UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO

Diego de Souza Calegário

Desenvolvimento de um drone operado remotamente via *Wi-Fi*

Vitória-ES

Dezembro/2018

Diego de Souza Calegário

Desenvolvimento de um drone operado remotamente via *Wi-Fi*

Parte manuscrita do Projeto de Graduação do aluno Diego de Souza Calegário, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Vitória-ES

Dezembro/2018

Diego de Souza Calegário

Desenvolvimento de um drone operado remotamente via *Wi-Fi*

Parte manuscrita do Projeto de Graduação do aluno Diego de Souza Calegário, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovado em 10 de Dezembro de 2018.

COMISSÃO EXAMINADORA:

Prof. MSc. Leonardo de Assis Silva
Instituto Federal do Espírito Santo
Orientador

Profª. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo
Coorientadora

Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo
Examinador

Eng. Ricardo da Mota Salvador
Examinador

Vitória-ES

Dezembro/2018

RESUMO

O desenvolvimento das tecnologias envolvendo veículos aéreos não tripulados, VANTs, vem crescendo muito nos últimos anos. Isso permitiu que pesquisas e aprimoramentos de suas finalidades pudessem ser exploradas. Ligado ao advento dos VANTs, popularmente chamados de drones, muitos países estão elaborando leis para regularizar o uso de tal tecnologia a fim de manter a segurança da população, pois a má operação de um drone pode resultar em acidentes com danos ao próprio equipamento ou a terceiros. Para que haja um uso mais eficiente dos drones na sociedade, novas técnicas de operação e manuseio são desenvolvidas a todo tempo. Por isso, cada vez mais componentes designados para o uso em drones estão sendo desenvolvidos, viabilizando um crescimento quase que constante para pesquisas na área.

O presente projeto consiste no desenvolvimento de um modelo de drone operado por *Wi-fi* em um ambiente interno. Essa aeronave permitirá trabalhos futuros no âmbito educacional e em pesquisas no ramo. Técnicas de engenharia são utilizadas para que decisões sejam tomadas, buscando uma maior eficiência do protótipo. Em todo desenvolvimento, na aquisição dos eletrônicos, será dada preferência a componentes que o autor já tiver disponível. Assim, para a implementação, foram utilizadas mais de uma plataforma de programação, sendo necessário trabalhar com linguagens diferentes em finalidades diferentes. Um programa também foi desenvolvido, utilizando a plataforma *Unity*, para que o usuário pudesse operar o drone em um local próximo, bastando que haja uma conexão *Wi-Fi*. Foram utilizadas métricas de segurança para evitar acidentes ou danos à aeronave elaborada. Os resultados obtidos permitem verificar que o drone apresentou um funcionamento adequado e o programa implementado pôde ser validado, apresentando uma interface amigável, com poucos comandos, sendo eles autoexplicativos. Além disso, é possível se obter imagens a partir do ponto de vista da aeronave por meio da câmera embarcada na sua estrutura.

Palavras-chave: Desenvolvimento de drones; Unity; Wi-fi; quadrimotores.

LISTA DE FIGURAS

Figura 1 – Levantamento de dados de *drones* cadastrados pela ANAC 11
Figura 2 – Diferença entre quadrimotores e helicópteros 15
Figura 3 – Movimento ascendente ou descendente. 16
Figura 4 – Forças atuando como resposta ao movimento de cabrar e picar. 17
Figura 5 – Forças atuando como resposta ao movimento de rolagem. 18
Figura 6 – Funcionamento da guinada. 19
Figura 7 – Modelo Zmr250 21
Figura 8 – Modelo Lumenier QAV250 21
Figura 9 – Eficiência de cada tipo de bateria 23
Figura 10 – CleanFlight SPRacing F3 25
Figura 11 – FLIP 32 - F4 - OMNIBUS REV.2 25
Figura 12 – Matek F4 F405 - AIO 26
Figura 13 – Conjunto de hélices 28
Figura 14 – Velocidades dos diferentes procolos ESC 29
Figura 15 – Modelo de ESC Bullet. 30
Figura 16 – ESC de 40A com UBEC de 3A. 31
Figura 17 – Câmera FPV do tipo *CloverLeaf*. 32
Figura 18 – Motor EMAX RS2205 33
Figura 19 – Eficiência do motor por velocidade 35
Figura 20 – Tensão da bateria no tempo durante o ensaio 36
Figura 21 – Corrente da bateria ao decorrer do ensaio 37
Figura 22 – Hélice 5045 38
Figura 23 – Esc Bullet da marca BEEROTOR 38
Figura 24 – Câmera FPV esolhida da marca Boldclash 39
Figura 25 – Diagrama do funcionamento do projeto 40
Figura 26 – NodeMCU ESP8266 41
Figura 27 – NodeMCU ESP32 41
Figura 28 – Modulação PPM 43
Figura 29 – Display OLED utilizado 44
Figura 30 – Sensor HC-SR04 45
Figura 31 – Sensor GP2Y0A02YK0F 46
Figura 32 – Curva de resposta do sensor GP2Y0A02YK0F 46
Figura 33 – Tela inicial do aplicativo criado 48
Figura 34 – Tela de controles 48
Figura 35 – Tela onde o *drone* é controlado 49
Figura 36 – Módulo receptor da câmera FPV 50

Figura 37 – Fluxograma de inicialização do sistema embarcado 53
Figura 38 – Fluxograma da rotina de Loop do sistema embarcado 54
Figura 39 – Controlador proposto para controle da altura 55
Figura 40 – Porcentagem de ruído no sinal variando janela de filtragem 56
Figura 41 – Rotina de inicialização no programa em Unity 57
Figura 42 – Rotina de armação da aeronave no programa em Unity 58
Figura 43 – Rotina de voo da aeronave no programa em Unity 59
Figura 44 – Rotina de pouso da aeronave no programa em Unity 60
Figura 45 – Protetor de teste 61
Figura 46 – Primeiro protótipo funcional 62
Figura 47 – Protetor final 63
Figura 48 – Suporte da câmera 64
Figura 49 – Projeto de placa confeccionada, superfície superior e inferior 65
Figura 50 – Primeiro modelo de trem de pouso criado 66
Figura 51 – Modelo de trem de pouso final 67
Figura 52 – Tela de controle do programa com a câmera em pleno funcionamento . 68
Figura 53 – Tempo de amostragem médio por rotinas utilizadas 69
Figura 54 – Eficiência dos motores por tração 71
Figura 55 – Versão final do projeto 72

LISTA DE TABELAS

Tabela 1 – Tabela de características das placas controladoras 26
Tabela 2 – Tabela de dados do motor escolhido 34
Tabela 3 – Custos de todas as peças necessárias para a construção do quadrimotor 70

LISTA DE ABREVIATURAS E SIGLAS

UFES	*Universidade Federal do Espírito Santo*
VANT	*Veículo Aéreo Não Tripulado*
VTOL	*Vertical Take-Off and Landing*
ESC	*Eletronic Speed Controller*
UBEC	*Universal Battery Eliminator Circuit*
BEC	*Battery Eliminator Circuit*
PWM	*Pulse Width Modulation*
GPS	*Global Positioning System*
LiPO	*Lithium Polymer*
MIT	*Massachusetts Institute of Technology*
FPV	*First Person View*
OSD	*On-screen Display*
MSP	*Multiwii Serial Protocol*
UART	*Universal asynchronous receiver/transmitter*
IMU	*Inertial measurement unit*

SUMÁRIO

1 INTRODUÇÃO . . . 10
1.1 Motivação . . . 10
1.2 Estado da Arte . . . 11
1.3 Objetivo . . . 14

2 TEORIA DE VOO APLICADA A QUADRIMOTORES . . . 15
2.1 Princípio de voo . . . 15
2.2 Ascender e descender . . . 16
2.3 Cabrar e picar . . . 16
2.4 Rolar . . . 17
2.5 Guinada . . . 18

3 CONSTRUÇÃO E DESENVOLVIMENTO DO PROTÓTIPO . . . 20
3.1 Componentes do *drone* . . . 20
3.1.1 Framekit . . . 20
3.1.2 Motores . . . 22
3.1.3 Baterias . . . 22
3.1.4 Controlador de voo . . . 24
3.1.5 Hélices . . . 27
3.1.6 Outros componentes . . . 28
3.2 Escolha dos componentes . . . 32

4 PROGRAMAÇÃO DOS SISTEMAS REMOTO E EMBARCADO . . . 40
4.1 Sistema embarcado . . . 41
4.2 Sistema remoto . . . 47
4.3 Testes dos sistemas embarcado e remoto . . . 51
4.4 Programação final . . . 52

5 AJUSTES FINAIS E EXPERIMENTO COMPLETO DA PLATAFORMA . . . 61
5.1 Definição do *layout* Inicial . . . 61
5.2 Construção do *layout* final . . . 62
5.3 Resultados de ensaios e procedimentos de adequação da estrutura . . . 65

6 DISCUSSÃO DOS RESULTADOS E POSSÍVEIS APLICAÇÕES DO PROTÓTIPO . . . 69
6.1 Discussão dos resultados . . . 69

6.2 **Possíveis aplicações** **72**

7 **CONCLUSÕES E TRABALHOS FUTUROS** **74**

REFERÊNCIAS **76**

1 INTRODUÇÃO

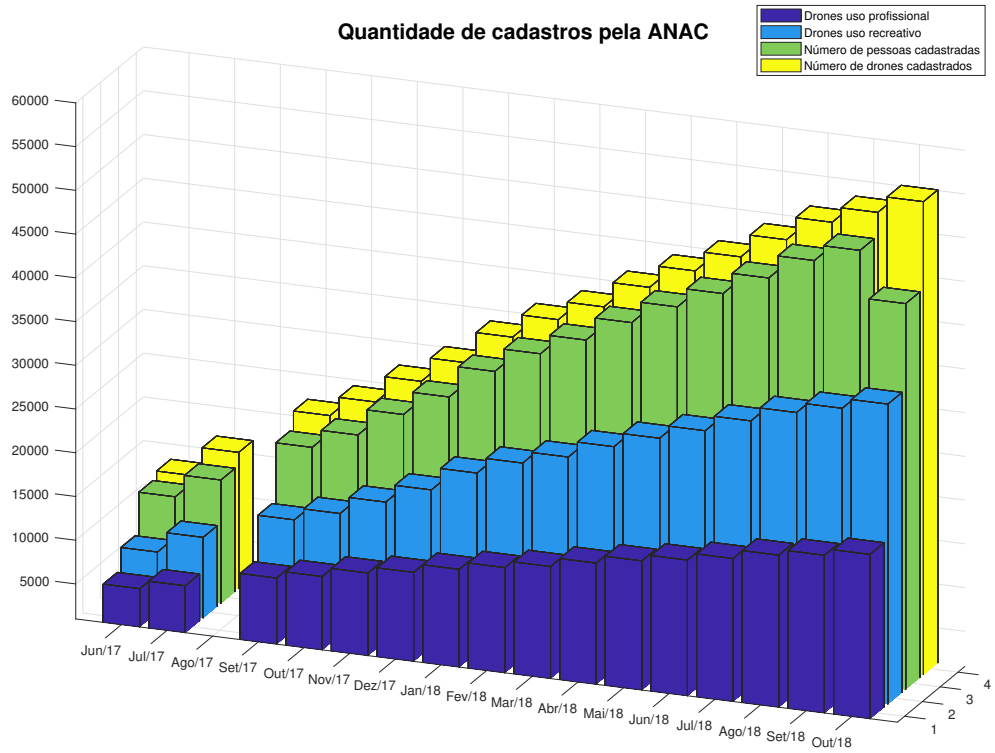
1.1 Motivação

O ser humano sempre teve o desejo de voar, e essa sua obsessão levou à criação de inúmeros modelos de aviões, helicópteros, espaçonaves, dentre outros. Por conta disso, também surgiram cursos de engenharia voltados para a área de criação e desenvolvimento de aeronaves, como a engenharia aeronáutica ou aeroespacial. E, atualmente, essas áreas se aproximaram mais da população em geral com a popularização de modelos operados por controle remoto.

Esses modelos podem ser categorizados de diversas maneiras. Entretanto, eles se concentram, basicamente, em aeronaves de asa fixa (e.g., aviões) e de asa rotativas (e.g., helicópteros). Dentre eles, os modelos de asa móvel mais utilizados hoje em dia são aqueles popularmente conhecidos como "*drones*". Eles fazem parte do grupo dos VANTs (Veículo aéreo não tripulado), em inglês UAVs (*Unmanned Aerial Vehicle*) de asas rotativas, e são um tipo de helicóptero com multiplos motores (em grande maioria quatro motores) para sua propulsão (CINNAMON, 2017). Neste projeto, o modelo adotado para estudo será o VANT do tipo VTOL (*Vertical Take-Off and Landing*) com quatro motores formando o conjunto motopropulsor, também chamado de quadricóptero.

Tanto o desenvolvimento deste projeto, quanto a aquisição de *drones* por grande parte da população se tornaram viáveis pelo avanço da tecnologia, que possibilitou a diminuição e barateamento de microcontroladores e outros componentes eletrônicos. Segundo a ANAC (Agência Nacional de Aviação Civil), em julho de 2017 haviam 12.514 pessoas, seja física ou jurídica, cadastradas como pilotos de *drones* com 13.256 *drones* cadastrados, sendo 7.881 alocados para fim recreativo. Em março de 2018, o número de cadastros subiu para 36.338 pessoas e 38.453 *drones*, sendo 24.644 para fim recreativo. Ou seja, um crescimento de aproximadamente 300% em menos de 1 ano, evidenciando assim o resultado do acelerado desenvolvimento do mercado de *drones* (CIVIL, 2018). Dados mais detalhados podem ser vistos na Figura 1. No mês de Agosto do ano de 2017 não houve levantamento de dados.

Figura 1 – Levantamento de dados de *drones* cadastrados pela ANAC



Fonte: (CIVIL, 2018)

1.2 Estado da Arte

Com o grande advento da tecnologia, adquirir um *drone* tornou-se mais viável economicamente. As principais aplicações dos *drones* hoje em dia, de acordo com (ISSOD, 2015), são:

1. **Imagens e videos aéreos**: São os modelos de *drones* que possuem um controle de estabilidade mais elaborado, para evitar filmagens com grandes ruídos e fotos tremidas. Geralmente, não atingem altas velocidades. Podem possuir um sensoriamento moderno, como GPS e sensores via imagem;

2. **Corridas e manobras acrobáticas**: Esta classe dispensa controladores modernos, uma vez que alta velocidade produz instabilidade. Neste caso, esta instabilidade é controlada pelo piloto. Existem eventos de corridas onde a habilidade do piloto

é colocada a prova. Altas velocidades são facilmente atingidas pela classe, que geralmente é de pequeno porte;

3. **Análises técnicas, montagens para fins acadêmicos**: São *drones* que geralmente são montados por peças que são escolhidas sob necessidade e geralmente utilizados em estudos. Podem possuir programação aberta com uma telemetria avançada;

4. **Comercial e não comercial**: Existem muitas aplicações comerciais para *drones* nos dias de hoje. Alguns exemplos são: inspeções de prédios e telhados, resgate e segurança, e, por final, mapeamento 2D e 3D;

5. **Hobbie**: Podem possuir os mais variados tamanhos, e mais variadas faixas de preço. Geralmente, podem se encaixar em todos os tópicos já mencionados.

No âmbito mais profissional e técnico, os VANTs podem ser usados em patrulhamento aéreo (e.g. utilização para equipes de resgate) (APVRILLE; TANZI; DUGELAY, 2014), inspeção de equipamentos e estruturas (e.g. geradores eólicos, linhas de transmissão e inspessão de oleodutos) (MCAREE; AITKEN; VERES, 2016; MARINHO et al., 2012), monitoramento rural ou urbano (JENSEN; ZELLER; APAN, 2011), e auxílio de equipes em terra ou veículos terrestres onde eles possuem limitações de acesso devido à presença de obstáculos, às características do terreno, ao limite visual em terra, ou ao risco envolvido na aproximação do local de interesse (SÁ et al., 2014).

Devido às diferentes aplicações, o desenvolvimento de um *drone* que possua as características adequadas à missão se torna um fator relevante para uma maior eficiência. A principal vantagem da criação de um *drone* próprio é a flexibilidade de escolhas, tanto para construção, quanto para obtenção de dados. Geralmente, no ramo da robótica, esses dados são: telemetria, tensão das baterias, visão do robô, etc. Além disso, desenvolver o próprio *drone* possibilita certas escolhas como: dimensionamento das baterias (para maior duração de operação) e o aproveitamento de componentes que o próprio desenvolvedor possua (o que facilita a sua manutenção).

Por isso, muitos laboratórios e universidades desenvolvem seus próprios protótipos, como é o caso do Grupo Viros, pertencente ao Laboratório Cisne na Universidade Federal do Espírito Santo. Como comentado, dependendo das tarefas a serem executadas, alguns componentes devem ser escolhidos no lugar de outros. O protocolo de comunicação, por exemplo, é um deles.

A operação de *drones* a partir de comunicação *Wi-fi* tornou-se comum com o grande crescimento de demanda por conectividade entre aparelhos *wireless*. Tal forma de comunicação abrange grandes quantidades de aplicações e modalidades de quaisquer tipo. O *Wi-fi* não se limita a curtas distâncias, podendo ter altas taxas de transmissão de dados com distâncias superiores a 6 *km*, com as devidas adaptações e equipamentos especializados (SANCHEZ-IBORRA, 2016). Entretanto, dependendo da aplicação, até mesmo um módulo de plug USB pode ser empregado, como é o caso do Dongle *Wi-fi*, utilizado por Salvador, Escalfoni e Assis (2017) para enviar comandos para seu *drone*. Inclusive, para aplicações em ambientes internos, o *Wi-fi* acaba sendo um forte canditado para uso, visto que está presente em todo lugar.

A tecnologia GSM também é utilizada, principalmente quando a comunicação é realizada à distância. Porém, seu uso não se limita à troca de informações, podendo ser aplicada em radares na detecção de *drones* que possam representar perigo ou ameaça (KNOEDLER; ZEMMARI; KOCH, 2016). Além dela, o aumento das distâncias de comunicação com VANTs tem estimulado o surgimento de outras inovações. Um exemplo é o LoRa, que aprensenta uma comunicação bem sucedida aplicada a *drones* com distâncias de até 10 *km* (GODOY et al., 2018).

Em relação ao hardware, um dos componentes eletrônicos mais importantes utilizados no *drones*, atualmente, é a placa controladora. Ela exerce a função de piloto automático, trabalhando entre o usuário e toda a estrutura da aeronave. Uma placa muito comum e utilizada no meio acadêmico é a PixHawk (SALVADOR; ESCALFONI; ASSIS, 2017). Esse modelo tem o atrativo de utilizar diversos sensores como giroscópio, acelerômetro, barômetro e magnetômetro, mesmo que de módulos externos, para a estimativa e controle de posição e velocidade.

As placas controladoras vem crescendo em variedade e tornando-se mais acessíveis, possibilitando todos os tipos de escolhas para as mais diferentes demandas, conforme a necessidade do usuário (LIANG, 2017b).

