



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA

JUDÁ TEIXEIRA SANTOS

**DESENVOLVIMENTO DE INTERFACE DE OPERAÇÃO PARA SISTEMA
AUTÔNOMO DE VARREDURA AÉREA 3D APLICADO À TERMELÉTRICA DO
PECÉM/CE**

FORTALEZA

2021

JUDÁ TEIXEIRA SANTOS

DESENVOLVIMENTO DE INTERFACE DE OPERAÇÃO PARA SISTEMA AUTÔNOMO
DE VARREDURA AÉREA 3D APLICADO À TERMELÉTRICA DO PECÉM/CE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Controle e Automação

Orientador: Prof. Dr. Fabrício Gonzalez Nogueira

Coorientador: Prof. Dr. Bismark Claude Torrico

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- S235d Santos, Judá Teixeira.
Desenvolvimento de interface de operação para sistema autônomo de varredura aérea 3D aplicado à Termelétrica do Pecém/CE / Judá Teixeira Santos. – 2021.
67 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2021.
Orientação: Prof. Dr. Fabrício Gonzalez Nogueira.
Coorientação: Prof. Dr. Bismark Claire Torrico.
1. Robótica Móvel. 2. Veículos Aéreos Não Tripulados. 3. Interface Gráfica. 4. VANT multi-rotor autônomo. 5. Android. I. Título.

CDD 621.3

JUDÁ TEIXEIRA SANTOS

DESENVOLVIMENTO DE INTERFACE DE OPERAÇÃO PARA SISTEMA AUTÔNOMO
DE VARREDURA AÉREA 3D APLICADO À TERMELÉTRICA DO PECÉM/CE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Controle e Automação

Aprovada em: 24 de março de 2021

BANCA EXAMINADORA

Prof. Dr. Fabrício Gonzalez Nogueira (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Bismark Claire Torrico (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulo Peixoto Praça
Universidade Federal do Ceará (UFC)

Prof. Dr. Jarbas Aryel Nunes da Silveira
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família, por todo o apoio que me deram durante esta jornada.

AGRADECIMENTOS

Agradecer primeiramente a minha família que sempre me apoiou em minhas decisões. Ao Jeferson Henrique por ouvir todas as minhas reclamações e reclamar junto comigo durante as últimas etapas do mestrado e ser um grande companheiro durante esta jornada. Aos meus amigos Lucas Batista e Silvio Furtado por aproveitarem as noites mal dormidas jogando comigo. Ao Paulo Passos pelas boas conversas e aconselhamentos durante todo esse período.

Aos meus professores orientadores Prof. Dr. Fabrício Gonzalez e Prof. Dr. Bismark Torrico e toda a equipe do programa de Pesquisa e Desenvolvimento do Grupo de Pesquisa em Automação, Controle e Robótica (GPAR) da Universidade Federal do Ceará. Equipe que foi de fundamental importância para desenvolvimento do projeto e que me acompanhou durante esses dois anos de mestrado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro durante o mestrado. À EDP Energia Pecém e ao programa de Pesquisa e Desenvolvimento Tecnológico do Setor de Energia Elétrica da ANEEL por todo apoio recebido durante a elaboração e execução do projeto PD-07267-0016/2018.

Ao Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

“The scariest moment is always just before you start.”

(Stephen King)

RESUMO

Usinas termoelétricas a carvão necessitam do constante controle da quantidade de carvão armazenada em seu estoque. Isto garante o abastecimento da usina e mantém a produção de energia. Técnicas como fotogrametria podem solucionar este problema, porém, apresentam algumas desvantagens quando comparado com um mapeamento a laser. O correto controle e operação destes sistemas é imprescindível para atualizar corretamente o inventário de estoque da usina. Este trabalho apresenta o desenvolvimento de uma interface gráfica de operação para sistema aéreo de mapeamento tridimensional de alta precisão. O sistema desenvolvido tem a finalidade de reconstruir e mapear o pátio de carvão da UTE Pecém através de nuvem de pontos para controle do inventário.

O sistema aéreo apresenta um sensor LiDAR (*Light Detection and Ranging*) para aquisição da nuvem de pontos. Utiliza também o sistema de posicionamento RTK (*Real-Time Kinematic*) para aumentar a precisão do posicionamento do sistema, o que também influencia na reconstrução da nuvem de pontos. Estes equipamentos são acoplados em um VANT (Veículo Aéreo Não Tripulado) junto de um computador de bordo para processamento dos dados e comunicação com a interface móvel.

Toda a operação do sistema de mapeamento aéreo é feita através de uma interface gráfica. A interface foi desenvolvida para dispositivos móveis *Android* e escrita na linguagem de programação *Java*. A interface móvel apresenta informações em tempo real do sistema de mapeamento tridimensional. A interface tem como finalidade facilitar a operação do VANT e viabilizar informações da aeronave para o operador em campo.

Ao final do trabalho são apresentados experimentos testes de validação da operação do sistema através da interface. Também são demonstrados experimentos de mapeamento do local de armazenamento de carvão com o sistema desenvolvido.

Palavras-chave: Robótica Móvel. Veículos Aéreos Não tripulados. Interface Gráfica. VANT multi-rotor autônomo. Android

ABSTRACT

Coal-fired power plants required constant control of the amount of coal stored in their stock. This guarantees the supply of the plant and maintains the production of energy. Techniques such as photogrammetry can solve this problem. However, they have disadvantages compared to laser mapping. The correct control and operation of these systems are essential to properly update the plant's inventory. This work presents the development of a graphical operator interface for the high-precision three-dimensional aerial mapping system. The developed system has the hardware to rebuild and map the coal yard of UTE Pecém through a point cloud to control the inventory.

The aerial system features a LiDAR (Light Detection and Ranging) sensor for acquiring the point cloud. It also uses a RTK (Real-Time Kinematic) positioning system to increase the positioning accuracy of the system, which also influences the reconstruction of the point cloud. This equipment is coupled to an UAV (Unmanned Aerial Vehicle) with an on-board computer for data processing and communication with the mobile interface.

The entire operation of the aerial mapping system is done through a graphical user interface. The interface was developed for mobile devices using *Android* and written in *Java*. The mobile interface presents real-time information from the three-dimensional mapping system. The interface has the purpose of facilitating the operation of the UAV and providing information from the aircraft to the operator in the field.

At the end of the work, experiments to validate the operation of the system through the interface are presented. Experiments to map the coal storage location with the developed system are also demonstrated.

Keywords: Mobile Robotics. Unmanned Aerial Vehicles. UAV autonomous multitoror. Graphical User Interface. Android

LISTA DE FIGURAS

Figura 1 – Diagrama Operacional do Sistema de Mapeamento.	19
Figura 2 – Sistema Aéreo de Mapeamento 3D.	20
Figura 3 – Teste de voo do VANT <i>Wind 4</i>	20
Figura 4 – Performance de propulsão do motor M10 Pro.	21
Figura 5 – Esquemático do controlador de voo A3	23
Figura 6 – Sensor LiDAR LD-MRS 420201.	24
Figura 7 – Região de leitura do sensor LD-MRS 420201.	24
Figura 8 – Sistema D-RTK completo.	26
Figura 9 – Manifold 2-G conectada ao VANT.	27
Figura 10 – Sistema desenvolvido em <i>ROS</i>	29
Figura 11 – Quadros de referência adotados.	30
Figura 12 – Link de telemetria via rádio.	32
Figura 13 – Sistema <i>Lightbridge 2</i> - Na esquerda o módulo terrestre e na direita o módulo aéreo.	33
Figura 14 – Chave de seleção do modo de voo.	35
Figura 15 – Diagrama de utilização MSDK e conectividade com o VANT.	36
Figura 16 – Ícone do aplicativo desenvolvido.	38
Figura 17 – Modelo do ciclo de vida de uma atividade.	38
Figura 18 – Interface móvel desenvolvida.	40
Figura 19 – Criação da lista dinâmica de missões.	43
Figura 20 – Janela para seleção da missão de voo.	43
Figura 21 – Mapa com a localização em tempo real do VANT.	44
Figura 22 – Programa TRMS instalado no computador para operação.	46
Figura 23 – <i>Dashboard</i> de controle de inventário.	47
Figura 24 – Sistema aéreo conectado ao simulador de voo.	49
Figura 25 – Resultado do teste ao desconectar sensor LiDAR.	49
Figura 26 – Mensagem de aviso ao operar com o sensor LiDAR.	50
Figura 27 – Mensagem do nível de bateria.	51
Figura 28 – Mensagem de conexão do OSDK.	51
Figura 29 – Mensagem de conexão com o sistema aéreo.	52
Figura 30 – Caminho percorrido na simulação.	53

Figura 31 – Log de voo gerado durante simulação.	53
Figura 32 – Nuvem de pontos resultante da simulação.	54
Figura 33 – Imagem obtida do local a ser mapeado no teste.	55
Figura 34 – Trajetória do experimento na interface.	56
Figura 35 – Trajetória definida para missão de mapeamento.	56
Figura 36 – Nuvem de pontos resultante do Centro de Tecnologia.	57
Figura 37 – Sistema aéreo de mapeamento tridimensional na termoeletrica.	58
Figura 38 – <i>Ground System</i> do sistema de posicionamento RTK e operador com controle <i>Lightbridge 2</i> e Tablet com interface gráfica.	58
Figura 39 – Interface durante testes.	59
Figura 40 – Processo de mapeamento na termoeletrica.	60
Figura 41 – Caminho percorrido pelo sistema aéreo.	60
Figura 42 – Nuvem de pontos resultante do pátio de carvão.	61
Figura 43 – Comparativo da nuvem de pontos com o local.	61

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ESC	Electronic Speed Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
MSDK	Mobile Software Development Kit
OSDK	Onboard Software Development Kit
PMU	Power Management Unit
PPTM	Porto do Pecém Transportadora de Minérios
ROS	Robot Operating System
RTK	Real-Time Kinematic
SDK	Software Development Kit
UAV	Unmanned Aerial Vehicle
UI	User Interface
UTE	Usina Termoelétrica
VANT	Veículo Aéreo Não Tripulado

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Sistema de Mapeamento Aéreo	14
1.2	Sistema de operação	15
1.3	Trabalhos relacionados	15
1.4	Problemática e justificativa	17
1.5	Objetivos	17
1.6	Organização do texto	18
2	SISTEMA AÉREO DE MAPEAMENTO TRIDIMENSIONAL	19
2.1	Hardware do Sistema de Mapeamento	19
<i>2.1.1</i>	<i>Drone Wind 4</i>	<i>19</i>
<i>2.1.1.1</i>	<i>Controlador de voo DJI A3</i>	<i>22</i>
<i>2.1.2</i>	<i>Sensor LiDAR LD-MRS</i>	<i>23</i>
<i>2.1.3</i>	<i>Sistema D-RTK</i>	<i>25</i>
<i>2.1.4</i>	<i>Computador de Bordo Manifold</i>	<i>26</i>
<i>2.1.4.1</i>	<i>Sistema de desenvolvimento OSDK</i>	<i>27</i>
2.2	Software do Sistema de Mapeamento	28
<i>2.2.1</i>	<i>ROS</i>	<i>28</i>
<i>2.2.1.1</i>	<i>Construção da nuvem de pontos</i>	<i>30</i>
3	INTERFACE GRÁFICA E SISTEMA DE COMUNICAÇÃO	32
3.1	Comunicação entre VANT e Operador - Telemetria	32
<i>3.1.1</i>	<i>Lightbridge 2</i>	<i>33</i>
3.2	Mobile Software Development Kit - MSDK	35
<i>3.2.1</i>	<i>Aplicativo Android Desenvolvido</i>	<i>36</i>
<i>3.2.1.1</i>	<i>MainActivity</i>	<i>37</i>
<i>3.2.1.2</i>	<i>Interface gráfica</i>	<i>39</i>
<i>3.2.1.2.1</i>	<i>Informações de comunicação e conectividade</i>	<i>39</i>
<i>3.2.1.2.2</i>	<i>Botões de operação</i>	<i>41</i>
<i>3.2.1.2.3</i>	<i>Log de voo</i>	<i>43</i>
<i>3.2.1.2.4</i>	<i>Mapa e progresso</i>	<i>44</i>
4	CÁLCULO DE VOLUME E CONTROLE DO INVENTÁRIO	46

5	RESULTADOS	48
5.1	Resultados de Simulação	48
5.1.1	<i>Testes de Operação da Interface</i>	48
5.1.1.1	<i>Teste de conectividade e mensagens de segurança</i>	48
5.1.2	<i>Teste de Mapeamento no Simulador</i>	52
5.2	Testes de Operação em Ambiente Externo	54
5.2.1	<i>Mapeamento do Centro de Tecnologia da UFC</i>	55
5.2.2	<i>Testes de Mapeamento na Termoelétrica</i>	57
6	CONCLUSÕES	63
	REFERÊNCIAS	64

1 INTRODUÇÃO

1.1 Sistema de Mapeamento Aéreo

Em usinas termoelétricas a carvão se faz necessário o controle de inventário da matéria-prima presente nos pátios de armazenamento devido à alta variação do estoque de sua matéria-prima. As discrepâncias na estimação do volume de carvão podem acarretar problemas de operação e eficiência de uma usina. Mesmo utilizando de recursos como balanças dinâmicas em correias transportadoras, tais equipamentos podem não oferecer uma medição precisa do carvão. Diante disto, pesquisadores e indústrias buscam maneiras mais eficazes de se mapear e calcular o volume do estoque de matéria-prima.

Uma maneira de se calcular e estimar a quantidade de carvão é através do método de fotogrametria. Este método consiste na utilização de diversas fotos tiradas por uma câmera, captadas de diversos ângulos, onde juntas formam uma única imagem. A aquisição dessas fotos pode ser feita de forma estacionária ou não, dependendo da aplicação. Porém, este método apresenta uma grande desvantagem que é o longo tempo de processamento das imagens adquiridas (TUCCI *et al.*, 2019).

Uma forma de melhorar o sistema de aquisição de dados é utilizando sensores *Light Detection And Ranging (LiDAR)*. Esses sensores conseguem aferir distâncias através da emissão e recepção de feixes de luz. Através dos pontos obtidos pelo sensor laser é possível se fazer um mapeamento de superfícies através de uma nuvem de pontos tridimensional. Esta nuvem é construída através da fusão sensorial dos dados obtidos pelo sensor *LiDAR* e os dados de posicionamento do sensor. Devido a isto, um sensor *LiDAR* é capaz de construir uma nuvem de pontos tridimensional a partir de um sistema estacionário que apresenta apenas uma variação de ângulo, como também de um sistema móvel, necessitando apenas dos dados de movimentação do sensor.

Os sistemas de mapeamento aéreo utilizam um Veículo Aéreo Não Tripulado (VANT), conhecido em inglês como *Unmanned Aerial Vehicle (UAV)*, com o auxílio de um sensor para sobrevoar o local e fazer um mapeamento da região. Dentre os modelos de drones se destacam os multi-rotor. Eles apresentam uma maior mobilidade e flexibilidade de movimento comparado a VANTs de asa fixa, por exemplo. Com o desenvolvimento dos sistemas de localização presentes nesses VANTs, além de uma maior robustez e capacidade de carga a ser levada, viu-se a possibilidade da utilização de sistemas aéreos para sensoriamento e mapeamento

aplicado diretamente para estimação de volume de carvão (HE *et al.*, 2019).

1.2 Sistema de operação

Diante dos diversos usos e aplicações dos sistemas aéreos, é de fundamental importância que esses sistemas sejam controlados e monitorados de maneira eficiente (ZHANG *et al.*, 2016). Sistemas de teleoperação, como interfaces de controle, podem ser utilizadas para controle de robôs e sistemas autônomos. A interface deve permitir a recepção de dados remotamente e tomadas de decisões através de comandos (FONG; THORPE, 2001). Dependendo do tipo de operação, o sistema de teleoperação pode ser desenvolvido para sistemas móveis através de aplicativos. Isto permite uma maior liberdade do operador durante o seu funcionamento (VALERO *et al.*, 2009).

Com o passar do tempo, houve um aumento notável na demanda de desenvolvedores para criar aplicativos cada vez mais específicos (JAWAD, 2019). Atualmente, diversos sistemas operacionais dominam o mercado de celulares através do mundo. Onde, cada sistema operacional apresenta um Kit de Desenvolvimento de Software (em inglês *Software Development Kit (SDK)*) próprio, possuindo bibliotecas de criação independentes. Além disso, cada sistema operacional apresenta uma linguagem de programação predominante associada (HOLZINGER *et al.*, 2012).

Um dos sistemas operacionais mais utilizados é o sistema *Android*. Esse sistema possui código aberto, pode ser chamado de *open-source*, e de ampla utilização (STRADOLINI *et al.*, 2016). Aplicativos *Android* podem ser escritos tanto em *Kotlin* quanto em *Java* no Ambiente de Desenvolvimento Integrado (do inglês *Integrated Development Environment (IDE)*), como por exemplo *Android Studio*. Sendo a linguagem *Java* uma das mais utilizadas (CHEON, 2019). Isso se dá devido a sua similaridade com *Android* nos quesitos: bibliotecas e Interfaces de Programação de Aplicação (em inglês *Application Programming Interface (API)*). A linguagem de programação *Java* é orientada a objetos, gratuita e de código-fonte aberto, o que facilita o compartilhamento de algoritmos entre desenvolvedores, (DEITEL *et al.*, 2015).

1.3 Trabalhos relacionados

Sistemas que combinam aquisição de dados e geoposicionamento são alvo de estudos e aplicação em diversas áreas. Como exemplo, é possível observar esses sistemas em transportes autônomos (ZHANG *et al.*, 2018) e no mapeamento de áreas urbanas (VACCA *et al.*, 2018).

Essa aquisição é feita por sensores LiDAR, que apresentam a vantagem na aquisição quanto a precisão dos dados adquiridos (KOTB *et al.*, 2018) e podem ser utilizados na aquisição de nuvens de pontos tridimensionais de superfícies (CUI *et al.*, 2019).

A aquisição feita por sensores LiDAR pode ser feita tanto de forma estacionária, abordados em (LU *et al.*, 2011) e (SANTOS *et al.*, 2019), quanto em sistemas aéreos, (SANTOS *et al.*, 2020) e (WALLACE *et al.*, 2012). Esse sistema demonstra ser tão eficiente quanto a estação total (ARANGO; MORALES, 2015), porém, apresentando vantagens quanto ao tempo de aquisição e facilidade do processo (BEMIS *et al.*, 2014).

A indústria de drones tem se desenvolvido e atualmente, diversos modelos são criados visando diversas aplicações. Com por exemplo, tem-se a utilização de VANTs para estudo de estruturas (GUERREIRO *et al.*, 2017), aquisição de dados para uso militar (ORFANUS *et al.*, 2016), construção civil (LIU *et al.*, 2014), dentre outros. Dentre esses modelos de drones se destacam os multi-rotor, que apresentam uma maior mobilidade e flexibilidade de movimento comparado a VANTs de asa fixa, por exemplo. Com o desenvolvimento dos sistemas de localização presentes nesses VANTs, além de uma maior robustez e capacidade de carga a ser levada, viu-se a possibilidade da utilização de sistemas aéreos para sensoriamento e mapeamento aplicado diretamente para estimativa de volume de carvão (HE *et al.*, 2019).

