Sistema para Controle de Voo de Drone com Restrição Espacial Baseada em Geolocalização

Matheus Braga, André Flores dos Santos
Curso de Ciência da Computação
UFN - Universidade Franciscana
Santa Maria - RS
matheus.braga@ufn.edu.br, andre.flores@ufn.edu.br

***Resumo*-Este artigo apresenta o desenvolvimento de um sistema para delimitação da área de voo de um drone em um espaço tridimensional, utilizando coordenadas geográficas. O projeto abrange tanto os componentes de *hardware* quanto de *software*, adotando a metodologia ágil FDD. Foi desenvolvido um circuito com módulo GPS responsável por fornecer os dados de localização do drone, enquanto o sistema realiza cálculos geográficos para controlar sua locomoção. O sistema foi testado e validado em um ambiente de voo controlado, permitindo avaliar sua eficácia e aplicabilidade em contextos operacionais.**

***Palavras-chave*: drone, geofencing, coordenadas, interface, ESP8266.**

I. INTRODUÇÃO

A atual revolução tecnológica tem colocado a robótica em evidência, consolidando-a como uma das áreas mais promissoras da ciência moderna [1]. Com a capacidade de desenvolver soluções inovadoras em *softwares* e *hardwares*, a robótica tem transformado diversos setores da indústria, desde a saúde até a exploração espacial. A presença de robôs na vida cotidiana tem se tornado cada vez mais comum, automatizando atividades rotineiras e ampliando as possibilidades de atuação em ambientes extremos, inclusive em outros planetas [2], [3].

Nesse contexto, os drones, uma categoria específica de robôs aéreos, surgem como ferramentas essenciais em uma ampla gama de aplicações. Eles são utilizados desde a captura de imagens aéreas e entrega de suprimentos em áreas remotas até operações de vigilância e monitoramento de zonas sensíveis [4]. Equipados com sistemas de navegação avançados, como o Sistema de Posicionamento Global (GPS - *Global Positioning System*), esses dispositivos são capazes de voar com precisão e autonomia, mesmo em condições adversas. O uso do GPS não apenas facilita o rastreamento e controle dos drones, como também permite a programação de rotas específicas e a coleta de dados de forma eficiente [5].

Entretanto, a integração entre as áreas de robótica, pilotagem de drones e tecnologia GPS apresenta desafios. Questões relacionadas à segurança e privacidade também se destacam, especialmente porque drones equipados com GPS podem coletar imagens ou vídeos não autorizados, exigindo um alto nível de responsabilidade em sua operação [6].

A. Justificativa

O desenvolvimento de um sistema para controlar um drone com tecnologia GPS dentro de uma área de voo delimitada por *geofencing* (limite geográfico virtual em torno de áreas específicas de interesse) justifica-se pela crescente importância da robótica e da automação em diversos setores da sociedade, além das novas regulamentações sobre o uso de drones em determinados locais. Segundo Alamouri [7], em janeiro de 2021, a Agência Europeia para a Segurança da Aviação (EASA - *European Union Aviation Safety Agency*) estabeleceu novas regras para o uso de drones, cujo impacto pode ser analisado a partir de três aspectos centrais: limitações de alcance visual, sistemas de *geofencing* e considerações de privacidade. A pesquisa de Alamouri indicou que 75% dos participantes relataram dificuldades para implementar as novas normas de voo em seus projetos. Ademais, a pesquisa também apontou que 44% dos entrevistados enfrentaram problemas relacionados à privacidade, o que afetou diretamente as missões de voo e levou ao cancelamento de alguns projetos.

Portanto, o foco deste projeto está no desenvolvimento de um sistema para realizar *geofencing*, capaz de atender às novas normas de segurança estabelecidas por órgãos reguladores, conforme descrito no trabalho de Alamouri [7]. A proposta visa não apenas garantir a conformidade com esses requisitos, mas também promover práticas seguras e confiáveis no uso de tecnologias automatizadas, contribuindo para a prevenção de riscos e o fortalecimento da credibilidade em aplicações tecnológicas cada vez mais presentes em setores estratégicos da sociedade [8], [9], [10].

B. Objetivos

1) Objetivo Geral: Projetar, implementar e avaliar um sistema para controle de voo com a funcionalidade de delimitação de área, uma cerca virtual, utilizando coordenadas geográficas que

garanta o controle e monitoramento do comportamento de um drone.

2) Objetivos Específicos:

- Programar e documentar o desenvolvimento de um sistema *desktop* para controle de voo para drones.
- Construir, programar e documentar um sistema embarcado GPS para a coleta de coordenadas geográficas.
- Programar e documentar métodos de conversão de coordenadas geográficas em medidas lineares (em metros).
- Programar e documentar técnicas de *geofencing* para drones.
- Definir e aplicar um ambiente de teste para avaliar o desempenho do sistema de controle, a precisão dos dados do GPS e a estabilidade de voo do drone.

Para uma melhor compreensão deste projeto, o texto está dividido nas seguintes seções: Referencial Teórico, que aborda os conteúdos utilizados no projeto; Trabalhos Correlatos, que demonstram temas e estudos que ajudaram neste projeto; Metodologia, que descreve os procedimentos adotados durante o desenvolvimento do projeto; Resultados, que apresenta uma análise geral dos dados obtidos, além de expor as conclusões alcançadas com o desenvolvimento do sistema proposto; e Referências, que reúne as obras utilizadas como base teórica para o desenvolvimento do texto e do projeto.

II. REFERENCIAL TEÓRICO

Nesta seção, são abordados os principais conceitos, tecnologias e práticas que sustentam o desenvolvimento deste trabalho. Apresentam-se também as diretrizes de segurança aplicáveis à sua implementação. Foram exploradas ainda as tecnologias relacionadas ao controle, à programação e à construção de circuitos, destacando como esses elementos se integram na criação de um sistema funcional de delimitação de área.

A. Metodologia Ágil

Para o desenvolvimento deste trabalho, adotou-se uma metodologia ágil, cujo objetivo é permitir respostas rápidas e adaptáveis às mudanças ao longo do processo. As metodologias ágeis favorecem a divisão do projeto em etapas incrementais, proporcionando uma visão progressiva do desenvolvimento, bem como promovendo uma comunicação contínua e ajustes constantes nos processos.

Considerando a necessidade de detalhamento de cada etapa para garantir melhor organização e compreensão do projeto, optou-se pela metodologia Desenvolvimento Orientado a Funcionalidades (FDD - *Feature-Driven Development*), desenvolvida por Jeff De Luca em parceria com Peter Coad, em 1997, durante um projeto de desenvolvimento de *software* para um grande banco em Singapura [11].