O uso de *drones* em ambientes *indoor*, por exemplo, possibilita que, mesmo com placas controladoras mais simples, sensores para evitar colisões e auxiliar no controle de posição no ambiente sejam usados. Exemplos disso são os sensores ultrassônicos e de infravermelho. Além deles, há modelos mais aprimorados como, por exemplo, os sistemas de radares por FMCWR (*Frequency-modulated continuous-wave*), que realizam a detecção de movimento por varredura (YAN et al., 2017). Câmeras também podem ser utilizadas, vide a quantidade de informações que podem ser extraídas das imagens.

Neste contexto, o presente trabalho consiste na criação de um quadricóptero operado via *Wi-Fi* em um ambiente interno. O protótipo deve apresentar facilidade de adaptação para que possa ser usado futuramente como um modelo de *drone* educacional e útil para pesquisas no ramo. Além disso, ele deverá possuir uma câmera e sensores, que auxiliem na operação do *drone* pelo usuário. O trabalho também consistirá no desenvolvimento de um *software* que permita ao usuário operar o *drone* de uma estação terrestre.

1.3 Objetivo

O objetivo deste projeto consiste na criação de uma aeronave não tripulada, do tipo quadrimotor, que seja operada via *Wi-Fi* em um ambiente interno. Em todo desenvolvimento, na aquisição dos eletrônicos, será dada preferência a componentes que o autor tiver disponível, o que justificará certas escolhas debatidas neste trabalho.

Um dos propósitos a ser considerado é a facilidade de adaptação que o protótipo deverá apresentar para ser usado futuramente como um modelo de *drone* educacional e útil para pesquisas no ramo. Outro ponto a ser observado é que o *drone* deverá possuir uma câmera para permitir que exerça funções de monitoramento e filmagem *indoor* em trabalhos futuros. Além disso, ele deve ter sensores para melhorar a eficiência e segurança de seu funcionamento, evitando colisões, enquanto transmite uma imagem real de todo o ambiente ao redor.

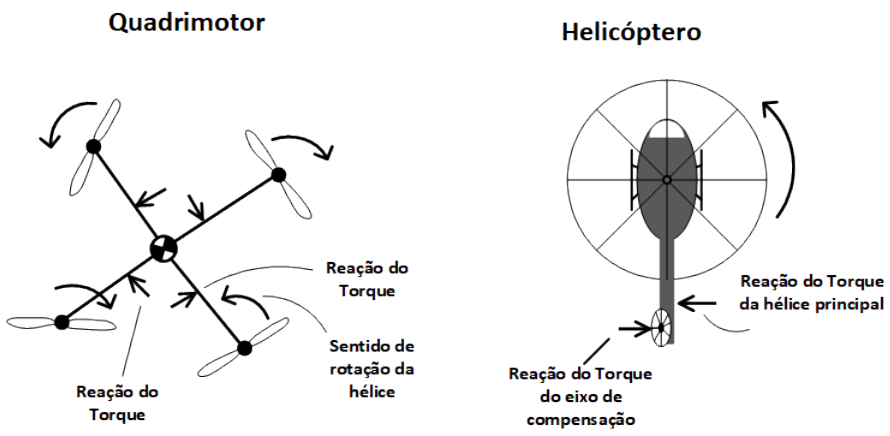
Para alcançar esse objetivo, é necessário que um *software* também seja desenvolvido, para que o usuário possa operar o *drone* de uma estação terrestre. Esse programa deve apresentar as teclas de comando da aeronave, assim como em controles remotos comerciais, além da visão da câmera presente no protótipo.

2 TEORIA DE VOO APLICADA A QUADRIMOTORES

2.1 Princípio de voo

O princípio de funcionamento de quadrimotores é o mesmo dos helicópteros: hélices que propulsionam a aeronave para cima. Entretanto, tais veículos possuem certas diferenças entre si. O quadrimotor possui, como o nome sugere, quatro motores que exercem uma tração vertical. Já os helicópteros possuem apenas um motor com essa finalidade. Essa hélice, ao girar, exerce uma reação na estrutura do helicóptero, forçando-a a girar no sentido contrário, para cancelar o torque gerado. Por isso, o helicóptero possui outro motor na trazeira, que compensa essa reação no plano horizontal (Figura 2). Os quadrimotores não precisam desse motor extra, visto que duas hélices giram em um sentido e as outras duas no sentido contrário, anulando o torque total. Para que o *drone* não fique desbalanceado, as hélices devem girar em sentidos contrários da forma mostrado, na Figura 2: vizinhos não giram no mesmo sentido.

Figura 2 – Diferença entre quadrimotores e helicópteros



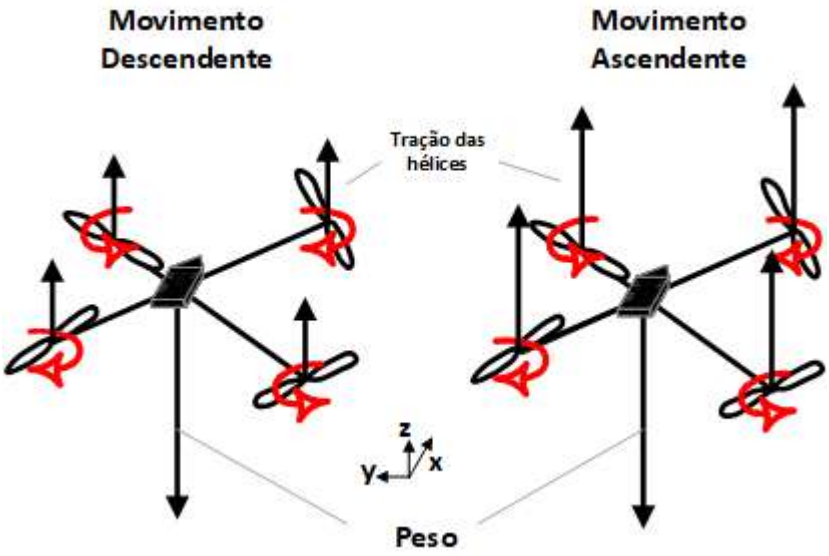
Fonte: Autor

Além disso, o *drone* pode se mover apenas utilizando a ação e reação de seus motores, reduzindo a rotação de um motor específico para a execução da manobra desejada. Essas manobras são comuns para todas as aeronaves, sendo listadas no dercorrer deste capítulo: ascender e descender (*Thrust*), cabrar e picar (*Pitch*), rolar (*Roll*) , e guinar (*Yaw*) (RODRIGUES, 2014).

2.2 Ascender e descender

Para mudar a posição vertical de um *drone*, os quatro motores trabalham juntos uniformemente, para continuar anulando as reações de torque. Quando se aumenta a velocidade de rotação, o *drone* sobe e quando se diminui, ele desce. O peso da estrutura compete contra as quatro trações, como apresentado na Figura 3. Ou seja, caso a velocidade de rotação gere uma tração de mesma magnitude que o peso da aeronave, ela permanece verticalmente parada.

Figura 3 – Movimento ascendente ou descendente.

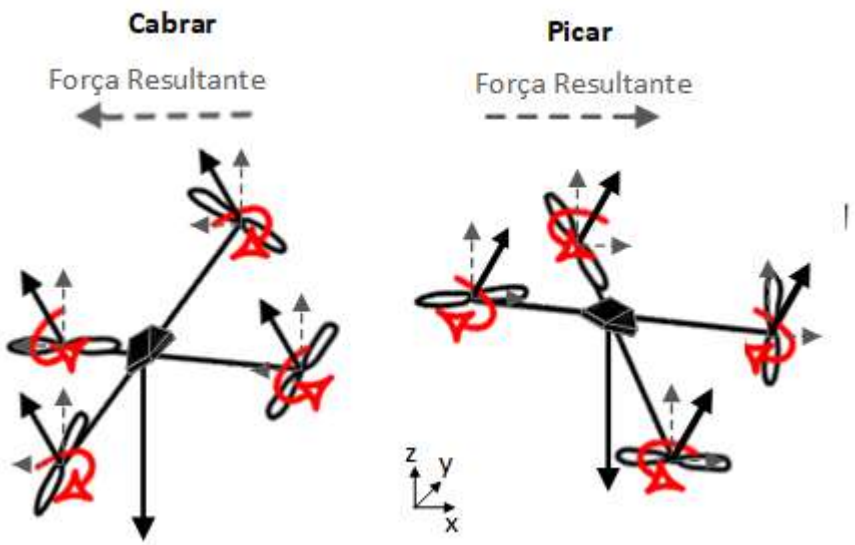


Fonte: Autor

2.3 Cabrar e picar

A ação de cabrar em um *drone* faz com que os motores dianteiros aumentem a rotação, enquanto os trazeiros mantêm a mesma velocidade. Tal ação, partindo de um estado inicial de repouso, faz com que ocorra um aumento na sustentação dos motores de trás, causando um desequilíbrio de forças e submetendo o *drone* a rotacionar até que fique em equilíbrio de forças novamente. Mas o estado final só é alcançado após o *drone* sofrer uma aceleração no plano horizontal e atingir uma velocidade limite. Geralmente, este movimento causa também uma variação na altura, já que a sustentação total também sofre algum tipo de mudança (Figura 4).

Figura 4 – Forças atuando como resposta ao movimento de cabrar e picar.



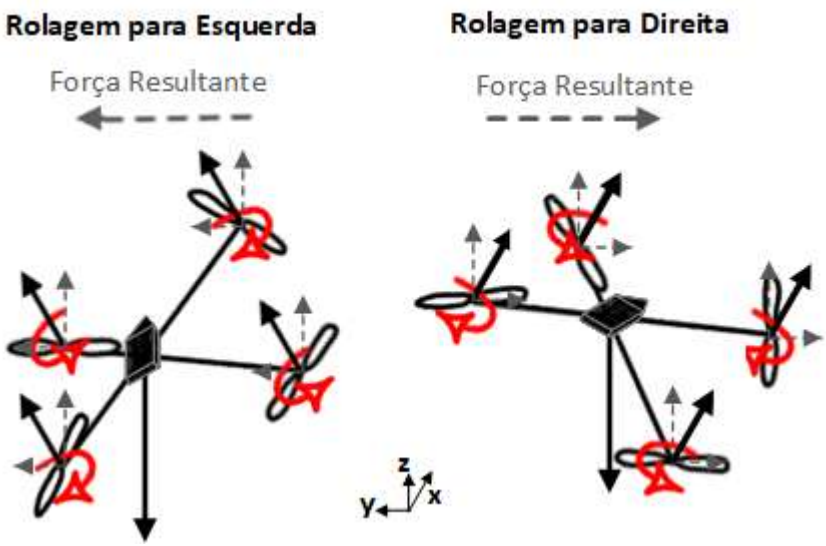
Fonte: Autor

A ação de picar, como mostrado na Figura 4, movimenta o *drone* para frente. Essas duas manobras são comumente chamadas de arfagem: ir para frente ou para trás.

2.4 Rolar

Rolar é muito semelhante a ação de cabrar e picar. Para que o quadrimotor role, basta que os motores de um dos lados aumentem a rotação, enquanto os outros dois mantenham a velocidade. Essa manobra está representada na Figura 5.

Figura 5 – Forças atuando como resposta ao movimento de rolagem.



Fonte: Autor

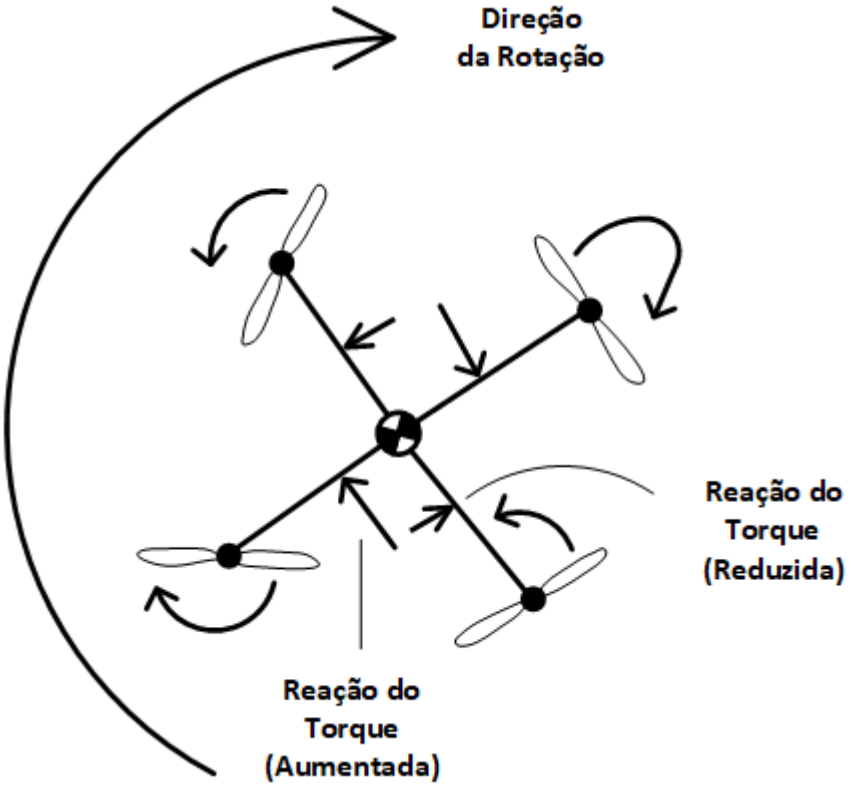
Com o movimento de arfagem e rolagem, o *drone* possui um controle completo nas duas direções no eixo horizontal.

2.5 Guinada

Como comentado na Seção 2.1, quadrimotores e helicópteros utilizam certos conceitos para anular a reação do torque das hélices: motor extra (helicópteros), e hélices que giram em sentidos opostos (quadrimotores). Caso contrário, ambos girariam em torno do próprio eixo. Entretanto, em certas ocasiões, é desejável que a aeronave realize esse tipo de rotação.

Por isso, quando se deseja que um *drone* gire em torno de si dois dos motores, os que giram no mesmo sentido, aumentam a velocidade de rotação, e os outros dois a diminuem, de forma a ainda manter a altitude do VANT. Esse processo está representado na Figura 6.

Figura 6 – Funcionamento da guinada.



Fonte: Autor

Este efeito ocorre apenas em *drones* que possuam um número par de motores, maior que dois. Devido ao fato de os motores vizinhos estarem sempre girando em sentidos opostos, os movimentos de cabrar, picar e rolar não promovem uma rotação da aeronave em torno do próprio eixo. Vale reafirmar que isso só ocorre pelo torque resultante estar sempre anulado.

3 CONSTRUÇÃO E DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo serão apresentados os principais componentes presentes nos *drones*, responsáveis pelo seu adequado funcionamento, e serão descritos os passos utilizados para o desenvolvimento do protótipo. Observa-se que o modelo de quadrimotor aqui desenvolvido se encaixa na categoria *Análises técnicas, montagens para fins acadêmicos*, descrita na Seção 1.2.

3.1 Componentes do *drone*

Os principais componentes que constituem um *drone* são:

- Framekit;
- Motores;
- Bateria;
- Controlador de voo;
- Hélices.

Esses componentes são responsáveis pelo funcionamento adequado da aeronave e serão selecionados conforme as necessidades do projeto, apontadas no objetivo deste trabalho. Cada um deles será melhor discutido nas seções que seguem.

3.1.1 Framekit

O *Framekit* constitui a parte física onde serão acoplados todos os outros componentes, isto é, a parte estrutual do *drone*. Eles variam de tamanho e formato de acordo com o que será embarcado. Portanto, a escolha do mais apropriado depende das outras escolhas do projeto, como os motores, placa controladora, etc. Os dois modelos de *framekit* disponibilizados pelo autor são:

- Zmr250 (Figura 7);

Figura 7 – Modelo Zmr250

Fonte: Autor

- QAV250 (Figura 8);

Figura 8 – Modelo Lumenier QAV250

Fonte: (LUMENIER, 2018)

Ambos os *framekit*s são feitos de fibra de carbono, por isso são muito resistentes a impactos e quedas, além de apresentarem baixo peso. Estes modelos disponíveis são utilizados

em *drones* de corridas ou em VANTs voltados para *hobbie*, modalidades que exigem tais características. Na Seção 3.2 será explicado os motivos para a escolha do *framekit*. Entretanto, é importante informar que ambas estruturas comportam todos os componentes do projeto. Os pontos importantes da escolha estariam nas diferenças de massa e espaço interno.

3.1.2 Motores

A escolha dos motores é um dos fatores determinates do projeto. Caso realizada de maneira inadequada, promove um mau rendimento do *drone*. Um motor superdimensionado requer ESCs maiores e mais potentes, além de uma bateria maior (com mais carga). Dessa maneira, o *drone* será mais pesado, exigindo novamente mais potência do motor. Isso também aumenta o consumo de energia, o que reduz o tempo de voo e a vida útil das baterias.

Para a construção de um *drone* com massa superior à 200 gramas, é preferível a utilização de motores *brushless*. Este tipo de motor não apresenta desgaste de escovas (por não terem), possui elevado rendimento, maior potência e menor peso, quando comparado a outros tipos de motores, como os motores de corrente contínua com escovas (BAICHTAL, 2015).

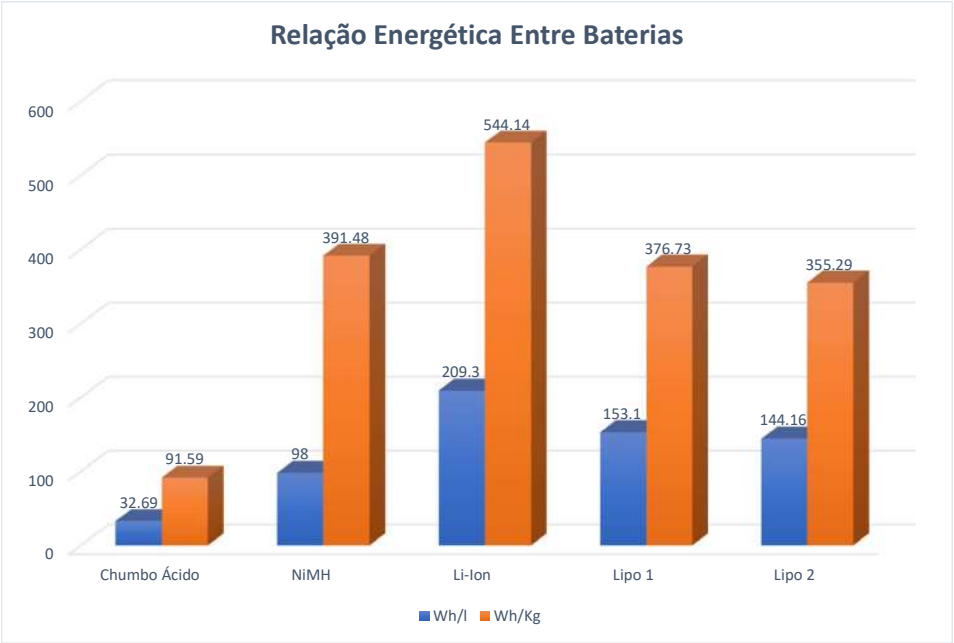
Muitos fabricantes disponibilizam uma tabela, como por exemplo a Tabela 2, com informações de quanta tração (peso) o conjunto motor-hélice consegue entregar à aeronave, além da corrente/potência necessária para isso. Dessa maneira, é possível determinar se um motor atende ou não à demanda de peso. Por fim, o motor escolhido define o ESC, a bateria e as hélices a serem utilizados.

