

**Sistema de monitoramento de métricas para Ambiente integrado UAV-FOG****Metric monitoring system for Integrated UAV-FOG environment**

DOI:10.34117/bjdv6n6-411

Recebimento dos originais: 08/05/2020

Aceitação para publicação: 17/06/2020

**João Lucas Ribeiro Barbosa**

Bacharel em Sistemas de Informação

Instituição: Universidade Federal do Sul e Sudeste do Pará.

Endereço: Folha 17, Quadra 04, Lote Especial, s/n.º - Nova Marabá, PA, Brasil.

E-mail: engjoaobarbosa@gmail.com

**Isaac Barros da Costa Moreira**

Bacharelado em Engenharia da Computação

Instituição: Universidade Federal do Sul e Sudeste do Pará.

Endereço: Folha 17, Quadra 04, Lote Especial, s/n.º - Nova Marabá, PA, Brasil.

E-mail: isaacx@unifesspa.edu.br

**Wendel Barros Serra**

Bacharelado em Engenharia da Computação

Instituição: Universidade Federal do Sul e Sudeste do Pará.

Endereço: Folha 17, Quadra 04, Lote Especial, s/n.º - Nova Marabá, PA, Brasil.

E-mail: wendel.barros@unifesspa.edu.br

**Warley Muricy Valente Junior**

Doutor em Ciência da Computação pela Universidade Federal de Pernambuco.

Instituição: Universidade Federal do Sul e Sudeste do Pará.

Endereço: Folha 17, Quadra 04, Lote Especial, s/n.º - Nova Marabá, PA, Brasil.

E-mail: wmvj@unifesspa.edu.br

**Hugo Kuribayashi**

Doutorando em Engenharia Elétrica Enf. Computação Aplicada

Instituição: Universidade Federal do Sul e Sudeste do Pará.

Endereço: Folha 17, Quadra 04, Lote Especial, s/n.º - Nova Marabá, PA, Brasil.

E-mail: hugo@unifesspa.edu.br

**João Victor Costa Carmona**

Doutor em Engenharia Elétrica Enf. Computação Aplicada

Instituição: Universidade Federal do Sul e Sudeste do Pará.

Endereço: Folha 17, Quadra 04, Lote Especial, s/n.º - Nova Marabá, PA, Brasil.

E-mail: jvictor@unifesspa.edu.br

**RESUMO**

Apesar dos avanços recentes que tornaram o uso de UAVs (*Unmanned Air Vehicles*) em diversos cenários uma realidade, há vários desafios a serem enfrentados para explorar todo o seu potencial. O uso de *Fog Computing* se torna uma saída para auxiliar estes dispositivos móveis com energia e hardware limitados a terceirizar toda a capacidade de processamento para um recurso remoto na borda da rede. Este trabalho apresenta o desenvolvimento e implementação de um sistema de

levantamento de métricas para avaliação de desempenho da técnica de *offloading* com uso de drone e servidor *Cloudlet* em ambiente de *Fog Computing* para a transmissão e processamento de imagens através de algoritmo tradicional de detecção e reconhecimento facial em tempo real, em diferentes cenários, visando a análise do consumo energético do drone, ao longo do processo, a intensidade de sinal entre o drone e o orquestrador, e o tempo de processamento gasto em cada entidade, drone e máquinas virtuais.

**Palavras-chave:** UAV, *Fog Computing*, *Offloading* e *Cloudlets*.

## ABSTRACT

Despite the recent advances that made possible the use of Unmanned Aerial Vehicles (UAVs) on diversified scenarios, there are still several challenges to be faced in order to accomplish its full potential. The use of *Fog Computing* becomes an alternative to support UAVs devices with limited power and hardware to offload all processing power to a remote resource. This work presents the development and implementation of a metric survey system for performance evaluation of *offloading* technique using drone and *Cloudlet* server in *Fog Computing* environment for image transmission and processing through traditional detection and recognition algorithm facial in real time, in different scenarios to analyze drone energy consumption throughout the process, signal strength between drone and orchestrator, and processing time spent on each entity, drone and virtual machines.