O sistema de aquisição aéreo, VANT mais sensores LiDAR, necessita da correta estimativa de posição e orientação da aeronave para se reconstruir e mapear tridimensionalmente uma região ou ambiente de maneira confiável (CHIANG *et al.*, 2017). Já que erros de medição e diferenças de altitude podem afetar a precisão dos pontos laser (WANG *et al.*, 2018). Assim, uma possível maneira de se obter com precisão a orientação e posição do sistema aéreo é através de sistemas *Real-Time Kinematic (RTK)*. Estes sistemas permitem a correção em tempo real da leitura obtida por satélites, o que garante uma precisão de centímetros (LANGLEY, 1998).

A aplicação de um sistema de mapeamento em uma pilha de carvão também é abordado em (ZHANG *et al.*, 2020). Porém, é um processo que não é prático nem eficiente, pois a aquisição feita em solo com uma pessoa levando o equipamento a pé. A busca em reduzir as diferenças entre os valores presentes nos livros de registros de carvão e no estoque da indústria é abordada em (SMITH *et al.*, 1981). Desde então, novos sistemas e métodos de aquisição são estudados para evitar a propagação de tamanhos erros.

1.4 Problemática e justificativa

A problemática diante do sistema proposto é motivada pela necessidade técnica de controle do estoque de matéria-prima apresentada pelo setor de operação e planejamento da Porto do Pecém Transportadora de Minérios (PPTM) que abastece a Usina Termoeletrica (UTE) Pecém, localizado em São Gonçalo do Amarante - Ceará. A UTE Pecém, da empresa EDP Brasil, apresenta uma capacidade total de 720 MW de potência ativa. Sendo equivalente é uma grande porção do parque elétrico gerador do Ceará.

É necessário um rigoroso controle do estoque para garantir a disponibilidade de operação da UTE Pecém, além de manter o correto planejamento financeiro da empresa e correta declaração nos relatórios de armazenamento e compra de material. A UTE Pecém conta com um pátio de carvão que é dividido em três grandes pilhas para armazenar a matéria prima. Atualmente, a PPTM contrata uma empresa terceirizada para fazer a estimativa de volume através de um sistema de fotogrametria aéreo. Porém, foi constatado que os meios atuais de estimação podem gerar erros de até 40 % no volume do estoque. Essa elevada inconsistência de valores pode ser determinada por diversos fatores, como equipamentos descalibrados e imprecisões na metodologia aplicada.

Esta dissertação foi desenvolvida no âmbito do projeto de P&D Energia Pecém/ANEEL número PD-07267 – 0016/2018. O projeto foi desenvolvido por uma equipe de três professores, três alunos de doutorado e três alunos de mestrado.

1.5 Objetivos

Este trabalho propõe o estudo e desenvolvimento de uma interface amigável para operação de um sistema de monitoramento aéreo para estimação de volume de carvão utilizando ferramentas *open-source* para reconstrução de modelos tridimensionais. Tanto o sistema de mapeamento quanto a interface de operação, visam contribuir para a redução dos erros na estimativa de estoque de carvão, otimizando assim o planejamento e controle do pátio de carvão da PPTM no CTE do Pecém. Onde os principais objetivos são apresentados a seguir:

- a) Desenvolvimento de sistema aéreo de mapeamento tridimensional utilizando sensor LiDAR acoplado em um Veículo Aéreo Não Tripulado;
- b) Desenvolvimento de interface gráfica de operação remota em *Java* para dispositivos móveis *Android* própria para a problemática e de fácil utilização;

- c) Testes de comunicação entre sistema aéreo de mapeamento e interface gráfica móvel;
- d) Testes de mapeamento e operação completa via interface gráfica móvel.

1.6 Organização do texto

O texto será dividido nos seguintes capítulos:

- a) Capítulo 2: apresenta o sistema de mapeamento tridimensional desenvolvido no projeto;
- b) Capítulo 3: apresenta a interface gráfica desenvolvida para operação do sistema de aquisição;
- c) Capítulo 4: apresenta brevemente a interface para controle do inventário;
- d) Capítulo 5: apresenta os testes e resultados feitos para validação do sistema desenvolvido e operação da interface;
- e) Capítulo 6: apresenta as conclusões obtidas pelo trabalho.

2 SISTEMA AÉREO DE MAPEAMENTO TRIDIMENSIONAL

O sistema de mapeamento desenvolvido é composto de três módulos, ou partes, que são apresentadas na Figura 1. A primeira etapa consiste no sistema aéreo desenvolvido e o processo de varredura da pilha de carvão. O segundo módulo consiste na interface gráfica móvel e operação do sistema em campo em tempo real. E por fim, a última etapa consiste no processamento de dados e controle de inventário.

Figura 1 – Diagrama Operacional do Sistema de Mapeamento.



Fonte: Próprio autor.

O VANT opera de forma automática utilizando um algoritmo de seguimento de trajetória seguindo pontos georreferenciados predefinidos. A aquisição de dados através do sensor laser também é feita de forma automática e utiliza destes pontos predefinidos como referência para início e fim da aquisição. Todo o processo de decolagem, mapeamento e pouso no ponto de partida também é feito de forma autônoma. Este processo é iniciado através da interface móvel de operação e detalhado na Seção 3.

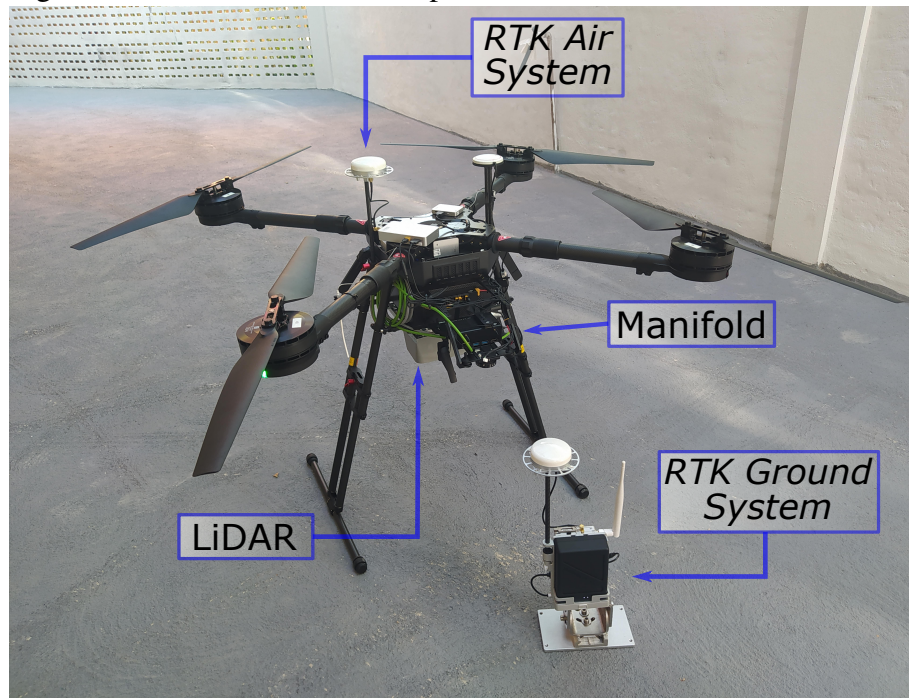
2.1 Hardware do Sistema de Mapeamento

Iniciando no primeiro módulo de operação do sistema desenvolvido, têm-se os equipamentos utilizados no sistema aéreo de mapeamento 3D, Figura 2. A descrição dos componentes de forma detalhada é apresentada nas subseções a seguir.

2.1.1 Drone Wind 4

O multi-rotor é um dos equipamentos fundamentais para o sistema de mapeamento aéreo. Por essa razão, foi-se escolhido um VANT industrial com grande robustez e compati-

Figura 2 – Sistema Aéreo de Mapeamento 3D.



Fonte: Próprio autor.

dade com diversos outros equipamentos industriais. O drone adotado é o *DJI Wind 4*, Figura 3, da empresa chinesa *Shenzhen DJI Sciences and Technologies Ltd.* A empresa se destaca no ramo com o domínio de aproximadamente 70% do mercado mundial de drones (JOSHI, 2019).

Figura 3 – Teste de voo do VANT *Wind 4*.

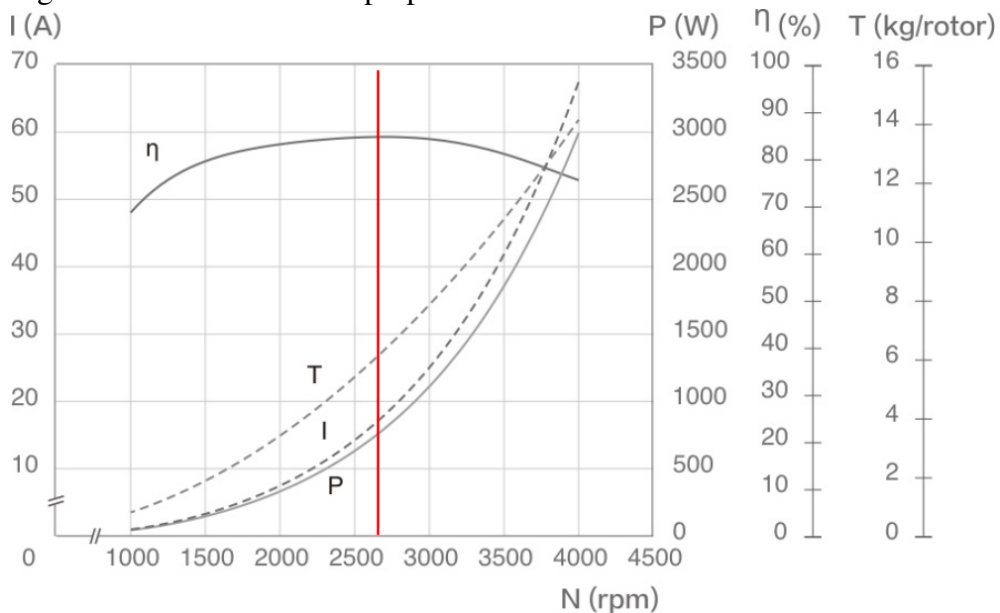


Fonte: Próprio autor.

Outra vantagem do equipamento escolhido é a bateria inteligente *DZ-12000*. Essa bateria apresenta LEDs para indicar o nível de carga da mesma, além ser capaz de mandar informações em tempo real para o computador de bordo da aeronave. Dentre as informações enviadas tem-se os valores de tensão, corrente, número de cargas, além de um histórico de possíveis erros da bateria, o que facilita na hora de identificar problemas do sistema aéreo. A bateria apresenta capacidade de 12000 *mAh* e opera com tensão de 22.2 *V*, o que garante aproximadamente 22 minutos de *hovering*, ou seja, planando.

Esta bateria, além de alimentar o sistema de controle do VANT e o sensor LiDAR, alimenta os 4 rotores M10 Pro, do sistema de propulsão E5000. Esses rotores, cujas curvas de performance são apresentadas na Figura 4, apresentam velocidade média de operação de aproximadamente 2660 *rpm*, representado pela linha em vermelho. Na Figura 4, *I* representa a corrente em amperes, *P* a potência de entrada, η a eficiência, *T* o empuxo ou propulsão e *N* é velocidade de rotação dos motores.

Figura 4 – Performance de propulsão do motor M10 Pro.



Fonte: (DJI, 2016)

O VANT *Wind 4* apresenta capacidade de carregar até 10 *Kg* de carga útil. O que é suficiente para transportar todos os equipamentos necessários para o sistema de mapeamento. Além de apresentar plataformas extras de expansão que permitem o acoplamento de outros equipamentos industriais, como o sensor LiDAR.

Por fim, o VANT *Wind 4* apresenta braços dobráveis e ajuste na altura dos pés de sustentação. O que permite acomodar o sistema aéreo em espaços menores dentro da indústria,

já que ele apresenta aproximadamente 1,05 m de envergadura sem considerar o tamanho das pás. Além de facilitar o transporte do mesmo na hora de usar em campo e guardar em local abrigado.

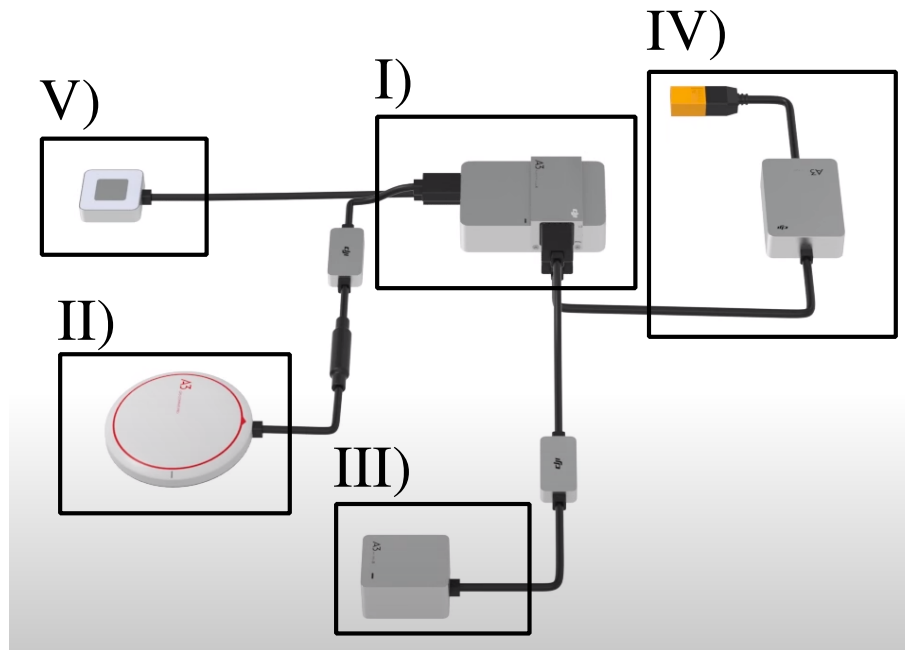
2.1.1.1 Controlador de voo DJI A3

Um destaque importante para o *Wind 4* é a presença do controlador de voo DJI A3. O seu robusto sistema de controle é fundamental na hora de manter a estabilidade do sistema aéreo durante o seu uso. Isso é possível devido aos avançados algoritmos de fusão sensorial presentes nele e nas leituras obtidas pelos sensores inerciais presente em um de seus módulos. Este controlador é capaz de manter comunicação com o sistema *DJI Lightbridge 2*, apresentado na Seção 3.1.1. Além de compatibilidade com o sistema de posicionamento via *Global Positioning System (GPS)* e D-RTK, apresentado na Subseção 2.1.3.

O Controlador de voo é composto por cinco módulos, apresentados na Figura 5. I) Apresenta o controlador de voo propriamente dito, contendo as portas de comunicação. Dentre essas portas estão presentes a entradas de comunicação com os controladores de velocidade (*Electronic Speed Control (ESC)*) da aeronave, portas de comunicação com os outros módulos e portas independentes e configuráveis. II) Módulo GPS para medidas de posicionamento. III) Módulo com *Inertial Measurement Unit (IMU)*, necessário para cálculo e medição da altitude do sistema. IV) Módulo de *Power Management Unit (PMU)* para alimentação de todo o sistema de controle de voo. V) Módulo de LED, para informações visuais durante o voo.

Após os componentes do controlador serem devidamente conectados e instalados dentro do VANT ele está pronto para ser configurado. Este processo pode ser feito através do *Software DJI Assistant 2*, disponibilizado gratuitamente pelo fabricante. Com este *Software* é possível fazer testes de simulação de voo do sistema aéreo e parametrização de ajustes de controle da aeronave. Durante toda a etapa de simulação uma aba com informações do sistema aéreo é apresentada. Dentre estas informações presentes, têm-se a velocidade e aceleração dos três eixos, ângulos de *roll*, *pitch* e *yall* além das posições de latitude e longitude, assim como a distância percorrida em metros a partir do ponto de origem. Também é possível adicionar um traçado com o caminho percorrido pelo VANT, uma ferramenta que auxilia na observação do modelo 3D durante missões de voo.

Figura 5 – Esquemático do controlador de voo A3



Fonte: (DJI, 2017a)

2.1.2 Sensor LiDAR LD-MRS

Outro equipamento de suma importância para o sistema de mapeamento aéreo é o sensor LiDAR. O LiDAR é um sensor laser que através da emissão e recepção dos feixes de luz que emite é capaz de medir distâncias. Sendo um sensor que virou foco de estudo e melhorias de implementação devido ao seu uso em veículos autônomos (CHESWORTH; HUDDLESTON, 2018). O que resultou em uma baixa no seu preço (DELEÓN *et al.*, 2020).

O LiDAR em questão utilizado no projeto é o modelo LD-MRS 420201 da empresa SICK, Figura 6. Sendo esta empresa uma das principais fabricantes de sensores para aplicações na indústria (SICK, 2020). Este sensor é acoplado na parte inferior ao VANT com o seu campo de visão voltado para o solo, como visto na Figura 2.

O sensor adotado é direcionado para aplicações em ambientes externos, possuindo 4 faixas de laser com capacidade máxima de leitura de até 300 m, como visto na Figura 7. O ângulo de abertura de é 85° para implementações com 4 faixas de laser e de 110° para 2 faixas de laser.

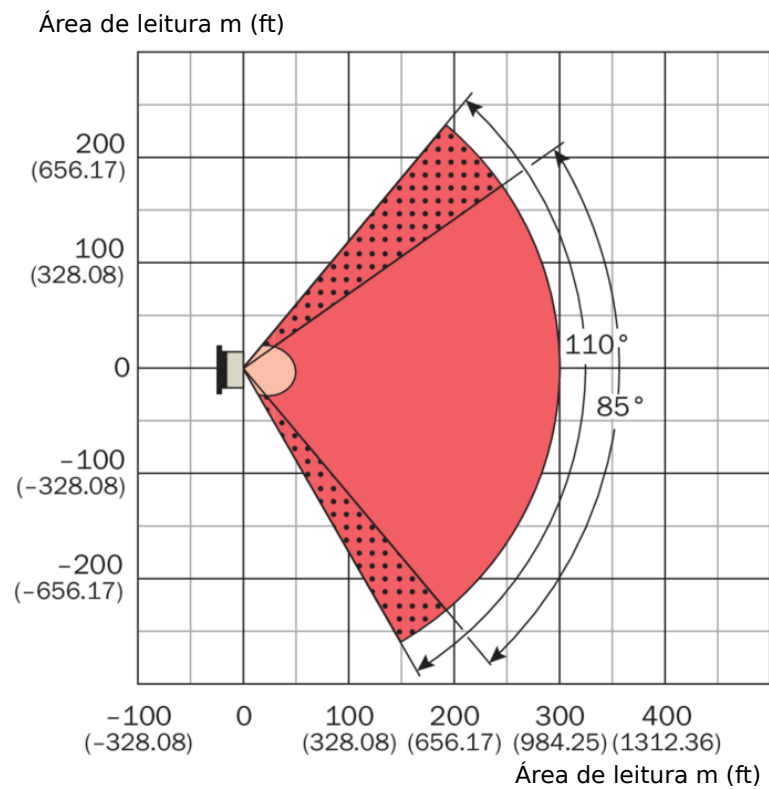
Apesar do sensor LiDAR ser capaz de operar de certa forma independente da iluminação local, (Van Genderen, 2010), para esta aplicação, o sensor LiDAR deve se manter próximo do material de leitura devido ao baixo nível de remissão de luz do carvão. Sendo assim, é ideal que o sistema não ultrapasse uma altura de 50 m da pilha de carvão para garantir que não

Figura 6 – Sensor LiDAR LD-MRS 420201.



Fonte: Próprio autor.