3.1.3 Baterias

Para escolher a melhor bateria, não basta apenas conhecer a potência/corrente máxima drenada pelos motores. É interessante analisar as relações energéticas entre cada tipo de bateria disponível no mercado. Em geral, quanto maior a bateria, mais energia e maior a massa. Porém, o aumento da massa da bateria faz com que a peso total do *drone* seja maior, exigindo dos motores uma maior tração, resultando em um maior consumo. Por isso, baterias de menor porte podem sim ter mais eficiência, se levado em consideração o tempo de voo.

Portanto, foram analisadas quatro tipos de baterias encontradas no mercado: Chumbo Ácido, NiMH, Li-Íon, e LiPo. Realizou-se medições de volume e massa, além de ensaios com descarga a corrente constante, medindo-se a tensão para se obter a potência instantânea. Os dados desse teste estão disponíveis no gráfico de barras da Figura 9.

Figura 9 – Eficiência de cada tipo de bateria



Fonte: Autor

Como se pode notar no gráfico da Figura 9, o tipo de bateria mais eficiênte é a Li-Íon. Porém, de acordo com o manual destas baterias, a taxa máxima de descarga fica em torno de 5C. Isso significa que, por exemplo, para baterias de *2200mAh*, a corrente máxima que pode ser drenada é de *11A*.

Entretanto, no site do fabricante dos motores escolhidos (EMAX, 2018), está informando que cada motor pode consumir até *20A* em *12V*, ou seja, o consumo total de 4 motores pode ser de até *80A* (Tabela 2, na Seção 3.2), valor este demasiadamente superior ao máximo da bateria do tipo Li-Íon. Portanto, para os casos onde apenas um motor principal é utilizado, como em automodelos e aeroplanos, este tipo de bateria possui capacidade para atender os requisitos, mas, no caso de um quadrimotor, a bateria escolhida deve ser do tipo LiPo.

As baterias LiPo possuem taxa de descarga de até 100C, sendo o tipo de bateria mais comum nesse meio. Outras baterias que, pela Figura 9, se compararíam com as do tipo LiPo são as de NiMH. Entretanto, após um certo número de ciclos, podem perder a capacidade de armazenamento (LINDEN, 1995).

Já as de Chumbo Ácido podem fornecer uma corrente alta, mas apresentam um peso muito elevado se comparadas às outras baterias. Geralmente, são utilizadas em situações em que são necessárias altas correntes e que o peso não é um fator importante.

3.1.4 Controlador de voo

Os controladores de voo são essenciais para qualquer robô ou veículo capaz de voar, uma vez que ao se tratar de voo existem muitos fatores externos e internos que influenciam direta e indiretamente na estabilidade. Diferente de veículos terrestres, onde não existe movimentação no eixo Z, isto é, variação de altura, quadrimotores possuem a necessidade de manter um controle de malha fechada sobre seus quatro motores, para que assim ele possa sempre evitar mover-se sem a real necessidade.

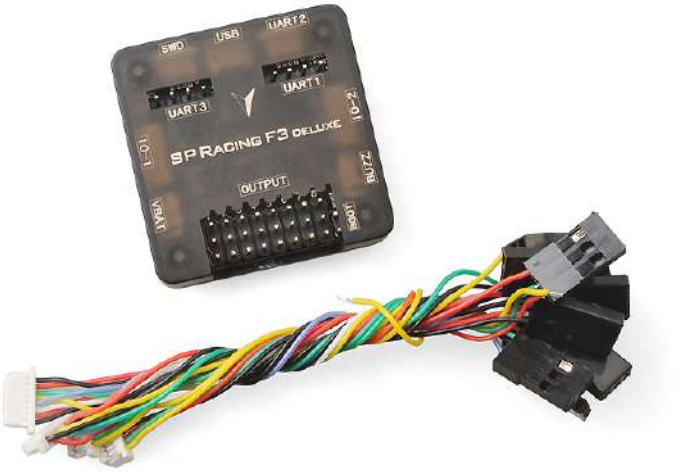
Um controlador de voo possui minimamente um giroscópio e um acelerômetro para que o controle possa ser realizado (BAICHTAL, 2015). O giroscópio permite ao *drone* estimar a sua orientação, enquanto o acelerômetro, permite estimar a movimentação em qualquer direção.

Existem modelos mais elaborados que possuem barômetro, magnetômetro e até GPS. O barômetro auxilia na estimativa de altitude, o GPS na posição global, e o magnetômetro no ângulo de guinada (orientação). Todos esses três componentes podem ter seus dados fundidos aos do giroscópio e do acelerômetro para permitir um melhor controle da aeronave.

Neste contexto, os modelos disponíveis para o autor são:

- CleanFlight SPRacing F3 (Figura 10);

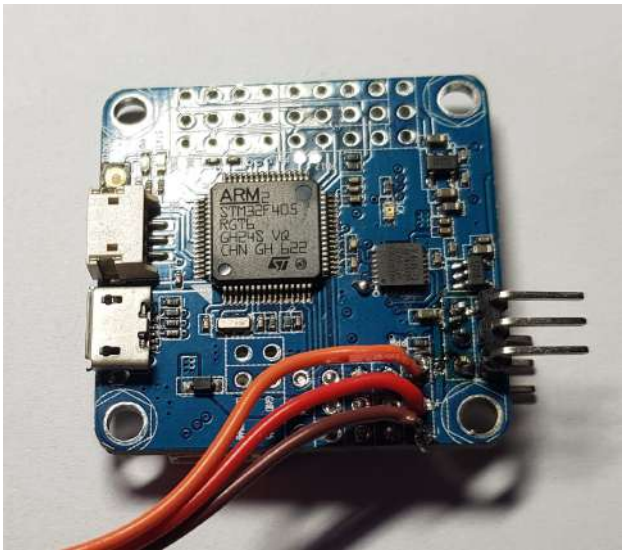
Figura 10 – CleanFlight SPRacing F3



Fonte: (SERIOUSLYPRO, 2018)

- FLIP 32 - F4 - OMNIBUS REV.2 (Figura 11);

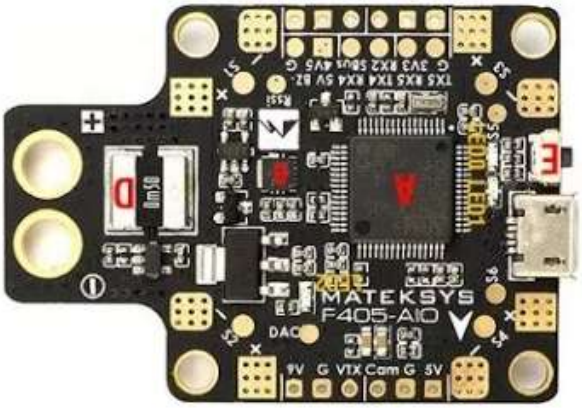
Figura 11 – FLIP 32 - F4 - OMNIBUS REV.2



Fonte: (QUADS, 2018)

- F405 - AIO (Figura 12);

Figura 12 – Matek F4 F405 - AIO



Fonte: (MATEK, 2018)

A Tabela 1 apresenta algumas características de interesse acopladas a cada um desses modelos.

Tabela 1 – Tabela de características das placas controladoras

Características de cada controlador			
	Flip 32	Sp Racing F3	F405
Magnetômetro	N	S	N
Barômetro	N	S	N
Acelerômetro	S	S	S
Giroscópio	S	S	S
OSD embutido	S	N	S
PDB	N	N	S
Memória Flash	16Mb	8Mb	-
UART	2	2	5
Caixa Preta via SD	N	N	S
Sensor de corrente	N	N	S
Sensor de tensão	S	S	S
SoftSerial	S	S	S
GPS embutido	N	N	N

Fonte: Adaptado de (SERIOUSLYPRO, 2018; MATEK, 2018; QUADS, 2018)

Como se pode notar na Tabela 1, todos os modelos apresentados possuem acelerômetro,

giroscópio, e sensor de tensão. Além disso, todos eles podem realizar uma comunicação via *SoftSerial*, permitindo ler informações da aeronave.

Outro ponto a ser observado são as características não comuns entre as placas. Por exemplo, a Sp Racing F3 possui magnetômetro e barômetro, enquanto a F405 é a única que possui sensor de corrente e uma placa de distribuição de força embutida. O controlador de voo deve ser escolhido de acordo com o objetivo do trabalho.

Como o objetivo é proporcionar ao usuário uma ferramenta de operação de *drone* com acesso visual da cena, as placas que possuem OSD (On-Screen Display) são mais interessantes.

As placas F4 F405 e F32 Omnibus inserem telemetria via OSD em suas transmissões de vídeo, o que possibilita ao usuário conferir informações de bateria, linha do horizonte (ângulos de rolagem e arfagem), etc. Ou seja, esse protótipo pode ser utilizado em ensino de pilotagem, ou para exemplos didáticos de controle de voo.

3.1.5 Hélices

As hélices são componentes diretamente atrelados aos motores do *drone*. Cada fabricante de motor estipula um valor de diâmetro e passo para as hélices, cuja capacidade de carga é diretamente ligada ao seu diâmetro e passo. Dessa maneira, a escolha da hélice utilizada será uma consequência do motor selecionado. Isso acarretará em um maior desempenho e menor consumo de energia (BAICHTAL, 2015). A Figura 13 apresenta um conjunto de hélices.

Figura 13 – Conjunto de hélices

Fonte: Autor

3.1.6 Outros componentes

Outro componente muito importante, cuja escolha depende diretamente do motor escolhido, é o ESC (*Eletronic Speed Controller*). O ESC é o responsável pelo controle da velocidade do motor. Ele produz em seus 3 fios ondas quadradas em frequências que se relacionam à velocidade requerida pelo usuário. O sinal de entrada, a referência de velocidade, pode ser enviado por diferentes tipos de comunicação. O mais comum deles é o *PWM* (*Pulse Width Modulation*), mas pode variar entre *Oneshot 125*, *Oneshot 42*, *Multishot* e *Dshot*.

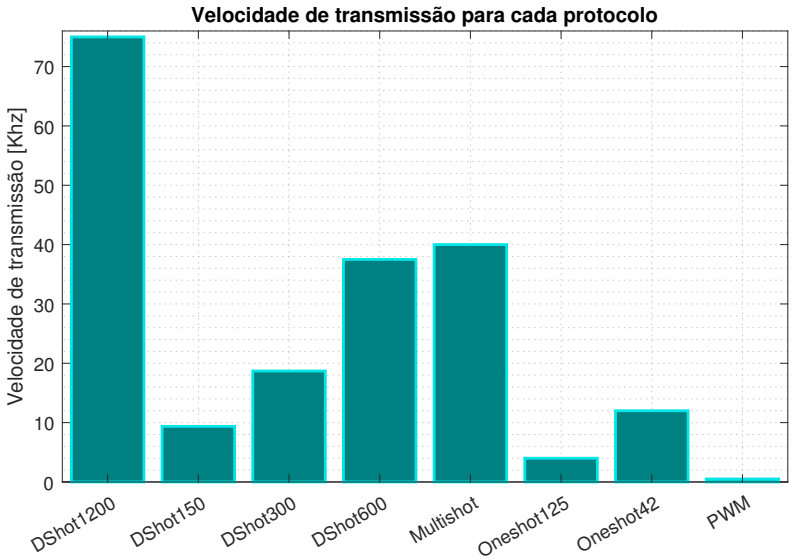
As comunicações que utilizam *DShot* são digitais, e possuem uma série de vantagens sobre as demais comunicações, como: não é necessário calibrar os ESCs, possuem um sinal mais preciso e robusto a ruídos, uma maior resolução de comandos, além de ser mais seguro, por conseguir detectar e rejeitar dados corrompidos. Este tipo de protocolo envia 16 *bits* de comunicação, sendo 11 referentes ao valor do comando, 1 para pedido de telemetria e 4 de *checksum* (verificação de erros) (LIANG, 2017a).

Os outros protocolos, PWM, *Oneshot125*, *Oneshot42* e *Multishot* são sinais analógicos (a

recepção do sinal se faz de forma analógica) e requerem um processo de calibração, devido à possibilidade de o ESC e a placa controladora possuírem *clocks* diferentes. Além disso, são mais suscetíveis a campos magnéticos, por serem sinais analógicos.

A Figura 14 apresenta uma comparação entre as velocidades dos diferentes protocolos de comunicação. Nota-se que a comunicação mais rápida é a digital (*DShot1200*).

Figura 14 – Velocidades dos diferentes procolos ESC



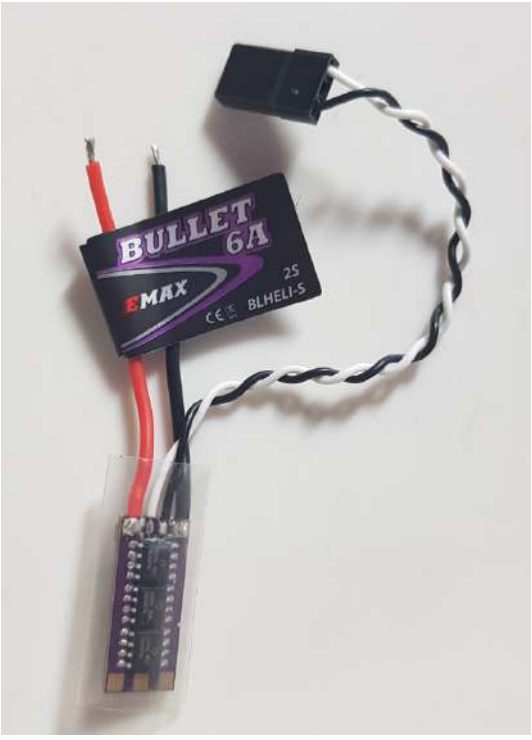
Fonte: Autor

Geralmente, as placas controladoras permitem certas opções de escolha de protocolo de comunicação. Nota-se também na Figura 14, que a comunicação por PWM quase não aparece. Isso se deve ao fato desse protocolo ser mais lento, transmitido em 500 Hz (LIANG, 2017a).

Os ESCs são conectados em paralelo à bateria, e sua principal característica é a corrente nominal fornecida ao motor. Entretanto, também podem se diferenciar entre os que possuem ou não UBEC (ou BEC).

Quando um ESC apresenta, além de seus cabos para sinal de entrada e referência (GND), um cabo que fornece uma alimentação externa de 5V, limitada de 1 a 5A, dizemos que ele possui UBEC (BAICHTAL, 2015). No exemplo mostrado na Figura 15, observa-se que a conexão do controle possui apenas dois fios (branco e preto entrelaçados), sinal de entrada e referência. Neste caso, o *ESC* não possui *UBEC*.

Figura 15 – Modelo de ESC Bullet.



Fonte: Autor

Já na Figura 16, tem-se um *ESC* com *UBEC*: Observa-se os três fios que se conectam ao controle.

Figura 16 – ESC de 40A com UBEC de 3A.

Fonte: Autor

Além do *ESC*, outro componente que está muito presente na maioria do *drones*, é a câmera, seja ela para fins de fotografia ou para se obter a visão do ponto de vista da aeronave. O tipo de câmera mais indicado para o projeto é a câmera utilizada em *drones* com FPV. O exemplo de câmera da Figura 17 apresenta baixo consumo, 25 mW de transmissão, e pouco peso, 4 g. Além disso, possui qualidade adequada, 700TVL, visto que o objetivo não consiste em obter fotos e vídeos de alta qualidade. Um exemplo de câmera FPV acoplada ao decodificador e antena de transmissão pode ser observada na Figura 17.

Figura 17 – Câmera FPV do tipo *CloverLeaf*.

Fonte: Autor

No exemplo da Figura 17, a antena é do tipo *CloverLeaf*, permitindo uma transmissão omnidirecional. O modelo tradicional permite uma distância maior de transmissão, mas exige que o sinal enviado seja direcionado para ela.

Esse tipo de câmera ainda possui um transmissor com 40 canais na faixa de 5,658 GHz a 5,917 GHz, permitindo que se altere a frequência de transmissão caso o sinal esteja sofrendo algum tipo de interferência. Ela é um produto comum, podendo ser comercializada por vários fabricantes.

3.2 Escolha dos componentes

O modelo de *framekit* escolhido para o desenvolvimento do projeto foi o *QAV250* pelo maior espaço interno para alocação de peças como a câmera, controlador de voo e microcontrolador programado. Além disso, possui um peso inferior ao outro, de cerca de 50 g, característica importante em *drones*, como já comentado na Seção 3.1.1. A Figura 8 apresenta o modelo escolhido.

Dentre os motores *brushless* de que o autor dispõe, optou-se pelo modelo *EMAX RS2205 RaceSpec Motor - Cooling Series 2300kv*. A Figura 18 mostra esse modelo.

Figura 18 – Motor EMAX RS2205

Fonte: (EMAX, 2018)

O motor selecionado possui características de tração e corrente que justificam sua escolha, conforme pode ser observado na Tabela 2. Os dados dessa tabela podem ser encontrados no site do fabricante (EMAX, 2018).