**Key words:** UAV, *Fog Computing*, *Offloading* and *Cloudlets*.

## 1 INTRODUÇÃO

A fim de tratar os problemas de aplicativos sensíveis a privacidade, largura de banda alta, geograficamente dispersos, e de latência ultra baixa, há uma necessidade de um paradigma de computação que ocorra mais próximo dos dispositivos conectados. A *Fog Computing*, da tradução para Computação em Névoa, tem sido introduzido em 2012 pela Cisco para permitir o processamento de aplicações em bilhões de dispositivos conectados na borda da rede (OPENFOGCONSORTIUM, 2017). Consequentemente, a Computação em Névoa pode ser considerada um dos principais facilitadores da Internet das Coisas e aplicações de big data (BONOMI et al., 2012) uma vez que oferece: 1) baixa latência e reconhecimento de localização devido à proximidade dos dispositivos de computação com a borda da rede, 2) distribuição geográfica ampla, 3) interconexão com um número muito grande de nós (por exemplo, sensores sem fio), e 4) suporte para aplicativos de streaming e de tempo real (BAHTOVSKI; GUSEV, 2014).

Por outro lado, drone, por definição, compreende-se como um veículo aéreo não tripulado, também chamado de UAV (Unmanned Aerial Vehicle). É todo e qualquer tipo de aeronave que não necessita de pilotos embarcados para ser guiada. Desse modo, esses aviões são controlados à distância por meios eletrônicos e computacionais sob a supervisão humana ou sem a sua intervenção, através de métodos sofisticados como IA (Inteligência Artificial). Inicialmente idealizados para fins militares, estas máquinas voadoras de última geração foram concebidas, projetadas e construídas a fim de serem utilizadas em missões muito perigosas para seres humanos. Assim, com sua arquitetura

principal composta de sistemas de voo, controle, monitoramento, processamento de dados e de pouso, os drones atuais são capazes de prover um leque variado de funcionalidades, tais como: navegação, aquisição de dados, processamento de imagens e transferência de informações para estações em solo.

Os UAVs podem ser usados em muitas aplicações civis devido à sua facilidade de implantação, baixo custo de manutenção, alta mobilidade, e capacidade de pairar. Esses veículos estão sendo utilizados para monitoramento em tempo real do tráfego rodoviário, fornecendo cobertura sem fio, sensoriamento remoto, operações de busca e salvamento, entrega de mercadorias, segurança e vigilância, e inspeção de infraestrutura civil (HAYAT; YANMAZ; MUZAFFAR, 2016).

Nesse sentido, a arquitetura de Computação em Névoa vem para fazer uma ponte entre dispositivos de IoT (tal como um drone, por exemplo) e a nuvem, permitindo gerenciamento computacional, de rede, armazenamento e de dados nos nós da rede próximos ao dispositivo de IoT, como mostrado por (YOUSEFPOUR et al., 2019). Computação em Névoa é definido pelo *OpenFog Consortium* como “*uma arquitetura horizontal de sistema que distribui poder computacional, armazenamento, funções de controle e rede próximo aos usuários, ao longo de todo o escopo que envolve nuvem e demais dispositivos*”. Desse modo, a integração entre dispositivos de IoT e Computação em Névoa se mostra promissora, visto que gera um grande leque de facilidades e oportunidades para cenários de aplicações reais, podendo, a título de exemplo, delegar funções de um drone para o dispositivo presente na névoa (OPENFOGCONSORTIUM, 2017).

## 2 TRABALHOS RELACIONADOS

Existe na literatura diversas pesquisas sobre a utilização da técnica de *offloading* computacional que envolve UAVs em cenários de ambiente integrado com paradigmas de computação em borda, tais como a MEC (Mobile Edge Computing), MCC (Mobile Cloud Computing) e FOG.