Figura 7 – Região de leitura do sensor LD-MRS 420201.



- Região máx. 300 m (984,25 ft)
- Região para objetos com remissão de 10% 50 m (164,04 ft)
- Leitura com 2 feixes
- Leitura com 4 feixes

Fonte: Adaptado de: (SICK, 2020)

haja problemas de performance por parte do sensor.

O LiDAR LD-MRS 420201 pode ser alimentado com tensão entre 9 V e 27 V consumindo 8 W de energia. Apresenta índice de proteção IP69K, ou seja, é à prova de poeira e protegido contra água proveniente de jatos de vapor e alta pressão. O modelo de sensor utilizado apresenta frequência de varredura entre 12,5 Hz e 50 Hz. Esta frequência de aquisição pode ser ajustada antes da operação com o sensor. Onde são emitidos 440 pulsos de luz com o sensor ajustado em uma frequência de 12,5 Hz e 880 pulsos de luz em uma frequência de 50 Hz.

2.1.3 Sistema D-RTK

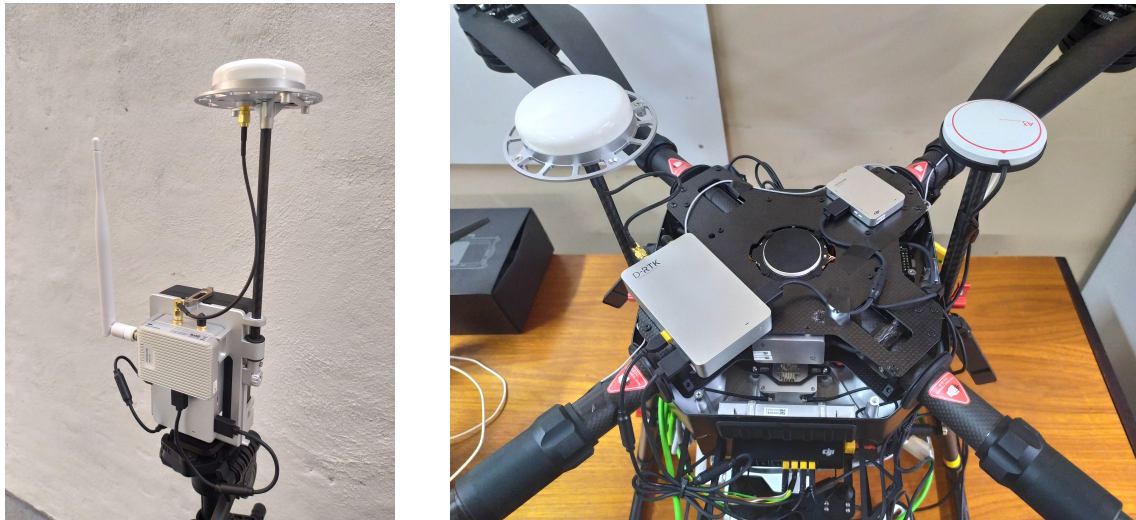
Outra parte importante para o sistema de mapeamento aéreo 3D é a correta estimativa de posição do VANT durante o voo. O sistema necessita dessa informação de forma precisa para obter uma nuvem de pontos fiel a região mapeada. Um sistema RTK consiste de dois receptores. Um receptor é estacionário, normalmente sendo uma estação fixa em solo, e o outro é móvel, acoplado no objeto que se quer medir a posição.

Pensando nessa necessidade da correta estimativa de posição a DJI desenvolveu um sistema de localização RTK próprio, conhecido como D-RTK. O equipamento de alta precisão e foi desenvolvido para ter total compatibilidade com o controlador A3. Além de apresentar um sistema de antenas que não é afetado por interferências magnéticas de estruturas metálicas (DJI, 2017b). Um experimento observando a precisão desse sistema RTK comparado ao do GPS para a reconstrução de nuvens de pontos 3D é apresentado em (SANTOS *et al.*, 2020).

Este sistema D-RTK consiste de dois módulos o *Ground Station*, Figura 8a), e o *Air Station*, Figura 8b). O primeiro representa a estação base a qual faz comunicação em tempo real com o VANT e com os satélites presentes na região. Tem como função fazer os cálculos para correção do posicionamento do sistema aéreo. O *Air Station* é montado no *Wind 4* e conectado diretamente ao controlador de voo A3, auxiliando na leitura de posição do mesmo.

O *Ground System* é composto do *DJI Datalink Pro 900*, uma antena para recepção do sinal e uma bateria inteligente. Este conjunto de itens é normalmente integrado a um tripé ou base de sustentação para manter fixo no chão. Já o *Air System* contém uma antena *Global Navigation Satellite System (GNSS)*, um processador D-RTK o *Datalink Pro Mobile Station*, todos acoplados no VANT.

Figura 8 – Sistema D-RTK completo.



a) *Ground System*

b) *Air System*

Fonte: Próprio autor.

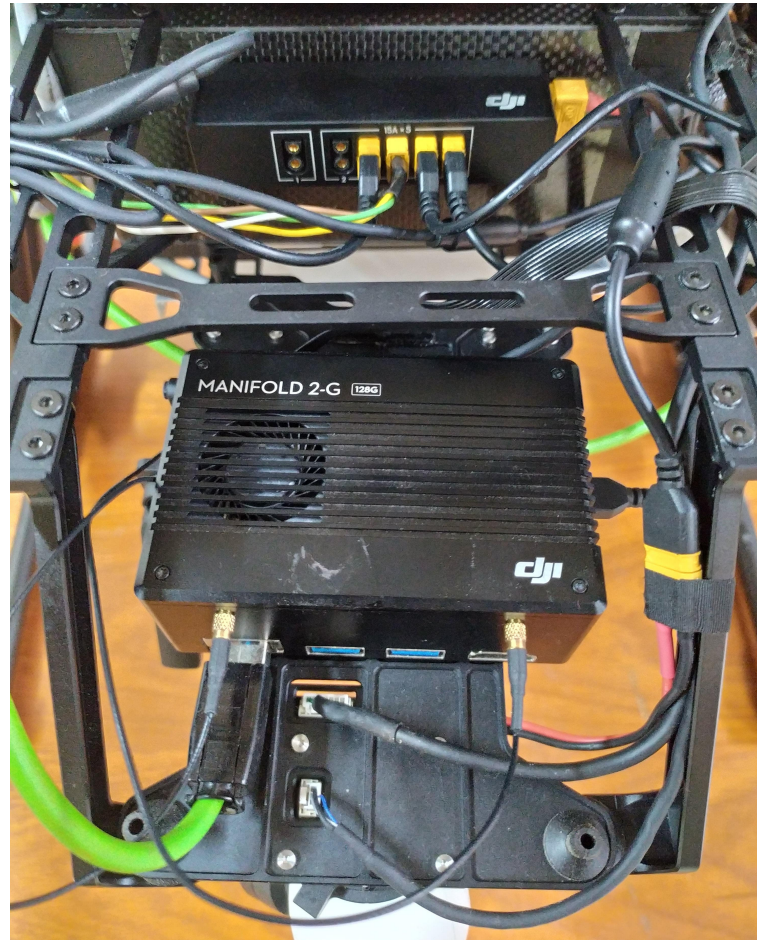
2.1.4 *Computador de Bordo Manifold*

O último item embarcado no sistema aéreo é um microcomputador. O computador embarcado tem como finalidade processar os dados obtidos pelos sensores durante um voo de aquisição e gerar uma nuvem de pontos resultante. Com este propósito em mente, foi escolhido o microcomputador *Manifold 2-G*, também da empresa DJI, Figura 9.

O *Manifold 2-G* possui um processador *NVIDIA Jetson TX2*. Este processador de alta potência garante um ótimo desempenho e velocidade no processamento paralelo de imagens e nuvem de pontos. Contém 8 *Gb* de memória RAM, 32 *Gb* de memória eMMC padrão 5.1, e 128*Gb* de memória de disco SSD. O microcomputador possui entrada Ethernet e capacidade de conexão via Wi-Fi. Além de 2 portas USB 3.0, 1 porta USB 3.0 micro-B, 2 portas CAN, 2 portas UART.

Os dados obtidos pelo sensor LiDAR são transmitidos pela entrada Ethernet para a *Manifold*. Através dele que a nuvem de pontos resultante é montada e filtros são aplicados. O processo é feito unindo as informações obtidas pelos sensores inerciais e de localização, junto dos pontos obtidos pelo LiDAR. Tudo isso é possível devido a presença do sistema operacional base Ubuntu 16.04. Assim, a construção da nuvem de pontos é feita através do *framework* de robótica *Robot Operating System (ROS)*, apresentado e explicado na Seção 2.2.

Figura 9 – Manifold 2-G conectada ao VANT.



Fonte: Próprio autor.

2.1.4.1 Sistema de desenvolvimento OSDK

Uma das vantagens de se utilizar o controlador de voo A3 junto de um computador embarcado com ROS é a versatilidade de aplicação devido a presença do *Onboard Software Development Kit (OSDK)*. Esse sistema junto do *Mobile Software Development Kit (MSDK)*, apresentado na Seção 3.2, permitem a criação de aplicativos e sistemas customizáveis próprios. Possibilitando desenvolver sistemas específicos e variados de acordo com a carga utilizada no VANT e na aplicação escolhida para o mesmo.

A comunicação entre o controlador de voo A3 e o computador de bordo ocorre pela porta serial UART1. Essa comunicação é feita pelo sistema *OSDK* desenvolvido presente no *Manifold*. Por ele é possível obter e mandar informações do controlador A3 que podem ser utilizadas de acordo com a necessidade do operador.

Através do sistema *OSDK* é possível montar missões de voo e enviá-las ao controlador A3. Essas missões são criadas através de pontos georreferenciados, os *Waypoints*. Por estes

pontos o controlador de voo cria uma trajetória que deverá ser seguida pelo VANT de forma automática. Os *Waypoints* também servem como referência para iniciar e parar a aquisição de dados feitas pelo sensor LiDAR também de forma automática. Informações como altitude e velocidade de voo também são informadas nas missões de voos presente no sistema *OSDK* e enviadas para o controlador A3.

Os *waypoints* presentes nas missões de voo devem ser escolhidos após uma análise do local para mapeamento. Deve ser levado em consideração o ângulo de abertura do sensor LiDAR e a segurança do equipamento no local. Com isso, é possível determinar uma altura ideal para o voo.

A plataforma *OSDK* também permite a comunicação com o controle remoto *Light-bridge 2*. Essa comunicação é feita mediante o controlador A3 e é de fundamental importância para completa integração com o sistema *MSDK* presente no dispositivo móvel conectado ao controle remoto do VANT.

2.2 Software do Sistema de Mapeamento

Com explicado anteriormente, no computador embarcado no sistema de mapeamento está presente o sistema operacional Ubuntu 16.04. Através dele é utilizado o *framework* de robótica *ROS*, onde foi desenvolvido um sistema para construção da nuvem de pontos 3D.

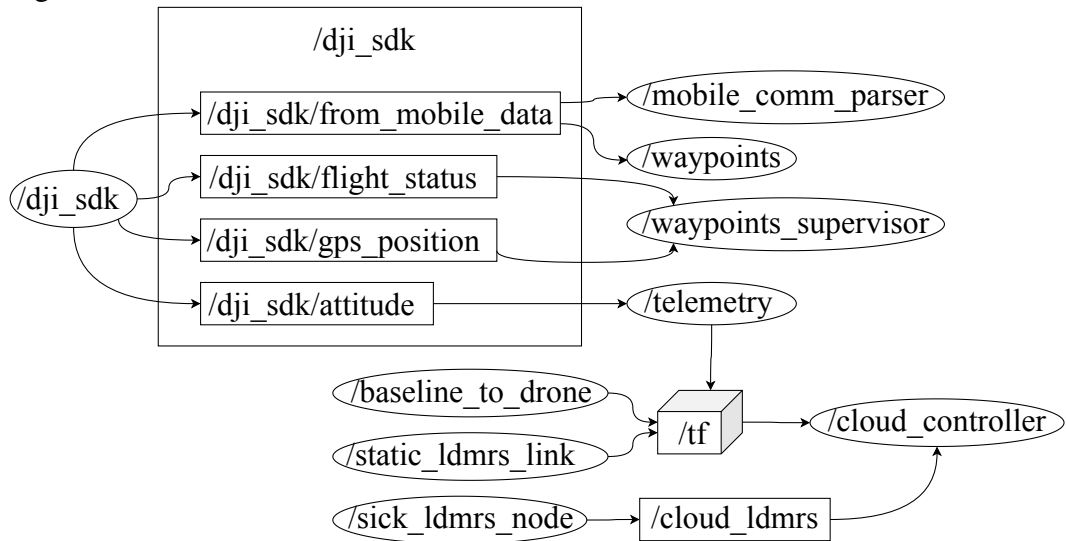
2.2.1 ROS

Sistemas *ROS* apresentam diversas aplicações devido ao seu ambiente colaborativo, modular e com diversas bibliotecas que permitem comunicação com sistemas de *hardware* e robótica. Assim, é possível desenvolver sistemas complexos e específicos de robótica através desta ferramenta, (QUIGLEY *et al.*, 2009).

Sistemas desenvolvidos em *ROS* são compostos e organizados em “nós”, que são módulos que processam ou calculam informações. Esses “nós” comunicam entre si através de mensagens, ou dados, publicadas em “tópicos”. Os “tópicos” podem ser lidos por outros “nós” que tenham interesse no que é transmitido por ele. O sistema desenvolvido para o projeto é apresentado na Figura 10. Os “nós” são apresentados nas elipses e os “tópicos” são representados em retângulos.

O pacote */dji_sdk* permite que o usuário consiga controlar informações de VANTs

Figura 10 – Sistema desenvolvido em ROS.



Fonte: Próprio autor.

que tenham o controlador de voo A3. Os “tópicos” que foram utilizados são apresentados a seguir:

- */dji_sdk/from_mobile_data* contém os dados recebidos do dispositivo móvel e envia para o VANT. Dentre esses dados estão os *waypoints* da missão de voo escolhido pelo operador e que são lidos pelo nó */waypoints*;
- */dji_sdk/flight_status* apresentam os *status* de voo do VANT em uma frequência de 50 Hz;
- */dji_sdk/gps_position* fornece a posição global do VANT;
- */dji_sdk/attitude* fornece a estimativa de posição a partir da fusão de dados inerciais;
- */waypoint_supervisor* calcula o progresso da missão e inicia automaticamente a aquisição via LiDAR no primeiro ponto definido no plano de voo. Ao chegar no último *waypoint* definido da missão a aquisição é parada e os dados são salvos;

A construção da nuvem de pontos é feita a partir dos “nós” e “tópicos” apresentados a seguir. Os dados do “tópico” */dji_sdk/attitude* apresenta os dados de posição tridimensional a partir da fusão de dados entre os sensores inerciais e os sensores de posicionamento GPS e D-RTK. Essa posição tridimensional está referenciada em um quadro de referência e os dados obtidos pelo sensor LiDAR, nó “tópico” */cloud_ldmrs*, em outro quadro. O processo completo é apresentado na subseção a seguir.

- */telemetry* combina os dados de */dji_sdk/attitude* e de */dji_sdk/gps_position* em matrizes de transformação entre quadros */tf*;
- */sick_ldmrs_node* publica os dados do LiDAR na forma de nuvem de pontos em relação ao quadro do sensor através do tópico */cloud_ldmrs*;

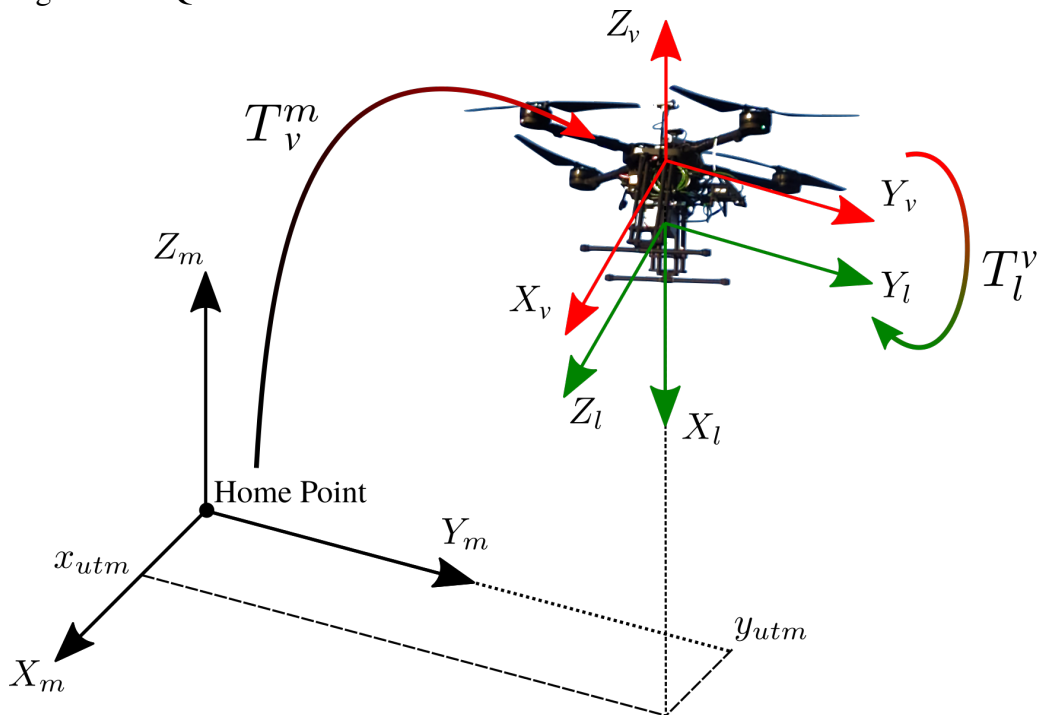
- */cloud_controller* de forma geral ele combina todas as transformações para construir uma única nuvem de pontos resultante do ambiente mapeado;

2.2.1.1 Construção da nuvem de pontos

Previamente foi falado da importância de saber o posicionamento exato do sistema de mapeamento aéreo. Esta informação é relevante para a construção da nuvem de pontos devido a transformação entre quadros de referência. Este processo de transformação é calculado através de cálculos de rotação e translação entre quadros. Para isto, é necessário se definir os quadros, ou *frames*, de referência que serão utilizados no processo e qual será utilizado na construção da nuvem de pontos tridimensional.

Para o sistema de mapeamento tridimensional desenvolvido foram definidos três quadros de referência, apresentados na Figura 11. São eles: o quadro *mapa*, o quadro *VANT* e o quadro *LiDAR*. O quadro *mapa* tem origem no ponto de decolagem (*Home Point*) do VANT e é representado pelo subscrito *m*. Esse ponto é georreferenciado no momento que se dá início a missão de mapeamento. O *frame* do VANT, subscrito *v*, é definido pela fusão de dados dos sensores inerciais do controlador A3 junto do sistema D-RTK. Por último, o quadro *LiDAR*, subscrito *l* é definido alguns centímetros abaixo do quadro *v* localizado no centro do sensor.

Figura 11 – Quadros de referência adotados.



Fonte: Próprio autor.