Tabela 2 – Tabela de dados do motor escolhido

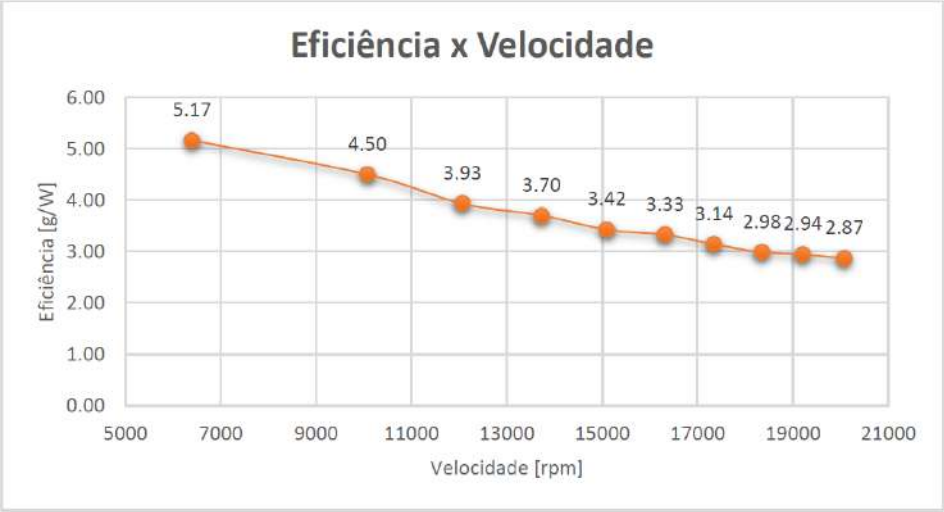
Tabela de dados do motor EMAX RS2205 utilizando hélice 5045						
Kv	Tensão(V)	Corrente (A)	Tração (g)	Potência (W)	Eficiência (g/W)	Velocidade (rpm)
2300	**12**	1	62	12	5.17	6400
		3	162	36	4.50	10080
		5	**236**	**60**	**3.93**	**12070**
		7	311	84	3.70	13730
		9.1	374	109.2	3.42	15100
		11	439	132	3.33	16320
		13	490	156	3.14	17350
		15.3	548	183.6	2.98	18350
		17.3	611	207.6	2.94	19210
		20.7	712	248.4	2.87	20080
	16	1	76	16	4.75	7220
		3	183	48	3.81	10790
		5	283	80	3.54	13030
		7.1	352	113.6	3.10	14720
		9.1	426	145.6	2.93	16180
		11	497	176	2.82	17150
		13	560	208	2.69	18460
		15	628	240	2.62	19270
		17	692	272	2.54	20270
		19	754	304	2.48	21060
		21	812	336	2.42	21840
		23.3	878	372.8	2.36	22590
		25.4	936	406.4	2.30	23210
		27.3	997	436.8	2.28	23920
		29.9	1024	478.4	2.14	24560

Fonte: (EMAX, 2018)

Esse motor permite ser alimentado numa faixa de 11,1 V a 14,8 V, ou seja, de $3S$ a $4S$ (3 a 4 células de baterias LiPo). Além disso, pesando 30 g, é capaz de produzir uma tração máxima de 1024 g (em 16 V).

Como a finalidade do *drone* (operações *indoor*) não são requeridas altas velocidades de deslocamento, espera-se que os motores trabalhem na região de maior eficiência (consumo energético por tração), isto é, baixas rotações e menor consumo, como pode ser visto na Figura 19.

Figura 19 – Eficiência do motor por velocidade



Fonte: Adapatado de (EMAX, 2018)

Com os motores definidos, é possível estabelecer as baterias, os ESCs e as hélices que serão utilizadas.

O procedimento de escolha da bateria se resume a acoplar o quadrimotor já equipado, incluindo a bateria e os eletrônicos, por uma corda ou barbante a uma base. Aplica-se um comando de ascender até que ele possa se manter parado no ar e, a partir disso, monitora-se o tempo que o *drone* fica capacitado a voar (CINNAMON, 2017). Esse teste é uma forma de determinar qual bateria possibilita um maior tempo de voo, sem a dependência de cálculos.

Como o motor disponível necessita que as baterias tenham de 3 a 4 células, os testes foram realizados com todas as baterias LiPo de 3 a 4 células disponíveis. Foram utilizadas as seguintes baterias:

- Zippy - 1000 mAh, 4 células, 35C
- Zippy - 2200 mAh, 3 células, 40C
- Eachine - 1300 mAh, 3 células, 30C
- Tattu - 1300 mAh, 3 células, 75C

Utilizando a placa *F405 AIO*, foi possível fazer a aquisição de todos os dados como: tempo de operação, corrente de cada motor, acelerômetro (X, Y, Z), giroscópio (X, Y, Z), comandos aplicados ao *drone*, nível de bateria, entre outros.

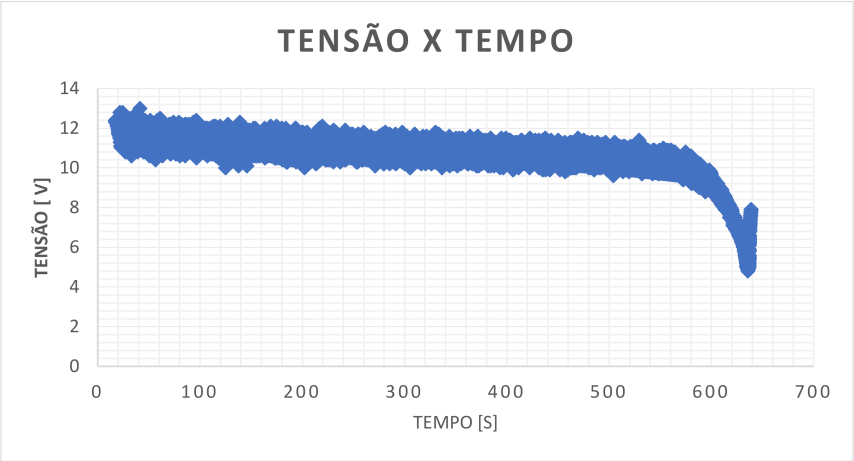
Os resultados de tempo obtidos para cada bateria foram:

- Zippy - 1000 mAh, 4s : *00:04:40*
- Zippy - 2200 mAh, 3s : *00:10:20*
- Eachine - 1300 mAh, 3s : *00:06:04*
- Tattu - 1300 mAh, 3s : *00:07:20*

Pode-se observar que a bateria de maior carga (2200 mAh) apresentou o maior tempo de voo, mesmo sendo, dentre as baterias disponíveis, a mais pesada.

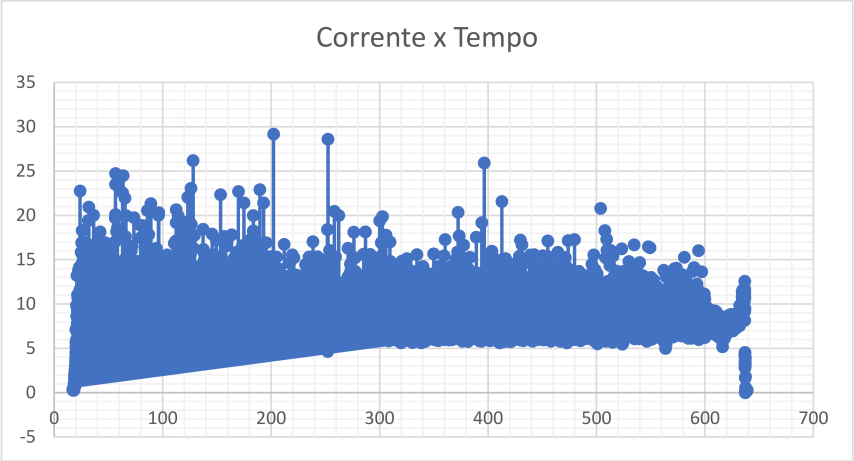
As Figuras 20 e 21 apresentam os dados de tensão e corrente no tempo, respectivamente, da bateria Zippy 2200 mAh.

Figura 20 – Tensão da bateria no tempo durante o ensaio



Fonte: Autor

Figura 21 – Corrente da bateria ao decorrer do ensaio



Fonte: Autor

A Figura 20 apresenta uma curva semelhante à fornecida pelo fabricante e, conforme os valores de corrente mostrados na Figura 21, o valor máximo foi de aproximadamente 30 A, sendo a corrente de máxima capacidade da bateria de 88 A. Isso mostra que a bateria está trabalhando dentro da sua faixa de segurança.

Já a escolha do controlador de voo baseou-se nos objetivos do trabalho, como comentado na Seção 3.1.4, optando-se, no final, pela placa F405. Entretanto, pela placa SP Racing F3 possuir uma IMU mais completa, foram realizados testes de forma a tentar utilizá-la, e obter as informações de telemetria, assim como as dadas pela F405, via *SoftSerial* (protocolo MSP).

Porém, como será visto na Seção 4.3, a obtenção desses dados de telemetria via MSP acaba sendo muito lenta, na faixa de 120 ms.

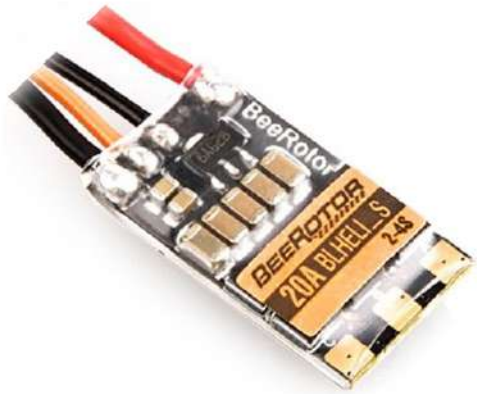
A determinação da hélice, como já comentado, segue as recomendações do fabricante para o motor escolhido. O modelo possui 5 polegadas de diâmetro e passo de 45 °, sendo descrito no mercado como modelo *5045*. A hélice escolhida, via disponibilidade, possui 3 pás, como mostrado na Figura 22.

Figura 22 – Hélice 5045

Fonte: Autor

Assim como a hélice, o ESC também deve seguir o limite de corrente apresentado na Tabela 2. Ou seja, para o uso de uma bateria de $3S$, um ESC com 20 A é suficiente para o projeto. Assim, escolheu-se um ESC do tipo *Bullet*, da marca *BeeRotor* que usa o protocolo *DShot300*, podendo ser atualizado futuramente (Figura 23). Nesse caso, não é necessário *UBEC*, uma vez que existe uma placa distribuidora de energia que se liga à bateria e alimenta todos os ESCs e os periféricos do *drone*, como a placa controladora, microcontrolador embarcado, câmera e servos.

Figura 23 – Esc Bullet da marca BEEROTOR



Fonte: Autor

Por fim, a câmera escolhida, via disponibilidade do autor, foi o modelo FPV da marca *Boldclash*, mostrada da Figura 24.

Figura 24 – Câmera FPV esolhida da marca Boldclash

Fonte: Autor

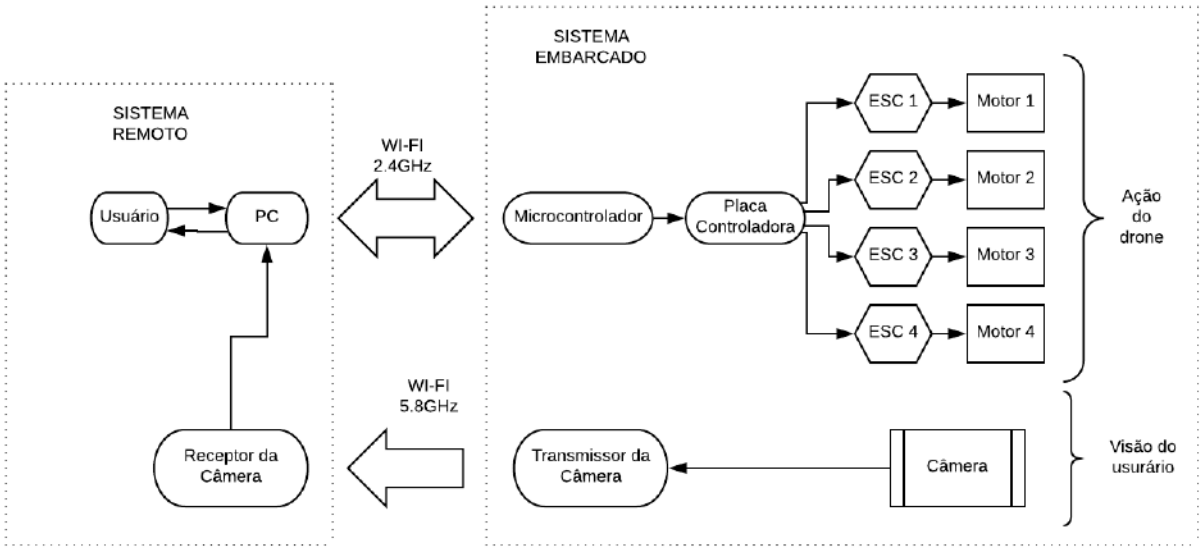
Esta câmera possui diferenças em relação àquela mostrada na Seção 3.1.6, sendo a principal o desacoplamento entre câmera e transmissor. Este desacoplamento permite que dados de telemetria possam ser introduzidos na imagem provinda da câmera antes de serem direcionados ao transmissor. Além disso, o modelo escolhido possui uma antena omnidirecional diferente (parecida com a direcional), que, de acordo com o objetivo do trabalho, não interfere no resultado final.

4 PROGRAMAÇÃO DOS SISTEMAS REMOTO E EMBARCADO

Apesar de utilizar um controlador de voo já programado, o projeto demanda um microcontrolador embarcado na aeronave para que o *drone* possa se comunicar com o usuário. No caso deste trabalho, essa comunicação será realizada por *Wi-Fi*, visto que o protótipo operará em ambiente interno. Além disso, é necessário o desenvolvimento de um programa de computador que funcionará como uma interface com o usuário, permitindo que ele controle o *drone* e receba as imagens da câmera.

O funcionamento do sistema está representado pelo diagrama de blocos na Figura 25.

Figura 25 – Diagrama do funcionamento do projeto



Fonte: Autor

Neste capítulo serão explicitados os desenvolvimentos dos sistemas embarcado e terrestre, além de apresentados os testes dessas plataformas e a estrutura do primeiro *layout* do protótipo. O sistema embarcado consiste no microcontrolador que servirá de conexão entre usuário e *drone*, além de adicionar o sensoreamento do protótipo.

4.1 Sistema embarcado

Para desenvolver o sistema embarcado para a comunicação, deve-se, primeiramente, escolher um microcontrolador. Dentre as possibilidades, buscou-se um microcontrolador que possuísse *Wi-Fi* acoplado a sua placa. As opções foram: NodeMCU ESP8266 (Figura 26) e NodeMCU ESP32 (Figura 27).

Figura 26 – NodeMCU ESP8266

Fonte: (AMAZON, 2018b)

Figura 27 – NodeMCU ESP32

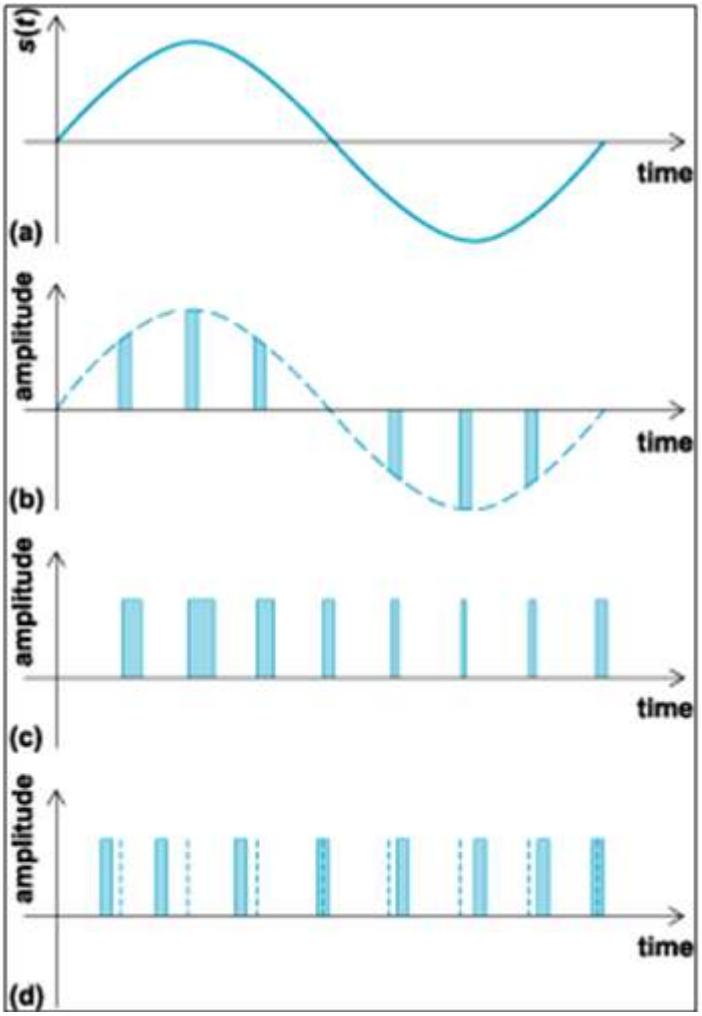
Fonte: (AMAZON, 2018a)

Apesar do microcontrolador ESP32 possuir maior capacidade de processamento que a outra opção, vide *clock* e memória *Flash*, a placa NodeMCU ESP8266 foi a escolhida para o projeto por possuir *hardware* com desempenho suficiente para o projeto e por ser mais acessível de programar para as funcionalidades descritas a seguir (protocolo de comunicação, leitura de sensores e escrita de saídas ao usuário). A comunicação a ser realizada entre o controlador de voo e o microcontrolador embarcado pode ser feita basicamente de 3 formas: PWM, PPM e SBUS.

A primeira, PWM é a forma mais básica de comunicação entre placas controladoras de voo do ramo. O PWM funciona por modulação da largura do pulso, sendo, portanto, um sinal modulado digitalmente e decodificado de forma analógica onde a largura do pulso especifica o comando. A largura do sinal geralmente varia entre $1000\mu s$ e $2000\mu s$. Ele requer 1 pino para cada canal de comunicação, sendo necessários pelo menos 4 canais principais e 4 auxiliares. A opçao PWM foi excluída, por ser ineficiente e necessitar de pelo menos 8 pinos para comunicação, o que resultaria em muitos fios.

A comunicação PPM, *Pulse-position modulation*, também é conhecida como CPPM ou PPMSUM. A vantagem deste tipo de comunicação é que ela requer apenas um fio para seus canais (tipicamente 8 canais). Ela é considerada uma comunicação digital, sendo seu sinal basicamente uma série de pulsos onde sua posição define o valor a ser transmitido. A Figura 28 apresenta essa ideia: um sinal analógico $s(t)$, mostrado em (a), é amostrado (b), e sua amplitude é convertida para PWM (c) e para PPM (d) (PAL, 2014). Essas duas conversões ilustram a diferença entre esses protocolos.

Figura 28 – Modulação PPM



Fonte: (PAL, 2014)

Já o SBUS, ou *Serial Bus*, é um tipo de protocolo de comunicação serial. Essa comunicação suporta até 18 canais utilizando apenas 1 fio. O SBUS utiliza UART para troca de dados. Como a quantidade de canais é superior, a implementação do protocolo SBUS é mais complexo, que a do PPM. Esse protocolo foi desenvolvido e padronizado pelas empresas Frsky e Futaba.

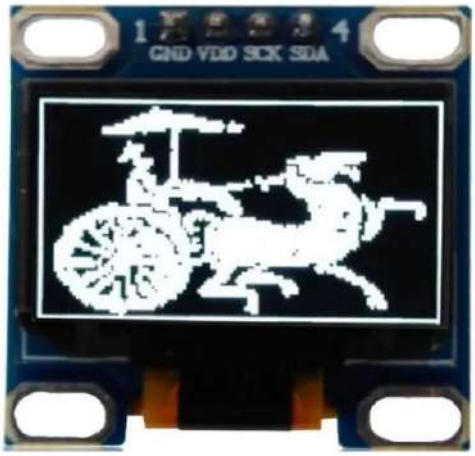
Como os dois métodos utilizam apenas 1 fio, o protocolo de comunicação PPM é mais simples e ambos levariam a um mesmo resultado, PPM foi escolhido para a aplicação.