O trabalho (LIMA; SILVA; VELLOSO, 2018) propõe uma experimentação de coleta de dados de dispositivos IEEE 802.11, de baixo custo, em uma ambiente de floresta com o auxílio de drones, visando uma análise de desempenho de uma rede sem fio, avaliando a capacidade da rede juntamente com o consumo energético do drone. Baseado na perspectiva da dificuldade de monitoramento de animais silvestre e da importância do mesmo para a área de ecologia, os autores propõe a implantação de um projeto denominado IoTWild, que se consiste na utilização de um drone em um ambiente de floresta, que sobrevoará próximo as copas das árvores, até se conectar com uma *cameraTrap*, entre várias, que estão distribuídas no ambiente, e tem como função a captação de imagens e a transmissão dos dados coletados para o drone, na qual o drone permanecerá estabilizado

no ar, somente alterando seu estado quando não houver mais dados para o dispositivo enviar para o drone, ou quando sua bateria sinalizar baixa carga energética.

Os autores (MOTLAGH; BAGAA; TALEB, 2017) propõem uma plataforma integrada de um *testbed*, na qual possui um drone como dispositivo de IoT, controlado por uma unidade orquestradora, e com um serviço de MEC, com o objetivo de comparar entre os índices obtidos do resultado do processamento de um algoritmo de reconhecimento facial em cenários de grandes multidões, destinado a identificar pessoas suspeitas a partir do servidor MEC, com relação ao processamento local no drone, avaliando as métricas de tempo de processamento e o consumo energético. Baseado no pressuposto da necessidade de controle e reconhecimento de pessoas suspeitas, considerando cenários com grandes multidões, como shows e jogos esportivos, os autores defendem sua proposta na qual a utilização do drone integrado com a rede LTE, dispõe de possíveis soluções para a problemática, como o reconhecimento facial em tempo real, que no experimento é realizado tanto pelo drone como pelo servidor MEC, em vídeos capturados pelo drone, onde os resultados mostram uma eficácia em deslocar o processamento do algoritmo do drone para o servidor MEC, que influência diretamente em melhores resultados nas métricas da economia da bateria do drone e no tempo de resposta da aplicação.

O artigo (SARA; JAWHAR; NADER, 2016) apresenta uma proposta de implementação de uma camada de software baseada em SDN (*Software Defined Networking*) em um ambiente integrado por uma Rede de Sensores Sem Fio, da sigla WSN (*Wireless Sensor Network*), que dispõe em sua arquitetura: sensores de umidade, drones, um orquestrador e serviços em nuvem, na qual a WSN objetiva-se para monitorar em tempo real os índices de umidade em uma área de atividade agrícola, para se necessário ativar os drones que tem função de irrigamento do local. Os autores defendem ao longo do trabalho as vantagens obtidas com a utilização da arquitetura proposta, principalmente a eliminação das restrições oriundas de estruturas fortemente acopladas, na qual as redes SDN são fracamente acopladas e possuem características como escalabilidade, reusabilidade, configurabilidade, tolerância a falhas, baixo custo e confiabilidade.

O trabalho (LUO et al., 2015) propõe a utilização de serviços de nuvem para fins de processamento de algoritmos de detecção de desastres. No experimento, os autores buscam identificar o derramamento de óleo em ambientes marinhos, em vídeos coletados por UAVs interligados por uma unidade de controle com conexão sem fio, em uma arquitetura cliente-servidor, na qual o cliente está hospedado no UAV, que coleta dados de vídeo e informações de contexto de seu *hardware*. A solução proposta pelos autores busca tanto a transferência do processamento do algoritmo de detecção e reconhecimento para o servidor em nuvem, como também a diminuição dos dados trafegados entre o drone e a estação base de controle, utilizando de uma máquina virtual suportada na nuvem,

que contém uma unidade de reconstrução dos vídeos, filtrados pelo drone após a captura, antes de passar para o algoritmo, diminuindo assim, significativamente a quantidade de bytes transmitidas e armazenadas no drone, e conseqüentemente o consumo energético do mesmo.

O artigo (VALENTINO; JUNG; KO, 2018) apresenta uma proposta de compartilhamento de poder computacional entre *clusters* de UAVs, onde uma entidade inserida no conjunto desempenhará a função que os autores denominam de *cluster head*, que controlará as ações e decisões do resto do *cluster*. Por fim, o trabalho (MOHAMED et al., 2017) apresenta uma análise de uma simulação da utilização de uma plataforma de Fog baseada em UAV para aplicativos de IoT, com o objetivo de agregar as vantagens e os recursos de ambas as tecnologias, Computação em Névoa e UAVs, para oferecer suporte eficaz a alguns aplicativos de IoT. Os autores da obra apresentam uma arquitetura de UAVFog, para aplicações de dispositivos de IoT, na qual fora implementada um protótipo em ambiente distribuído de agentes Java para o UAVFog orientado a serviços.