O processo de transformação entre quadros deve ocorrer de maneira a levar os pontos obtidos pelo LiDAR, referenciados no quadro l , para o quadro do *mapa*. Esta etapa, pode ser representada por uma matriz de transformação T contendo uma matriz de rotação R e uma matriz de translação A . A Equação 2.1 apresenta matriz de Transformação entre os quadros genéricos a e b como exemplo, onde a seria o *frame* de origem e b o *frame* final,

$$T_b^a = \begin{bmatrix} R_{3 \times 3} & A_{3 \times 1} \\ 0_{1 \times 3} & 1_{1 \times 1} \end{bmatrix}. \quad (2.1)$$

Na Figura 11 também é apresentado as duas transformações feitas no sistema, sendo elas T_l^v e T_v^m . Com isso, a nuvem de pontos resultante é construída no quadro de referência m , sendo necessário que todos os 3 quadros sejam devidamente georreferenciados para uma reconstrução fidedigna do modelo tridimensional.

Durante o período de aquisição da nuvem os dados vão sendo salvos na memória RAM da *Manifold*. Após a missão de mapeamento ser concluída é que a nuvem é gravada no disco de memória do computador embarcada. Junto da nuvem de pontos é criado um arquivo com informações referentes a missão de voo com dados do sistema aéreo.

3 INTERFACE GRÁFICA E SISTEMA DE COMUNICAÇÃO

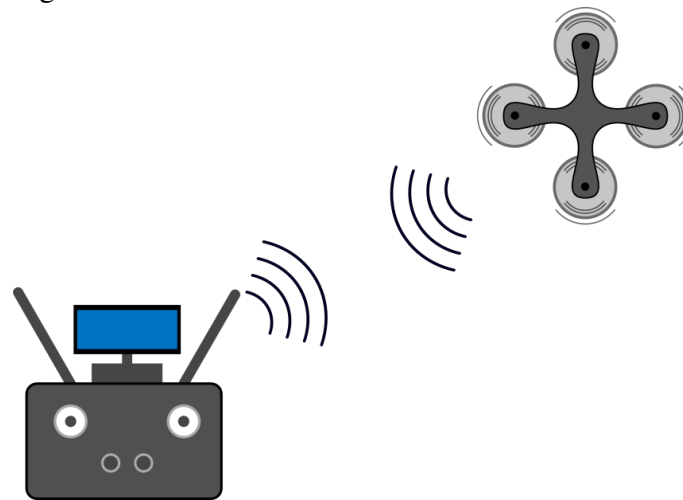
Seguindo para a segunda etapa apresentada na Figura 1, têm-se a apresentação da interface gráfica de operação e o sistema de comunicação entre o VANT e o operador em solo. Este sistema visa permitir a operação em tempo real do VANT com poucos comandos.

3.1 Comunicação entre VANT e Operador - Telemetria

O processo de operação a distância, ou remota, sem fio apresenta diversas aplicações e consiste no envio e recepção de comandos e medidas através de ondas de rádios ou sinais via satélite, (ARGYROUDIS *et al.*, 1998). Frequentemente conhecida como telemetria, este sistema possibilita criar um link direto entre o sistema analisado e o operador. O que viabiliza o controle de um sistema de forma segura e eficiente mesmo sem a necessidade de contato visual com o mesmo.

Em sistemas de robótica, tanto aéreo quanto terrestres, este sistema de telemetria é representado pelos controles remotos, Figura 12. Onde o sistema de controle por muitas vezes é feito de forma *wireless* através de ondas de rádio. Através dele é possível controlar velocidade, orientação e outras funções presentes no controlador interno ao sistema robótico. Quanto mais complexo o sistema mais informações de controle e operação é necessário na mão de um operador capacitado.

Figura 12 – Link de telemetria via rádio.

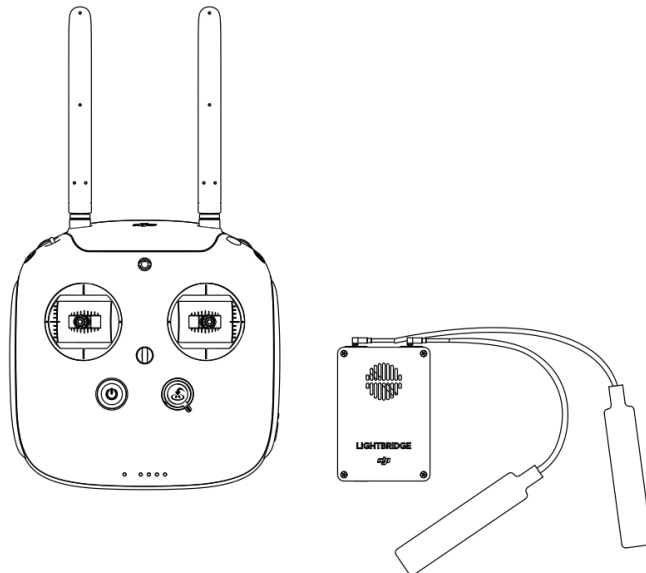


Fonte: Próprio autor.

3.1.1 Lightbridge 2

O sistema de telemetria presente no VANT *Wind 4* é o *Lightbridge 2* da DJI. O *DJI Lightbridge 2* é capaz de manter comunicação a longas distâncias, até 5 Km, e opera com frequência entre 5,725 GHz e 5,825 GHz. Sendo capaz de fazer transmissão de vídeo em 1080p60, (DJI, 2017c). O sistema de telemetria é composto por dois módulos, um terrestre e um aéreo, Figura 13. Através dele, o VANT tem capacidade de comunicação simultânea com mais um módulo terrestre.

Figura 13 – Sistema *Lightbridge 2* - Na esquerda o módulo terrestre e na direita o módulo aéreo.



Fonte: Manual do usuário. (DJI, 2017c)

O módulo aéreo é composto de um receptor embarcado ao VANT. Este receptor apresenta entrada para duas antenas para recepção do sinal do sistema terrestre. Apresenta entrada para conexão com o controlador de voo A3, esta ligação é essencial para o devido controle do sistema aéreo através do controle remoto. O módulo aéreo também apresenta porta de comunicação para sistemas de captação de imagem que podem ser embarcados no sistema aéreo.

O módulo terrestre é o próprio controle remoto utilizado pelo operador para controle manual do VANT. O controle apresenta duas antenas para comunicação com o sistema aéreo. Contém dois *joysticks* para controle de movimentação do VANT e botão para retornar automaticamente ao ponto de decolagem. Apresenta também botões para controle do sistema de aquisição de imagens, caso presente no VANT. Entrada USB para conexão de aparelho móvel

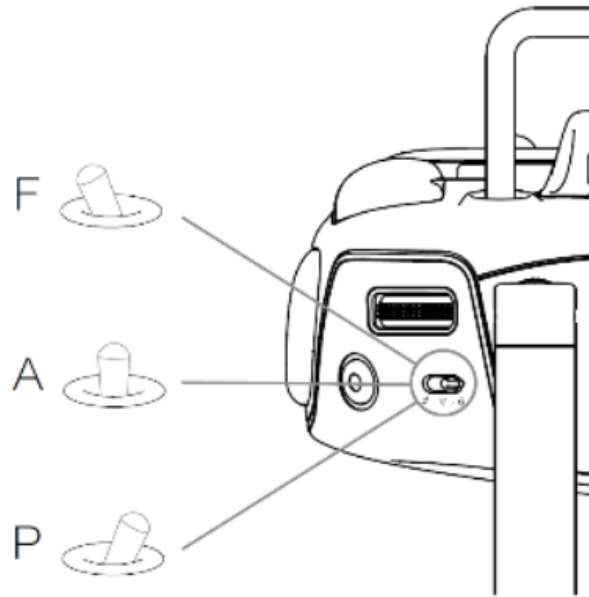
com aplicativo de operação e um suporte para dispositivos móveis. O controle também conta com um conjunto de quatro LEDs para indicar o nível de bateria do controle remoto e um LED para indicar o status de conectividade entre o sistema aéreo e o terrestre do *Lightbridge*.

O sistema *Lightbridge 2* apresenta uma chave para escolha do modo de voo da aeronave. Esta chave determina quais ferramentas a aeronave usa para manter o posicionamento, além de determinar se a missão de voo automática está ativa. Assim, a chave consegue selecionar quais sensores de posicionamento serão utilizados entre três modos distintos. A Figura 14 apresenta o esquemático do controle e da chave de seleção do modo de voo e as características de cada modo de voo é apresentado a seguir.

- **Modo P (*Positioning*):** O VANT utiliza todos os sistemas de posicionamento presente na aeronave durante os voos autônomos. Entre eles está o GPS, barômetro, sistema visual de posicionamento e sistema de desvio de obstáculos, caso presentes no sistema aéreo. Este modo também permite o uso de missões de voo autônomo;
- **Modo A (*Attitude*):** O VANT utiliza apenas o barômetro para obter a altitude. Normalmente utilizado para reaver o controle manual durante um voo autônomo. Neste modo, o sinal do GPS é utilizado apenas quando é acionado o botão para retornar ao ponto de partida;
- **Modo F (*Function*):** Este modo opera semelhante ao modo P, utilizando todos os sistemas de posicionamento, porém, este modo pode ser configurado utilizando o software *DJI Assistant 2*;

O *Firmware* do módulo aéreo do *Lightbridge 2* pode ser atualizado através do *Software DJI Assistant 2*. Através dele tanto o controlador de voo quanto o sistema de telemetria podem ser atualizados. Além de ser possível fazer testes de comando e operação do sistema no simulador de voo presente neste *Software*. A DJI também oferece gratuitamente o aplicativo *DJI Go*. Por ele é possível operar os equipamentos da DJI através de um dispositivo móvel conectado ao controle remoto via USB. Com ele se tem diversas informações diretamente da aeronave que são captadas através do link de telemetria. Porém, pensando nas diversas possibilidades de aplicação que suas aeronaves e controladores de voo apresentam, a DJI desenvolveu o *MSDK*, apresentado a seguir.

Figura 14 – Chave de seleção do modo de voo.



Fonte: Manual do usuário. (DJI, 2017c)

3.2 *Mobile Software Development Kit - MSDK*

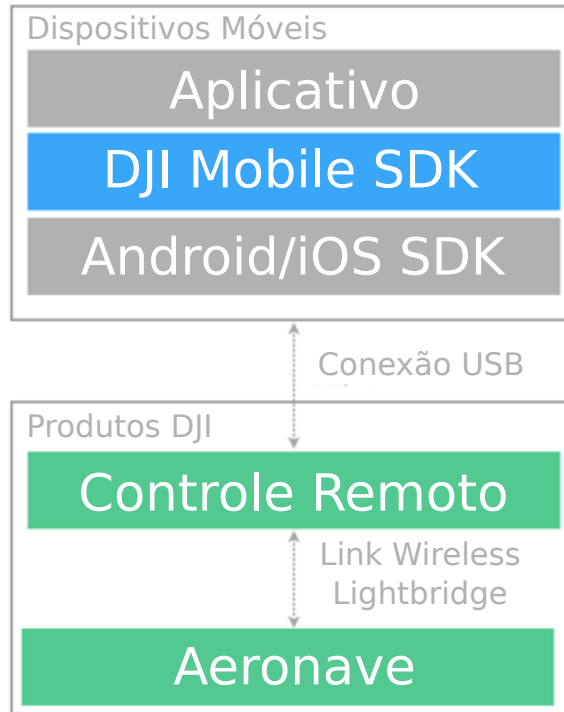
O kit de desenvolvimento de software para sistemas móveis, ou MSDK, da DJI foi projetado para oferecer aos desenvolvedores as mais diversas funcionalidades presentes nos produtos da DJI. Este SDK facilita o processo de desenvolvimento de aplicativos trazendo as informações de maneira simples e direta. Isto permite um foco maior na aplicação sem necessidade de muito conhecimento técnico sobre robótica e sistemas embarcados (DJI, 2020).

O MSDK inclui bibliotecas que podem ser importadas em aplicativos tanto para *Android* quanto para *iOS*. Essas bibliotecas permitem monitorar o estado dos componentes presentes no VANT e de todas as funcionalidades presentes no controlador de voo. Esse sistema SDK também conta com a compatibilidade de informações presentes no simulador de voo do *DJI Assistant 2*. Além de apresentar documentação completa de todas as informações possíveis a serem obtidas pelo SDK no site da DJI destinada aos desenvolvedores. O site também apresenta códigos e tutorias que podem ser testados de maneira rápida dentro do simulador de voo ou em campo junto ao VANT.

Devido a este conjunto de funcionalidades e informações presentes no MSDK, desenvolvedores conseguem projetar aplicativos para dispositivos móveis que utilizem destes dados. Onde o aplicativo utiliza aquilo que for de interesse para a aplicação planejada. O esquemático geral de como o sistema MSDK opera e se conecta com o VANT é apresentado na

Figura 15.

Figura 15 – Diagrama de utilização MSDK e conectividade com o VANT.



Fonte: Adaptado de: (DJI, 2020)

3.2.1 Aplicativo Android Desenvolvido

Analisando os objetivos gerais do sistema de mapeamento elaborado, foi desenvolvido um aplicativo para dispositivos móveis que opera de maneira simples e direta. O aplicativo busca ter uma interface gráfica amigável desenvolvida em *Java* para dispositivos *Android* projetada no *Android Studio*.

O *Software Android Studio* é atualmente uma das melhores ferramentas de desenvolvimento para *Android*. Apresenta como diferencial um editor de *layout* visual, que permite desenvolver uma interface apenas arrastando os elementos de Interface do Usuário (ou *User Interface (UI)*) sobre o *layout*. O arquivo de *layout XML* permite o desenvolvimento da interface tanto escrevendo o código, quanto arrastando e configurando os elementos de UI diretamente. Também apresenta um editor de código inteligente, que complementa partes do código em *Java*, tornando o processo de criação algo rápido.

O aplicativo desenvolvido pode ser dividido e apresentado em duas partes principais. A primeira consiste da classe *Activity*, representa a tela inicial do aplicativo contendo os métodos

e *callbacks* da parte principal do aplicativo. A segunda parte consiste da interface gráfica que permite a interação com o usuário.

3.2.1.1 *MainActivity*

O aplicativo desenvolvido para o projeto foi construído a partir de um modelo de demonstração presente na documentação do MSDK da DJI. A partir deste modelo as configurações e itens presentes foram alterados de acordo com as necessidades do projeto. As primeiras configurações alteradas estão no arquivo *build.gradle (Module:app)*. Através dele é possível de alterar a versão SDK do *Android* que será utilizada no projeto. Onde foi adotada a versão 27 que apresenta otimização no uso de memória RAM do dispositivo. Além de novos recursos relacionados a parte visual do aplicativo. Entre as dependências presentes no arquivo *gradle*, tem-se a implementação sistema SDK da DJI escolhido. Onde, a versão 4.10 do SDK da DJI apresenta todas as ferramentas utilizadas para comunicação com o VANT.

Após a configuração do *gradle* é possível focar na classe principal do aplicativo, denominada de *MainActivity*. Através dessa classe é feito o registro de informações presentes no manifesto, conhecido como *AndroidManifest*, e gerenciar os ciclos de vida, ou estados, presentes na atividade.

Primeiramente, no *AndroidManifest*, são apresentadas as permissões de acesso para partes protegidas do sistema que o aplicativo necessita para operação, dentre elas o acesso a internet. Também são apresentadas as características do aplicativo, como por exemplo o nome escolhido para o aplicativo e o ícone que irá aparecer no dispositivo móvel, Figura 16. O *Manifest* também apresenta o conjunto de chaves, ou *API_KEY*, que servem para identificar o aplicativo desenvolvido. Para o projeto é necessário duas dessas chaves. Uma para o uso do SDK da DJI e outra para uso das funções do GoogleMaps. Ambas obtidas de forma gratuita. Por fim, no *AndroidManifest* também são declaradas todas as *Activities* presentes no aplicativo. Por ela, é definida a *MainActivity* do projeto, onde configurações particulares podem ser definidas. Como por exemplo, o uso da entrada USB do dispositivo.

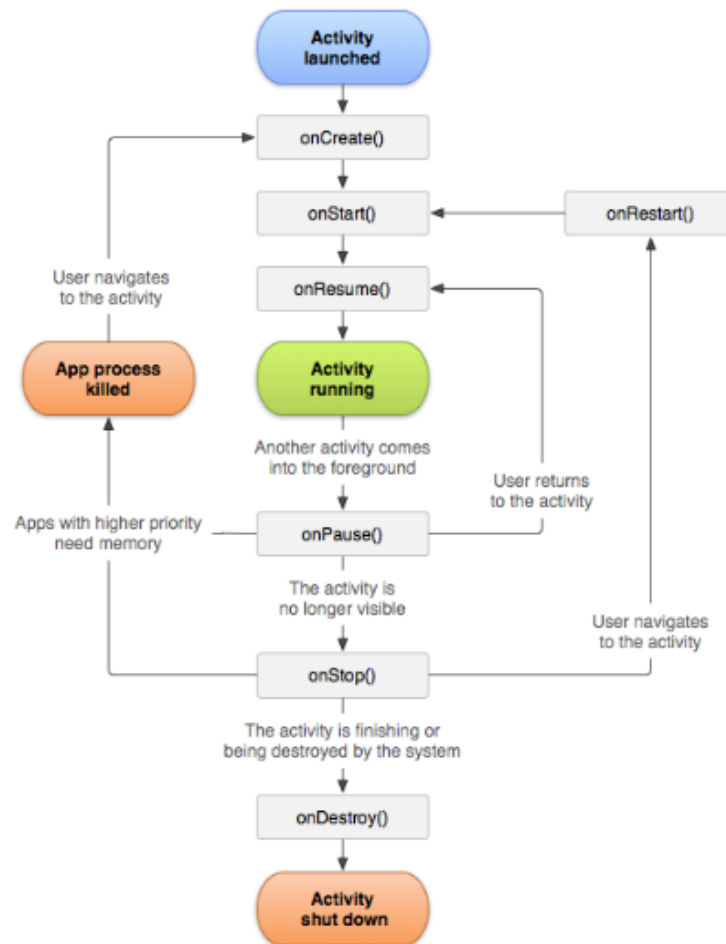
Após definir as configurações da *MainActivity* é possível desenvolver o restante do aplicativo dentro dos ciclos de vida da atividade. Para lidar com cada estado da atividade são utilizados alguns *callbacks* definidos, que operam sempre que a atividade muda de estado, Figura 17. Os estados de operação são: *onCreate()*, *onStart()*, *onResume()*, *onPause()*, *onStop()*, *onRestart()* e *onDestroy()*.

Figura 16 – Ícone do aplicativo desenvolvido.



Fonte: Próprio autor.

Figura 17 – Modelo do ciclo de vida de uma atividade.



Fonte: (GOOGLE DEVELOPERS, 2020)

Ao se criar a atividade *MainActivity* o *Callback onCreate()* é chamado. Nele os seguintes processos ocorrem: é definido o *layout* utilizado nessa atividade; são checadas as permissões do aplicativo; iniciado o *mapFragment* do GoogleMaps; e chama o método para iniciar os parâmetros. Ao checar as permissões do dispositivo ocorre o processo de registrar o

SDK da DJI.

O método *startSDKRegistration()* tem como finalidade checar a *API_KEY* definida no *Manifest* através da internet. Após este processo é checada a conectividade com algum produto da DJI via USB. Quando essa conexão é identificada é chamado o método, *getProduct()* que retorna um objeto da classe *BaseProduct*. Este objeto é o link entre o sistema OSDK e MSDK, pois, através dele é possível enviar e receber informações do sistema aéreo. Este objeto é o link das informações obtidas do VANT para a interface.