Após a escolha do protocolo de comunicação entre sistema embarcado e a placa controladora, deve-se criar uma forma de comunicação entre sistema embarcado e o sistema remoto. Como definido no início do projeto, esta comunicação se dará via *Wi-Fi*.

A princípio, antes da realização dos testes, a comunicação foi realizada criando-se um servidor pelo *IP* do microcontrolador. Neste caso, era possível, à escolha do usuário, que o sistema embarcado se conectasse a uma rede *Wi-Fi* existente ou criasse uma própria. A criação de uma rede própria implica em uma comunicação mais rápida e com menor latência. Porém, devido a alguns problemas, dentre eles uma latência de comunicação alta, foi preciso implementar outra maneira de comunicação de dados, utilizando *Socket*. Os motivos serão melhor explicados na Seção 4.3.

Este microcontrolador também permitiu que uma tela OLED fosse adicionada ao protótipo, facilitando, assim, o acesso ao IP do *drone*, o que permite que o usuário se conecte à aeronave facilmente. O pequeno monitor pôde ser utilizado também para avisos de erros ou de falhas. A Figura 29 apresenta o display OLED utilizado neste trabalho.

Figura 29 – Display OLED utilizado



Fonte: Autor

Após a implementação da comunicação entre sistema remoto e placa controladora, o passo seguinte consistiu na implementação do sensoreamento de proteção.

Para evitar colisões, foram escolhidos os sensores ultrassônicos *HC-SR*04 (Figura 30). Três desses sensores foram dispostos no *drone* nas seguintes posições: frente e lados direito e esquerdo. Eles demandam, cada um, 2 pinos digitais de comunicação: *Trigger* e *Eccho*.

Figura 30 – Sensor HC-SR04

Fonte: Autor

Como os sensores $HC\text{-}SR04$ funcionam com a emissão e recepção de uma onda sonora de alta frequência que vaga pelo ar com uma velocidade de aproximadamente $340m/s$, existe um certo atraso a ser considerado, principalmente quando se utilizam 3 sensores ultrassônicos. Por isso, na sua biblioteca (disponibilizada juntamente ao código de todo o projeto), existe a possibilidade de configurar uma distância máxima, para quando o tempo de retorno for maior do que um tempo pré determinado. Assim, optou-se por um atraso de $3ms$ com uma distância máxima de $51cm$. Qualquer distância acima de $51cm$ será considerada como a distância limite, restringindo assim o tempo gasto com essa leitura no loop de controle.

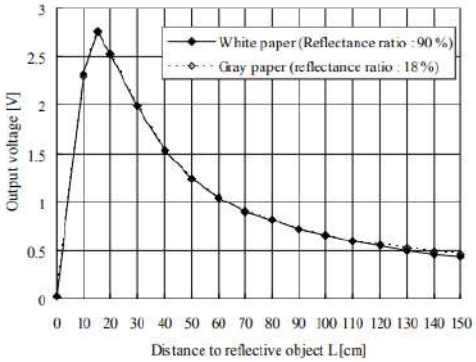
Também foi adicionado um sensor infravermelho $GP2Y0A02YK0F$ da marca Sharp (Figura 31) para auxiliar o *drone* a manter sua altura constante. Esse sensor funciona com uma saída analógica, dessa forma, apenas um pino ADC é suficiente para a transmissão da informação de distância.

Figura 31 – Sensor GP2Y0A02YK0F

Fonte: (SHARP, 2006)

Ele possui duas características muito relevantes: sua curva de tensão não linear em resposta à distância, possuindo uma faixa de operação de $20cm$ a $150cm$, e seu desempenho em relação a superfícies refletoras ou escuras é praticamente o mesmo. A curva de tensão em resposta à distância pode ser vista na Figura 32.

Figura 32 – Curva de resposta do sensor GP2Y0A02YK0F



Fonte: (SHARP, 2006)

É importante destacar que ambos sensores possuem uma medição ruidosa, sendo necessária a implementação de um filtro para remover tais valores de erro. A estratégia inicial de filtragem foi a utilização de uma janela móvel de médias. Esta lógica mostrou-se promissora pois exige um baixo processamento e lógica simples, além de poder reduzir o tempo de resposta, dependendo do tamanho da janela escolhida. De forma empírica, o tamanho da janela escolhido para os sensores ultrassônicos e infravermelho foi de 15 e 30 valores, respectivamente.

Durante a construção e programação, foram consideradas estratégias de se realizar odome-

tria, para que o *drone* pudesse ter um controle de posição, sem que a inércia de movimento ou a irregularidade do ambiente fizessem com que ele viesse a criar uma velocidade, mesmo que baixa, que causasse uma mudança de posição não controlada.

A primeira estratégia se baseava em fazer a comunicação da placa controladora com o restante do sistema embarcado utilizando o protocolo MSP. Esta comunicação utiliza uma UART e pode ser aplicada em muitas plataformas. É uma das formas de comunicação entre a placa controladora e outros dispositivos, via *SoftSerial*. Essa comunicação é utilizada em alguns modelos de rádios de controle para que o usuário tenha acesso a qualquer variável ou sensor disponível na placa. Além de fornecer dados da IMU, fornece comandos de entrada (comandos do usuário), rotações dos motores, posição, caso possua GPS acoplado, tensão, corrente e potência drenada da bateria, dentre outros. Um maior detalhamento dessa primeira abordagem será dado na Seção 4.3.

Posteriormente, uma vez que a primeira tentativa de se realizar a odometria não foi bem sucedida, outra tentativa foi utilizar uma IMU própria, isto é, um conjunto composto de um acelerômetro e um giroscópio foi instalado e utilizado para se calcular as posições do *drone*. Assim como o caso anterior, não foram obtidos resultados satisfatórios, e seu detalhamento será realizado na Seção 4.3.

4.2 Sistema remoto

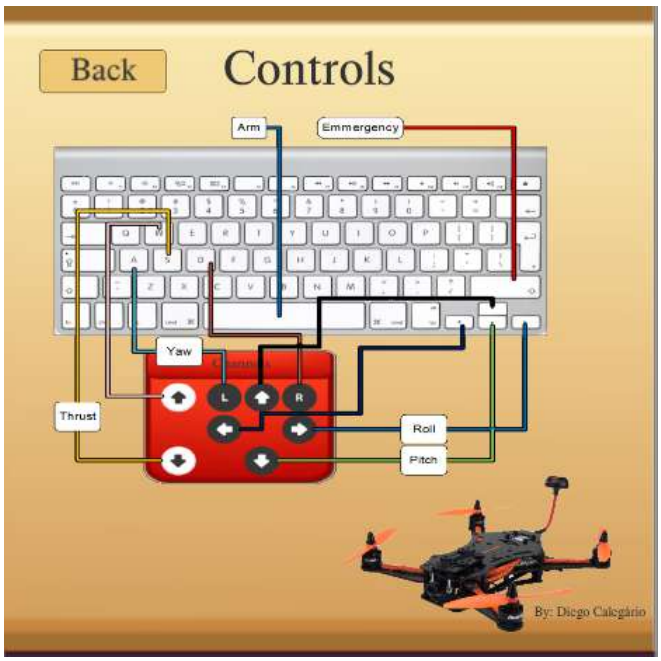
A plataforma escolhida para o desenvolvimento do sistema remoto (base terrestre) foi a *Unity*. Sua escolha se deu por uma série de fatores: ser um programa gratuito, facilidade de utilização, muitas possibilidades de implementação, e possuir uma vasta comunidade disposta a tirar dúvidas e ajudar com *bugs* e erros. Inicialmente, a plataforma *Unity* foi criada para desenvolvimento de jogos, mas não se limita mais apenas a isso (UNITY, 2018).

A programação foi implementada na linguagem *C#*. Criou-se uma interface simples e dinâmica, que pudesse ser controlada pelo *mouse* e pelo teclado. Vale ressaltar que a presença de uma câmera no *drone* é um fator importante, e por isso seu uso já é considerado pelo aplicativo local. As Figuras 33, 34 e 35 apresentam as três janelas presentes no programa. A Figura 34 mostra a janela de ajuda, que informa os controles do *drone* pela interface e pelo teclado.

Figura 33 – Tela inicial do aplicativo criado

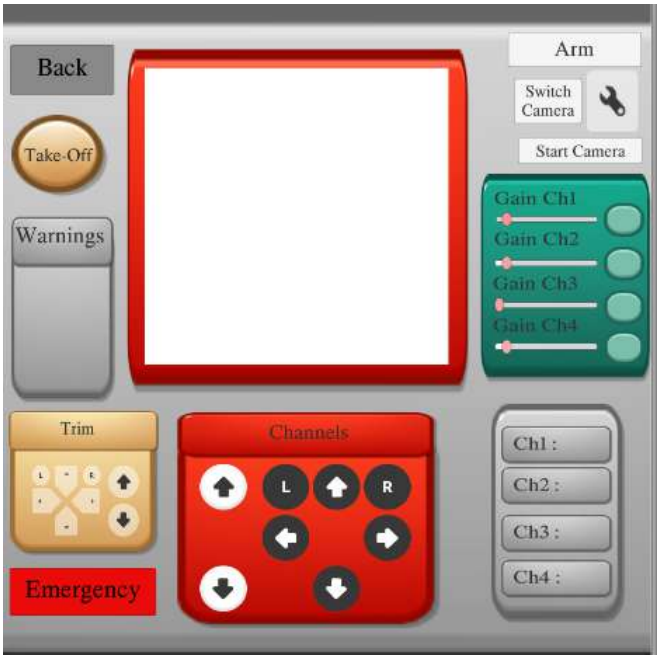
Fonte: Autor

Figura 34 – Tela de controles



Fonte: Autor

Figura 35 – Tela onde o *drone* é controlado



Fonte: Autor

Para iniciar, o usuário deve fornecer o IP do *drone* (Figura 33), presente em sua tela OLED. Em seguida, espera-se a conexão se estabelecer e, então, o programa ficará aguardando os comandos do usuário. Na tela de controle (Figura 35), existe um quadro que permite ao usuário ajustar os ganhos de cada botão quando pressionado, isto é, o controle da intensidade do comando. Além disso, possui comandos que permitem ao usuário iniciar a recepção da imagem da câmera, e a possibilidade de trocar de câmera, visto que muitas vezes, dependendo do computador, a *WebCam* é iniciada por acidente.

A câmera, referente à tela do aplicativo, se encontra acoplada à placa controladora do *drone*, que insere dados OSD como: tensão da bateria, potência consumida, angulação, linha do horizonte e avisos. No sistema local, deverá haver um receptor acoplado via USB (Figura 36), fazendo com que a imagem vinda da câmera seja reconhecida pelo computador como uma *WebCam*. Então, diferentemente dos dados que são enviados ao sistema embarcado no *drone* (que utilizam conexão *Wi-Fi*), a transmissão da imagem da câmera não depende da qualidade da rede ou processamento do sistema embarcado. Isso resulta em um tempo de resposta muito pequeno e praticamente invariante à taxa de comandos enviados ao *drone*. A transmissão é feita na frequência de 5,8 GHz.

Figura 36 – Módulo receptor da câmera FPV

Fonte: Autor

Por fim, também foram inseridas medidas de segurança para a aeronave, como por exemplo:

- Não se desconectar, via pedido do usuário, enquanto o *drone* estiver armado, ou seja, pronto para voar, com os motores energizados;
- Não desarmar de forma regular, isto é, não emergencial, enquanto estiver em voo;
- Não exercer comandos com ganhos elevados sem alertar o usuário;
- Avisos sobre a proximidade dos sensores de ultrassom;
- Não movimentar-se em direção a objetos quando detectados pelo sensor;
- Caso ocorra uma desconexão entre o sistema embarcado e o sistema remoto, o *drone* irá executar uma rotina de pouso de emergência.

4.3 Testes dos sistemas embarcado e remoto

As rotinas implementadas no código do microcontrolador, considerando a primeira abordagem, de comunicação via servidor Web, são executadas segundo a ordem:

1. Leitura dos argumentos (comandos de voo) do servidor Web;
2. Medição das distâncias com o ultrassom;
3. Tratamento com o filtro de janela móvel;
4. Leitura dos dados de odometria via MSP;
5. Verificação da validade dos dados obtidos pelo servidor Web;
6. Transmissão dos comandos via PPM para a placa controladora e atualização dos dados no monitor OLED.

Todos os itens citados são executados em *Loop*, e o período de amostragem mostrou-se na ordem de 300 ms, quando o *drone* não recebia comandos novos. Esse tempo pôde chegar a 600 ms, quando novos comandos eram executados, ocorrendo um gargalo nos dados obtidos via MSP. Foi então percebido que o solicitação de dados via MSP causava um atraso relativamente grande no período de amostragem. Portanto, suas medições de aceleração nos 3 eixos (X, Y e Z) não eram feitas a uma frequência que pudessem ser medidas pequenas variações de posição. Assim, a segunda alternativa foi utilizar um acelerômetro, o sensor MPU 6050. Este sensor é muito utilizado em vários projetos onde acelerômetros são requisitados, por possuir um custo baixo.

Após testes iniciais, percebeu-se que seu período de medição alcançava em torno de 10 ms, um período satisfatório, comparado ao caso anterior. Porém, por ser um sensor barato, suas medições possuíam muito ruído, e, por isso, pequenas variações de posição não eram percebidas pelo sensor, uma vez que o ruído estava com amplitude maior que os dados.

Existem no mercado IMUs mais precisas e mais eficientes, como a $Bno055$ do fabricante Adafruit, que possui um circuito dedicado para filtrar e fundir os dados de forma a disponibilizar a odometria a uma taxa de 100 Hz (SENSORTEC, 2014).

Entretanto, essas IMUs possuem um custo mais elevado. No caso da $Bno055$, ela está custando no mercado brasileiro por volta de R$$260,00$. Por isso, decidiu-se não realizar a odometria do *drone*, uma vez que este possuirá sensores para evitar que ocorram colisões.

Mesmo após remover a medição da odometria via MSP ou leitura do acelerômetro, o período de atualização estava próximo de 150 ms a 200 ms. Assim, foi necessário mudar a abordagem de comunicação entre sistema remoto e sistema embarcado, antes feita via *Web Server*. A nova abordagem utiliza *Socket*.

Nessa nova abordagem, assume-se que o *drone* está utilizando comunicação TCP/IP. Cada nó de conexão (servidores e clientes) possui um endereço de IP. Portas lógicas são usadas para organizar a comunicação. Programas podem rodar em portas específicas. *Sockets* são a combinação lógica do endereço e da porta que é aberta pelos nós (comumente clientes) para se comunicar com outros nós (geralmente servidores). Assim que um *Socket* é aberto, uma comunicação bidirecional ocorre.

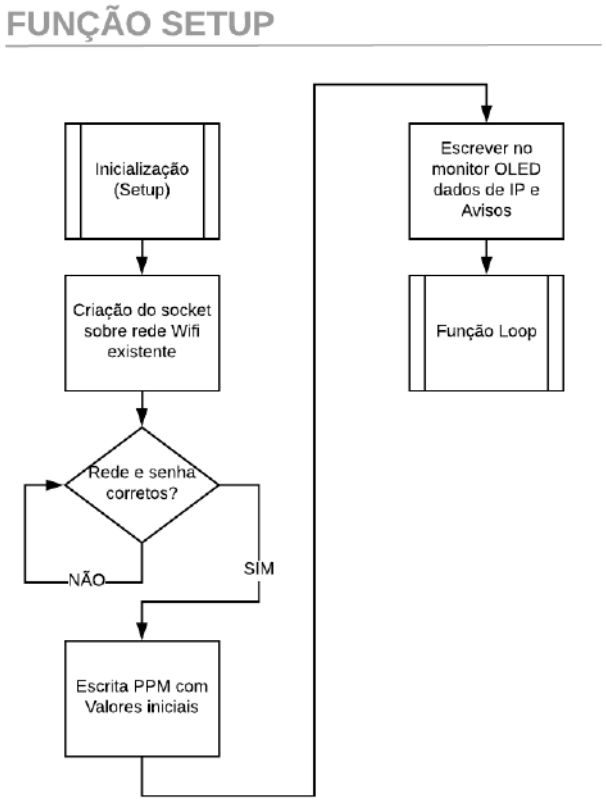
Após as mudanças nos códigos do sistema embarcado e no sistema remoto, o período de amostragem teve uma redução para aproximadamente 60 ms, variando muito pouco, mesmo quando comandos são enviados. Esse tipo de comunicação permite inclusive que menos argumentos sejam enviados. Por exemplo, utilizando-se *Web Server*, o último argumento enviado na comunicação era o *tempo real*, para que o sistema embarcado fizesse a análise e verificação quando esse tempo parasse de mudar. Isso permitia ao sistema embarcado saber que houve uma perda de conexão com o sistema local. Utilizando-se *Socket*, por ser uma comunicação bidirecional, não há a necessidade de enviar argumentos de verificação. Em testes onde o sistema local se desconectava, o sistema embarcado percebia isso de forma rápida e executava o procedimento de segurança para a dada situação.

4.4 Programação final

Devido ao grande número de linhas e rotinas diferentes, os códigos estão disponíveis na plataforma *GitHub* para fácil acesso e melhor replicação dos componentes necessários para executar o trabalho aqui apresentado. O link de acesso é <https://github.com/diegocalegario/Drone-Controller>.

O código utilizado pelo sistema embarcado, em sua rotina de *setup*, está representado por meio de um fluxograma na Figura 37.

Figura 37 – Fluxograma de inicialização do sistema embarcado

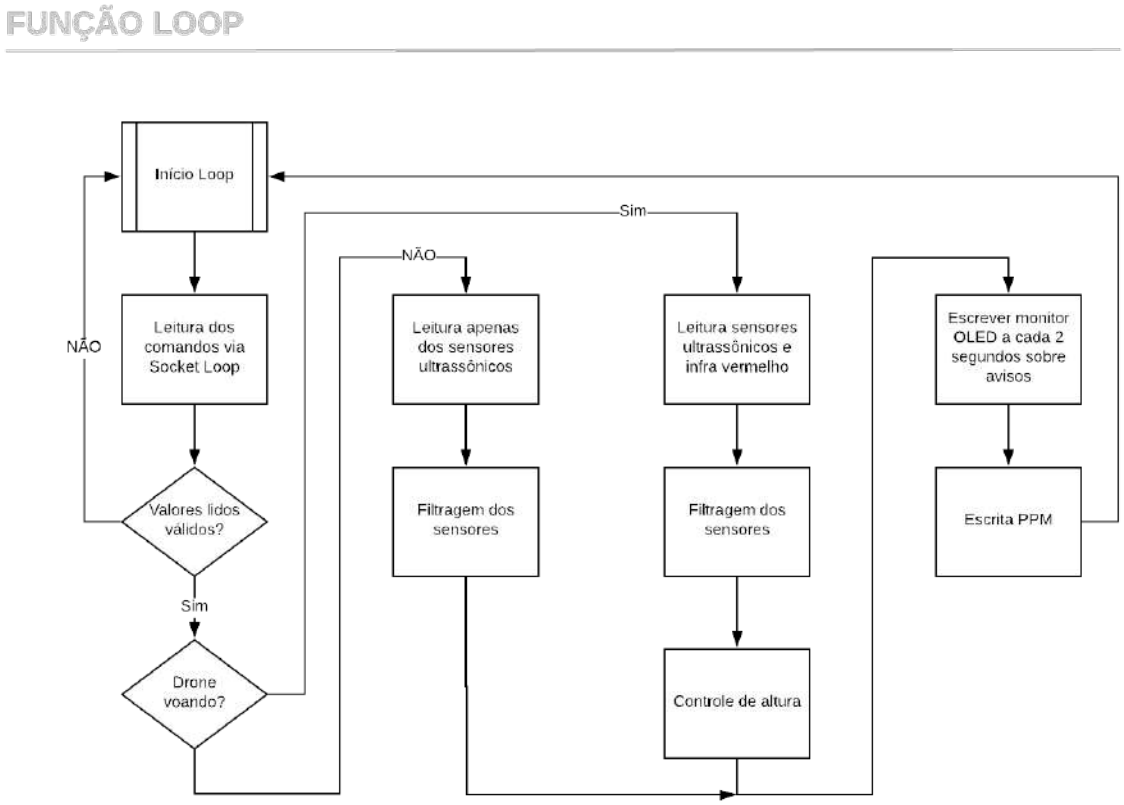


Fonte: Autor

De acordo com o fluxograma da Figura 37, a rotina de inicialização acessa uma rede *Wi-Fi* conhecida, e, a todo tempo, mostra em seu monitor OLED, presente no *drone*, seu *status*. Isto é, ele apresenta se a conexão foi bem sucedida ou não, além de informar o número de IP que deve ser utilizado para a conexão.