Essa abordagem facilita a estruturação do trabalho em tarefas menores e bem definidas, contribuindo para um desenvolvimento mais eficiente e rastreável. Além disso, combina práticas ágeis com uma estruturação mais detalhada, sendo especialmente adequada para equipes maiores e projetos de alta complexidade. O FDD foca na entrega de funcionalidades específicas que trazem valor ao usuário final, proporcionando um processo de desenvolvimento contínuo e incremental [12].

A Figura 1 ilustra detalhadamente os passos da metodologia FDD que serão seguidos na construção e desenvolvimento do projeto.

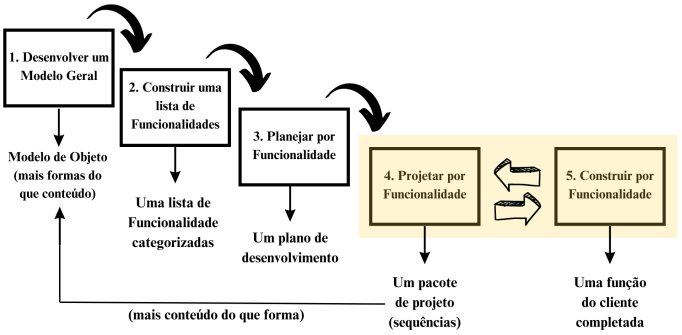

Figura 1. Exemplo do modelo FDD. Fonte: criação do autor

B. Drone DJI Tello

O Tello, drone desenvolvido pela Ryze Tech em parceria com a DJI e a Intel, destaca-se como uma plataforma acessível voltada para fins educacionais, experimentais e de pesquisa em robótica e automação. Com dimensões reduzidas e peso leve (cerca de 80 gramas), o dispositivo é amplamente utilizado em ambientes internos, favorecendo atividades supervisionadas e prototipagem de sistemas. Sua interface amigável e facilidade de operação tornam o Tello uma escolha comum em instituições de ensino, especialmente em projetos introdutórios sobre navegação autônoma e controle de voo [13].

Uma das principais vantagens do Tello, para este projeto, é sua compatibilidade com linguagens de programação como Scratch e Python, além da possibilidade de integração com o Sistema Operacional do Robô (ROS - *Robot Operating System*) por meio de bibliotecas mantidas por comunidades acadêmicas. Essa flexibilidade permite o desenvolvimento e a validação de sistemas de controle, algoritmos de geolocalização, *geofencing* e funcionalidades básicas de voo autônomo. Mesmo com limitações de hardware quanto à autonomia e capacidade de carga (13 minutos de voo), o Tello se mostra eficaz como ferramenta didática para experimentos em automação, visão computacional e inteligência artificial embarcada [14].

Adicionalmente, o drone conta com ampla documentação técnica, bibliotecas de código aberto e suporte ativo da comunidade de desenvolvedores, o que facilita sua aplicação em projetos multidisciplinares. Esse ecossistema de apoio

contribui significativamente para a popularização do Tello em contextos acadêmicos e educacionais, potencializando o aprendizado prático em áreas como Internet das Coisas (IoT - *Internet of Things*), robótica móvel e monitoramento ambiental em pequena escala [15, 16].

C. Global Positioning System

O Sistema de Posicionamento Global (GPS - *Global Positioning System*) é uma tecnologia baseada em satélites que permite a localização geográfica de um objeto em qualquer ponto da superfície terrestre. Desenvolvido originalmente para fins militares, o GPS foi posteriormente liberado para uso civil e hoje é amplamente empregado em aplicações que exigem navegação, rastreamento e sincronização de tempo. Seu funcionamento baseia-se na triangulação de sinais emitidos por uma constelação de satélites, permitindo que receptores no solo calculem sua posição com base na distância de pelo menos quatro satélites. Em projetos de engenharia e robótica, como o controle de drones, é crucial o uso de GPS para fornecer dados de latitude, longitude e altitude, possibilitando o monitoramento contínuo e a automação de rotas [17].

D. Matemática do GPS

A conversão de coordenadas geográficas para medições lineares é uma etapa fundamental em aplicações que exigem o cálculo preciso de distâncias. As coordenadas geográficas, expressas em latitude, longitude e altitude, representam posições sobre a superfície terrestre com base em um *datum* geodésico — um modelo matemático que descreve a forma da Terra e define o ponto de origem, a orientação e o sistema de referência utilizado para as medições [18]. Latitude e longitude, por serem medidas angulares, não são diretamente adequadas para cálculos de distância em unidades métricas.

Para superar essa limitação, uma abordagem amplamente utilizada é a fórmula de Haversine [19], que permite calcular a distância entre dois pontos sobre a superfície da Terra utilizando apenas suas latitudes e longitudes. Essa fórmula considera a curvatura da Terra ao estimar a menor distância sobre a superfície esférica entre dois pontos, com base em funções trigonométricas e no raio médio terrestre. Por sua simplicidade e eficiência computacional, o método é especialmente útil em aplicações embarcadas, como sistemas de geolocalização e monitoramento de trajetórias em tempo real.

A altitude, por sua vez, representa a componente vertical (Z) e é geralmente expressa em metros em relação a um *datum* vertical. Quando combinada com a distância horizontal obtida pela fórmula de Haversine, é possível estimar deslocamentos tridimensionais e realizar análises espaciais mais completas, como o cálculo de trajetórias, delimitação de áreas e verificação de limites de geofencing em ambientes controlados.

E. Tecnologias para Captação de Coordenadas

O setor de tecnologias para automação, controle e captura de dados inclui diversas ferramentas que auxiliam na otimização de processos por meio de comunicação remota, identificação eficiente e processamento de dados. Dentre as tecnologias utilizadas, destacam-se o módulo ESP8266, empregado em soluções de conectividade sem fio, e o módulo GNSS AT6558 (GPS), utilizado para obtenção de dados de localização.

1) NodeMCU Esp8266: A placa de prototipagem NodeMCU ESP8266 é um módulo de comunicação sem fio amplamente utilizado em projetos de Internet das Coisas (IoT - *Internet of Things*). Baseada em um microcontrolador com capacidade de conectividade Wi-Fi, a ESP8266 permite a comunicação de dispositivos com redes locais e a internet, proporcionando uma interface eficiente para aplicações em automação, monitoramento remoto e controle de sistemas. Seu baixo custo, juntamente com o suporte a protocolos de comunicação como o Protocolo de Controle de Transmissão/Protocolo de Internet (TCP/IP - *Transmission Control Protocol/Internet Protocol*), torna a ESP8266 uma plataforma acessível e versátil para desenvolvimento de protótipos e soluções embarcadas. Além disso, a vasta comunidade de desenvolvedores contribui com bibliotecas e tutoriais que facilitam a integração do módulo com outras tecnologias e sistemas. Sua flexibilidade e capacidade de processamento, aliadas à facilidade de programação, fazem do ESP8266 uma escolha popular em ambientes acadêmicos e industriais para pesquisa e desenvolvimento em tecnologias de comunicação sem fio [20].

