ficou claro que a utilização desses drones, aliada a plataformas de programação em blocos (Scratch) e linguagens textuais (Python), potencializa tanto a aprendizagem inicial de algoritmos quanto aplicações avançadas, como voos autônomos e interação via visão computacional.

Um dos grandes diferenciais constatados é a facilidade de adaptação das atividades propostas: desde alunos do ensino técnico, que podem iniciar seus estudos na lógica de programação de forma mais lúdica, até graduandos em engenharia e computação, capazes de explorar recursos avançados das bibliotecas OpenCV e MediaPipe para processamento de imagens e reconhecimento de gestos. A abordagem incremental adotada, com transição gradual do ambiente visual para o textual, mostrou-se eficaz para reduzir barreiras iniciais e estimular a curiosidade dos estudantes. Além disso, o acesso a uma ferramenta tangível e de baixo custo operacional – considerando que o IFAM já dispõe de drones que se encontram subaproveitados – torna a adoção desse método ainda mais factível e escalonável em sala de aula.

A possibilidade de trabalhar conceitos complexos de inteligência computacional, como algoritmos de aprendizagem de máquina e processamento de imagens em tempo real, reforça a relevância dos drones Tello EDU para fomentar a pesquisa aplicada. Alunos e professores podem desenvolver projetos interdisciplinares, conectando áreas como eletrônica, automação, robótica, programação e até mesmo ciências exatas e biológicas, dependendo do tipo de investigação proposta. Nesse sentido, o Tello EDU não só desperta maior interesse e engajamento, mas contribui diretamente para a formação de competências consideradas essenciais no século XXI, como pensamento crítico, criatividade, resolução de problemas e colaboração.

Para trabalhos futuros e perspectivas, espera-se que este TCC sirva como base para a implementação de práticas efetivas e inovadoras no IFAM, evidenciando a efetividade dos drones como ferramentas pedagógicas e de pesquisa. Os resultados aqui apresentados reforçam a importância de prosseguir com investigações futuras, ampliando o número de turmas participantes e aprofundando o uso de técnicas de inteligência computacional. Dessa forma,

além de otimizar a infraestrutura já disponível na instituição, cria-se um ambiente de ensino dinâmico, capaz de preparar os alunos para os desafios tecnológicos e científicos atuais, confirmando a versatilidade e o potencial formativo dos drones Tello EDU.

REFERÊNCIAS

ABRANTES, Maria Gracielly Lacerda de. **O pensamento computacional na formação de professores de matemática na educação profissional e tecnológica: do currículo a prática.** 2023. 150 f. Dissertação (Mestrado em Educação) – Universidade Federal do Rio Grande do Norte, Natal, 2023.

ALVES, Washington; BREMGARTNER, Vitor. **Uso de drones para ensino de Lógica de Programação e estímulo do Pensamento Computacional.** In: WORKSHOP DE INFORMÁTICA NA ESCOLA (WIE), 28., 2022, Manaus. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022. p. 168-178. DOI: <https://doi.org/10.5753/wie.2022.225311>.

BRADSKI, G. R.; KAEHLER, A. **Learning OpenCV: Computer Vision with the OpenCV Library.** 1. ed. Sebastopol: O'Reilly Media, 2008.

BRASIL. ANAC: Agência Nacional de Aviação Civil. **Drones.** Disponível em: <https://www.gov.br/anac/pt-br/assuntos/drones>. Acesso em: 27 nov. 2024.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular Educação Básica.** Brasília, DF, out 2018.

BRENNAN, Karen; RESNICK, Mitchel. **New frameworks for studying and assessing the development of computational thinking.** In: ANNUAL MEETING OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION, 2012, Vancouver. Anais... Vancouver: AERA, 2012. p. 25.

CAÑAS, J. M.; MARTÍN-MARTÍN, D.; ARIAS, P.; VEGA, J.; ROLDÁN-ÁLVAREZ, D.; GARCÍA-PÉREZ, L.; FERNÁNDEZ-CONDE, J. **Open-Source Drone Programming Course for Distance Engineering Education.** Electronics, [S. l.], v. 9, n. 12, p. 2163, dez. 2020. Disponível em: <<https://doi.org/10.3390/electronics9122163>>. Acesso em: 07 jan. 2025. DOI: 10.3390/electronics9122163.