Conforme descrito nos parágrafos anteriores, grande parte das propostas visam melhorar as operações de *offloading* computacional a partir de duas principais métricas de medição: energia e tempo de processamento; cujo o ambiente de avaliação é bastante heterogêneo, o que revela a diversidade da aplicação de UAVs em sinergia com a paradigmas de computação na borda. Baseado neste dados, podemos concluir que nenhuma das propostas contempla o desenvolvimento e implementação de um sistema de monitoramento capaz de coletar e armazenar métricas que mensuram o desempenho das três elementos de um ambiente de Computação em Névoa, Dispositivo UAV, Rede Sem fio, e *Cloudlet*. Além do mais, nenhum dos trabalhos anteriormente mencionados, avaliaram o desempenho de aplicações complexas, tais como detecção e reconhecimento facial, em ambientes internos e abertos, considerando diversos fatores, tais como distância, tamanho das imagens, e barreiras.

### 3 PROPOSTA

#### 3.1 ARQUITETURA DE MONITORAMENTO

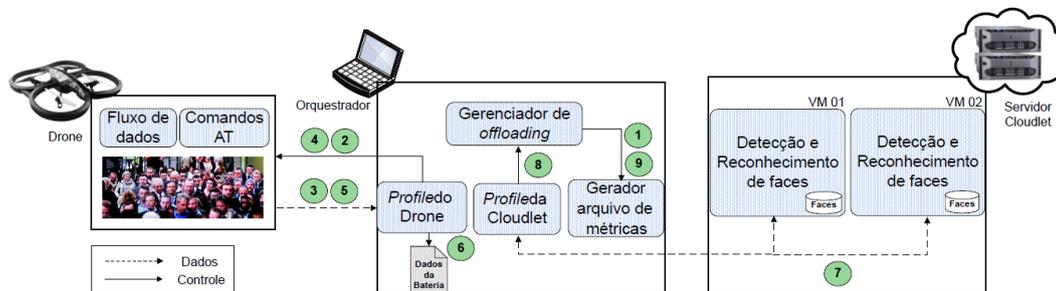
A Figura 1 apresenta a arquitetura utilizada no experimento, ilustrando os relacionamentos e ações feitas pelas entidades, que são identificadas de forma cronológicas pelos índices numéricos de 1 a 8 (círculos verdes), no qual as relações descritas que utilizando-se a seta pontilhada representam transmissão de dados, e as setas sólidas representam comunicação e controle. A proposta define três camadas de escopo: 1ª dispositivos inseridos no contexto de IoT (UAV, Smart TVs, Apple Watch e Wearables fitness), 2ª processos de monitoramento e registro de informações, gerenciamento de atividades e fluxo de informações e fluxo de dados e controle de dispositivos, e a 3ª Servidores *Cloudlet* para fins de processamento de imagens.

**Drone:** Captura e transmite imagens de diferentes tamanhos. Possui dois componentes: fluxo de dados e comandos AT. O componente fluxo de dados permite a captura e transmissão de imagens e *stream* de vídeo entre o drone e o orquestrador. Comandos AT é destinado para controle e configuração do drone em comunicação com o Orquestrador.

**Orquestrador:** Controla as ações e movimentos do drone, monitora seus respectivos índices de atividade, gerencia e manipula as informações coletadas e transmitidas durante todo o processo. Possui quatro componentes: gerenciador de *offloading*, gerenciador de arquivo de métricas, *profile* do drone e *profile* da *Cloudlet*. O gerenciador de *offloading*. O gerenciador de arquivo de métricas realiza ações relacionadas a criação e registro das informações coletadas, assim como atualizações e manipulação das mesmas. O *profile* do drone manipula e gerencia recursos e informação pertinentes ao drone (bateria, taxa de vazão e RSSI), assim como recursos de controle do mesmo. O *profile* da *Cloudlet* realiza comunicação da nuvem com o orquestrador.