2) GPS Gnss AT6558: O módulo GPS utilizado neste projeto é baseado no chip AT6558, um componente de posicionamento multimodal de alta integração que oferece suporte a múltiplos Sistemas Globais de Navegação por Satélite (GNSS - *Global Navigation Satellite System*). Esse módulo é capaz de rastrear até 32 canais simultaneamente e receber sinais de seis Sistemas de Navegação por Satélite: o *Beidou* (China), o GPS (Estados Unidos), o GLONASS (Rússia), o *Galileo* (Europa), o QZSS (Japão). Essa ampla compatibilidade permite a realização de operações conjuntas de posicionamento, navegação e temporização com maior precisão e confiabilidade. Seu projeto compacto facilita a integração em aplicações com restrições de espaço, como drones, veículos autônomos, dispositivos portáteis e vestíveis. A comunicação do módulo é feita por meio de interface serial, o que o torna compatível com diversas plataformas de prototipagem e desenvolvimento, incluindo Arduino, Raspberry Pi e STM32 [21].

F. Tecnologias para Programação

No desenvolvimento de sistemas automatizados, as linguagens de programação são utilizadas para estruturar algoritmos, controlar dispositivos e viabilizar a integração entre os componentes do sistema. Python é amplamente utilizada devido à sua versatilidade e robustez, sendo aplicada em diversas áreas, incluindo inteligência artificial e automação. Já

linguagens como C e C++ são frequentemente empregadas na programação de sistemas embarcados, desde a implementação de funcionalidades baseadas em IA até o controle motor de braços robóticos. Para a comunicação entre os sistemas e hardware de robôs, são utilizados modelos de comunicação, como o TCP/IP.

1) Python: O Python é uma linguagem de programação de alto nível, reconhecida por sua simplicidade, legibilidade e versatilidade. Sua sintaxe intuitiva facilita o aprendizado para iniciantes, ao mesmo tempo que oferece poderosas bibliotecas e *frameworks* que a tornam ideal para uma ampla gama de aplicações, como automação, análise de dados, inteligência artificial e desenvolvimento *web*. Python é popular em áreas que envolvem processamento de dados e prototipagem rápida devido à sua flexibilidade e à extensa comunidade de desenvolvedores que contribuem com recursos e ferramentas. Sua compatibilidade com diversas plataformas e sua capacidade de integração com outras linguagens e sistemas fazem do Python uma escolha preferencial tanto para desenvolvedores iniciantes quanto experientes [22].

2) C/C++: As linguagens de programação C e C++ são muito utilizadas em contextos que exigem controle direto sobre o *hardware* e alta performance. A linguagem C, devido à sua simplicidade e eficiência, é comumente empregado no desenvolvimento de sistemas embarcados e em software que necessitam de manipulação de baixo nível, como sistemas operacionais e dispositivos de hardware. A linguagem C++, por sua vez, introduz o paradigma de programação orientada a objetos, permitindo a organização de código mais complexos e a reutilização de componentes, sendo frequentemente utilizado em áreas como desenvolvimento de sistemas em tempo real e aplicações que exigem uma grande quantidade de processamento. Ambas as linguagens continuam a ser amplamente aplicadas em robótica e sistemas de controle, onde a eficiência computacional e o gerenciamento de recursos são fatores críticos [23].

3) Modelo de Comunicação TCP/IP: De acordo com Mendes [24], o modelo TCP/IP é um conjunto de normas e protocolos utilizado para a troca de dados em redes de computadores, servindo como base tanto para a Internet quanto para redes locais (LANs - *Local Area Network*). Esse modelo possui aplicações em diversas áreas, como automação residencial e IoT. Entre seus elementos fundamentais estão os conceitos de *socket* e porta lógica. Um *socket* é um ponto de comunicação que permite a troca de dados entre dois dispositivos em uma rede; ele combina um endereço IP com um número de porta para identificar processos específicos. Cada processo que necessita estabelecer comunicação cria um *socket* para o envio e recebimento de dados. Ao receber um pacote, o sistema utiliza o endereço IP de destino e o número da porta para identificar o processo receptor, possibilitando que múltiplos processos operem simultaneamente em um mesmo dispositivo por meio de diferentes portas. Dessa forma, o modelo TCP/IP, por meio de sua estrutura modular, viabiliza a comunicação entre dispositivos e sistemas em redes computacionais de maneira organizada e funcional.

III. TRABALHOS CORRELATOS

Esta seção apresenta estudos que fundamentaram o desenvolvimento do sistema proposto, abordando tecnologias relacionadas à robótica móvel, *geofencing* e comunicação entre dispositivos embarcados e servidores. Os artigos apresentados nesta seção trazem informações sobre o uso do drone Tello DJI como plataforma para pesquisa e educação em robótica móvel e engenharia de controle, implementação de tecnologia de *geofencing* para rastreio de animais, e, por fim, a comunicação entre ESP8266 e servidor Python.

1. DJI Tello Quadrotor as a Platform for Research and Education in Mobile Robotics and Control Engineering: O estudo de Giernacki [25] destaca o uso do drone DJI Tello [13] como uma plataforma experimental acessível e eficaz em ambientes acadêmicos. Devido ao seu baixo custo, tamanho compacto e integração com o *Robot Operating System* (ROS), o Tello tem sido amplamente adotado em projetos educacionais e de pesquisa. Entre os exemplos notáveis estão o "*TelloBird*", um gerador de trajetórias baseado em código de barras (QR Code - *Quick Response Code*), e o "*Sentry drone*", desenvolvido para detectar e rastrear pessoas que entram em áreas específicas. Esses projetos ilustram a versatilidade do Tello em aplicações como controle de trajetória e vigilância autônoma, reforçando seu papel como ferramenta de prototipagem rápida em robótica móvel e engenharia de controle .

2. Geofencing technology implementation for pet tracker using Arduino based on Android: No projeto de Deni Setiawan [26] foi desenvolvido um sistema de rastreamento de animais de estimação utilizando tecnologia de *geofencing*, baseado em Arduino e integrado a um aplicativo Android. O sistema emprega o módulo GPS U-Blox Neo 6M para obtenção de coordenadas geográficas e o módulo GSM SIM800L para transmissão de dados via rede GPRS. O microcontrolador Arduino Pro Mini gerencia os processos de entrada e saída do sistema. O aplicativo Android recebe notificações quando o animal ultrapassa os limites predefinidos da *geofencing*. Os testes indicaram que o sistema funciona conforme o esperado, fornecendo alertas em tempo real sobre a movimentação dos animais.