CHUOLING. **MediaPipe Hands.** 2023. Disponível em: <<https://chuoling.github.io/mediapipe/solutions/hands.html>>. Acesso em: 23 fev. 2025.

ESCOTÉ, D. **DJITelloPy: Library for interacting with DJI Ryze Tello drones [repositório online].** (2025). Disponível em: <<https://github.com/damiafuentes/DJITelloPy>>. Acesso em: 10 jan. 2025.

DJI. **DJI Store.** Disponível em: <<https://store.dji.com/>>. Acesso em: 27 nov. 2024.

DJI-SDK. **Tello-Python**. Disponível em: <<https://github.com/dji-sdk/Tello-Python?tab=readme-ov-file>>. Acesso em: 05 nov. 2024.

DORLING, Mark; WHITE, Dave. **Scratch: A way to logo and python**. In: ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, 46., 2015, Kansas City. Anais... New York: ACM, 2015. p. 191-196.

ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. 1. ed. New York: John Wiley & Sons, 2007.

GARCÍA, P.; ARRIBAS, L.; PUEYO, P.; RIAZUELO, L.; MURILLO, A. C. **Exploring DJI Tello as an Affordable Drone Solution for Research and Education**. In: 7th IBERIAN ROBOTICS CONFERENCE (ROBOT), Madrid, Spain, 2024, p. 1-6. doi: 10.1109/ROBOT61475.2024.10796954.

GHAZI, Georges; VOYER, Julien. **Use of a DJI Tello Drone as an Educational Platform in the Field of Control Engineering**. Proceedings of the Canadian Engineering Education Association (CEEAA), [S. l.], 2024. DOI: 10.24908/pceea.2023.17061. Disponível em: <<https://ojs.library.queensu.ca/index.php/PCEEA/article/view/17061>>. Acesso em: 10 nov. 2024.

GIERNACKI, W.; SKWIERCZYŃSKI, M.; WITWICKI, W.; WROŃSKI, P.; KOZIERSKI, P. **Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering**. In: 22ND INTERNATIONAL CONFERENCE ON METHODS AND MODELS IN AUTOMATION AND ROBOTICS (MMAR), 2017, Miedzyzdroje, Polônia. Anais... Miedzyzdroje, Polônia: s.n., 2017. p. 37-42. DOI: 10.1109/MMAR.2017.8046794.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital image processing**. 4. ed. Harlow: Pearson, 2018.

GOOGLE AI EDGE. **MediaPipe**. 2023. Disponível em: <<https://github.com/google-ai-edge/mediapipe>>. Acesso em: 21 out. 2024.

GOOGLE DEV. **MediaPipe Solutions Guide**. 2023. Disponível em: <<https://ai.google.dev/edge/mediapipe/solutions/guide?hl=pt-br>>. Acesso em: 25 out. 2024.

GUARDA, Graziela Ferreira; PINTO, Sérgio Crespo C. S. **Dimensões do Pensamento Computacional: conceitos, práticas e novas perspectivas**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 31., 2020, Online. Anais [...]. Porto Alegre:

Sociedade Brasileira de Computação, 2020. p. 1463-1472. DOI: <https://doi.org/10.5753/cbie.sbie.2020.1463>.

HAN, Jeong-Seop; LEE, Choong-Iyeol; YOUN, Young-Hwa; KIM, Sung-Jun. **A Study on Real-time Hand Gesture Recognition Technology by Machine Learning-based MediaPipe**. Journal of System and Management Sciences, v. 12, n. 2, p. 462-476, 2022. doi: 10.33168/JSMS.2022.0225.

JESUS, Camille; SANTANA, Bianca; BITTENCOURT, Roberto. **Oficinas de Aprendizagem de Programação com Scratch e Python em um Curso de Engenharia de Computação**. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27, 2019, Belém. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 31-40. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2019.6614>.

MAJEED, Kazemitabaar; CHYHIR, Viktar; WEINTROP, David; GROSSMAN, Tovi. **CodeStruct: Design and Evaluation of an Intermediary Programming Environment for Novices to Transition from Scratch to Python**. In: Interaction Design and Children (IDC '22), 2022, Braga, Portugal. Proceedings [...]. New York, NY, USA: ACM, 2022. Disponível em: <https://doi.org/10.1145/3501712.3529733>. Acesso em: 12 dez. 2024.