3.2.1.2 Interface gráfica

Após o processo bem-sucedido de obter informações do VANT através do aplicativo agora é possível desenvolver a interface gráfica para comunicação. A interface busca trazer uma abordagem simples e direta. Onde o operador pode acionar o sistema em poucas etapas. Além de ter na mão todas as informações fundamentais para utilização do sistema. A interface é desenvolvida utilizando os *layouts* e *views* disponíveis para *Android*. Como exemplo: *RelativeLayout*, *LinearLayout*, *TextView*, *ImageView*, *Button* e *Switch*.

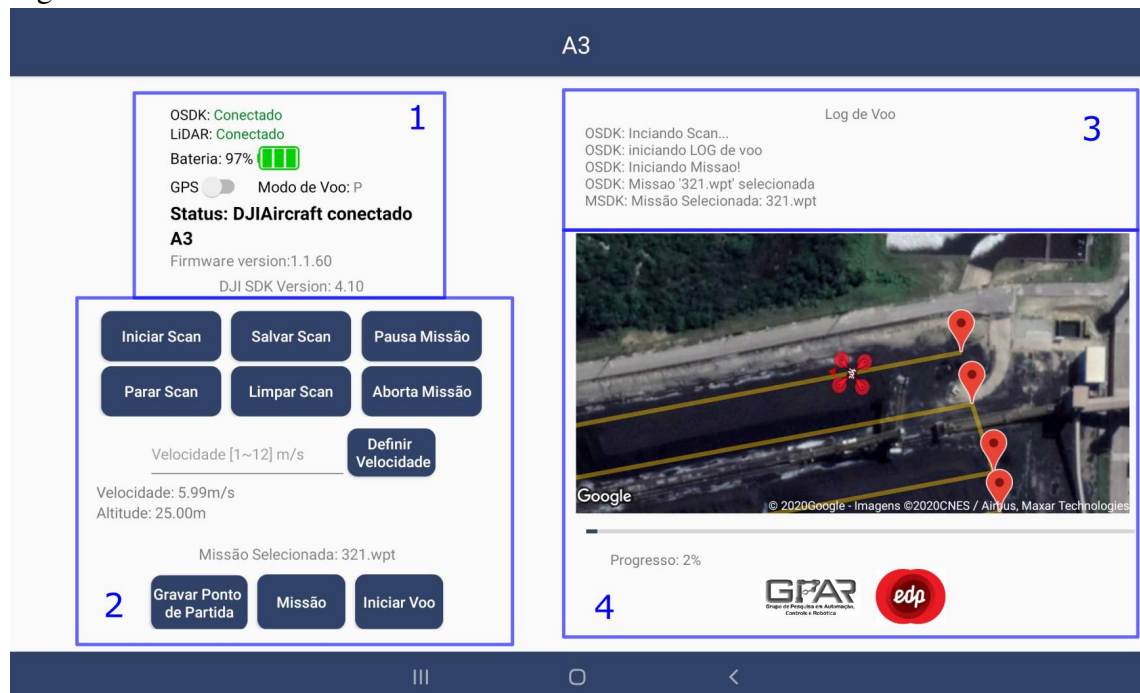
Para o projeto da interface foi feito um estudo de caso para a aplicação da usina. Foi levado em consideração todas as informações que seriam importantes para o operador durante o controle do sistema. Assim, um estudo prévio das bibliotecas e da documentação do MSDK permitiu planejar a estrutura da interface. Contendo informações fundamentais para localização do VANT, progresso de mapeamento, botões de operação, além de um *log* de voo com informações do sistema. A interface móvel desenvolvida é apresentada na Figura 18. O sistema é dividido e apresentado nos blocos presentes nas subseções a seguir.

3.2.1.2.1 Informações de comunicação e conectividade

O primeiro bloco, presente no canto superior direito do aplicativo, apresenta informações de conectividade do sistema. Serve como confirmação visual de que o sistema está pronto para operação.

O primeiro conjunto de informações visíveis é o estado de conectividade com o sistema OSDK e o sensor LiDAR. Esta conectividade é testada através de um *callback* presente na classe *FlightController* da DJI. Este *callback* é acionado toda vez que o sistema OSDK envia alguma informação, em *bytes*, para o MSDK. Dentro deste *callback* uma *flag booleana* recebe *true* como parâmetro para indicar que houve conectividade com o sistema OSDK.

Figura 18 – Interface móvel desenvolvida.



Fonte: Próprio autor.

Dentro deste *callback* também é chamado um método que analisa os dados recebidos do sistema OSDK. Então, foi definido uma padronização das mensagens que seriam recebidas. Onde os três primeiros caracteres da mensagem indicam a categoria da informação. Se o início da *string* for **PRG:** significa que a mensagem recebida avalia o progresso da missão de voo. Caso a *string* comece com **STA:** indica uma mensagem com relação ao status do sistema aéreo. Se a *string* começar por **LOG:** indica que é uma mensagem a ser impressa na área do *log* de voo.

Dentre as informações recebidas pela *string* “STA” duas delas são importantes para este bloco de comunicação e conectividade. São elas: o nível de bateria do drone e se o LiDAR está ou não conectado. Com estas informações, uma *Thread* foi criada e fica responsável por atualizar o *TextView* que indica se o sistema OSDK e o LiDAR estão conectados ou desconectados. A porcentagem da bateria também é atualizada de acordo com as informações recebidas pelo “STA:”. Onde foi feita uma lógica no código para alterar o ícone da bateria de acordo com a porcentagem restante. Sendo verde quando a bateria está entre 100 % e 70 %, amarelo para 69 % e 25 %, e vermelho quando inferior a 25 %.

As informações seguintes neste bloco indicam o sistema de posicionamento e o modo de voo. Através da chave é possível alternar entre o sistema RTK e o GPS para localização do drone. Por mais que o modo RTK seja mais eficiente quanto a localização do VANT, o modo GPS ainda é necessário para se trabalhar com simulações de voo no *DJI Assistant 2* como

também pode ser utilizado caso algum dos módulos do sistema RTK não esteja operando. O modo de voo faz a leitura do posicionamento da chave presente no controle VANT. Sendo apenas uma ferramenta visual direta de qual modo de operação está ativo no momento.

Logo em seguida tem o status de conexão com a aeronave. Este *TextView* altera de acordo com a detecção de conexão do sistema *Lightbrige 2* com alguma aeronave. Através dessa conexão ele também consegue observar qual o controlador de voo presente na aeronave. Por fim, este bloco também apresenta a versão do *firmware* do VANT e do sistema DJI SDK utilizado. Sendo uma ferramenta útil quando novas atualizações do sistema forem divulgadas.

3.2.1.2.2 Botões de operação

O segundo conjunto da interface é composto pelos botões de operação e controle do sistema. Este bloco da interface é organizado através de *RelativeLayout* para dispor os botões na ordem apresentada e *LinearLayout* para alinhar os elementos seguintes.

O primeiro conjunto de botões é composto por quatro botões para operação do sensor LiDAR e dois botões para controle da missão de voo. Os botões funcionam enviando um caractere para o sistema OSDK. Cada um deles envia uma letra e o sistema OSDK consegue identificar qual ação operar através disso. Este envio é feito através de um *callback* da classe *FlightController* da DJI SDK.

Os quatro botões de operação com o sensor LiDAR têm o objetivo de fazer uma operação “manual” do sistema de aquisição. Através dele é possível determinar onde e quando começar, parar, salvar ou limpar uma aquisição pelo sensor LiDAR. Os dois botões do lado direito operam para controle da missão de voo autônomo. Podendo interromper completamente uma missão com o botão “Aborta Missão” ou pausa e continuar uma missão com o botão “Pausa Missão”. Estes dois botões só ficam ativos e clicáveis quando o VANT se encontra em uma missão.

Logo abaixo destes botões é possível ver a área de controle de velocidade. As missões de voo autônomo são definidas com uma velocidade base definida em 8 m/s , porém, este valor pode ser alterado com este *EditText* e o botão “Definir Velocidade”. A nova velocidade é mantida até que missão atual seja concluída e uma nova se inicie ou caso seja determinado outro valor de velocidade pelo operador. Esta função apresenta uma limitação imposta com relação à velocidade máxima permitida para operação ser de 12 m/s . Este limite foi definido devido aos resultados não satisfatórios obtidos com a nuvem de pontos resultante para velocidades acima de

12 *m/s*.

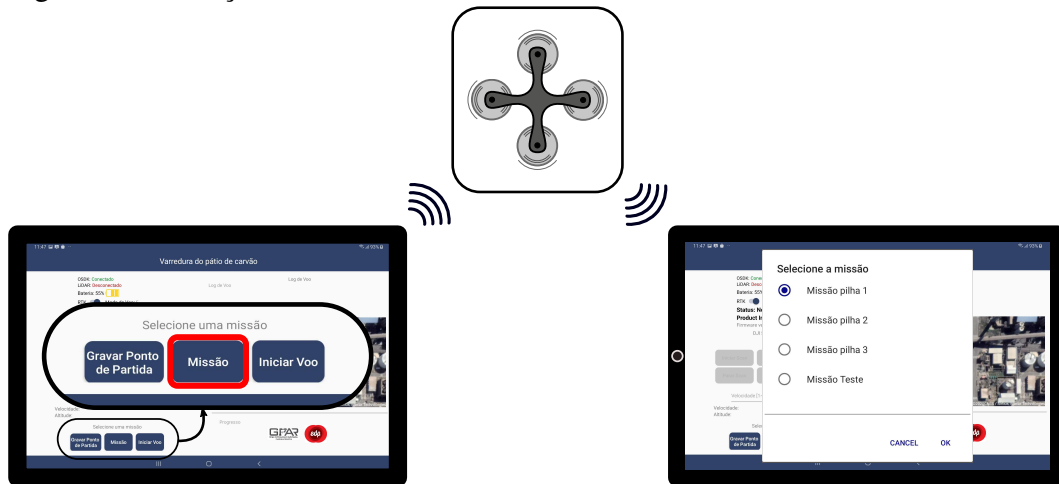
Em seguida, é informado os valores da velocidade do VANT em *m/s* e a altitude do mesmo em *m*. Essas duas informações são obtidas em tempo real e servem como um indicativo de localização do sistema aéreo. Onde é possível observar se o drone mantém a altitude e a velocidade constantes durante a missão escolhida. A altura do VANT pode ser alterada diretamente através do *Lightbridge 2* e será indicado imediatamente na interface. Porém, não é indicado uma alteração da altitude durante o processo de aquisição da nuvem de pontos pelo LiDAR.

Por fim, os últimos elementos deste bloco é composto por um *TextView* que informa a missão de voo selecionada pelo operador e três botões para início do voo. O primeiro botão define o Ponto de Partida (do inglês *HomePoint*). Ao pressionar este botão um *callback* próprio da DJI é acionado e define para o sistema as coordenadas do ponto de partida do VANT. É ideal que este ponto de partida seja um espaço aberto e que permita o sistema aéreo alçar voo e pousar em segurança. Este local de partida também é importante para a criação do quadro de referência *mapa*, como apresentado na Subseção 2.2.1.1.

O segundo botão só fica disponível para ser clicado após ser definido o ponto de partida. Este botão abre uma janela para seleção das missões de voo possíveis para o sistema aéreo voltado para o pátio de carvão. Ao pressionar o botão um sinal é enviado para o sistema aéreo que retorna todas as missões do voo salvas na sua memória através do sistema OSDK. A Figura 19 apresenta o esquemático de funcionamento de criação desta lista de missões dinâmica e a lista de missões para escolha do operador é apresentada na Figura 20. Dentre as missões, é possível escolher uma única pilha entre as três presente, ou até mesmo uma combinação entre as pilhas. Onde também é possível escolher uma missão para mapeamento das três pilhas em sequência.

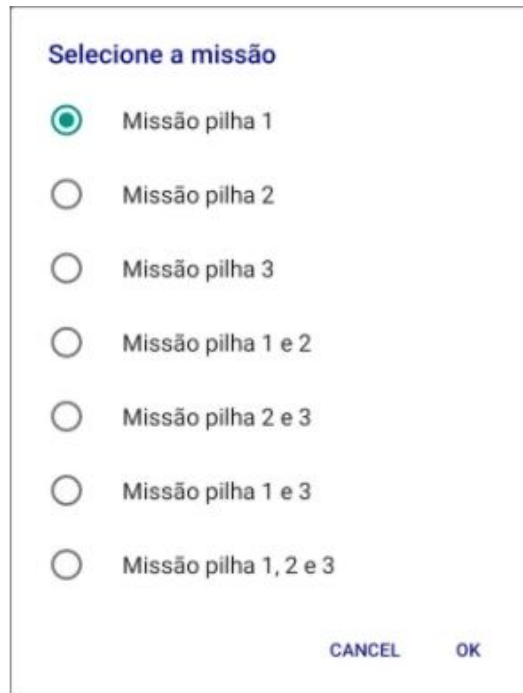
O último botão é liberado para operação após uma missão ser escolhida. Ao se apertar este botão uma *string* contendo a missão selecionada é enviada para o sistema OSDK. Ele então processa qual a missão de voo foi escolhida e dá início ao voo autônomo. As missões de voo têm definidas a velocidade e altura de voo e todos os *waypoints* pré-definidos. Assim, o controlador de voo A3 se encarrega de montar o plano de voo seguindo a trajetória definida pelos *waypoints*.

Figura 19 – Criação da lista dinâmica de missões.



Fonte: Próprio autor.

Figura 20 – Janela para seleção da missão de voo.



Fonte: Próprio autor.

3.2.1.2.3 Log de voo

O terceiro bloco da interface é composto apenas do *log* de voo. Este módulo é composto por um *TextView* que atualiza constantemente com as mensagens recebidas pelo OSDK e MSDK, sendo indicada no início de qual dos dois SDK veio a mensagem. Como explicado anteriormente, uma das mensagens recebidas do sistema OSDK é a *string* iniciada com “Log:”. Esta mensagem é então adicionada em sequência e impressa na área destinada para isto. Documentando as últimas cinco mensagens recebidas pelo sistema aéreo ou pelo sistema *mobile*.

Por ele, informações como início da aquisição pelo sensor LiDAR, ou possíveis mensagens de erro, são apresentadas ao operador.

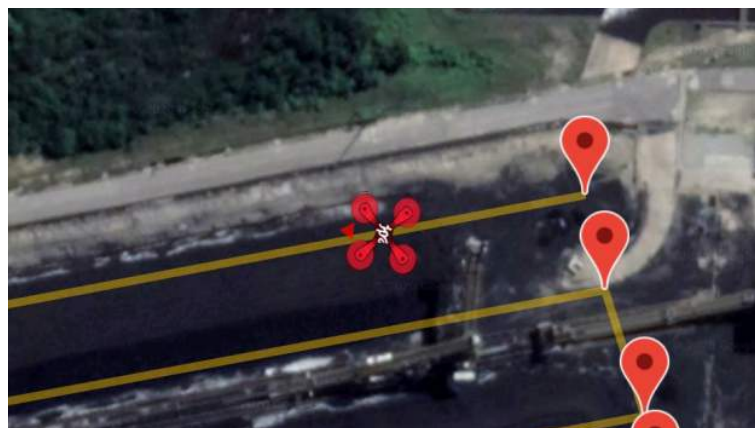
3.2.1.2.4 Mapa e progresso

Por fim, o último dos quatro blocos é composto pelo mapa e pela barra de progresso. O mapa é implementado através de um *SupportMapFragment* com *OnMapReadyCallback* para inicializar as configurações do mapa. Dentre as configurações iniciais do mapa tem a localização da posição atual do VANT e centralização do mapa nesta localização. Caso o VANT ainda não esteja ligado o mapa é centralizado nas coordenadas do pátio de carvão da EDP e um ícone com a logo da EDP representando o VANT é adicionado ao mapa.

O mapa tem como função apresentar a localização do VANT em tempo real durante todo o tempo que ele está ligado. Para isto, foi gerado um método automático para leitura da latitude, longitude e ângulo *yaw* do VANT. Quando ocorre uma mudança nos valores dessa leitura o ícone do VANT no mapa é atualizado e novamente o mapa é centralizado nesta posição. Outra ferramenta implementada no mapa é a trajetória que o quadricóptero irá percorrer na missão escolhida. Quando o botão de “Iniciar Voo” é pressionado os pontos georreferenciados, *waypoints*, e a trajetória a ser percorrida pelo sistema aéreo são implementados no mapa.

A Figura 21 apresenta o mapa e todos os elementos descritos acima. O ícone do VANT com uma seta para indicar a parte frontal do sistema de mapeamento que segue as informações obtidas do ângulo *yaw*. Os *Waypoints* representados pelos marcadores em vermelho e a trajetória a ser percorrida em amarelo.

Figura 21 – Mapa com a localização em tempo real do VANT.



Fonte: Próprio autor.

Por fim, há a barra de progresso. O cômputo do progresso da missão é calculado pelo sistema OSDK. Esta porcentagem é calculada a partir da distância percorrida pelo VANT, tendo os *waypoints* como referência para este cálculo. Onde é calculado a distância total entre os *waypoints* da missão e a distância percorrida pelo VANT. Este cálculo é feito em tempo real enquanto o VANT percorre a trajetória e é enviado constantemente para o sistema MSDK. Como apresentado anteriormente, o aplicativo ao receber uma mensagem iniciada com “PRG:” alimenta o elemento *ProgressBar* da interface e atualiza o *TextView* com o valor de porcentagem recebido.

4 CÁLCULO DE VOLUME E CONTROLE DO INVENTÁRIO

Após o sistema aéreo operar uma missão de varredura da pilha de carvão os dados obtidos devem ser processados para a construção da nuvem de pontos, conforme apresentado na Figura 1. Após o VANT retornar ao solo os dados obtidos pela *Manifold* devem ser enviadas ao servidor que opera o sistema de controle do estoque da empresa.

O repasse das informações obtidas após a aquisição é campo é feito de forma automática através de um programa desenvolvido para o projeto. O programa, chamado de **TRMS**, detecta se o sistema de mapeamento aéreo está conectado a mesma rede wi-fi, ou conectado via cabo, com o computador onde o TRMS está instalado. Após essa confirmação um botão é liberado para transmitir os dados das missões concluídas que permanecem no computador de bordo *Manifold*. O sistema é apresentado na Figura 22.

Figura 22 – Programa TRMS instalado no computador para operação.

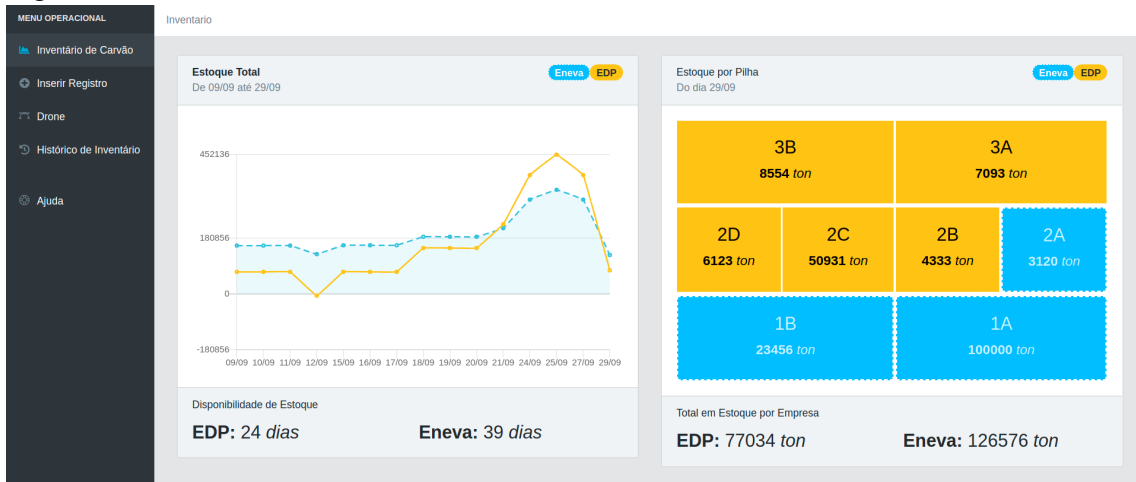


Fonte: Próprio autor.