3. Sistema de aquisição de dados utilizando o módulo ESP8266 NodeMCU: O trabalho desenvolvido por Pedro Cannavale Graça [27] apresenta um sistema de aquisição de dados que integra sensores de temperatura, umidade e luminosidade a um microcontrolador ESP8266 NodeMCU. Os dados coletados são transmitidos via Wi-Fi para uma plataforma IoT, permitindo tanto a visualização em tempo real

quanto o armazenamento para análises futuras. O acesso a essas informações é realizado remotamente por meio de um navegador *web*, proporcionando ao usuário a capacidade de monitorar as variáveis ambientais de forma eficiente e prática.

O trabalho de Giernacki [25] nos ajuda a entender o funcionamento do drone Tello, escolhido para este projeto, assim como tecnologias para controlá-lo. O trabalho de Setiawan [26] é o único que mostra a aplicação da tecnologia de *geofencing* em um sistema para delimitar a área e emitir avisos. Enquanto o trabalho de Pedro [27] aborda a integração de módulos e sensores a um microcontrolador, o ESP8266, que será usado no para o circuito do GPS neste projeto.

Os trabalhos correlatos apresentados oferecem uma base consistente para o desenvolvimento deste projeto, que será conduzido com o uso da metodologia ágil FDD. Essa abordagem possibilita a entrega incremental das principais funcionalidades do sistema, favorecendo a organização e o acompanhamento das etapas de implementação.

IV. METODOLOGIA

Esta seção descreve detalhadamente a metodologia adotada para o desenvolvimento do projeto, a qual compreende a condução de um estudo de caso, o levantamento e a definição de requisitos, o projeto de *software*, a elaboração de protótipos e a montagem do sistema, além da implementação do *software*. Material suplementar do projeto pode ser acessado em um repositório público [28].

A. Estudo de Caso

O estudo de caso é uma abordagem empírica que permite aproximar a teoria aprendida com a prática, avaliando a aplicabilidade e eficácia em um contexto real [12].

Com base na definição da metodologia ágil FDD, foi elaborado um esboço visual da interface do projeto com o objetivo de representar os componentes necessários para a construção do sistema. A Figura 2 apresenta esse esboço, correspondente à interface do sistema de *geofencing*. Nessa representação, são indicados os *frames*, áreas da tela que contém botões, textos e imagens, os quais estarão presentes durante a execução do sistema.

Na Figura 2, o *Frame 1* apresenta os componentes relacionados à configuração do *geofencing*, incluindo o tipo e a medida linear da área a ser delimitada. O *Frame 2* exibe os status do sistema referentes à conexão com o drone e ao monitoramento de suas coordenadas. O *Frame 3* mostra a posição do drone em relação à área previamente delimitada. Por fim, o *Frame 4* contém os controles de voo, que permitem a execução dos movimentos do drone durante o voo.

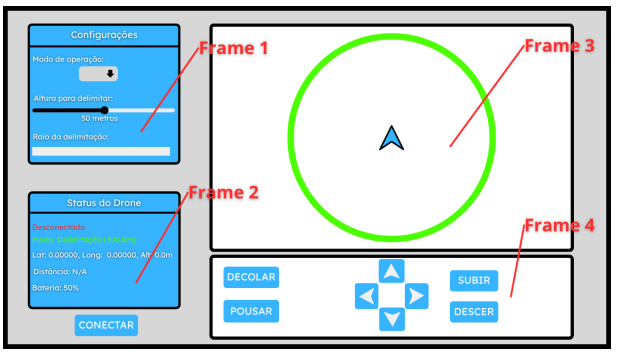

Figura 2. Esboço da interface para a construção do sistema. Fonte: criação do autor

A construção do esboço proporcionou uma visão clara do funcionamento do projeto, essencial para compreender os componentes e suas interações. Essa visualização facilitou o levantamento de requisitos ao revelar necessidades específicas de *hardware* e *software*, além de antecipar possíveis desafios técnicos.

B. Levantamento e Definição de Requisitos

As especificações de um sistema solicitadas pelo cliente devem ser organizadas em dois grupos. De acordo com Sommerville [29], os Requisitos Funcionais (RF) correspondem às especificações das funcionalidades que o sistema deve ser capaz de desempenhar. Já os Requisitos Não Funcionais (RNF) definem como o sistema deve operar, abrangendo aspectos como desempenho, segurança, usabilidade e confiabilidade. Neste projeto, foram identificados requisitos funcionais, como a delimitação da área de voo do drone e a montagem do circuito GPS, além de requisitos não funcionais relacionados a atributos de qualidade e desempenho do sistema, demonstrados na Tabela I.

Tabela I
RESULTADO DO LEVANTAMENTO DE REQUISITOS

RF01	O drone deve estabilizar o voo automaticamente.
RF02	O drone pode se mover baseado nas coordenadas do GPS.
RF03	O drone deve realizar movimentos direcionais (frente, trás, lados e giros).
RF04	O drone deve permanecer no ar por mais de um minuto.
RF05	O drone deve permanecer dentro da área definida pelo usuário.
RF06	O Software deve limitar a área de voo com base no valor informado pelo usuário.
RNF01	O drone deve ser controlado por rede Wi-Fi.
RNF02	O drone deve ser controlado por uma aplicação *desktop*.
RNF03	O drone deve ter quatro hélices.
RNF04	A placa utilizada deve ser uma ESP8266.
RNF05	O protocolo de comunicação deve ser TCP/IP.
RNF06	O módulo GPS utilizado deve ser o AT6558.
RNF07	O circuito do GPS deve estar fora do *hardware* do drone.

C. Projeto de Software

Na fase inicial deste projeto, foi criado e adaptado um diagrama de caso de uso com o objetivo de demonstrar, de forma simples, as funcionalidades do sistema de *geofencing*, facilitando o entendimento do seu funcionamento como um todo. A Figura 3, a seguir, apresenta o diagrama de caso de uso, que contém componentes acessíveis ao usuário, como: pilotar o drone, definir a área de delimitação e visualizar coordenadas. O diagrama também representa os componentes com os quais o sistema interage, como o circuito GPS externo, responsável por enviar as coordenadas ao *software*, que realiza a validação da área restrita.