KHOZA, N.; OWOLABI, P.; MALELE, V. **Drone Gesture Control using OpenCV and Tello**. In: CONFERENCE ON INFORMATION COMMUNICATIONS TECHNOLOGY AND SOCIETY (ICTAS), 2024, Durban. Anais do ICTAS 2024. Durban: Organizing Committee, 2024. p. 204-208. DOI: 10.1109/ICTAS59620.2024.10507111.

KINIVI. **Hand Gesture Recognition Mediapipe**. 2023a. Disponível em: <<https://github.com/kinivi/hand-gesture-recognition-mediapipe>>. Acesso em: 13 nov. 2024.

KINIVI. **Tello Gesture Control**. 2023b. Disponível em: <<https://github.com/kinivi/tello-gesture-control>>. Acesso em: 10 jan. 2025.

LATIF, B.; BUCKLEY, N.; SECCO, E. L. **Hand Gesture and Human-Drone Interaction**. In: ARAI, K. (eds) Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems, v. 544. Cham: Springer, 2023. doi: 10.1007/978-3-031-16075-2_20.

LOBO, D.; Patel, D.; Morainville, J.; Shekhar, P.; Abichandani, P. **Preparing Students for Drone Careers Using Active Learning Instruction**. IEEE Access, [S. l.], v. 9, p. 126216-126230, 2021. Disponível em: <<https://doi.org/10.1109/ACCESS.2021.3110578>>. Acesso em: 2 fev. 2025.

LUGARESI, C.; TANG, J.; NASH, H.; MCCLANAHAN, C.; UBOWEJA, E.; HAYS, M.; ZHANG, F.; CHANG, C.; YONG, M. G.; LEE, J.; CHANG, W.; HUA, W.; GEORG, M.; GRUNDMANN, M. **MediaPipe: A Framework for Building Perception Pipelines**. ArXiv, 2019, arXiv:1906.08172. Disponível em: <<https://arxiv.org/abs/1906.08172>>. Acesso em: 5 fev. 2025.

MARJI, Majed. **Aprenda a programar com Scratch**. Tradução de Lúcia Kinoshita. São Paulo: Novatec, 2014.

MENEZES, Nilo Ney Coutinho. **Introdução à programação com Python: algoritmos e lógica de programação para iniciantes**. São Paulo: Novatec, 2010. p. 16-23. ISBN 978-85-7522-250-8.

MITRA, S.; ACHARYA, T. **Gesture Recognition: A Survey**. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, v. 37, n. 3, p. 311–324, 2007. DOI: 10.1109/TSMCC.2007.893280.

MLADENOVIC, Monika; KR PAN, Divna; MLADENOVIC, Sasa. **Introducing programming to elementary students novices by using game development in Python and Scratch**. In: INTERNATIONAL CONFERENCE ON EDUCATION AND NEW LEARNING TECHNOLOGIES, 2016, Barcelona, Espanha. Anais... Barcelona, Espanha: EDULEARN, 2016. Disponível em: <<https://doi.org/10.21125/edulearn.2016.1323>>. Acesso em: 27 nov. 2024

MORAES, Guilherme; EBERMAM, Elivelto; RIOS, Renan; PULINI, Igor; NARDI, Julio. **Uma abordagem de ensino de programação de computadores utilizando Scratch e Python**. In: V SINECT – V SIMPÓSIO NACIONAL DE ENSINO DE CIÊNCIA E TECNOLOGIA; II SEMANA ACADÊMICA DA LICENCIATURA INTERDISCIPLINAR EM CIÊNCIAS NATURAIS, 2016, Ponta Grossa – PR, Brasil, novembro 2016. Anais... Ponta Grossa – PR, Brasil: s.n., 2016.

OLIVEIRA, M. R. **Desenvolvimento da tecnologia de veículos aéreos não tripulados no mercado brasileiro**. 2022. 198 f. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2022.

OPENCV. **OpenCV**. (2023a). Disponível em: <<https://opencv.org/>>. Acesso em: 16 nov. 2024.

OPENCV. **Opencv-python**. (2023b). Disponível em: <<https://github.com/opencv/opencv-python>>. Acesso em: 18 out. 2024.

PAPERT, Seymour. **Mindstorms: children, computers, and powerful ideas**. New York: Basic Books, 1980.