Em seguida, executa-se a rotina de *Loop*, apresentada pelo fluxograma da Figura 38.

Figura 38 – Fluxograma da rotina de Loop do sistema embarcado

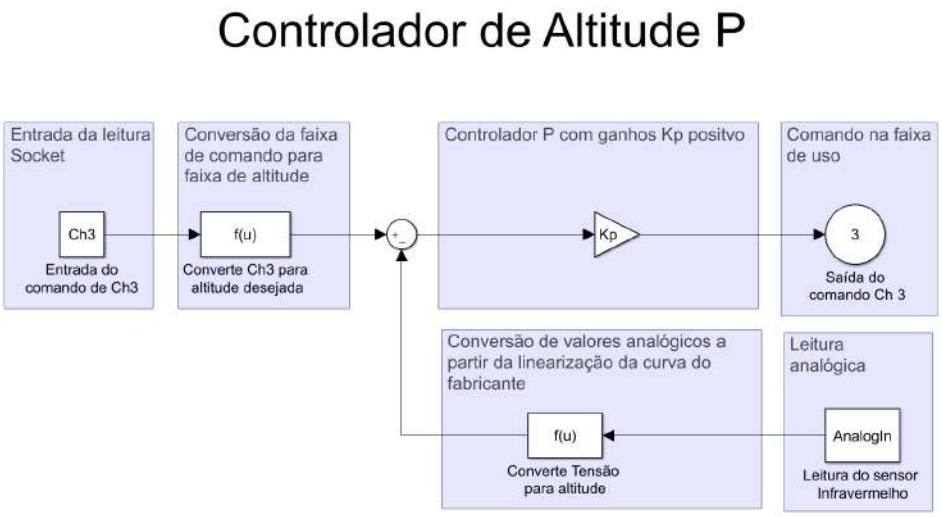


Fonte: Autor

A rotina apresenta um período de amostragem regular, na faixa de 60 ms, e apresenta uma condição de verificação se o *drone* encontra-se voando. Isso se deve ao fato do sensor de infravermelho não reproduzir saídas de tensão fiéis à parte linear da curva de resposta à distância apresentada na Figura 32. A conversão da saída de tensão do sensor foi tratada a partir de uma linearização, da faixa de 20 cm a 150 cm em uma curva polinomial de ordem 3, obtida por meio da ferramenta *Matlab*. Assim, para que a linearização faça sentido, é necessário que o quadrimotor esteja obrigatoriamente voando acima de 20 cm do chão.

A verificação é um dos parâmetros enviados pelo programa, que tem conhecimento prévio de qual é o valor de comando aplicado aos motores para que a aeronave esteja pairando acima do nível mínimo desejado, próximo de 30 cm, obtido a partir de ensaios. Então, o controle de altitude é executado para manter o *drone* numa faixa de estabilidade de altura a partir do controlador, mostrado na Figura 39.

Figura 39 – Controlador proposto para controle da altura

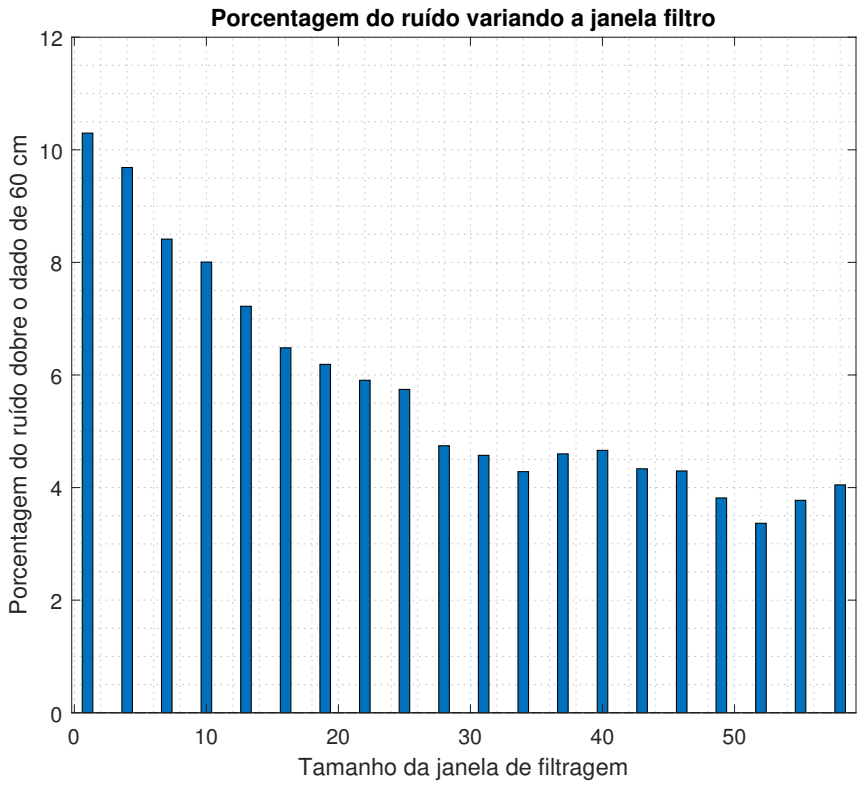


Fonte: Autor

Foram realizados testes para averiguar o melhor tipo de controlador e seus respectivos ganhos. Constatou-se que com a presença da ação de um controle integral ou proporcional, por exemplo PI ou P, o sistema torna-se instável. Como os dados lidos pelo sensor, mesmo utilizando um filtro, possuem uma quantidade de ruído significativa, não foram realizados testes com ação de controle derivativa. Mesmo testando diversos valores de ganho proporcional K_p, as oscilações não pararam de ocorrer.

Acredita-se que a principal causa é o tempo de resposta devido o tipo de filtro implementado no sensor infravermelho. Como a quantidade de ruído é muito grande, a janela acaba precisando aumentar também, ou seja, isso aumenta o tempo de resposta do filtro. Variando o tamanho da janela, o valor a partir 30 foi um valor minimamente satisfatório para se ter um filtro que não transmitisse ruídos acima de 5% do valor medido (Figura 40). Porém, como a janela é preenchida com um valor novo a cada *loop*, cujo período é de 60 ms, o tempo para que 60% dessa janela fosse preenchida com valores novos é de aproximadamente $1s$, valor demasiadamente alto para a dinâmica do *drone*.

Figura 40 – Porcentagem de ruído no sinal variando janela de filtragem



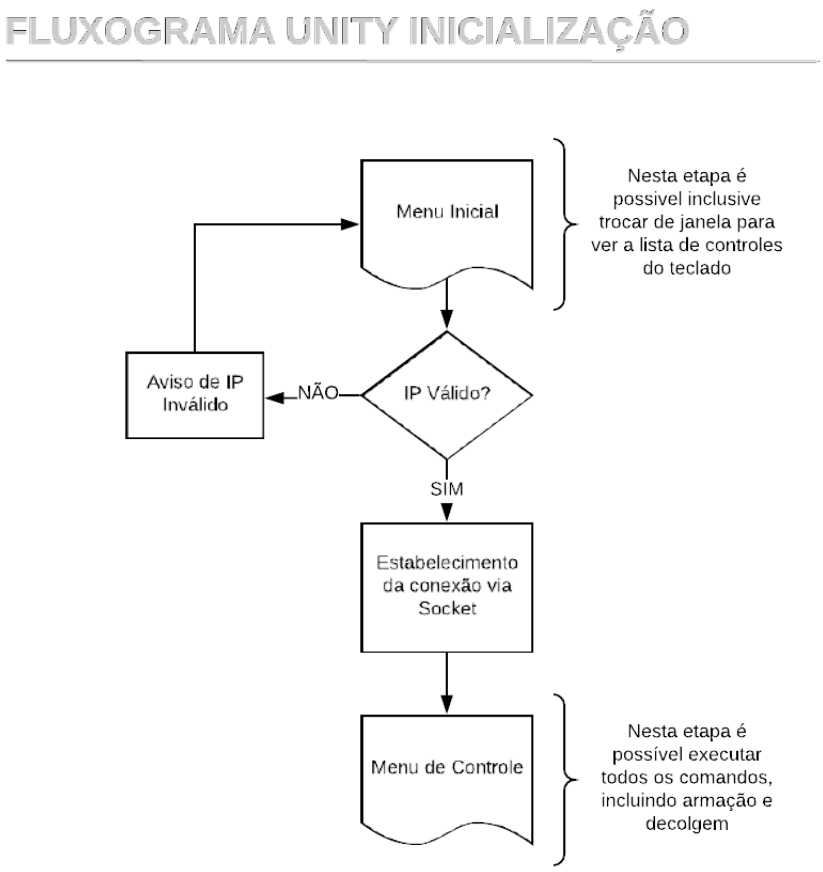
Fonte: Autor

O vídeo que apresenta o resultado oscilatório do controlador tipo P pode ser conferido no *Link* do *YouTube* <https://youtu.be/cQOA0gR6Fus>.

Devido a esses problemas, manteve-se o controle direto em malha aberta. A resposta a comandos com ganhos pequenos apresentou resultado satisfatório acompanhado de uma baixa oscilação de altitude. O resultado em vídeo se encontra na Seção 5.3.

Com relação ao código implementado na plataforma *Unity*, a rotina de inicialização está representada no fluxograma da Figura 41.

Figura 41 – Rotina de inicialização no programa em Unity



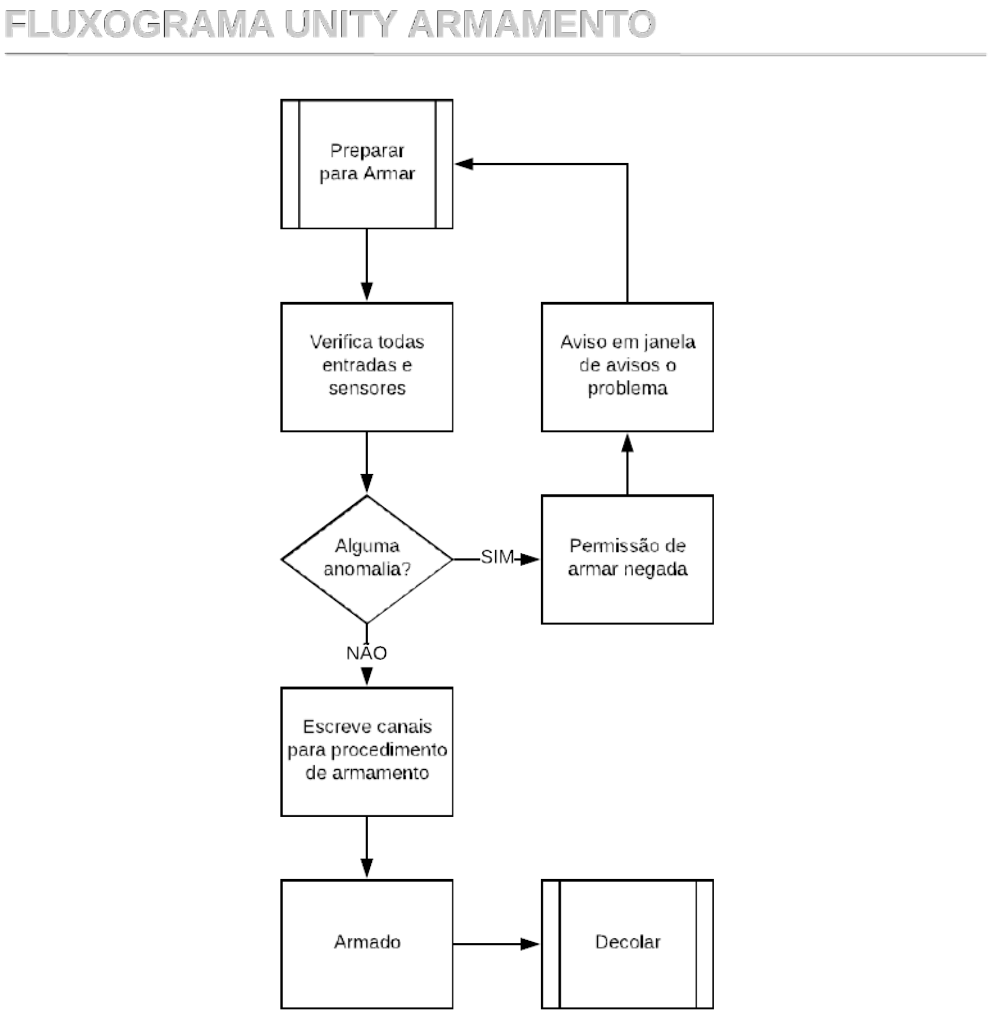
Fonte: Autor

Como comentado na Seção 4.2, o usuário deve inserir um IP válido na janela inicial (Figura 33), para que seja movido para a janela de operação do *drone*. Pode-se notar no fluxograma da Figura 41, que, caso o número IP esteja incorreto, o programa irá alertar ao usuário de uma falha na conexão. Caso o usuário queira se familiarizar com os comandos do teclado, basta clicar no ícone inferior e será remanejado para a janela da Figura 34.

Para iniciar um voo, o usuário deverá primeiramente armar a aeronave. O processo de armar é fundamental para assegurar o ambiente. Isto é, antes de armar, a aeronave se encontra completamente inofensiva, não necessitando de cuidados com suas partes móveis, momento perfeito para transportes ou verificações da integridade de suas partes. Após armado, o *drone* já apresenta um controle contra variação de velocidades angulares, isto é, ele já reage a movimentos de arfagem e rolagem, mesmo com seu comando de tração em estado mínimo. Por isso, armar requer que o ambiente esteja seguro. A rotina de armação

pode ser observado na Figura 42.

Figura 42 – Rotina de armação da aeronave no programa em Unity



Fonte: Autor

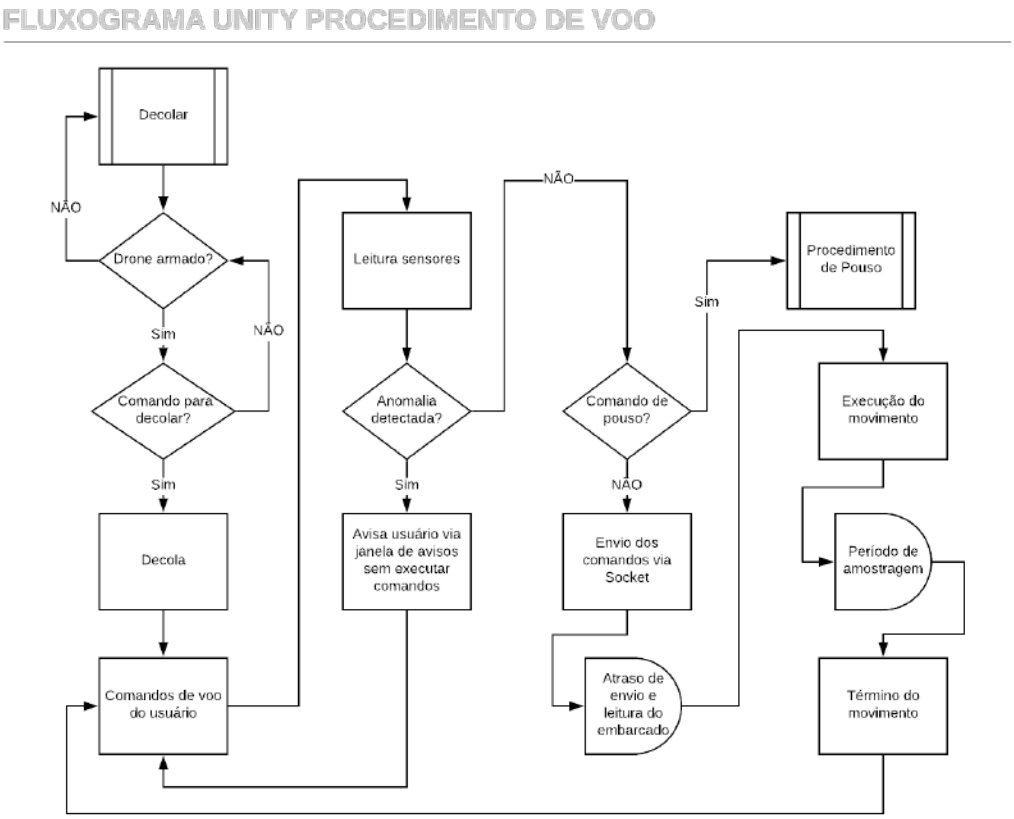
Como apresentado na Figura 42, após armado, o *drone* pode executar sua decolagem, de forma manual, ou seja, com o próprio usuário aumentando o comando de tração dos motores uniformemente.

Os ganhos, que podem ser mudados conforme a vontade do usuário, estão localizados ao lado direito da Figura 35. Não é mostrado nos fluxogramas, mas caso o comando de emergência seja pressionado, não importando a situação, o *drone* se desliga, evitando assim algum dano a terceiros ou a si mesmo. Caso o usuário pressione o botão de decolagem, este irá automaticamente garantir que o quadrimotor suba a uma altitude previamente

definida.

Caso o *drone* esteja executando algum movimento, sem que tenha havido um comando para tal, basta o usuário ajustar seu *Trim*[1], localizado na parte inferior à esquerda da Figura 35. O fluxograma que descreve seu funcionamento em voo pode ser visto na Figura 43.