**Servidor Cloudlet:** Recursos de computação em nuvem destinados a processamento dos algoritmos de detecção e reconhecimento facial. Possui dois componentes: detecção e reconhecimento de faces na máquina virtual um (recursos de hardware baixo) e a máquina virtual dois (recursos de hardware alto).

**Figura 1** - Arquitetura de monitoramento

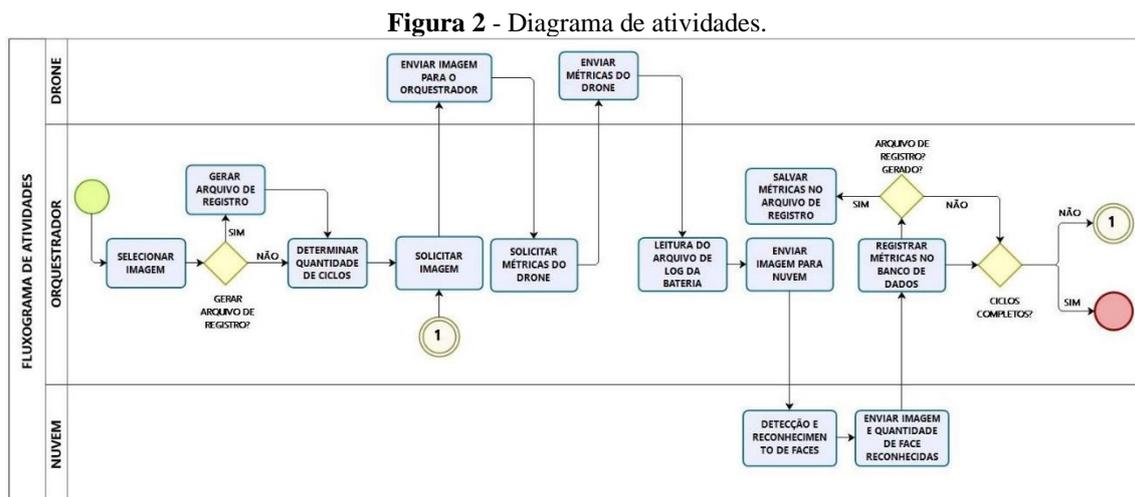


De acordo com a Figura 1, as entidades com seus componentes fazem interação sequencial a partir de vários fatores, apresentados a seguir: 1 - Criação do arquivo de registro. 2 - solicitação de envio da imagem (1, 2 ou 3) do drone para o orquestrador. 3 - Envio da imagem solicitada no segmento anterior, no canal de comunicação entre o drone e o orquestrador. 4 - Execução dos comandos para captura dos valores da vazão, tamanho da imagem, RSSI e tempo de transferência da imagem. 5 - Envio dos valores da vazão, tamanho da imagem, RSSI e tempo de transferência da imagem no canal de comunicação entre o drone e o orquestrador. 6 - leitura do arquivo de log com a informação do percentual da bateria do drone. 7 - Envio da imagem recebida do drone para a nuvem, onde é realizada a detecção e reconhecimento de faces e posteriormente a imagem é devolvida ao drone pela nuvem juntamente com o valor de quantidades reconhecidas na imagem. 8 - Recebimento

da imagem processada na nuvem. 9 - Inserção dos valores capturados no arquivo de registro e registro dos valores em banco.

### 3.1.1 Diagrama: Atividades

A Figura 2 apresenta o diagrama de atividades do sistema de monitoramento, ilustrando o fluxo de interação entre as entidades citadas na arquitetura da proposta, mostrando todo ciclo de vida de execução da solução e seu comportamento para coleta das informações necessárias, criação e manipulação do arquivo de registro, relacionamento entre drone, orquestrador e nuvem. A sequência de interações e ações mostrada pela figura segue da seguinte forma, com suas respectivas definições: Geração do arquivo de registro, fase onde é gerado. Solicitar imagem. Enviar imagem para o orquestrador. Solicitar métricas do drone. Enviar métricas do drone. Leitura de arquivo de log da bateria. Enviar imagem para a nuvem. Detecção e reconhecimento de faces. Enviar imagem e quantidade de faces reconhecidas. Salvar valores no arquivo de registro.