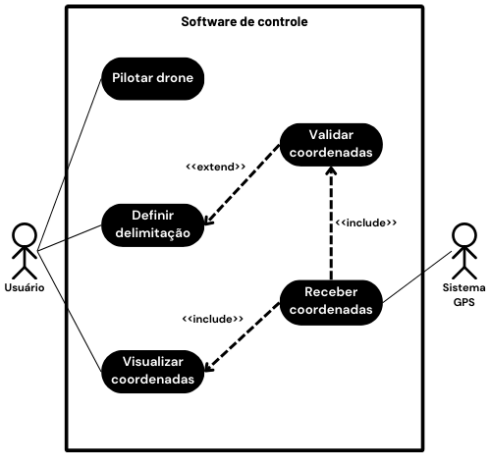

Figura 3. Diagrama do caso de uso. Fonte: Criação do autor.

Para detalhar o funcionamento do sistema de delimitação proposto, em conjunto com o circuito GPS e o drone Tello, foi elaborado um Diagrama de Atividades. Esse diagrama tem como objetivo esclarecer o papel de cada componente no ciclo de vida da aplicação, facilitando a compreensão do sistema e a identificação das responsabilidades funcionais. Ele inclui os seguintes elementos: Usuário, responsável por definir comandos e restrições de voo; Software, encarregado dos cálculos de delimitação e do controle do drone; Circuito GPS, que fornece e transmite as coordenadas geográficas ao software; e Drone, que executa os comandos de voo recebidos. O Diagrama de Atividades é apresentado na Figura 4.

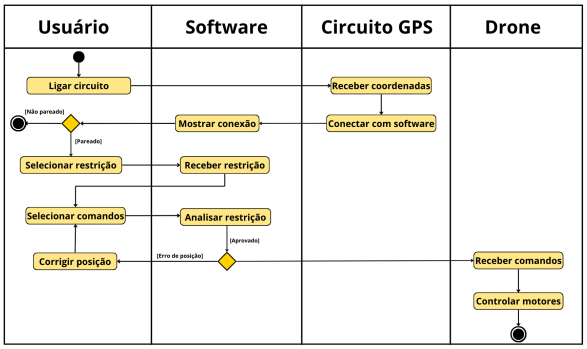

Figura 4. Diagrama de atividade. Fonte: Criação do autor.

D. Protótipos e Montagem

Durante o desenvolvimento do trabalho, a prototipagem foi adotada em conjunto com a metodologia FDD como método de validação dos requisitos do projeto. Com isso, foi possível garantir as principais funcionalidades do sistema.

Foi implementado o sistema embarcado GPS para a coleta de coordenadas geográficas, utilizando um microcontrolador ESP8266 programada em C++. A placa está conectada a um módulo GPS, responsável por captar as coordenadas, e é alimentada por uma bateria de 3,7V e 1000 mAh (miliampere). A Figura 5 apresenta o esquema eletrônico dos componentes. Embora o GPS seja representado no diagrama como um módulo NEO-6M, a aplicação real utiliza um módulo GNSS AT6558, uma escolha que atende à necessidade de compactar o circuito para que possa ser integrado externamente ao chassi do drone.

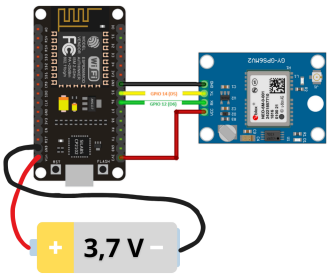

Figura 5. Esquema eletrônico. Fonte: Criação do autor.

A etapa de montagem do circuito teve como objetivo viabilizar a obtenção de coordenadas geográficas pelo sistema de *geofencing*. Essas coordenadas serão analisadas pelo sistema *desktop*, que, com base nessa análise, permitirá ou não o controle do drone pelo usuário dentro da área delimitada.

E. Implementação do Software

Para o desenvolvimento deste trabalho, foi implementado um sistema *desktop* utilizando a linguagem de programação Python. Foram empregadas as bibliotecas *CustomTkinter*, para a construção da interface gráfica; *math*, para a realização de cálculos matemáticos; *threading*, com o objetivo de viabilizar a execução simultânea de múltiplas tarefas no código; e *socket,* para a comunicação e controle de voo do drone Tello [22].

Para a implementação do *geofencing* no sistema, utilizou-se a fórmula de Haversine [19], a qual permite calcular a distância entre dois pontos na superfície esférica da Terra com base em suas coordenadas geográficas. Para isso, foi desenvolvido um código em C++ para o módulo ESP8266, conforme ilustrado na Figura 5, com a finalidade de transmitir as coordenadas geográficas do sistema embarcado ao sistema *desktop*, onde são realizados os cálculos da fórmula de Haversine para o funcionamento do *geofencing*.

O sistema utiliza uma comunicação baseada no protocolo TCP/IP, onde o ESP8266 atua como cliente e o computador atua como servidor. Os dados GPS são extraídos via interface serial com um módulo GPS, processados pela biblioteca *TinyGPS* e enviados como *strings* de texto no formato LAT: x , LON: y, ALT: z. A conexão é mantida via rede wi-fi, e a cada segundo, uma nova leitura é transmitida para permitir o monitoramento em tempo de execução da posição do drone. Logo abaixo na *Listagem 1* representa o código utilizado para o envio destas coordenadas.

```
// se tiver novas informações
  if (gps.location.isUpdated()) {
    String message = "";
    message += "LAT:" + String(gps.location.lat(), 6) + ",";
    message += "LON:" + String(gps.location.lng(), 6) + ",";
    message += "ALT:" + String(gps.altitude.meters(), 2);

    Serial.println("Enviando: " + message);
    client.println(message); // envia para o servidor

    delay(1000); // delay de 1 segundo para nova leitura/envio
  }
```

Listagem 1: Implementação do envio das coordenadas em C++ [29].

O trecho de código apresentado na *Listagem 2* ilustra a implementação, na linguagem Python, de uma versão adaptada da fórmula de Haversine [19], com o objetivo de realizar operações de *geofencing* em ambientes tridimensionais. Diferentemente da fórmula original, que considera apenas latitude e longitude para o cálculo de distâncias sobre a superfície terrestre, esta adaptação também leva em conta a altitude, permitindo uma análise espacial mais precisa em cenários que envolvem variações de elevação, como edificações, drones ou terrenos acidentados. A lógica do algoritmo baseia-se na comparação entre as coordenadas geográficas — latitude, longitude e altitude — coletadas em tempo de execução por um circuito embarcado, e um ponto de referência previamente definido. A partir desse cálculo, é possível verificar se o objeto monitorado permanece dentro dos limites estabelecidos de uma zona geográfica virtual.

```
def haversine_3d(lat1, lon1, alt1, lat2, lon2, alt2):
    R = 6371000  # raio médio da Terra, em metros
    phi1 = math.radians(lat1)
    phi2 = math.radians(lat2)
    delta_phi = math.radians(lat2 - lat1)
    delta_lambda = math.radians(lon2 - lon1)

    a = math.sin(delta_phi / 2)**2 + \
        math.cos(phi1) * math.cos(phi2) * \
        math.sin(delta_lambda / 2)**2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    horizontal_distance = R * c  # distância no plano horizontal

    delta_alt = alt2 - alt1  # diferença das altitudes

    # distância 3D combinando distância horizontal e vertical
    return math.sqrt(horizontal_distance**2 + delta_alt**2)
```

Listagem 2: Implementação do cálculo de *geofencing* em Python [29].