PETROVIČ, Pavel; VERČIMÁK, Peter. **Using Programmable Drone In Educational Projects And Competitions**. In: INTERNATIONAL CONFERENCE ON EDUCATION AND NEW LEARNING TECHNOLOGIES (EDULEARN), 15., 2023, Palma. Anais...[S. l.]: IATED, 2023. p. 5624-5633. ISBN 978-84-09-52151-7. ISSN 2340-1117. Disponível em: <<https://doi.org/10.21125/edulearn.2023.1472>>. Acesso em: 08 fev. 2025

PIAGET, J. **Seis estudos de psicologia**. São Paulo: Forense Universitária, 1964.

PYTHON SOFTWARE FOUNDATION. **PSF Landing**. Disponível em: <<https://www.python.org/psf-landing/>>. Acesso em: 13 jan. 2025.

ROCHA, João Victor Medeiros. **Controle De Posicionamento De Robô Aéreo Autônomo Utilizando Visão Computacional**. Projeto Final de Curso (Graduação em Engenharia Eletrônica e de Telecomunicações) - Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Campus Patos de Minas, Patos de Minas, 2023.

RYZE ROBOTICS. **Ryze**. Disponível em: <<https://www.ryzerobotics.com/>>. Acesso em: 13 jan. 2025.

SAINI, T. **Manoeuvring drone (Tello and Tello EDU) using body poses or gestures**. 2021. Projecte Final de Màster Oficial – UPC, Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels, Departament d'Enginyeria Telemàtica, Castelldefels, Espanha, 29 out. 2021. Disponível em: <http://hdl.handle.net/2117/356499>. Acesso em: 27 nov. 2024.

SZELISKI, Richard. **Computer vision: algorithms and applications**. 1. ed. Berlin: Springer-Verlag, 2010.

VENTURA, A. A. de O.; ALBUQUERQUE, J. de L.; GOMES, K. R. F. P.; NASCIMENTO, S. M. do.; LEITE, E. F...; ALVES, J. L.; DINIZ, J. R. B.; FRANÇA, S. V. A. **Educational robotics and the use of drones in education: a systematic mapping of the literature**. Research, Society and Development, [S. l.], v. 11, n. 17, p. e251111739115, 2022. DOI: 10.33448/rsd-v11i17.39115. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/39115>>. Acesso em: 9 jan. 2025.

WING, Jeannette M. **Computational thinking and thinking about computing**. Philosophical Transactions of the Royal Society A, London, v. 366, n. 1881, p. 3717-3725, jul. 2008. Disponível em: <<https://doi.org/10.1098/rsta.2008.0118>>. Acesso em: 08 jan. 2025.

WING, Jeannette M. **Computational thinking**. Communications of the ACM, New York, v. 49, n. 3, p. 33-35, mar. 2006. Disponível em: <<https://doi.org/10.1145/1118178.1118215>>. Acesso em: 08 jan. 2025.

YADAV, Aman; MAYFIELD, Chris; ZHOU, Ninger; HAMBRUSCH, Susanne; KORB, John T. **Computational Thinking in Elementary and Secondary Teacher Education**. ACM Trans. Comput. Educ., v. 14, n. 1, art. 5, mar. 2014, 16 p. DOI: 10.1145/2576872.

Yepes, I. (2020). **Uso de drones como Tecnologia pedagógica em disciplinas STEAM: um enfoque voltado ao aprendizado significativo com metodologias ativas** (Tese de Doutorado). Universidade Federal do Rio Grande do Sul, Centro Interdisciplinar de Novas Tecnologias na Educação, Porto Alegre.

YEPES, I.; BARONE, D. **Robótica Educativa: Drones e Novas Perspectivas**. RENOTE, Porto Alegre, v. 16, n. 2, p. 210–219, 2018. DOI: 10.22456/1679-1916.89293. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/89293>. Acesso em: 13 jan. 2025.

ZHANG, F.; BAZAREVSKY, V.; VAKUNOV, A.; TKACHENKA, A.; SUNG, G.; CHANG, C.; GRUNDMANN, M. **MediaPipe Hands: On-device Real-time Hand Tracking**. ArXiv, abs/2006.10214, 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:219792872>. Acesso em: 27 nov. 2024.