Após esse processo, de forma automática, o programa segmenta a nuvem de pontos de acordo com divisões já estabelecidas pela empresa. Essas divisões são apresentadas para o usuário, que de forma manual pode redefinir um seguimento de pilha. Por fim, ao finalizar a divisão da nuvem de pontos de acordo com as pilhas predefinidas o volume de carvão é apresentado.

O cálculo de volume é obtido através de códigos em Python e C++. Sendo este cômputo apresentado através de uma *Dashboard* para controle de inventário desenvolvida em *JavaScript*. A *Dashboard*, apresentada na Figura 23, tem o intuito de facilitar a análise dos dados de estoque de carvão na indústria. Sendo constituído de formulários e gráficos para facilitar a visualização e monitoramento dos pátios de carvão e o seu estoque.

Figura 23 – *Dashboard* de controle de inventário.



Fonte: Próprio autor.

Essa etapa do sistema é apresentada de maneira mais aprofundada em (SOUSA, 2020).

5 RESULTADOS

Após apresentação detalhada dos equipamentos a serem utilizados na operação do sistema de mapeamento tridimensional testes foram feitos. Este capítulo aborda os seguintes tópicos que são detalhados nas seções a seguir: (i) testes em laboratório de conexão entre a interface o sistema aéreo utilizando do simulador de voo como ambiente para testes; (ii) apresentação de testes práticos de operação do sistema de mapeamento completo.

5.1 Resultados de Simulação

5.1.1 Testes de Operação da Interface

Os primeiros testes do sistema foram feitos em um ambiente controlado e visam observar todas as funcionalidades da interface desenvolvida antes de uma aplicação em campo. O ambiente para testes está presente no *Software DJI Assistant 2* que apresenta uma aba de simulação de voo para os VANTs com controladores da DJI. Durante a simulação pode ser testado os modos de voo presentes no sistema, testar o controle manual de um modelo tridimensional de uma aeronave como também testar missões de voo. Através deste simulador todas as funcionalidades da interface gráfica podem ser testadas mantendo a segurança dos equipamentos.

5.1.1.1 Teste de conectividade e mensagens de segurança

Com o VANT conectado a um computador com o simulador de voo ligado é possível testar a conectividade do sistema aéreo com o aplicativo desenvolvido. Estes testes ocorrem sem o desgaste dos motores e da bateria, já que o consumo de carga durante a simulação é mínimo. A Figura 24 apresenta o sistema de mapeamento aéreo conectado via USB ao computador contendo o simulador de voo. A imagem também apresenta o sistema *Lightbridge* com a interface pronta para testes.

O primeiro experimento feito testou as informações presentes no primeiro bloco de operações da interface, apresentadas na Subseção 3.2.1.2.1. O teste consistiu em observar o tempo de resposta da interface ao detectar as mudanças do sistema de mapeamento. Para isso, de maneira manual, foram retirados os cabos de conexão do sensor LiDAR. A Figura 25 apresenta um trecho da interface ao se desconectar um dos cabos do sensor LiDAR.

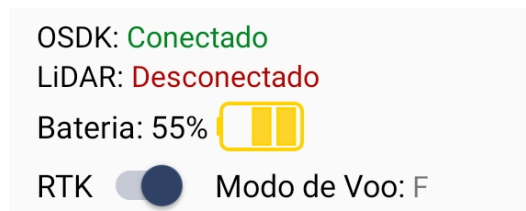
A conexão com o sistema é processada em tempo real, porém, o *status* só é atualizado

Figura 24 – Sistema aéreo conectado ao simulador de voo.



Fonte: Próprio autor.

Figura 25 – Resultado do teste ao desconectar sensor LiDAR.



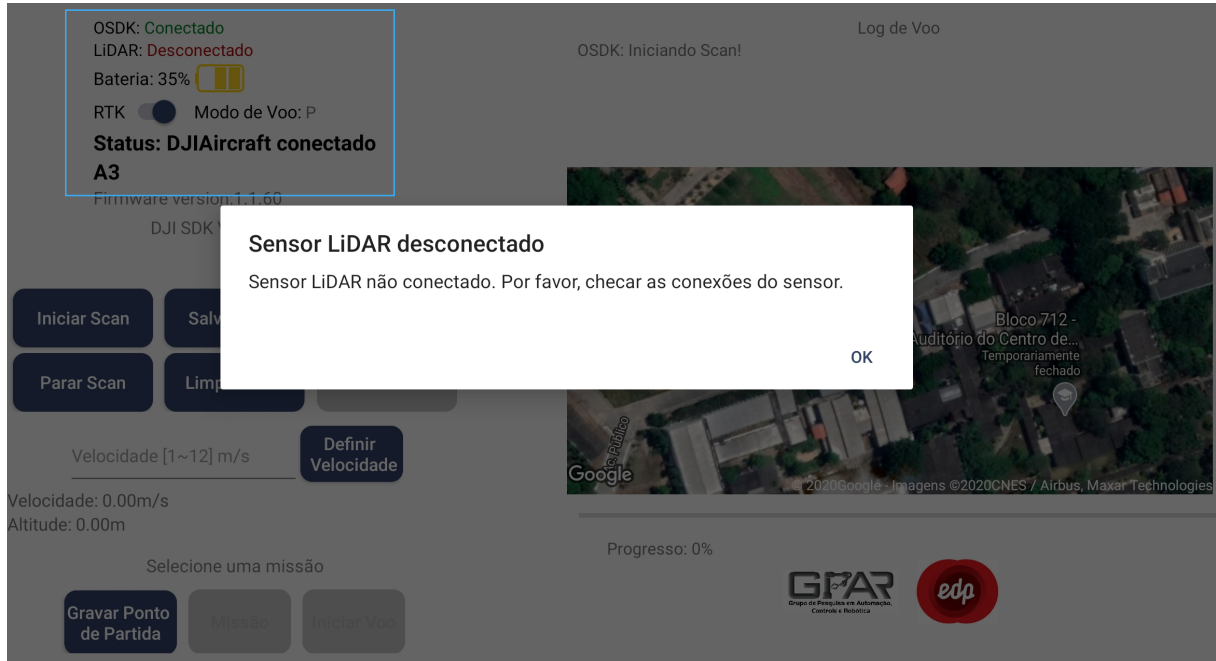
Fonte: Próprio autor.

quando a interface detecta a conexão por mais de 1 s. Este intervalo de tempo foi ajustado como forma de segurança e visa garantir que os equipamentos estão devidamente conectados. Com a informação de que o LiDAR está desconectado o operador deve checar as conexões do sistema aéreo antes de se iniciar a operação. Após a reconexão do sensor LiDAR esta informação é enviada para interface que muda o *status* para “Conectado” na cor verde dentro de 1 s após a reconexão. O mesmo intervalo de tempo é definido para atualização do *status* de conexão com o sistema OSDK.

Pensando na possibilidade de que a informação de que o sensor não está devidamente conectado passe despercebido pelo operador, uma mensagem de segurança foi implementada. A mensagem de segurança surge na tela da interface caso algum dos botões de operação do sensor LiDAR seja pressionado. A Figura 26 apresenta o resultado da tela ao se pressionar o botão “Iniciar Scan” com o sensor LiDAR desconectado. A mensagem também surge ao tentar

selecionar uma missão de voo, o que impede que o sistema tente fazer um voo de aquisição sem o LiDAR. A mensagem de segurança só para de aparecer após a devida reconexão do sensor.

Figura 26 – Mensagem de aviso ao operar com o sensor LiDAR.



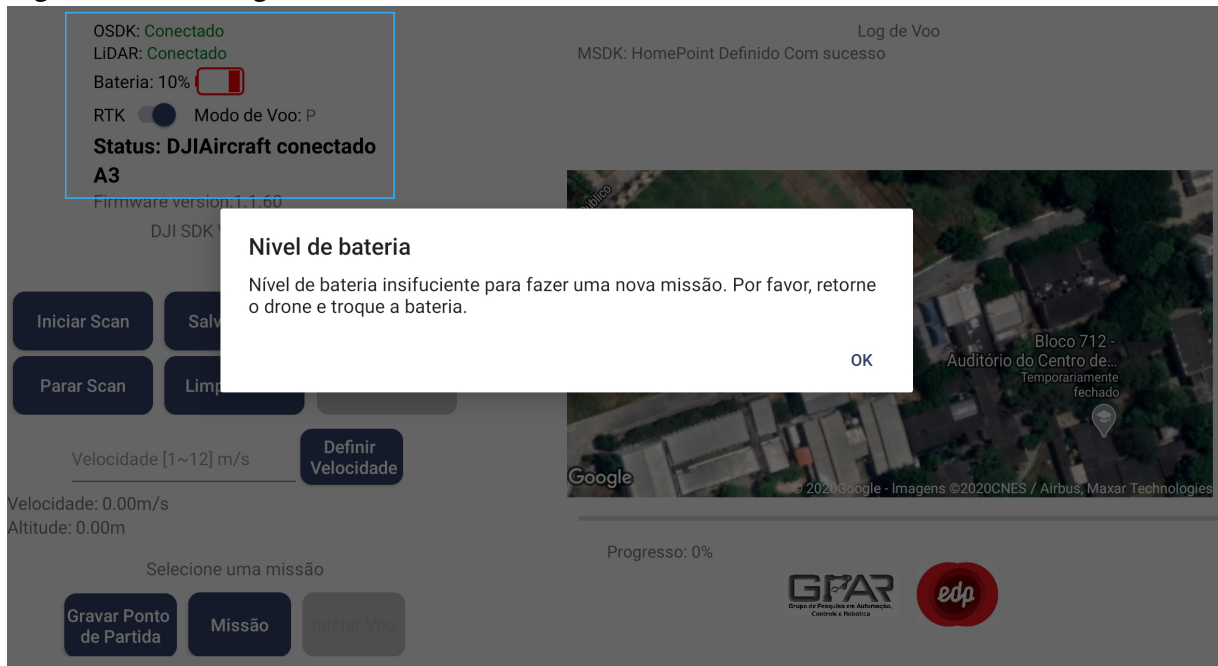
Fonte: Próprio autor.

A Figura 25 também apresenta outras informações diferentes da Figura 18. As variações do nível de bateria, modo de voo e chave de posicionamento RTK/GPS são atualizadas na interface de maneira dinâmica e em tempo real.

As variações do nível de bateria são recebidas do sistema aéreo e processadas, tendo o valor da porcentagem e o ícone como auxílio visual. Por exemplo, quando a bateria fica abaixo de 25 % o ícone fica em vermelho. Ao entrar neste estado a interface bloqueia automaticamente o início de uma nova missão de voo. Junto deste bloqueio aparece uma mensagem, Figura 27, avisando para o operador mudar a bateria do VANT para uma completamente carregada.

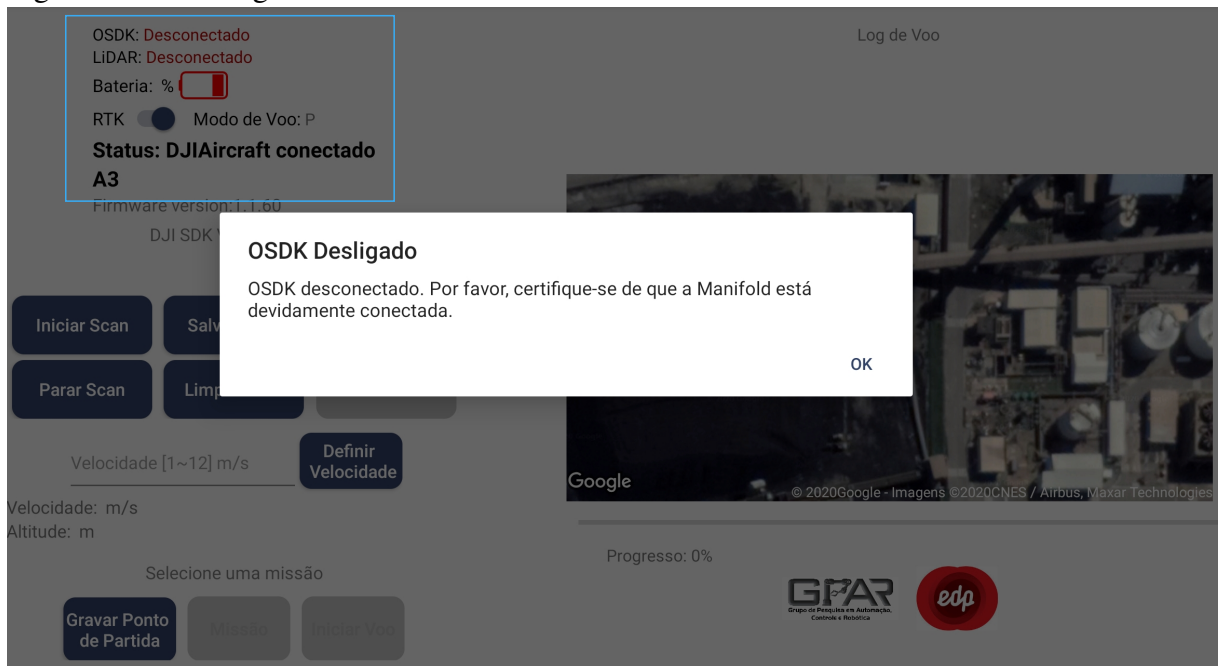
Outro experimento feito no simulador foi desconectar o computador embarcado *Manifold*. Ao perder a conexão com o sistema OSDK perde-se todos os dados de sensores presentes no VANT. A interface altera o status de conexão da OSDK para “Desconectado” e o texto em vermelho. Quando isto ocorre também é perdido o controle sobre o sensor LiDAR, que também altera seu status. Mais uma vez, como medida de segurança uma mensagem aparece, Figura 28, caso algum botão seja pressionado ou tente se iniciar uma missão de mapeamento. Esta mensagem de segurança é implementada com o intuito de impedir que o sistema aéreo opere sem que os dados sejam transmitidos para a interface.

Figura 27 – Mensagem do nível de bateria.



Fonte: Próprio autor.

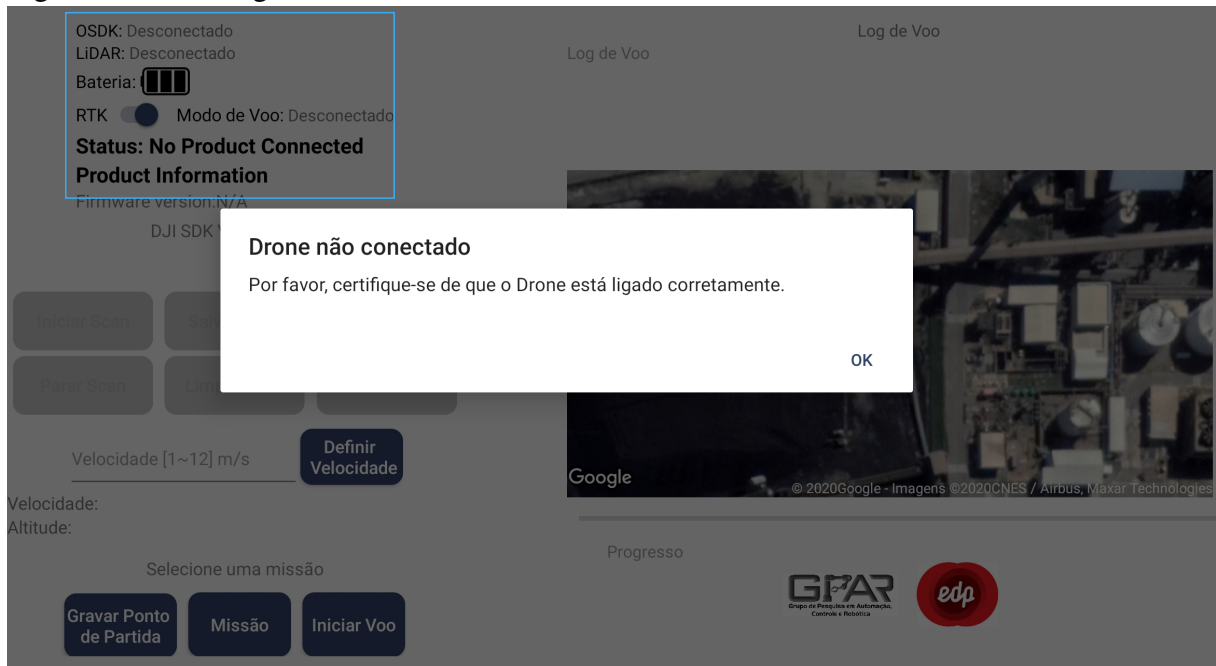
Figura 28 – Mensagem de conexão do OSDK.



Fonte: Próprio autor.

Em último caso, considerando que a bateria do VANT não foi bem fixada e o sistema não inicializou o controlador de voo, uma mensagem de segurança também é ativada. O alerta apresentado na Figura 29 visa informar ao operador que a alimentação do sistema aéreo apresenta uma falha e deve ser corrigida para a devida operação do sistema.

Figura 29 – Mensagem de conexão com o sistema aéreo.



Fonte: Próprio autor.

5.1.2 Teste de Mapeamento no Simulador

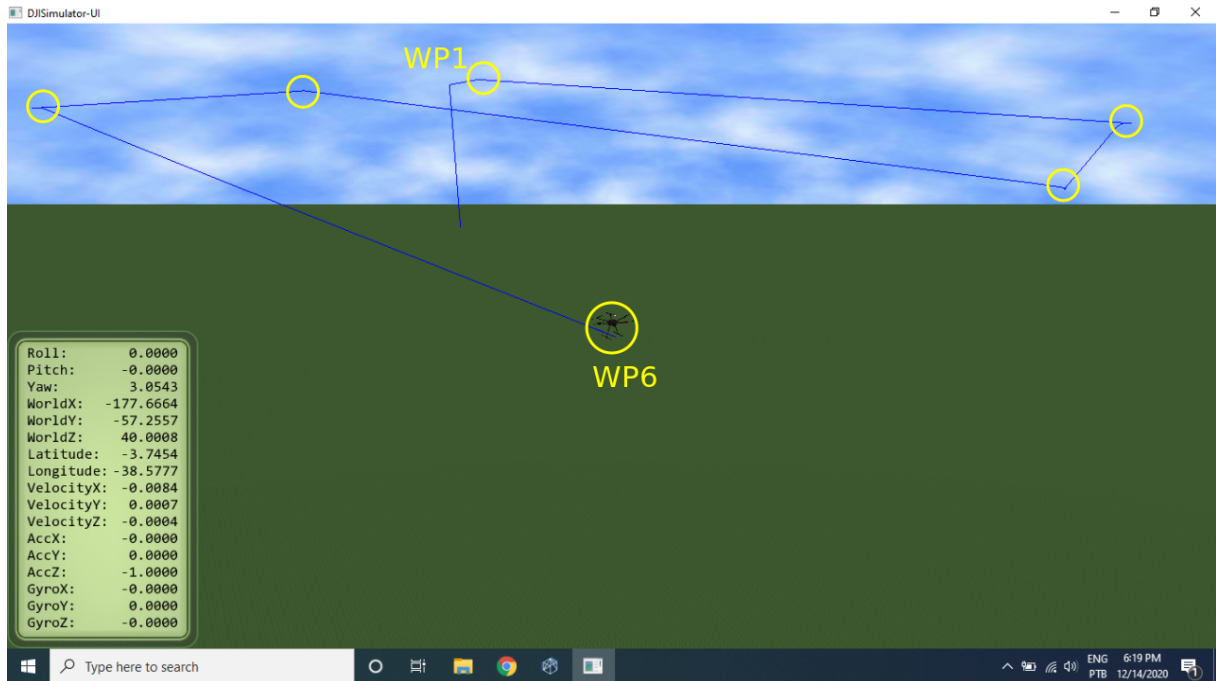
Após os testes dos botões de operação e do sistema de comunicação da interface o sistema de mapeamento aéreo precisa de mais um teste antes de ir para campo. O experimento final em laboratório visa simular a operação completa que será feita na termoelétrica para mapeamento das pilhas de carvão.