O método *haversine_3d* (linha 1) recebe como parâmetros os valores lat1, lon1 e alt1, que representam a posição atual do drone, e lat2, lon2 e alt2, que correspondem à posição de referência definida pelo usuário para a realização do *geofencing* através de cálculos matemáticos realizado no método, utilizando a biblioteca *math*. Esses dados são fornecidos pelo módulo GPS conectado ao sistema.

```
# verificar a geofencing apenas no modo estático
if self.operation_mode.get() == "static":
    distance = self.haversine(self.origin_lat,
                              self.origin_lon,
                              self.origin_altself.current_pos["lat"],
                              self.current_pos["lon"],
                              self.current_pos["alt"])

    self.in_geofence = distance <= self.geofence_radius.get()
    if not self.in_geofence and self.drone_connected:
        self.land()
```

Listagem 3: Implementação da condição da *geofencing* em Python [29].

Com a adaptação e implementação da fórmula de Haversine concluídas, foi possível desenvolver diferentes métodos de controle de voo para o drone. Para isso, utilizou-se a biblioteca *socket*, responsável pela comunicação entre o drone Tello e o sistema de *geofencing*, permitindo seu controle por meio da interface desenvolvida. A partir da estrutura condicional apresentada na *Listagem 3*, definiu-se uma lógica que identifica quando o drone ultrapassa os limites da área previamente delimitada pelo usuário, bem como alguma ação a ser executada nessa situação.

Para este sistema, também optou-se pela implementação de *threads*, de modo a permitir a execução simultânea de múltiplas tarefas [22]. Em particular, uma *thread* é dedicada ao recebimento, a cada segundo, das coordenadas atualizadas da posição do drone, enquanto as demais funcionalidades são executadas em paralelo.

Em seguida, iniciou-se a fase de testes, com o objetivo de validar os critérios estabelecidos neste trabalho.

V. RESULTADOS

Inicialmente, foi proposto para este projeto a adaptação de um ESP-Drone, desenvolvido pela empresa Espressif [30], com a adição de um módulo GPS ao seu *hardware*, além da adaptação de um aplicativo *mobile* para controle e monitoramento do drone. O objetivo era realizar testes com um sistema de *geofencing*, permitindo que decisões fossem tomadas com base na localização geográfica do drone.

Durante o desenvolvimento, utilizou-se a metodologia ágil FDD para assegurar que as principais funcionalidades do sistema estivessem operando corretamente, tais como o envio de coordenadas para o sistema e a validação de cálculos geográficos realizados para aplicar o *geofencing*. No entanto, ao alcançar a etapa relacionada ao voo do drone, o ESP-Drone apresentou instabilidades em seus movimentos, não

conseguindo se manter no ar de forma adequada para uma análise precisa das coordenadas obtidas via GPS.

Diante desse problema, optou-se pela substituição do ESP-Drone pelo drone Tello, da empresa DJI, que passou a desempenhar o papel de transportar o sistema embarcado GPS para a coleta das coordenadas. Com essa mudança, tornou-se necessário remodelar algumas etapas do projeto, uma vez que o módulo GPS não poderia mais ser integrado diretamente ao *hardware* do drone, pois o Tello possui estrutura fechada. Assim, foi projetado o circuito GPS externo, responsável por enviar as coordenadas ao sistema por meio do protocolo TCP/IP, permitindo a tomada de decisões quanto ao movimento do drone.

O circuito GPS foi construído com base no diagrama apresentado na Figura 5 e adaptado para ser acoplado ao chassi do drone Tello. Na Figura 6, é possível observar o drone Tello com o circuito GPS acoplado de forma externa.

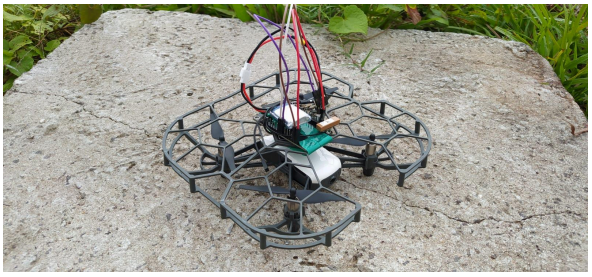

Figura 6. Drone Tello e circuito GPS [29].

O controle e monitoramento da posição do drone foi realizado por meio da interface, com uma aplicação *desktop* desenvolvida em Python, e integrando o módulo ESP8266 via protocolo TCP/IP, garantindo comunicação em tempo de execução. A Figura 7 apresenta os principais componentes analisados para a execução do projeto, como as coordenadas do drone, modo de operação, área delimitada, posição do drone em relação à área definida, status de conexão e controles de voo.

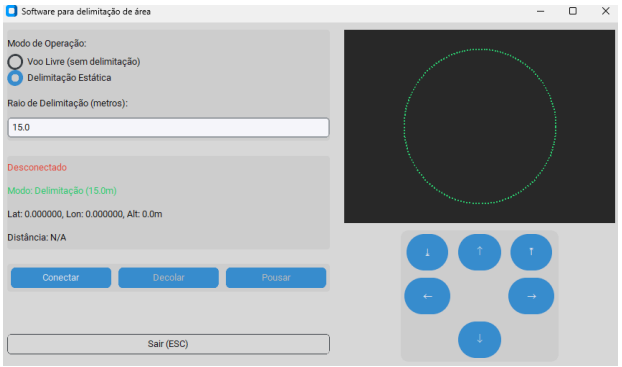

Figura 7. Componentes da interface do sistema de *geofencing* [29].

Foi realizado um teste de voo com o objetivo de avaliar a precisão do sistema de *geofencing*. Para isso, delimitou-se uma área com raio de 15 metros em ambiente aberto, conforme ilustrado na Figura 8. Essa distância foi medida entre o ponto de partida do drone (P1) e o limite máximo permitido (P2). O sistema foi configurado para reconhecer essa área com base na posição inicial do drone. Caso o limite de 15 metros fosse ultrapassado, o sistema deveria emitir um alerta ao usuário indicando a violação do perímetro e, em seguida, acionar automaticamente o comando de pouso. Essa resposta foi previamente programada pelo autor com o objetivo de validar o funcionamento do controle de área.