Figura 43 – Rotina de voo da aeronave no programa em Unity

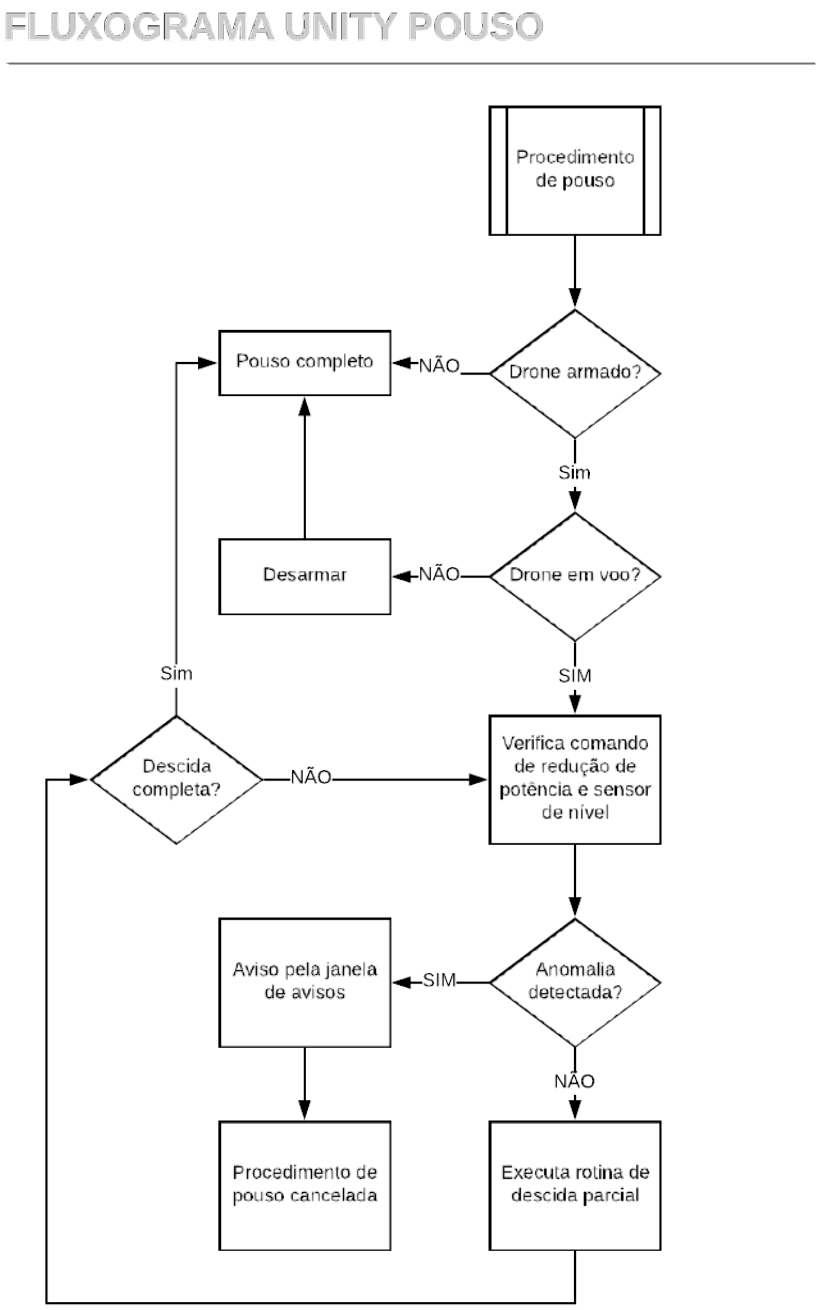


Fonte: Autor

Percebe-se no fluxograma da Figura 43 que é possível executar a rotina de pouso a qualquer instante. Assim como a rotina de decolagem, pousar faz com que o *drone* reduza seus motores uniformemente, até que este chegue a zero. O pouso, diferente da decolagem, exige um movimento muito mais sutil e delicado, evitando que o *drone* simplesmente caia, causando um risco enorme de danificação. Por isso, a rotina de pouso é feita em etapas que possuem um tempo maior que a decolagem. O fluxograma referente à rotina de pouso que explica como é realizado o comando no sitema local pode ser analisado na Figura 44.

[1] O Trim realiza ajustes finos que acertam pequenas imperfeições não planejadas, como centro de massa fora de posição.

Figura 44 – Rotina de pouso da aeronave no programa em Unity



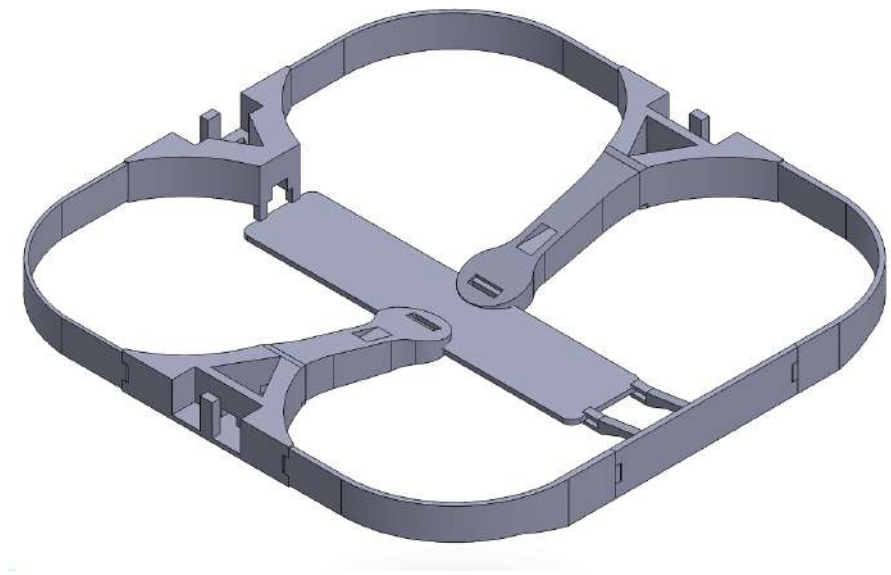
Fonte: Autor

5 AJUSTES FINAIS E EXPERIMENTO COMPLETO DA PLATAFORMA

5.1 Definição do *layout* Inicial

Para melhorar a proteção do *drone*, além do encaixe de seus sensores, foi projetado um protetor, utilizando a ferramenta *Solid Works*, que preserve tanto a aeronave, quanto terceiros de danos provindos principalmente de suas hélices. Essa primeira versão está apresentada na Figura 45, sendo utilizada para a realização dos testes.

Figura 45 – Protetor de teste



Fonte: Autor

O primeiro protótipo utiliza uma *protoboard* como local de acoplamento entre sistema embarcado, sensores e conexões com a placa controladora. No primeiro modelo funcional, foi observado que o protetor impresso em 3D estava com certas irregularidades, mas apresentava um peso baixo, suficiente para que o *drone* pudesse decolar com pouco menos da metade do comando. Esse *drone* pode ser conferido na Figura 46.

Figura 46 – Primeiro protótipo funcional

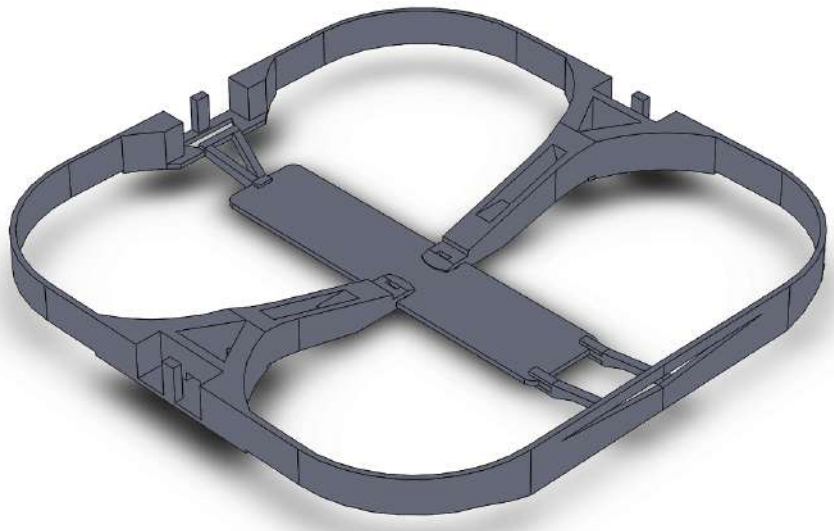
Fonte: Autor

Um rádio destinado para o *hobbie* de dronemodelismo foi utilizado para controlar o *drone* em um teste inicial de voo. Dessa maneira, pode-se averiguar a forma que o modelo voava, a fim de analisar qualitativamente o protótipo contruído. Essa análise se baseou nos conhecimento prévios de pilotagem do próprio autor.

5.2 Construção do *layout* final

O *layout* do protetor foi alterado para que fosse utilizado menos material plástico, de maneira a, praticamente, não reduzir sua resistência, além de melhorar o encaixe de suas peças. A versão final pode ser vista na Figura 47.

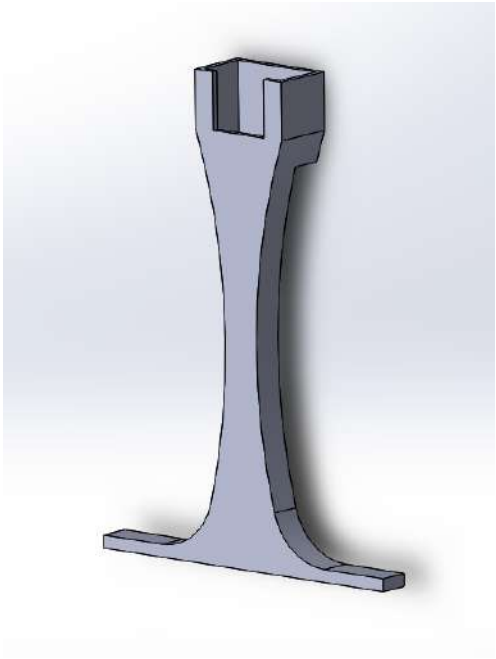
Figura 47 – Protetor final



Fonte: Autor

Elaborou-se também o suporte da câmera que irá situar-se na parte frontal, acima do protetor. Seu desenho pode ser conferido na Figura 48.

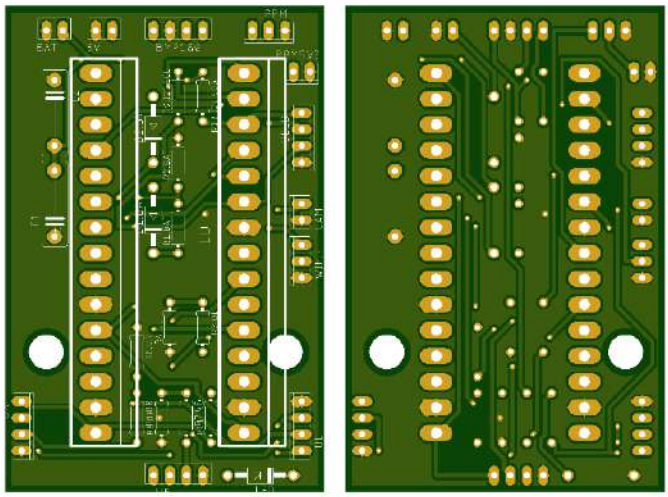
Figura 48 – Suporte da câmera



Fonte: Autor

Após testes realizados na *protoboard*, foi desenvolvida uma placa de circuito impresso no *software Eagle*. A placa criada precisava encaixar-se dentro do *framekit*, para evitar contatos desnecessários com as hélices. Por conta do pequeno espaço disponível, a placa precisou de dois *layers*, o que dificultou sua fabricação de forma manual. Sendo assim, o projeto foi enviado para ser confeccionado em um site de manufatura de placas. O desenho da placa final desenvolvida pode ser observado na Figura 49.

Figura 49 – Projeto de placa confeccionada, superfície superior e inferior



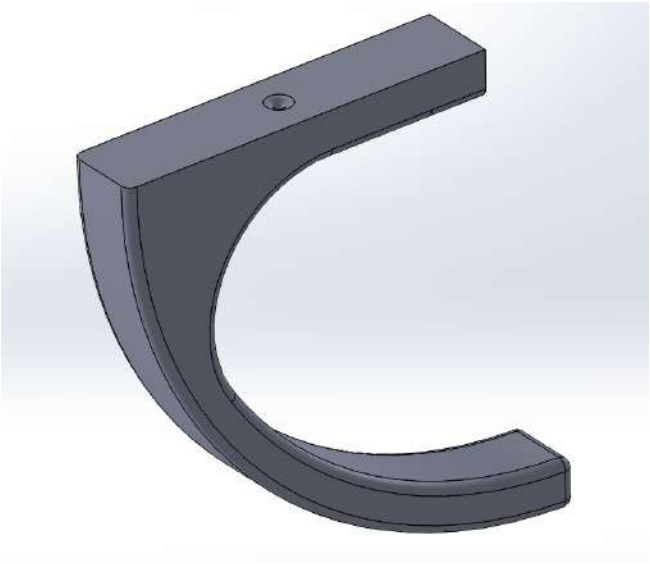
Fonte: Autor

5.3 Resultados de ensaios e procedimentos de adequação da estrutura

Os ensaios e testes referentes à estrutura final foram fundamentais para o seu aprimoramento e segurança. O teste inicial executa a rotina de decolagem e pouso. O único ponto de melhoria que foi considerado, foi a modificação do trem de pouso já existente. Por conta de choques mecânicos muito fortes, além de possuir uma altura com relação ao chão muito baixa, a substituição do trem de pouso original do *framekit* foi fundamental. O aumento de tamanho do trem foi necessário, pois antes, pequenas inclinações faziam com que o protetor tocasse o chão.

O trem de pouso inicialmente projetado possuía o formato observado na Figura 50.

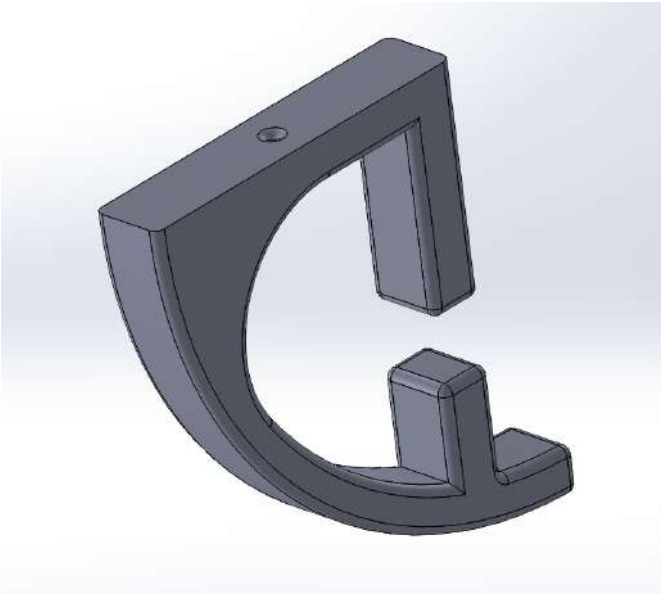
Figura 50 – Primeiro modelo de trem de pouso criado



Fonte: Autor

Porém, após alguns testes, o modelo de trem de pouso mostrou-se mal dimensionado para a função que iria exercer, pois todo o conjunto de 4 peças quebrou em um dos ensaios. O motivo foi a elevada deformação da peça mediante o peso considerável do quadrimotor. A solução foi, além de aumentar a largura do material próximo ao local de rompimento, inserir um batente para que não houvesse uma deformação elevada, e a peça resistisse melhor ao impacto criado pelas pequenas quedas durante o pouso. A versão final do trem de pouso pode ser vista na Figura 51.

Figura 51 – Modelo de trem de pouso final



Fonte: Autor

O segundo teste foi realizado para verificar a capacidade do *drone* de evitar colisões utilizando seus sensores ultrassônicos.

Os procedimentos e execuções dos ensaios foram realizados em um ambiente controlado, sem risco de danificação de componentes ou terceiros. O experimento completo pode ser visto nos *links* do *YouTube* <https://youtu.be/bkbHZbFxHeQ> e <https://youtu.be/Cd3gf-xj3K0>.

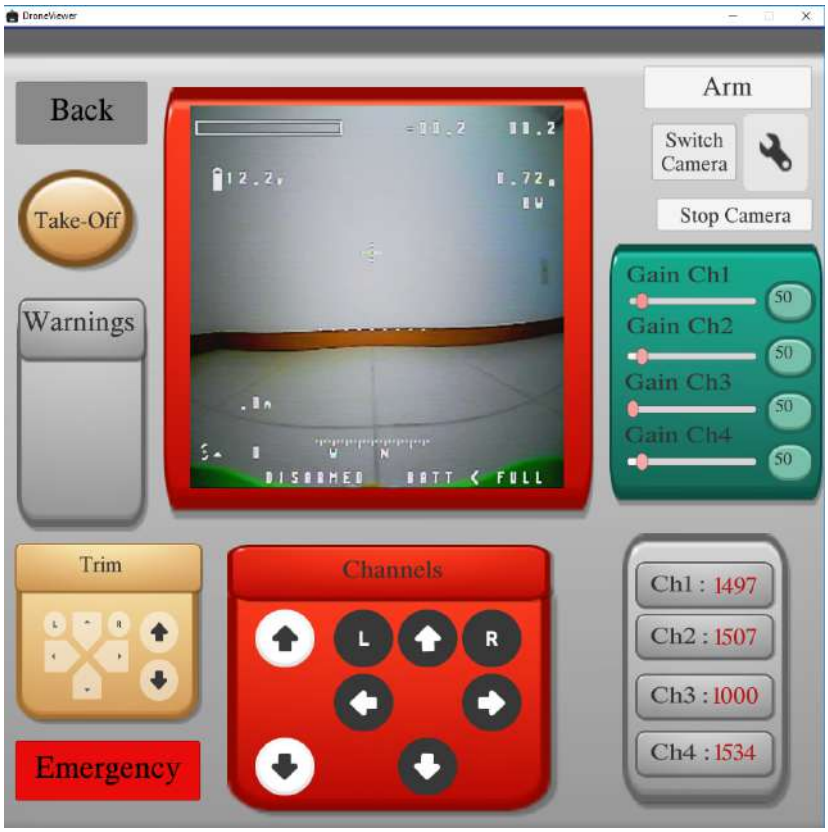
No primeiro vídeo, o *drone* evitou as paredes e avisou o usuário, por meio do *Software*, sobre a proximidade da parede quando o comando era exercido ($0:47\ min$). No segundo vídeo, o *drone* apresenta a resposta à operação em malha aberta da altura, com resultados satisfatórios para um sistema sem controle em malha fechada. No vídeo do *YouTube*<https://youtu.be/rZBEuxTJZlM>, é possível ver toda a rotina de inicialização do programa desenvolvido e a visão do quadrimotor.

Por fim, pode-se concluir que os resultados foram adequados. Apesar de o controle de altitude em malha fechada não ter sido implementado, sua ausência não impossibilitou o controle do *drone* pelo usuário, objetivo inicialmente proposto.