APÊNDICE

APÊNDICE A – CAPTURA DE VÍDEO COM DRONE TELLO

```
from djitellopy import Tello # Importa a biblioteca DJITelloPy para controle do drone
import cv2 # Importa a biblioteca OpenCV para processamento de imagens

# Inicializa o drone
# Criamos uma instância do objeto Tello para controlar o drone

drone = Tello()
drone.connect() # Conecta ao drone

drone.streamon() # Inicia a transmissão de vídeo

while True:
    frame = drone.get_frame_read().frame # Captura um frame da câmera do drone
    frame = cv2.resize(frame, (640, 480)) # Redimensiona o frame para 640x480 pixels
    cv2.imshow("Tello Camera", frame) # Exibe o frame em uma janela chamada "Tello
    Camera"

    if cv2.waitKey(1) & 0xFF == ord('q'): # Aguarda pressionamento da tecla 'q' para sair do
    loop
        break

cv2.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV
drone.streamoff() # Desativa a transmissão de vídeo do drone
```

APÊNDICE B - CAPTURA DE VÍDEO EM TONS DE CINZA

```
from djitellopy import Tello # Importa a biblioteca DJITelloPy para controle do drone
import cv2 # Importa a biblioteca OpenCV para processamento de imagens

# Inicializa o drone
# Criamos uma instância do objeto Tello para controlar o drone

drone = Tello()
drone.connect() # Conecta ao drone

drone.streamon() # Inicia a transmissão de vídeo

# Loop para capturar e exibir os frames em tons de cinza
while True:
    frame = drone.get_frame_read().frame # Captura um frame da câmera do drone
    frame = cv2.resize(frame, (640, 480)) # Redimensiona o frame para 640x480 pixels
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Converte o frame para tons de cinza
    cv2.imshow("Gray Image", gray) # Exibe o frame em uma janela chamada "Gray Image"

    if cv2.waitKey(1) & 0xFF == ord('q'): # Aguarda pressionamento da tecla 'q' para sair do
loop
        break

cv2.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV
```

APÊNDICE C - DETECÇÃO DE BORDAS COM CANNY

```

from djitellopy import Tello # Importa a biblioteca DJITelloPy para controle do drone
import cv2 # Importa a biblioteca OpenCV para processamento de imagens

# Inicializa o drone
# Criamos uma instância do objeto Tello para controlar o drone

drone = Tello()
drone.connect() # Conecta ao drone

drone.streamon() # Inicia a transmissão de vídeo

# Loop para capturar e exibir os frames com detecção de bordas usando o algoritmo de
Canny
while True:
    frame = drone.get_frame_read().frame # Captura um frame da câmera do drone
    frame = cv2.resize(frame, (640, 480)) # Redimensiona o frame para 640x480 pixels
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Converte o frame para tons de cinza
    edges = cv2.Canny(gray, 50, 150) # Aplica o algoritmo de detecção de bordas Canny
    cv2.imshow("Edges", edges) # Exibe o frame processado em uma janela chamada "Edges"

    if cv2.waitKey(1) & 0xFF == ord('q'): # Aguarda pressionamento da tecla 'q' para sair do
loop
        break

cv2.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV

```

APÊNDICE D – CONTORNO E SEGMENTAÇÃO DE OBJETOS

```

from djitellopy import Tello # Importa a biblioteca DJITelloPy para controle do drone
import cv2 # Importa a biblioteca OpenCV para processamento de imagens

drone = Tello()
drone.connect() # Conecta ao drone
drone.streamon() # Inicia a transmissão de vídeo

# Loop para capturar e processar os frames para detecção de contornos
while True:
    frame = drone.get_frame_read().frame # Captura um frame da câmera do drone
    frame = cv2.resize(frame, (640, 480)) # Redimensiona o frame para 640x480 pixels

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Converte o frame para tons de cinza
    blurred = cv2.GaussianBlur(gray, (5, 5), 0) # Aplica um desfoque para suavizar a imagem

    # Aplica a detecção de bordas
    edges = cv2.Canny(blurred, 50, 150) # Detecta bordas na imagem

    # Encontra os contornos na imagem processada
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Desenha os contornos na imagem original
    cv2.drawContours(frame, contours, -1, (0, 255, 0), 2)

    # Exibe a imagem com contornos detectados
    cv2.imshow("Contours", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'): # Aguarda pressionamento da tecla 'q' para sair do
loop
        break

cv2.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV

```

APÊNDICE E – DETECÇÃO DE MÃOS PELO DRONE

```

import cv2 # Biblioteca OpenCV para manipulação de imagens
import mediapipe as mp # Biblioteca MediaPipe para detecção de mãos
from djitellopy import Tello # Biblioteca DJITelloPy para comunicação com o drone

# Inicializa o drone

drone = Tello() # Cria uma instância do drone Tello
drone.connect() # Conecta ao drone
drone.streamon() # Inicia a transmissão de vídeo do drone

# Inicializa os módulos do MediaPipe Hands
mp_drawing = mp.solutions.drawing_utils # Utilitário para desenhar landmarks das mãos
mp_drawing_styles = mp.solutions.drawing_styles # Estilos para as landmarks das mãos
mp_hands = mp.solutions.hands # Módulo do MediaPipe para detecção de mãos

# Inicializa o modelo MediaPipe Hands para detecção de mãos em tempo real
with mp_hands.Hands(
    model_complexity=0, # Define a complexidade do modelo (0 para melhor desempenho)
    min_detection_confidence=0.5, # Confiância mínima para considerar a detecção de uma
    mão
    min_tracking_confidence=0.5) as hands: # Confiância mínima para o rastreamento de
    mãos

    while True:
        # Captura um frame da câmera do drone
        frame = drone.get_frame_read().frame # Obtém o frame atual da câmera do drone
        frame = cv2.resize(frame, (640, 480)) # Redimensiona o frame para 640x480 pixels

        # Converte a imagem para RGB, pois o MediaPipe requer este formato
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False # Otimiza o processamento marcando a imagem como
        não editável
        results = hands.process(image) # Processa a imagem no modelo para detectar mãos

        # Desenha os pontos das mãos detectadas na imagem
        image.flags.writeable = True # Reativa a capacidade de edição da imagem
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # Converte de volta para BGR para
        exibição no OpenCV
        if results.multi_hand_landmarks: # Se houver mãos detectadas
            for hand_landmarks in results.multi_hand_landmarks:
                mp_drawing.draw_landmarks(
                    image, hand_landmarks, mp_hands.HAND_CONNECTIONS,
                    mp_drawing_styles.get_default_hand_landmarks_style(), # Estilo dos pontos da
                    mão

```

```
mp_drawing_styles.get_default_hand_connections_style()) # Estilo das conexões  
entre os pontos
```

```
# Exibe a imagem processada com as detecções de mãos  
cv2.imshow('MediaPipe Hands - Tello Camera', image)  
if cv2.waitKey(5) & 0xFF == 27: # Pressione ESC para sair do loop  
    break
```

```
# Finaliza o processo  
cv2.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV  
drone.streamoff() # Desativa a transmissão de vídeo do drone
```