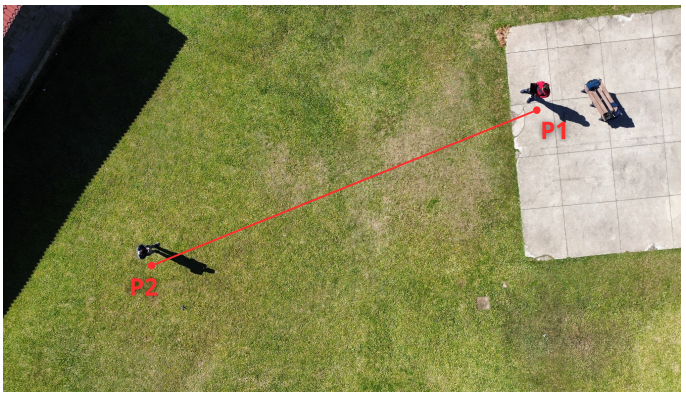

Figura 8. Área de teste da *geofencing* [29].

Com o monitoramento das coordenadas do drone sendo realizado em tempo de execução, a Figura 9 apresenta a notificação emitida quando o drone ultrapassa a área delimitada. Após essa detecção, o sistema aciona automaticamente a funcionalidade de pouso, conforme previamente configurado.

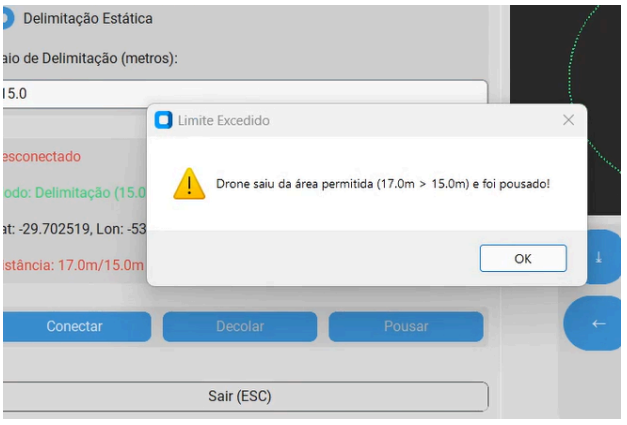

Figura 9. Alerta da *geofencing*[29].

A partir dos testes de voo realizados com o drone, foi possível analisar o desempenho do módulo GPS na delimitação da área. Observou-se um mau desempenho em ambientes muito fechados, o que comprometeu a precisão das coordenadas utilizadas para os cálculos no sistema. O GPS também apresentou variações em relação à sua posição original, impactando diretamente na tomada de decisão do

sistema quanto à definição dos limites da área. A Tabela II apresenta os valores de imprecisão do GPS coletados em diferentes locais, com o objetivo de analisar quais condições são mais adequadas para a aplicação da *geofencing*.

Tabela II
DADOS DE IMPRECISÃO DO GPS

Ambiente	Leituras	Imprecisão GPS (metros)	Desvio padrão (m/s)	Tempo para conectar GPS	Observação
Sala de aula I	30	Não conectado	Não conectado	Não conectado	Janelas fechadas
Sala de aula II	30	34,5	6,2	37,32 segundos	Janelas abertas
Área aberta I	30	9,4	2,7	28,74 segundos	Muitos edifícios
Área aberta II	30	0,8	0,3	4,24 segundos	Poucos edifícios

As métricas, apresentadas na Tabela II, foram utilizadas para avaliação do sistema em cada ambiente, as métricas são: número de leituras do GPS por segundo, valor da imprecisão do GPS em metros, desvio padrão das leituras (variação por segundo), tempo necessário para a conexão com o sinal de GPS e observações qualitativas sobre as condições do ambiente.

No ambiente denominado "Sala de Aula I", a área possuía aproximadamente 10×10 metros e localizava-se no térreo de um prédio com três andares. A quantidade de paredes ao redor, a presença de edifícios vizinhos e as janelas fechadas dificultaram a conexão com o sinal de GPS. Após a abertura da janela em "Sala de Aula II", mesmo local da "Sala de Aula I", foi possível estabelecer conexão e receber coordenadas, embora com elevada imprecisão. No ambiente "Área Aberta I", área de aproximadamente 10×10 metros, apesar de ser um espaço ao ar livre, havia edificações ao redor, o que permitiu a conexão com o GPS, mas resultou em imprecisão e variações nas coordenadas. Já no ambiente "Área Aberta II", área de 15×15 metros, tratava-se de um local a céu aberto com poucos edifícios ao redor, o que possibilitou uma conexão rápida com o GPS, além de apresentar baixa imprecisão e menor variação nos dados coletados.

VI. CONCLUSÃO

Os resultados obtidos ao longo deste projeto atendem as necessidades para a delimitação de área. O sistema de *geofencing* demonstrou ser capaz de delimitar a área de voo do drone, além de executar os comandos de controle de voo. A implementação de uma interface gráfica contribuiu para a organização e o monitoramento das operações, disponibilizando funcionalidades como exibição das coordenadas do drone, modo de operação, área delimitada, posição do drone em relação à zona definida, status de conexão e controles de voo.

No entanto, alguns desafios foram observados, como a variação na precisão do GPS apresentando variações altas em ambientes muito fechados em relação à posição real do drone. Dessa forma, para trabalhos futuros, recomenda-se a utilização de GPS e antenas que possuem maiores níveis de precisão para uma melhor eficiência do sistema e a utilização dos mesmos em áreas abertas.

O projeto foi desenvolvido de forma a facilitar a sua adaptação a diferentes contextos, inclusive trabalhos futuros. Com a aplicação da Programação Orientada a Objetos (POO), o sistema pode ser implementado tanto em drones de diferentes marcas quanto na integração de um módulo GPS, não apenas de forma externa, mas também incorporado ao *hardware* do próprio drone, a fim de facilitar o desenvolvimento de novas soluções alinhadas às regulamentações demonstradas por Alamouri [7].