Para este experimento, foram escolhidas coordenadas próximas às pilhas de carvão como parâmetros de entrada do simulador no *DJI Assistant 2*. Após isto, iniciou-se a operação. Quando o modelo tridimensional aparece no simulador o ícone do VANT surge no mapa da interface, o que confirma a localização do modelo simulado. Após isto, o sistema está pronto para ser operado através da interface.

Primeiro, deve-se observar a conectividade do sistema OSDK e do sensor LiDAR e mais uma vez a localização do sistema aéreo através do mapa. Também deve ser checado o nível de bateria, o modo de voo deve estar em “P” e o a chave ativada no GPS para operação dentro do simulador. Feito isto, é possível gravar o Ponto de Partida, escolher a missão deseja e por fim iniciar o voo. Ao se iniciar o voo o caminho percorrido e os *Waypoints* da missão são marcados no mapa da interface. O modelo do simulador então começa a seguir o plano de voo recebido pelo OSDK do sistema, Figura 30.

A partir deste ponto os botões de operação do sensor LiDAR podem ser testados.

Figura 30 – Caminho percorrido na simulação.

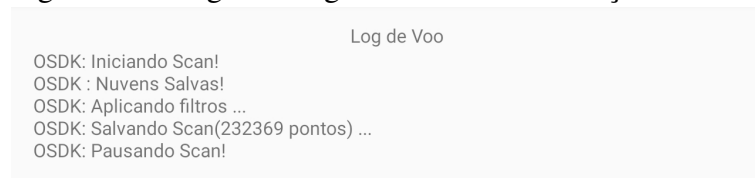


Fonte: Próprio autor.

Isto é feito para observar como o sistema responde durante uma missão de voo. Outra maneira de observar o devido funcionamento do sistema é a presença de novos arquivos de nuvem de pontos que são gerados pelo sensor mesmo durante o período de simulação. Todos esses comandos operados na interface imprimem uma mensagem no *log* de voo, assim como as mensagens de confirmação recebidas do sistema OSDK.

O teste com os botões de operação do sensor LiDAR consistiu em: i) iniciar uma missão de voo que automaticamente inicia a aquisição de dados no primeiro *Waypoint*; ii) Pausar a aquisição; iii) Salvar os dados; iv) Iniciar uma nova aquisição do sensor LiDAR e deixar a missão ser concluída. A Figura 31 apresenta o resultado pelo *log* de voo da interface.

Figura 31 – Log de voo gerado durante simulação.



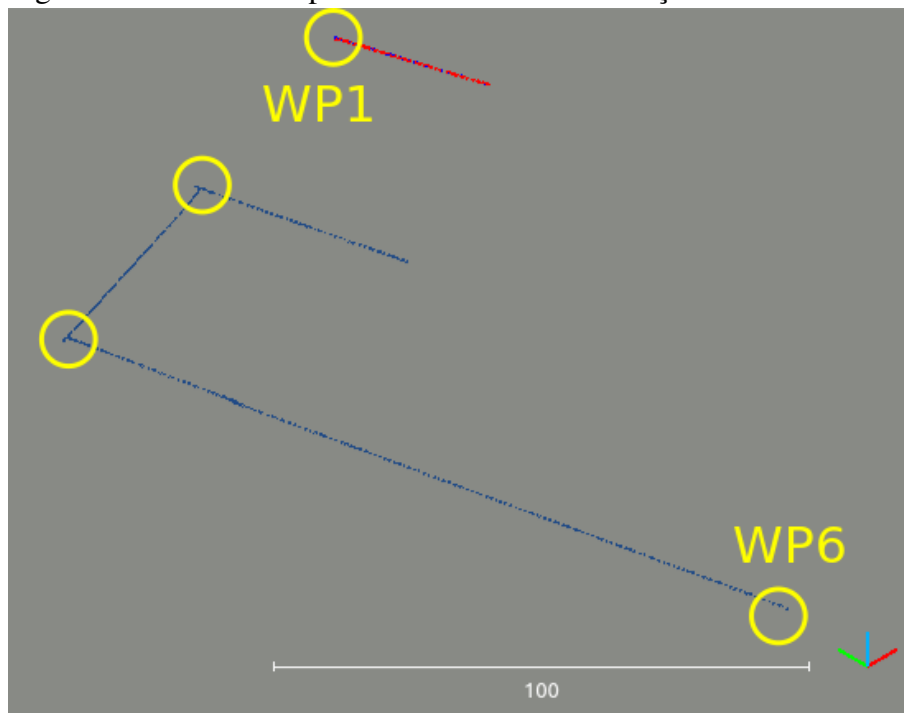
Fonte: Próprio autor.

Durante as missões de teste também é possível testar os botões de Pausar/Continuar Missão e o botão de Abortar missão. Este último botão encerra o plano de voo, limpando os *Waypoints* presentes no mapa e deixando o VANT em modo de *Hovering* esperando o comando para voltar ao ponto de partida através do *Lighbridge*.

Por fim, é possível fazer a confirmação de que o sistema aéreo já está em solo. Isto é feito observando o valor de altitude e velocidade do VANT, posição do ícone do VANT de volta a origem no mapa e mensagem de confirmação do fim da missão no *log*. Todos os itens presentes na interface móvel.

Ao conectar o sistema aéreo na rede é possível observar os dados obtidos pelo sensor LiDAR durante a simulação. Como o simulador gera um posicionamento artificial uma nuvem de pontos é gerada ao iniciar o *Scan*. A Figura 32 apresenta o resultado do ensaio devido ao uso dos botões de operação do sensor LiDAR, descrito anteriormente. A aquisição tem início em “WP1”, em seguida a aquisição é parada e a nuvem salva (nuvem com pontos vermelhos). Algum tempo depois a aquisição de pontos é retomada e ao chegar no “WP6” os dados são salvos novamente (nuvem com pontos azuis).

Figura 32 – Nuvem de pontos resultante da simulação.



Fonte: Próprio autor.

5.2 Testes de Operação em Ambiente Externo

Após os testes em laboratório o sistema estava pronto para operação completa em um ambiente externo. Para isto, foram feitos testes de mapeamento nas mediações da Universidade Federal do Ceará e na termoelétrica.

5.2.1 Mapeamento do Centro de Tecnologia da UFC

Os primeiros testes completos do sistema de mapeamento tridimensional ocorreram nas dependências da Universidade Federal do Ceará, mais precisamente no Centro de Tecnologia do Campus do Pici. O experimento visou reproduzir uma nuvem de pontos semelhante à Figura 33.

Figura 33 – Imagem obtida do local a ser mapeado no teste.



Fonte: Imagem obtida pelo GoogleMaps

Para isto, foi utilizada uma missão adotando seis *Waypoints* de referência sobre o Centro de Tecnologia. Os pontos de referência foram dispostos em um formato de “S” e mantendo a altura do sistema aéreo constante em 40 m. O VANT foi alocado em uma área aberta e ampla para poder decolar. Ali foi definido o seu Ponto de Partida e a missão apresentada anteriormente foi escolhida para dar início ao voo. Através da interface foi definida a velocidade de operação de 8 m/s e o processo pode ser acompanhado através do mapa e do *log* de voo, Figura 34.

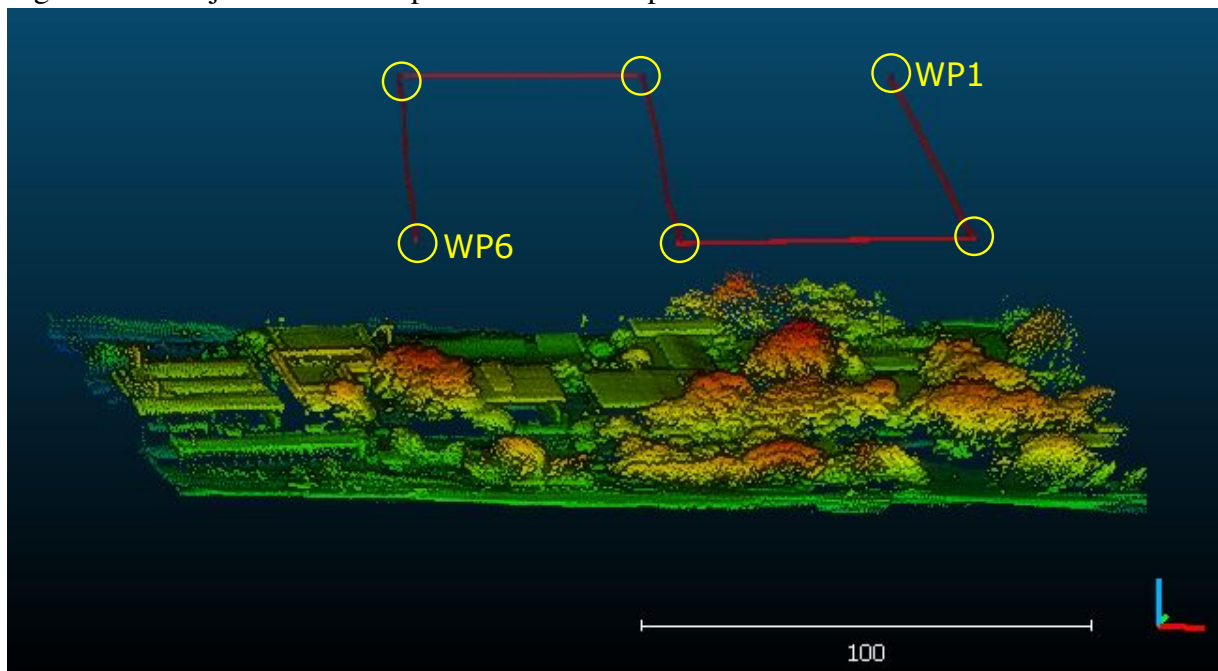
A trajetória percorrida pelo sistema aéreo é apresentada na Figura 35. O “WP1” indica o primeiro *Waypoint* escolhido e o local de início da aquisição pelo sensor LiDAR. Já o “WP6” indica o último ponto da trajetória automática e o final da aquisição do sensor LiDAR. A nuvem de pontos resultante após o retorno do sistema aéreo para Ponto de Decolagem é apresentada na Figura 36. Nesta imagem é possível observar a reconstrução, com detalhes, das construções do local, telhados, terraços e árvores.

Figura 34 – Trajetória do experimento na interface.



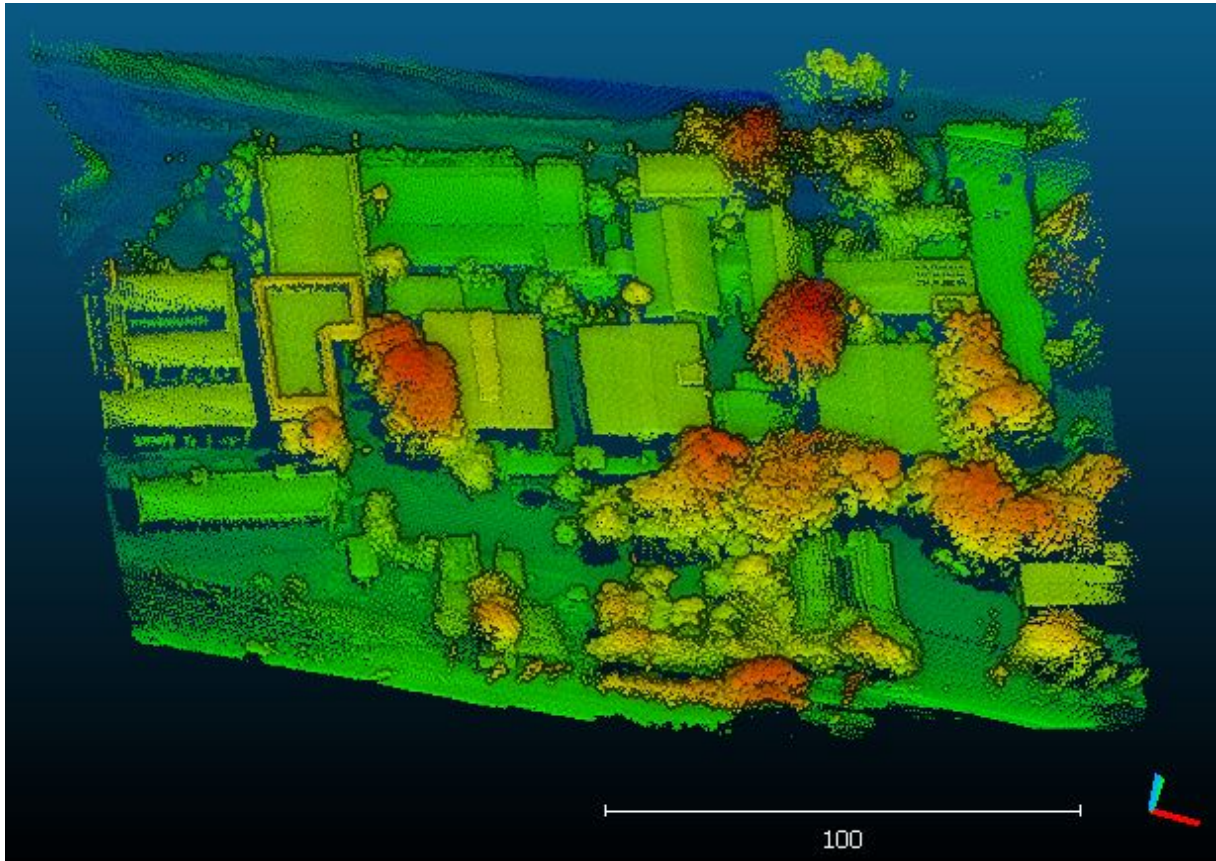
Fonte: Próprio autor.

Figura 35 – Trajetória definida para missão de mapeamento.



Fonte: Próprio autor.

Figura 36 – Nuvem de pontos resultante do Centro de Tecnologia.



Fonte: Próprio autor.

5.2.2 Testes de Mapeamento na Termoelétrica

Após os ensaios em laboratório e testes de operação e mapeamento foram feitos experimentos na Termoelétrica. O teste visa analisar a nuvem de pontos resultante do mapeamento. Também sendo observado a operação do sistema através da interface móvel. E por fim, observar como o sistema aéreo responde ao tempo de duração da varredura e consumo de bateria.

Primeiramente, o VANT é devidamente alocado em ambiente aberto e próximo às pilhas de carvão que vão ser mapeadas. O ponto escolhido para este teste é apresentado na Figura 37. O local é de baixo fluxo de pedestres e a uma distância segura da pista. Neste local é conectada a bateria de alimentação do sistema aéreo. Após isto, as conexões do sensor LiDAR e dos componentes do *air system* RTK são checadas.

Em um local próximo, utilizando a interface é possível observar se o sistema aéreo e os sensores estão devidamente conectados, Figura 38 a). O segundo componente do sistema RTK, o *ground system*, também deve ser instalado em um local próximo ao VANT, como visto na Figura 38 b). O sistema RTK que irá garantir a precisão da nuvem de pontos devido ao correto georreferenciamento do quadro de referência do VANT e do sensor LiDAR.

Figura 37 – Sistema aéreo de mapeamento tridimensional na termoeletrica.



Fonte: Próprio autor.

Figura 38 – *Ground System* do sistema de posicionamento RTK e operador com controle *Lightbridge 2* e Tablet com interface gráfica.



a)

b)

Fonte: Próprio autor.

Através da interface a missão de voo é escolhida. As missões de voo para sobrevoar as pilhas são construídas e salvas no computador de bordo previamente. Uma análise do local foi feita previamente. Dentre as informações analisadas e levadas em consideração, tem-se a

altura do maquinário presente nas pilhas de carvão, a largura e o comprimento das pilhas, como também o ângulo de abertura do sensor. Com isto, define-se a altura para aquisição no pátio de carvão em 30 metros.

Para este ensaio foi escolhido a “Missão pilha 1, 2 e 3” para fazer o mapeamento de todas as três pilhas em um único voo. Após o início do voo, a interface é atualizada com as informações dos *waypoints* a serem seguidos. Antes do sistema aéreo chegar no primeiro *waypoint* a velocidade do VANT é definida em 6 m/s.

Todas essas informações estão presentes na Figura 39. Nela é possível ver que o sistema está devidamente conectado. O sistema operando em modo de voo “P” e o sistema de posicionamento RTK ativo. A missão e a velocidade foram definidas com sucesso, como visto no *log* de voo. Na parte inferior é possível ver que o sistema está fazendo a aquisição em 30 m de altura e que a velocidade está em 5,87 m/s. Este valor de velocidade é ocasionado pela tentativa do VANT em manter a trajetória mesmo com a ação do vento forte presente no local.

Figura 39 – Interface durante testes.

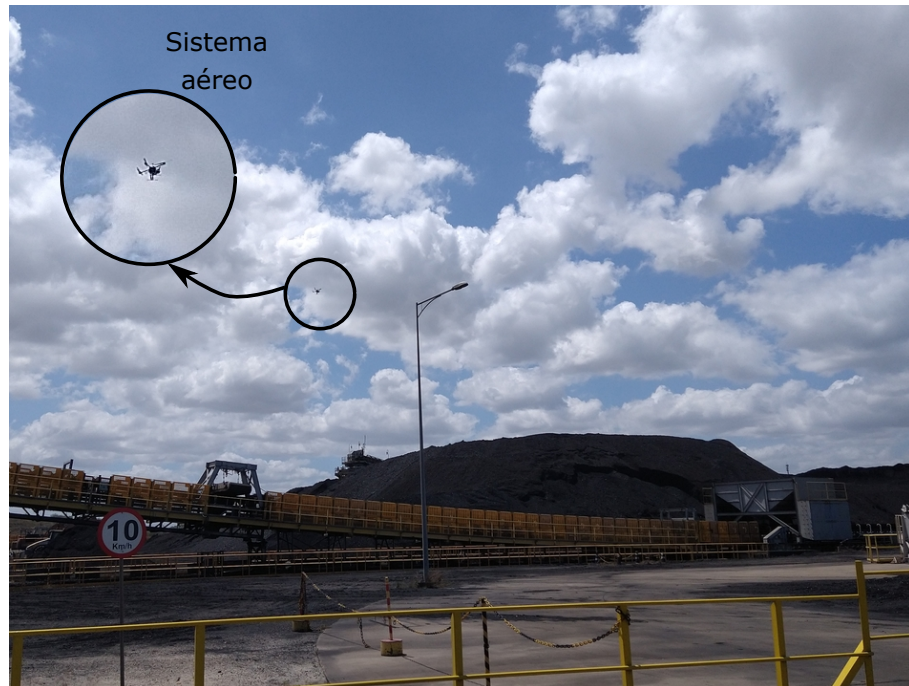


Fonte: Próprio autor.