Além disso, o funcionamento do programa criado na plataforma local pôde ser validado e apresentou uma interface amigável ao usuário, com poucos comandos, sendo eles autoex-

plicativos. A Figura 52 mostra o resultado obtido pela imagem da tela de controle durante o funcionamento da câmera.

Figura 52 – Tela de controle do programa com a câmera em pleno funcionamento



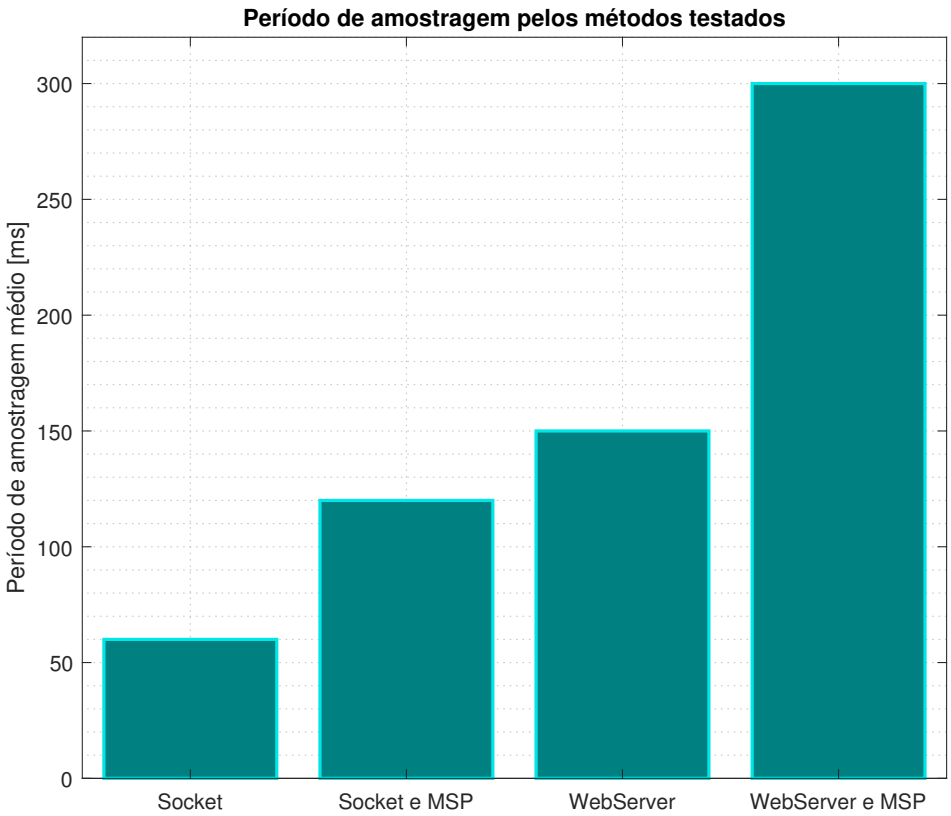
Fonte: Autor

6 DISCUSSÃO DOS RESULTADOS E POSSÍVEIS APLICAÇÕES DO PROTÓTIPO

6.1 Discussão dos resultados

O quadrimotor desenvolvido apresenta um controle simples e funcional para o objetivo proposto. O sistema possui apenas um microcontrolador para realizar todas as funções de comunicação, leitura de sensores e escrita de comandos, resultando em um tempo de amostragem relativamente alto. Os períodos de amostragem das abordagens discutidas neste trabalho podem ser vistos na Figura 53.

Figura 53 – Tempo de amostragem médio por rotinas utilizadas



Fonte: Autor

O atraso do tempo de amostragem dificulta que o controle de altitude tenha um melhor

desempenho, visto que a dinâmica do quadrimotor, enquanto voa, varia rapidamente. Porém a não existência de um controlador de altitude não foi considerado um problema, por conta dos resultados dos testes. Uma possível solução na implementação do controlador de altitude seria a mudança do tipo de filtro empregado. Um modelo de filtro adequado é o filtro de Kalman, que utiliza dados estatísticos para determinar o valor lido.

Geralmente, outros quadrimotores no mercado possuem mais de um controlador embarcado, possibilitando uma resposta mais rápida a qualquer ação. Mesmo assim, foi possível implantar sensores que evitassem a colisão do *drone* em paredes, função de grande importância para segurança do protótipo.

Seu custo final, obervando as principais peças adquiridas, pode ser observado na Tabela 3.

Tabela 3 – Custos de todas as peças necessárias para a construção do quadrimotor

Tabela de análise de custos			
Componente	Preço	Quantidade	Total
QAV250	U$129,99	1	R$ 493,96
Matek F405	R$ 191,00	1	R$ 191,00
Esc 20A BeeRotor	R$ 45,00	4	R$ 180,00
2200 mAh Zippy	R$ 76,00	1	R$ 76,00
NodeMCU esp8266	R$ 27,00	1	R$ 27,00
Sharp Gp2y0a02yk0f	R$ 56,00	1	R$ 56,00
Camera FPV Tiny Whoop	R$ 104,00	1	R$ 104,00
Display OLED	R$ 23,00	1	R$ 23,00
Emax Brushless Mt2204 2300kv	R$ 49,99	4	R$ 199,96
4 Hélices 50x45 X 3	R$ 13,00	1	R$ 13,00
Total			R$ 1.363,92

Na Tabela 3, não foram considerados os gastos na confecção da proteção, produzidos pela impressora 3D da universidade. Apesar do elevado custo, o autor já possuia os componentes utilizados. Os valores foram levantados com base nos produtos vendidos no Brasil, exceto o primeiro item, por não ser encontrado disponível no mercado nacional.

O protótipo final possui um peso de 595 g, somando-se à bateria, que pesa 180 g, aproximadamente 780 g. Assim, para que os quatro motores tenham tração suficiente para que a aeronave decole seja um valor acima de 780 g. De acordo com o fabricante, o consumo de cada motor no intante que a aeronave atinja esse valor de tração pode ser observado na Tabela 2, onde os valores estão destacados em negrito. Observando a Figura 54, onde a relação de eficiência dos motores e a tração por eles fornecida é apresentada, pode-se

notar que os motores escolhidos foram bem dimensionados para o protótipo final. O ponto de decolagem é o ponto onde a tração somada dos motores é maior que o peso.

Figura 54 – Eficiência dos motores por tração

Fonte: Autor

De acordo com os dados da Tabela 2 e da Figura 54, sabe-se que os motores conseguem produzir uma tração superior ao peso do *drone*. Com isso, caso existam modificações ou adição de componentes extras/cargas (*payload*) de até 2 kg, o *drone* ainda seria capaz de voar.

A versão final do protótipo encontra-se na Figura 55.

Figura 55 – Versão final do projeto

Fonte: Autor

6.2 Possíveis aplicações

A aeronave desenvolvida pode ter aplicações acadêmicas como, por exemplo, a programação em uma plataforma mais dinâmica, fazendo com que a aeronave execute uma rota pré programada ou, utilizando a câmera embarcada, realize análises de imagens e execute comandos conforme o programado.

Além disso, como um dos objetivos é a flexibilidade do protótipo a fim de ser uma ferramenta para utilização futura de forma educacional, o *drone* pode ser empregado no ensino de pilotagem e controle de aeronaves.

Um *drone* controlado remotamente, via *Wi-Fi*, por exemplo, pode modernizar as formas de monitoramento de segurança em ambientes protegidos, sem a necessidade de que pessoas precisem estar presentes no local, fazendo então seu custo operacional reduzir e o nível de segurança aumentar.

O modelo foi desenvolvido a partir de peças encontradas facilmente no mercado e de peças projetadas pelo autor. Ele possui uma potência instalada de aproximadamente 2,6 *cv* em seu conjunto de motores. Por ter essa alta capacidade, o *drone* pode ser também aplicado em ambientes externos.

Essa mudança exigiria apenas algumas adequações, como a mudança da comunicação, que atualmente é *Wi-Fi*, para algum tipo que possibilite um alcance maior de controle, como por exemplo o LoRa (*Long Range*) e o GSM. LoRa é uma tecnologia de comunicação de dados digital sem fio patenteada e desenvolvida pela Cycleo de Grenoble, França, e adquirida pela Semtech em 2012. A LoRa usa bandas de radiofrequência sub-gigahertz sem licença como 169 MHz, 433 MHz, 868 MHz e 915 MHz (ALLIANCE, 2018).

A placa utilizada (Figura 12) também possui entradas para acoplamento de um módulo GPS, possibilitando uma melhora no seu controle de posição em ambiente externo.

7 CONCLUSÕES E TRABALHOS FUTUROS

O quadrimotor desenvolvido apresentou um controle simples e funcional para o objetivo proposto. Ele permite que o usuário consiga realizar manobras e operá-lo como se o estivesse comandando por um controle remoto comercial.

O funcionamento do programa criado na plataforma remota apresentou uma interface amigável ao usuário, com poucos comandos, sendo eles autoexplicativos. Além disso, ele possibilita ao usuário ter a visão do *drone* pela comunicação do transmissor da câmera acoplada em sua estrutura.

A comunicação *Wi-Fi* por *Socket*, utilizada no sistema final, também foi um ponto importante para que essa operação remota fosse realizada com sucesso.

Por fim, o protótipo proporciona o desenvolvimento de algumas aplicações, podendo, inclusive, ser aproveitado em trabalhos com o foco em ambientes externos. Nesses casos, são necessários motores potentes que permitam a aeronave vencer fortes rajadas de ventos, característica existente no protótipo desenvolvido.

Entretanto, para aprimorar o trabalho, seria interessante diminuir a taxa de execução das funções, isto é, leitura da entrada, sensores e escrita na saída do controle. Para tal, deve-se introduzir outro microcontrolador, dividindo tarefas, como mencionado anteriormente na etapa de resultados, Seção 6.1. A inserção de outro microcontrolador, além de permitir um desempenho muito mais eficiente e rápido, por conta da divisão de tarefas e redução de gargalos, como comunicação e leituras de sensores, facilitaria toda a parte de controle do quadrimotor, principalmente a odometria.

Outra maneira de auxiliar na estimativa de odometria, seria a aquisição de uma IMU de maior qualidade (e.g., *Bno*055), como citado na Seção 4.3.

Paralela à solução de uma nova IMU, outra opção seria utilizar uma placa controladora com sensores mais precisos e comunicação via protocolo mais simples e rápida de implementar. Deste modo, não seria necessária a instalação e acoplamento de mais um componente ao sistema. Porém, este tipo de placa controladora é muito mais cara que os modelos disponíveis no projeto. Um exemplo de placa avançada é a *Naza-M V2 Flight Controller*, que tem um preço médio de R$1.500,00. Mesmo assim, existem modelos de placas de desenvolvimento que se aplicariam adequadamente, como a *PixHawk*, que possui um preço

mais acessível na faixa de R$ 300,00.

Outra possibilidade de implementar o controle de posição do *drone* baseando-se em visão computacional. Utilizando OpenCV, uma biblioteca multiplataforma que possui muitas funções desenvolvidas nesta área (OPENCV, 2018). Operando a câmera instalada no quadrimotor, juntamente com a bibliotecas fornecidas pelo OpenCV, é possível criar um controle por imagem, baseado em informações visuais da cena.

Esse tipo de implementação não foi possível de se integrar no sistema remoto pelo fato das bibliotecas serem desenvolvidas em linguagens incompatíveis com a plataforma *Unity*, que suporta apenas C# ou *JavaScript*. Um ponto negativo deste método é que ele requer que o *drone* possua pontos específicos para se ater como referência. Outra forma de executar um controle por imagem seria a partir de um ambiente controlado, com câmeras instaladas em posições bem definidas que monitorassem o ambiente. A partir de pontos de cores pré determinadas instalados em seu protetor, o *drone* poderia ser identificado e o controle de posição poderia ser realizado.

Existe a possibilidade ainda de que sejam realizados mais de um método simultâneo, tornando o sistema ainda mais completo. Quando não houvessem pontos de referência por imagem, para o *drone*, utilizar-se-ia então o médodo via sensores odométricos. E assim que pontos de referência surgissem na imagem, realizar-se-ia a calibração dos sensores, reduzindo então os erros somados.

Após realizar o controle de forma mais elegante e eficiente, a próxima etapa consiste em tornar seu funcionamento mais autônomo, podendo ser o desenvolvimento de uma base onde o *drone* possa se autocarregar em situações que forem necessárias. Em trabalhos futuros poder-se-ia, também, aplicar rotinas básicas de movimentação e a criação de uma base inteligente, com carregamento automático.

REFERÊNCIAS

ALLIANCE, L. *LoRa Alliance^{TM}*. 2018. Acessado em 20 de outubro de 2018. Disponível em: <https://lora-alliance.org>. Citado na página 73.

AMAZON. *HiLetgo-ESP-WROOM-32-Development-Microcontroller-Integrated*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <https://www.amazon.com/HiLetgo-ESP-WROOM-32-Development-Microcontroller-Integrated/dp/B0718T232Z>. Citado na página 41.

AMAZON. *HiLetgo-Internet-Development-Wireless-Micropython*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <https://www.amazon.com/HiLetgo-Internet-Development-Wireless-Micropython/dp/B010O1G1ES>. Citado na página 41.

APVRILLE, L.; TANZI, T.; DUGELAY, J.-L. Autonomous drones for assisting rescue services within the context of natural disasters. In: IEEE. *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*. [S.l.], 2014. p. 1–4. Citado na página 12.

BAICHTAL, J. *Building Your Own Drones: A Beginners' Guide to Drones, UAVs, and ROVs*. [S.l.]: Que Publishing, 2015. Citado 4 vezes nas páginas 22, 24, 27 e 29.

CINNAMON, R. S. K. . F. T. I. *DIY Drones for the EVIL GENIUS*. [S.l.]: McGraw-Hill Education, 2017. ISBN 978-1-25-986147-5. Citado 2 vezes nas páginas 10 e 35.

CIVIL, A. Agência Nacional de A. *Quantidade de Cadastros*. 2018. Acessado em 12 de abril de 2018. Disponível em: <http://www.anac.gov.br/assuntos/paginas-tematicas/drones/quantidade-de-cadastros>. Citado 2 vezes nas páginas 10 e 11.

EMAX. *brushless-motors*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <https://www.emaxmodel.com/emax-rs2205-racespec-motor.html>. Citado 4 vezes nas páginas 23, 33, 34 e 35.

GODOY, J. et al. A new approach of v2x communications for long range applications in uavs. In: IEEE. *2018 2nd URSI Atlantic Radio Science Meeting (AT-RASC)*. [S.l.], 2018. p. 1–4. Citado na página 13.

ISSOD, C. *Getting started with hobby quadcopters and drones*. [S.l.]: Seattle, WA: Createspace, 2015. Citado na página 11.

JENSEN, T.; ZELLER, L.; APAN, A. The use of an unmanned aerial vehicle as a remote sensing platform in agriculture. *Australian Journal of Multi-disciplinary Engineering*, Taylor & Francis, v. 8, n. 2, p. 139–146, 2011. Citado na página 12.

KNOEDLER, B.; ZEMMARI, R.; KOCH, W. On the detection of small uav using a gsm passive coherent location system. In: IEEE. *Radar Symposium (IRS), 2016 17th International*. [S.l.], 2016. p. 1–4. Citado na página 13.

LIANG, O. *Esc Firmware protocols*. 2017. Acessado em 20 de outubro de 2018. Disponível em: <https://oscarliang.com/esc-firmware-protocols/>. Citado 2 vezes nas páginas 28 e 29.

LIANG, O. *HOW TO CHOOSE FLIGHT CONTROLLER FOR QUADCOPTER*. 2017. Acessado em 20 de outubro de 2018. Disponível em: <https://oscarliang.com/best-flight-controller-quad-hex-copter/>. Citado na página 13.

LINDEN, D. Handbook of batteries. In: *Fuel and Energy Abstracts*. [S.l.: s.n.], 1995. v. 4, n. 36, p. 265. Citado na página 24.

LUMENIER. *The QAV250 Airframe*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <https://www.lumenier.com/products/legacy/qav250>. Citado na página 21.

MARINHO, C. A. et al. In-service flare inspection by unmanned aerial vehicles (uavs). In: *18th World Conference on Nondestructive Testing. Durban, South Africa*. [S.l.: s.n.], 2012. Citado na página 12.

MATEK, S. *FLIGHT CONTROLLER F405-AIO*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <http://www.mateksys.com/?portfolio=f405-aio>. Citado na página 26.

MCAREE, O.; AITKEN, J. M.; VERES, S. M. A model based design framework for safety verification of a semi-autonomous inspection drone. In: IEEE. *Control (CONTROL), 2016 UKACC 11th International Conference on*. [S.l.], 2016. p. 1–6. Citado na página 12.

OPENCV, T. *OpenCV*. 2018. Acessado em 22 de setembro de 2018. Disponível em: <https://opencv.org>. Citado na página 75.

PAL, K. *What does PPM (pulse position modulation) mean in drone terminology?* 2014. Acessado em 30 de setembro de 2018. Disponível em: <https://www.quora.com/What-does-PPM-pulse-position-modulation-mean-in-drone-terminology>. Citado 2 vezes nas páginas 42 e 43.

QUADS, R. T. F. *FLIP 32 - F4 - OMNIBUS REV.2*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <http://www.readytoflyquads.com/flip-32-f4-omnibus>. Citado 2 vezes nas páginas 25 e 26.

RODRIGUES, L. Fundamentos da engenharia aeronáutica com aplicações ao projeto sae-aerodesign: Aerodinâmica e desempenho. *Salto-SP*, v. 1, 2014. Citado na página 15.

SÁ, F. B. de et al. Planejamento de trajetoria para um robô movel usando imagens capturadas por um vant: Conceitos e resultados preliminares. 2014. Citado na página 12.

SALVADOR, R. da M.; ESCALFONI, V. H. de M.; ASSIS, L. de. Modulo de voo para quadrimotor baseado em visao computacional e ros para seguimento de padrao movel. 2017. Citado na página 13.

SANCHEZ-IBORRA, R. Wifi networks on drones. 2016. Citado na página 13.

SENSORTEC, B. *Data sheet BNO055-Intelligent 9-axis absolute orientation sensor*. 2014. Citado na página 51.

SERIOUSLYPRO. *SP RACING^{TM} F3 FLIGHT CONTROLLER.NEXT-GENERATION HARDWARE FOR YOUR MULTI-ROTOR*. 2018. Acessado em 10 de agosto de 2018. Disponível em: <http://seriouslypro.com/spracingf3>. Citado 2 vezes nas páginas 25 e 26.

SHARP. *Distance Measuring Sensor Unit Measuring distance: 20 to 150 cm Analog output type*. 2006. Acessado em 22 de setembro de 2018. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf>. Citado na página 46.

UNITY, T. *Imagine, corie e tenha sucesso com Unity*. 2018. Acessado em 22 de setembro de 2018. Disponível em: <https://unity3d.com/pt>. Citado na página 47.

YAN, J. et al. Indoor range-direction-movement sar for drone-based radar systems. In: IEEE. *Microwave Conference (APMC), 2017 IEEE Asia Pacific*. [S.l.], 2017. p. 1290–1293. Citado na página 13.