REFERÊNCIAS

1. J. A. Fenerick e C. R. Volante, “A EVOLUÇÃO DAS INDÚSTRIAS, OS BENEFÍCIOS DA AUTOMAÇÃO E AS PERSPECTIVAS DO MERCADO DA ROBÓTICA NO BRASIL E NO MUNDO”, *INFA*, vol. 17, nº 1, p. 734–745, ago. 2020.
2. P. G. D'Alte, “IMAGOLOGIA ROBÓTICA: robôs e inteligência artificial nas narrativas para crianças e jovens”, *Rev. Obs.*, vol. 5, n.º 5, pp. 586–614, agosto de 2019. Consult. 2024-10-14.
3. F. W. Trevizan e L. N. d. Barros, “Robótica cognitiva: programação baseada em lógica para controle de robôs”, *Sba*, vol. 18, n.º 2, pp. 187–198, junho de 2007. Consult. 2024-10-14.
4. A. Rejeb, A. Abdollahi, K. Rejeb, and H. Treiblmaier, "Drones in agriculture: A review and bibliometric analysis," *Computers and Electronics in Agriculture*, vol. 198, 2022, Art. no. 107017, doi: 10.1016/j.compag.2022.107017.
5. I. Um, S. Park, H. T. Kim, and H. Kim, "Configuring RTK-GPS architecture for system redundancy in multi-drone operations," *IEEE Access*, vol. 8, pp. 76228-76242, 2020, doi: 10.1109/ACCESS.2020. c2989276.
6. G. Zeng, K. Cui, Q. Quan, W. Lin, and Y. Lei, "An airport airspace flow control method for drones," *2019 IEEE International Conference on Unmanned Systems (ICUS)*, Beijing, China, 2019, pp. 178-182, doi: 10.1109/ICUS48101.2019.8996012.
7. A. Alamouri, A. Lampert, and M. Gerke, “Impact of drone regulations on the use of drones in geospatial

applications and research: focus on visual line-of-sight conditions, geofencing, and privacy considerations," *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 91, pp. 381–389, 2023.doi: 10.1007/s41064-023-00246-y.

8. J.-P. Yaacoub *et al.*, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet of Things*, vol. 11, p. 100218, 2020.
9. M. P. Sousa *et al.*, "Avanços e aplicações de drones na gestão de recursos naturais e monitoramento ambiental no semiárido brasileiro," *Revista de Gestão e Secretariado*, vol. 15, no. 7, pp. e4030-e4030, 2024.
10. A. M. Klidzio, M. H. A. Kageyama, S. H. Oliva, and S. O. Silveira, "Uso de drones em logística," in *XI FATECLOG: Os desafios da logística real no universo virtual*, FATEC Jornalista Omair Fagundes de Oliveira, Bragança Paulista, SP, Brasil, 2020, p. 23.
11. B. B. Fernandes *et al.*, "Metodologia FDD aplicada a especificação dos requisitos no processo GRE do nível G de maturidade do modelo MPS. BR," in *ENCOINFO - Congresso de Computação e Tecnologias da Informação*, ENCOINFO, 2011, pp. 199-208.
12. M. S. Dharrao, T. Bhamare, T. Pagar, e V. Sonwane, "Lidar micro-drone with proximity sensing," *Journal Name*, vol. 04, no. 04, pp. 583-586, Apr. 2024.
13. DJI, "Tello – Feel the Fun," Ryze Tech, DJI, 2023. [Online].Available:https://www.ryzerobotics.com/tello
14. D. Pinheiro, J. S. Silva, and M. L. Pereira, "A programmable drone for educational robotics: the case of DJI Tello," *IEEE Latin America Transactions*, vol. 18, no. 12, pp. 2137–2145, 2020.
15. R. Gomes, F. Carvalho, and P. Silva, "Aplicações acadêmicas com drones: uma análise do potencial do DJI Tello," *Revista Brasileira de Ensino de Ciência e Tecnologia*, vol. 13, no. 3, pp. 88–101, 2022.
16. S. Ahmad, A. Ikram, and T. Khan, "A ROS-based control framework for low-cost educational drones," in *2021 International Conference on Robotics and Automation in Industry (ICRAI)*, IEEE, pp. 45–50, 2021.
17. DJI, "Navegação Precisa: a revolução dos drones com GPS integrado!," Blog Loja DJI Brasil, 1 de março de 2024, acesso: 7 de maio de 2025
18. DronEng Drones e Engenharia, "O que são sistemas de referência e de coordenadas?", DronEng, 2024. Disponível:https://blog.droneng.com.br/geodesia-sistemas-de-referencia-e-de-coordenadas/, acessado em 8 de maio de 2025.
19. A. S. Munari, M. Y. H. Setyawan, and M. N. Fauzan, *Panduan Lengkap Algoritma Haversine Formula Pada Sistem Monitoring Mahasiswa Internship Berbasis GPS*. CV. Kreatif Industri Nusantara, 2020.
20. Espressif Systems, ESP8266EX Overview – Wi-Fi SoC, Espressif Systems, 2024. https://docs.arduino.cc/arduino-cloud/guides/esp32/, Acessado: 11 de maio de 2025.
21. DFRobot, *GPS + BDS BeiDou Dual Module (SKU: TEL0132)*, DFRobot Wiki, 2022. [Online]. https://wiki.dfrobot.com/GPS_+_BDS_BeiDou_Dual_Module_SKU_TEL0132, acessado: 11 de maio de 2025.
22. Python Software Foundation. *What is Python?* Executive Summary. Disponível em: https://www.python.org/doc/essays/blurb. Acessado em: 11 de maio de 2025
23. Microsoft, *C/C++ Documentation – Visual C++*, Disponível:https://learn.microsoft.com/pt-br/cpp/?view=msvc-170. Acesso em: 11 de maio de 2025.
24. Douglas Rocha Mendes. Redes de computadores: teoria e prática. Novatec editora, 2020.
25. W. Giernacki, J. Rao, S. Sladic, A. Bondyra, M. Retinger e T. Espinoza-Fraire, "DJI Tello Quadrotor como uma plataforma para pesquisa e educação em robótica móvel e engenharia de controle", *Conferência Internacional de Sistemas de Aeronaves Não Tripuladas (ICUAS) de 2022* , Dubrovnik, Croácia, 2022, pp. 735-744, doi: 10.1109/ICUAS54217.2022.9836168.
26. D. Setiawan, M. W. Sari e R. H. Hardyanto, "Geofencing technology implementation for pet tracker using Arduino based on Android," *Journal of Physics: Conference Series*, vol. 1823, art. 012055, 2021, doi: 10.1088/1742-6596/1823/1/012055.
27. Graça, Pedro Cannavale. Sistema de aquisição de dados utilizando o módulo ESP8266 NodeMCU. 2017. 42 f. Trabalho de conclusão de curso - Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2017.
28. M. Braga, *geofencing-system* [Repositório GitHub]. Disponível:https://github.com/matheuskbraga/geofencing-system. Acesso em: 18 de junho de 2025.
29. I. Sommerville. *Engenharia de software*. Pearson, 2018.
30. Espressif Systems, *ESP32 Datasheet*. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 18 jun. 2025.