APÊNDICE F – QR CODE PARA PLAYLIST DA APLICAÇÃO

ANEXOS

ANEXO A – ATIVIDADES REALIZADAS PELO ALUNO A

1-1



1-2



1-3



1-4



1-5



ANEXO B – ATIVIDADES REALIZADAS PELO ALUNO B





ANEXO C – ATIVIDADES REALIZADAS PELO ALUNO C

TMI11

1-1 →



1-2 →



1-3 →



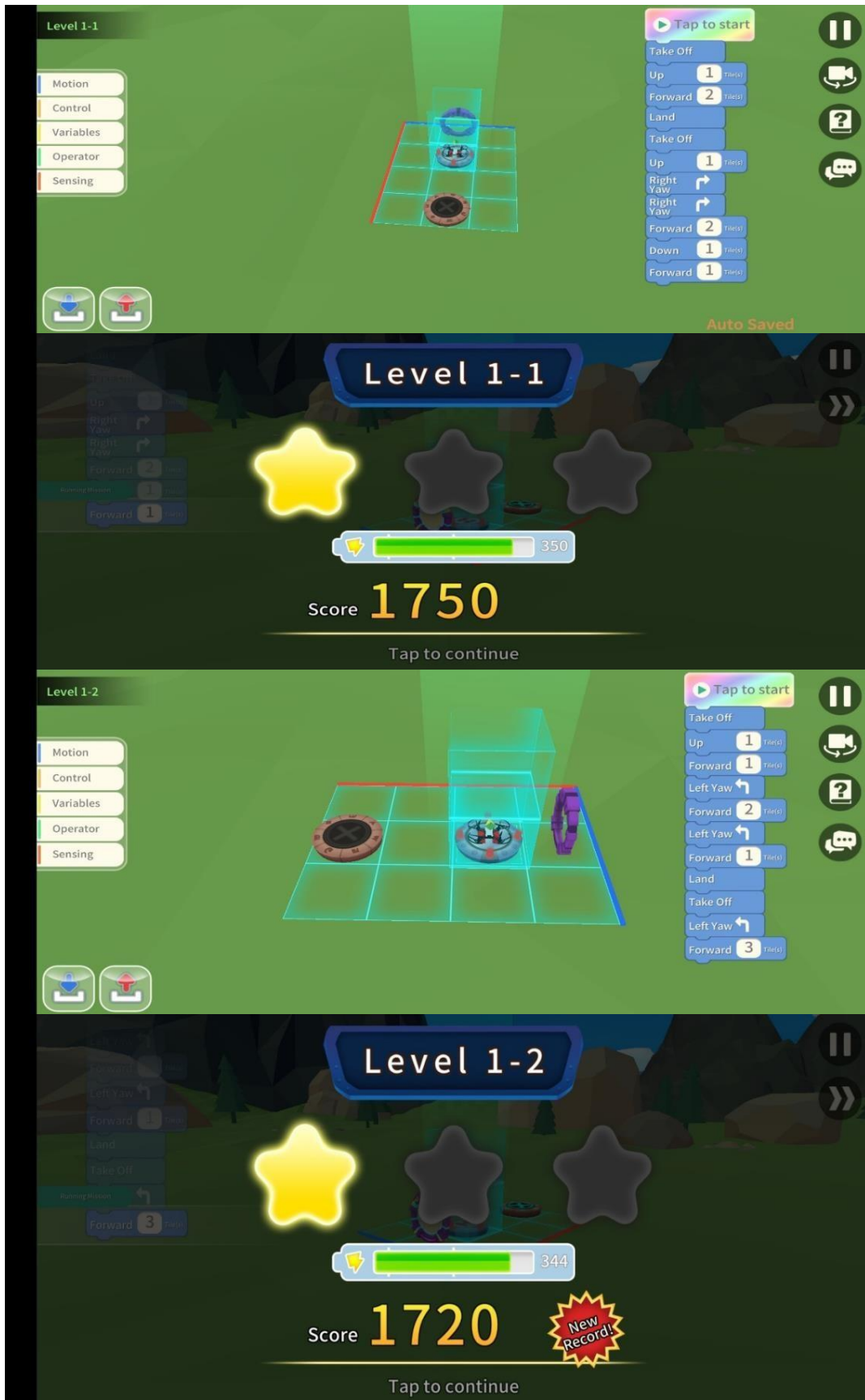
1-4 →



1-5 →



ANEXO D – ATIVIDADES REALIZADAS PELO ALUNO D





The image displays three sequential screenshots from a game interface, likely a block-based programming environment.

Top Screenshot (Level Editor): Shows a 3D environment with a grid floor and several cardboard boxes. A small robot is positioned on the grid. On the left, a menu lists categories: Motion, Control, Variables, Operator, and Sensing. On the right, a vertical stack of blue blocks represents a programmed sequence: Back (2), Land, Take Off, Forward (4), Land, Take Off, Back (2), Left (2), Forward (2), Land, Take Off, Back (2), Left (3), and Forward (2). Each block has a 'Tier(s)' indicator. A toolbar on the right contains icons for undo, redo, help, and chat. At the bottom left, there are two icons representing different robot modes.

Middle Screenshot (Level Completion): Features a central blue banner with the text "Level 1-5". Below it, a yellow number "2" is displayed next to the text "Total Block(s)". Three grey stars are shown in a row. Below the stars, the text reads "Level Completed!" followed by two conditions: "Use no more than 29 blocks." and "Use no more than 21 blocks." The background shows the 3D environment with boxes and a robot.

Bottom Screenshot (Leaderboard): Shows a leaderboard for the "Earth" category. It lists five levels with their top scores and star ratings:

- Level 1-1: Top Score 1750, 3 stars
- Level 1-2: Top Score 1720, 3 stars
- Level 1-3: Top Score 5843, 4 stars
- Level 1-4: Top Score 1555, 3 stars
- Level 1-5: Top Score 2937, 3 stars

Navigation arrows and a "1/1" indicator are visible. A text box at the bottom contains the message: "When facing new challenges, boldly try out new solutions. Get familiar with the tools and level-beating techniques."

ANEXO E – ATIVIDADES REALIZADAS PELO ALUNO E