Após definir todos os parâmetros o sistema está pronto para fazer o voo autônomo. A Figura 40 apresenta o sistema aéreo fazendo a missão de mapeamento. Pela figura é possível observar a importância das informações presentes na interface que apresentam, em tempo real, dados de voo e a localização do sistema aéreo, mesmo quando há a perda do contato visual com o mesmo. A Figura 41 apresenta a trajetória definida pelos *waypoints* no mapa e o percurso feito

pelo sistema aéreo em nuvem de pontos.

Figura 40 – Processo de mapeamento na termoeletrica.



Fonte: Próprio autor.

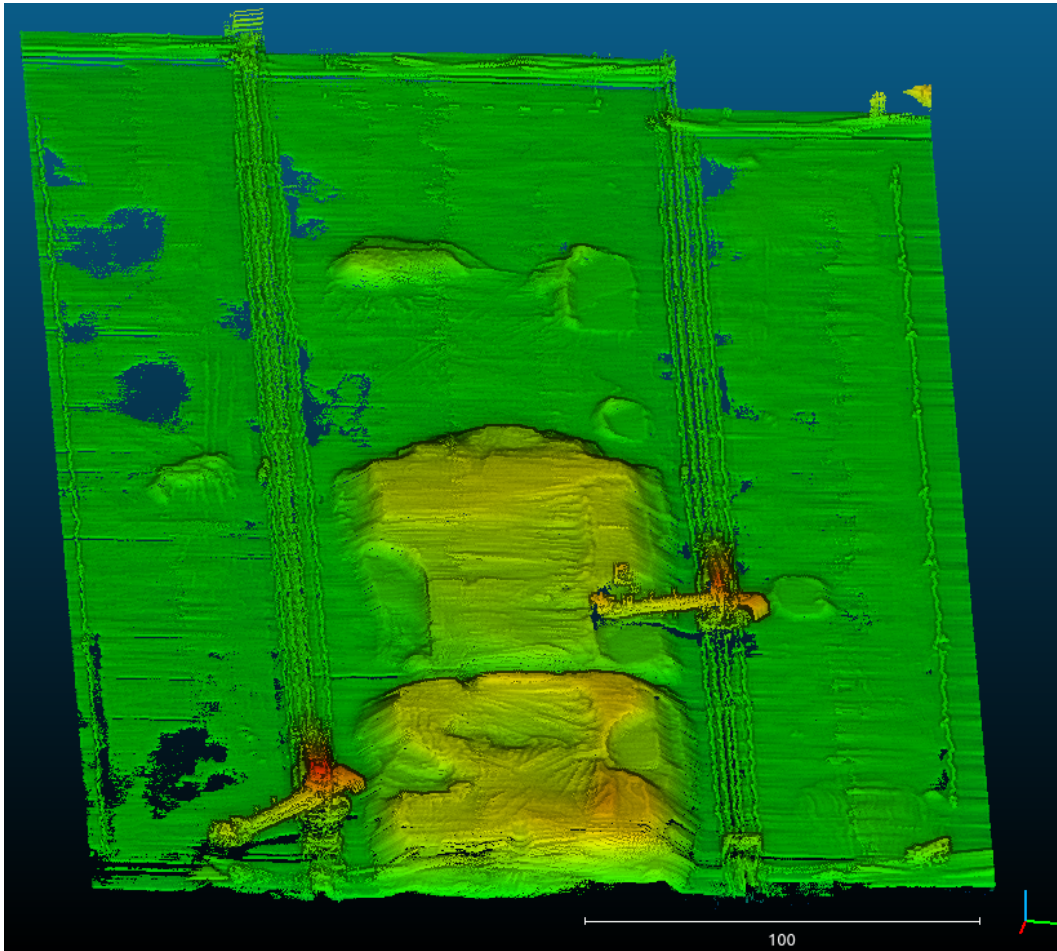
Figura 41 – Caminho percorrido pelo sistema aéreo.



Fonte: Próprio autor.

Após a conclusão da missão, o VANT retorna para o local de partida e pouso de maneira estável no solo. A partir daí, o sistema pode ser recolhido e a nuvem de pontos ser analisada. A Figura 42 apresenta a nuvem de pontos gerada após o mapeamento com 4.350.536 pontos. Nela é possível observar o maquinário operando sobre a pilha de carvão central. Os espaços vazios presentes nas pilhas laterais são devido à presença de poças de água. O pátio como um todo é constantemente irrigado para diminuir a liberação de pó de carvão no ambiente. Estas poças se acumulam quando o terreno está plano e isto faz com que os feixes de luz emitidos pelo sensor LiDAR não retornem como esperado.

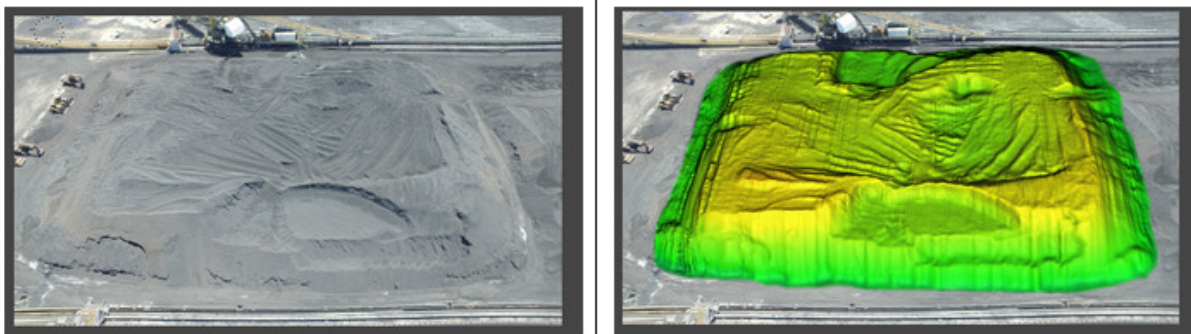
Figura 42 – Nuvem de pontos resultante do pátio de carvão.



Fonte: Próprio autor.

Porém, mesmo com estas lacunas, a pilha de carvão central pode ser observada com detalhes. A Figura 43 apresenta um comparativo lado a lado. A imagem da esquerda foi tirada utilizando uma câmera acoplada ao sistema aéreo. Esta é uma confirmação visual para o sistema de aquisição desenvolvido.

Figura 43 – Comparativo da nuvem de pontos com o local.



Fonte: Próprio autor.

Por fim, a nuvem resultante pode ser filtrada, para retirada do maquinário, e assim ter

o seu volume calculado. Os resultados são armazenados no servidor da termoelétrica e podem ser acessados através da *Dashboard* apresentada anteriormente. O volume resultante obtido passa então para análise da termoelétrica.

O teste experimental na termoelétrica também visou analisar o tempo de duração da missão completa, assim como o consumo de bateria pelo sistema. Do início ao final do voo, o sistema percorreu 4,37 Km em 15 : 05 min. Inicialmente a bateria estava com 100 % da carga e encerrou em 40 %. Conforme apresentado, o sistema consome por volta de 60 % da carga da bateria. Visando a segurança do sistema e garantia do mapeamento completo do pátio, considerando este o ensaio que mais exige do sistema, tem-se a necessidade de alterar as condições da interface. Assim, a interface passa a bloquear o início de novas missões quando o nível da bateria está abaixo de 70 %.

6 CONCLUSÕES

O desenvolvimento de um sistema de mapeamento aéreo tridimensional operado por uma interface gráfica para dispositivos móveis foi apresentado. O sistema de mapeamento é automático e de grande precisão na reconstrução de nuvens de pontos 3D. O sistema foi desenvolvido para o mapeamento de pilhas de carvão de uma termoelétrica. Entretanto, o sistema de geração da nuvem de pontos pode ser aplicado em diversas áreas da indústria. Podendo servir como material de estudos e controle de inventário devido aos resultados precisos obtidos por este sistema.

A interface gráfica desenvolvida em *Java* para aparelhos *Android* foi criada voltada para este sistema. A interface traz de forma eficiente e simplificada todas as informações essenciais para operação do sistema de mapeamento em campo. Os experimentos práticos mostraram que a interface permite que o operador tenha controle de qual pilha será feita o mapeamento. Podendo mapear completamente o pátio de carvão em uma única missão de voo.

Também foram feitos testes de operação da interface. Os resultados apresentados mostram que a interface desenvolvida é robusta, apresentando avisos e bloqueando certas funções. Isto garante que o sistema opere de maneira eficaz e reduz o desgaste do equipamento aéreo. A interface desenvolvida apresenta capacidade de operação com outros sistemas que apresentem controladores da DJI. O sistema como um todo apresenta a vantagem de ser de fácil adaptação por utilizar de ferramentas *open-source* no seu desenvolvimento.

Os resultados obtidos nas nuvens de pontos foram satisfatórios para a termoelétrica e serve de grande auxílio no controle de inventário da sua matéria prima. A precisão dos resultados obtidos permite inferir que não seja necessário o uso de algoritmos extras para alinhamento da nuvem de pontos. Esta precisão é obtida através da robustez do sistema aéreo utilizado e do sistema de localização RTK.

REFERÊNCIAS

- ARANGO, C.; MORALES, C. Comparison between multicopter uav and total station for estimating stockpile volumes. **The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences**, Copernicus GmbH, v. 40, n. 1, p. 131, 2015.
- ARGYROUDIS, P.; BAYLEY, G.; GROB, M. S.; PHILLIPS, M.; SOLIMAN, S. S. **Wireless remote telemetry system**. [S.l.]: Google Patents, 1998. US Patent 5,748,104.
- BEMIS, S. P.; MICKLETHWAITE, S.; TURNER, D.; JAMES, M. R.; AKCIZ, S.; THIELE, S. T.; BANGASH, H. A. Ground-based and uav-based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology. **Journal of Structural Geology**, v. 69, p. 163 – 178, 2014. ISSN 0191-8141. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0191814114002429>.
- CHEON, Y. Multiplatform application development for android and java. In: **2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)**. [S.l.: s.n.], 2019. p. 1–5.
- CHESWORTH, A. A.; HUDDLESTON, J. Precision optical components for lidar systems developed for autonomous vehicles. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Next-Generation Optical Communication: Components, Sub-Systems, and Systems VII**. [S.l.], 2018. v. 10561, p. 105610J.
- CHIANG, K.; TSAI, G.; LI, Y.; EL-SHEIMY, N. Development of lidar-based uav system for environment reconstruction. **IEEE Geoscience and Remote Sensing Letters**, v. 14, n. 10, p. 1790–1794, Oct 2017.
- CUI, Y.; LI, Q.; YANG, B.; XIAO, W.; CHEN, C.; DONG, Z. Automatic 3-d reconstruction of indoor environment with mobile laser scanning point clouds. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 12, n. 8, p. 3117–3130, Aug 2019.
- DEITEL, H.; DEITEL, P.; DEITEL, A.; TORTELLO, J. **Android: COMO PROGRAMAR**. 2. ed. ARTMED, 2015. ISBN 9788582603475. Disponível em: <https://books.google.com.br/books?id=NdZVvgAACAAJ>.
- DELEÓN, R.; VICENTE, G.; ZOPPOLO, R.; ARNAUD, A.; MIGUEZ, M. Lidar based, tree row volume estimation for phytosanitary products reduction in fruit trees orchards. In: **2020 IEEE International Symposium on Circuits and Systems (ISCAS)**. [S.l.: s.n.], 2020. p. 1–4.
- DJI. **E5000 PRO Tuned Propulsin System**. 1. ed. <https://www.dji.com/e5000/info#specs>, 2016.
- DJI. **A3/A3 PRO User Manual**. 1.4. ed. <https://www.dji.com/a3/info#downloads>, 2017.
- DJI. **D-RTK User Manual**. 1. ed. <https://www.dji.com/d-rtk/info#downloads>, 2017.
- DJI. **DJI Lightbridge 2 User Manual**. 1.6. ed. <https://www.dji.com/lightbridge-2/info>, 2017.
- DJI. **Mobile SDK Introduction**. DJI Developer MSDK, 2020. Disponível em: <https://developer.dji.com/document/4cd08995-3952-4db6-ab6e-bc3a754da153>. Acesso em: 26 nov. 2020.

FONG, T.; THORPE, C. Vehicle teleoperation interfaces. **Autonomous robots**, Springer, v. 11, n. 1, p. 9–18, 2001.

GOOGLE DEVELOPERS. **Understand the Activity Lifecycle**. Android Developers, 2020. Disponível em: <https://developer.android.com/guide/components/activities/activity-lifecycle>. Acesso em: 30 nov. 2020.

GUERREIRO, B. J.; SILVESTRE, C.; CUNHA, R.; CABECINHAS, D. Lidar-based control of autonomous rotorcraft for the inspection of pierlike structures. **IEEE Transactions on Control Systems Technology**, IEEE, v. 26, n. 4, p. 1430–1438, 2017.

HE, H.; CHEN, T.; ZENG, H.; HUANG, S. Ground control point-free unmanned aerial vehicle-based photogrammetry for volume estimation of stockpiles carried on barges. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 16, p. 3534, 2019.

HOLZINGER, A.; TREITLER, P.; SLANY, W. Making apps useable on multiple different mobile platforms: On interoperability for business application development on smartphones. In: SPRINGER. **International Conference on Availability, Reliability, and Security**. [S.l.], 2012. p. 176–189.

JAWAD, H. M. Android mobile app development as a motivation towards computer programming. In: **2019 IEEE International Conference on Electro Information Technology (EIT)**. [S.l.: s.n.], 2019. p. 169–175.

JOSHI, D. **Here are the world's largest drone companies and manufacturers to watch and stocks to invest in 2020**. Business Insider, 2019. Disponível em: <https://www.businessinsider.com/drone-manufacturers-companies-invest-stocks>. Acesso em: 10 nov. 2020.

KOTB, A.; HASSAN, S.; HASSAN, H. E.-Z. A comparative study among various algorithms for lossless airborne lidar data compression. **2018 14th International Computer Engineering Conference (ICENCO)**, p. 17–21, 2018.

LANGLEY, R. B. Rtk gps. **Gps World**, v. 9, n. 9, p. 70–76, 1998.

LIU, P.; CHEN, A. Y.; HUANG, Y.-N.; HAN, J.-Y.; LAI, J.-S.; KANG, S.-C.; WU, T.-H.; WEN, M.-C.; TSAI, M.-H. A review of rotorcraft unmanned aerial vehicle (uav) developments and applications in civil engineering. **Smart Structures and Systems**, Techno-Press, v. 13, n. 6, p. 1065–1094, 2014.

LU, T.-F.; ZHAO, S.; XU, S.; KOCH, B.; HURDSMAN, A. A 3dof system for 3 dimensional stockpile surface scanning using laser. In: IEEE. **2011 6th IEEE Conference on Industrial Electronics and Applications**. [S.l.], 2011. p. 1–5.

ORFANUS, D.; FREITAS, E. P. de; ELIASSEN, F. Self-organization as a supporting paradigm for military uav relay networks. **IEEE Communications Letters**, v. 20, n. 4, p. 804–807, 2016.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE, JAPAN. **ICRA workshop on open source software**. [S.l.], 2009. v. 3, n. 3.2, p. 5.

SANTOS, J. T.; FORTE, M. D. N.; PAULA, A. R. de; SOUSA, N. R. T. de; LIMA, T. A.; FILHO, M. P. de A.; BRANCO, I. R. A. C.; CORREIA, W. B.; TORRICO, B. C.; NOGUEIRA, F. Desenvolvimento de instrumentação para geração de nuvem de pontos usando sensores inerciais e lidar. In: **Anais do 14º Simpósio Brasileiro de Automação Inteligente**. [S.l.: s.n.], 2019. v. 1, p. 1901–1907. ISSN 2358-4483.

SANTOS, J. T.; FORTE, M. D. N.; SOUSA, N. R. T. de; BRANCO, I. R. A. C.; TORRICO, B. C.; NOGUEIRA, F. Sistema vant para reconstrução tridimensional na indústria. In: **Anais do XXIII Congresso Brasileiro de Automática**. [S.l.: s.n.], 2020. v. 2. ISSN 2525-8311.

SICK. **Data sheet do produto LD-MRS420201**.

https://cdn.sick.com/media/pdf/4/44/644/dataSheet_LD-MRS420201_1085081_pt.pdf, 2020.

SMITH, S.; VOORHIS, W.; YOUNG, J. **Coal pile density studies for inventory control.[Coal; measuring methods]**. [S.l.], 1981.

SOUSA, N. R. T. de. **Desenvolvimento de Plataforma Web para Gerenciamento de Dados do Pátio de Estocagem de uma Usina Termelétrica à Carvão**. Dissertação (Mestrado) — Universidade Federal do Ceará, Ceará, Brasil, 2020.

STRADOLINI, F.; RIARIO, S.; BOERO, C.; BAJ-ROSSI, C.; TAURINO, I.; SURREL, G.; MICHELI, G. D.; CARRARA, S. Wireless monitoring of endogenous and exogenous biomolecules on an android interface. **IEEE Sensors Journal**, IEEE, v. 16, n. 9, p. 3163–3170, 2016.

TUCCI, G.; GEBBIA, A.; CONTI, A.; FIORINI, L.; LUBELLO, C. Monitoring and computation of the volumes of stockpiles of bulk material by means of uav photogrammetric surveying. **Remote Sensing**, v. 11, p. 1471, 06 2019.

VACCA, G.; FURFARO, G.; DESSI, A. The use of the uav images for the building 3d model generation. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, p. 217–223, 8 2018.

VALERO, A.; RANDELLI, G.; SARACINI, C.; BOTTA, F.; MECELLA, M. The advantage of mobility, mobile tele-operation for mobile robots. In: **Proc. AISB-HRI Symposium” New Frontiers in Human-Robot Interaction (To appear)**. [S.l.: s.n.], 2009.

Van Genderen, J. L. **Airborne and terrestrial laser scanning**. United Kingdom: CRC Press, 2010. ISBN 978-1904445-87-6.

WALLACE, L.; LUCIEER, A.; WATSON, C.; TURNER, D. Development of a UAV-LiDAR system with application to forest inventory. **Remote Sensing**, v. 4, n. 6, p. 1519–1543, 2012. ISSN 2072-4292.

WANG, J.; XU, L.; FAN, Y.; LIU, X.; TIAN, Z.; WANG, X.; CHENG, Y. A method for compensating platform attitude fluctuation for helicopter-borne lidar: Performance and effectiveness. **Measurement**, Elsevier, v. 125, p. 37–47, 2018.

ZHANG, W.; FELTNER, D.; SHIRLEY, J.; SWANGNETR, M.; KABER, D. Unmanned aerial vehicle control interface design and cognitive workload: A constrained review and research framework. In: IEEE. **2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.], 2016. p. 001821–001826.

ZHANG, W.; YANG, D.; LI, Y.; XU, W. Portable 3D laser scanner for volume measurement of coal pile. In: **Communication, Signal Processing, and Systems. Lecture Notes in Electrical Engineering**. [S.l.: s.n.], 2020. v. 517, p. 340–347. ISSN 1876-1119.

ZHANG, Y.; ZHANG, G.; FIERRO, R.; YANG, Y. Force-driven traffic simulation for a future connected autonomous vehicle-enabled smart transportation system. **The Transactions on Intelligent Transportation Systems**, 2018.