## 4 METODOLOGIA

### 4.1 AMBIENTE DE DESENVOLVIMENTO

Para a realização dos experimentos deste trabalho, dois softwares foram desenvolvidos: aplicação para executar juntamente com o orquestrador e aplicação para executar juntamente na *Cloudlet*. Sendo estas interagindo por meio de *sockets*. Estas aplicações executam tarefas tais como, controle, transmissão de dados, detecção e reconhecimento facial, coleta de informações, e comunicação entre diferentes dispositivos. A Tabela 2 apresenta os dispositivos utilizados para a realização dos experimentos. São descritos quanto suas especificações de hardware e de software.

Tabela 1 – Configuração de hardware e software nos ambientes de execução.

Dispositivo	Hardware utilizado	Software utilizado
Drone	Parrot AR.Drone 2.0 Elite Edition; Processador: ARM Cortex A8 1 GHz 32-bit com DSP video 800 MHz TMS320DMC64x; RAM: DDR2 1 GB @ 200 MHz; USB: High-speed USB 2.0; Wi-Fi b/g/n; Bateria: 1000mAh	Linux 2.6.32
Orquestrador	Processador: Intel (R) Core i5-4300M CPU @ 2.60GHz x 4; RAM: 4GB;	Ubuntu 18.04.2 LTS NetBeans 8.2 CV Drone
Cloudlet	<b>VM1:</b> vCPU E5-2623 v3 @ 3.00GHz x 2; Cache de 10MB; RAM: 4GB;	Ubuntu 18.04.1 LTS OpenCV 3.2.0 Eclipse v.4.10.0
	<b>VM2:</b> vCPU E5-2623 v3 @ 3.00GHz x 4; Cache de 10MB; RAM: 8GB;	Ubuntu 18.04.1 LTS OpenCV 3.2.0 Eclipse v.4.10.0

O OpenCV é uma biblioteca multiplataforma *open-source* que foi desenvolvida para atuar em aplicações de visão computacional. Através dela foi possível realizar o processo de detecção e reconhecimento facial das imagens enviadas pelo orquestrador. Nesse processo, a imagem quando chega em uma das VM's é automaticamente capturada de seu diretório por um *script* Java que, primeiramente, com o auxílio de um algoritmo classificador previamente treinado advindo da biblioteca, realiza a detecção de todas as faces presentes naquela imagem. Ao realizar esse processo e armazenar todas as faces detectadas, um outro algoritmo classificador previamente treinado também advindo biblioteca, realiza um primeiro processo de reconhecimento, conferindo a cada imagem armazenada um rótulo identificador. Após terminado essa etapa, um segundo processo de reconhecimento é realizado para verificar se os rótulos das imagens são equivalentes aos presentes no algoritmo de reconhecimento e, dessa forma, possibilitar uma avaliação do desempenho do algoritmo classificador.

Na Figura 4a mostra o primeiro cenário em ambiente interno, que foi realizado no galpão de laboratórios da UNIFESSPA, tanto os testes com barreiras, quanto os sem barreiras. Nos testes sem barreiras utilizou-se a área aberta do galpão, já no com barreira também utilizou-se a área aberta do galpão para voo do drone e uma sala próxima onde o orquestrador foi posicionado, deixando a parede como barreira entre os dois dispositivos. A Figura 4b mostra a área externa do galpão e dos blocos de salas do campus, ressaltada a necessidade da localização dos testes apresentarem proximidade com relação aos blocos e prédios que se dá pelo motivo da conexão cabeada entre o orquestrador e os servidores *Cloudlet*, na qual foi executado os testes de ambiente externo, utilizando as árvores como barreiras entre o drone e o orquestrador.

Figura 3 – Montagem do equipamento para testes.



- (a) Drone com o orquestrador  
 (b) Drone iniciando voo em ambiente aberto sem barreiras.

A Tabela 3 mostra informações detalhadas dos parâmetros utilizados, como o tamanho das imagens utilizadas, apresentando os valores do tamanho em megapixels (MP), a quantidade de faces, o algoritmo usado, o número de execuções por teste, e as distâncias aplicadas nos experimentos.

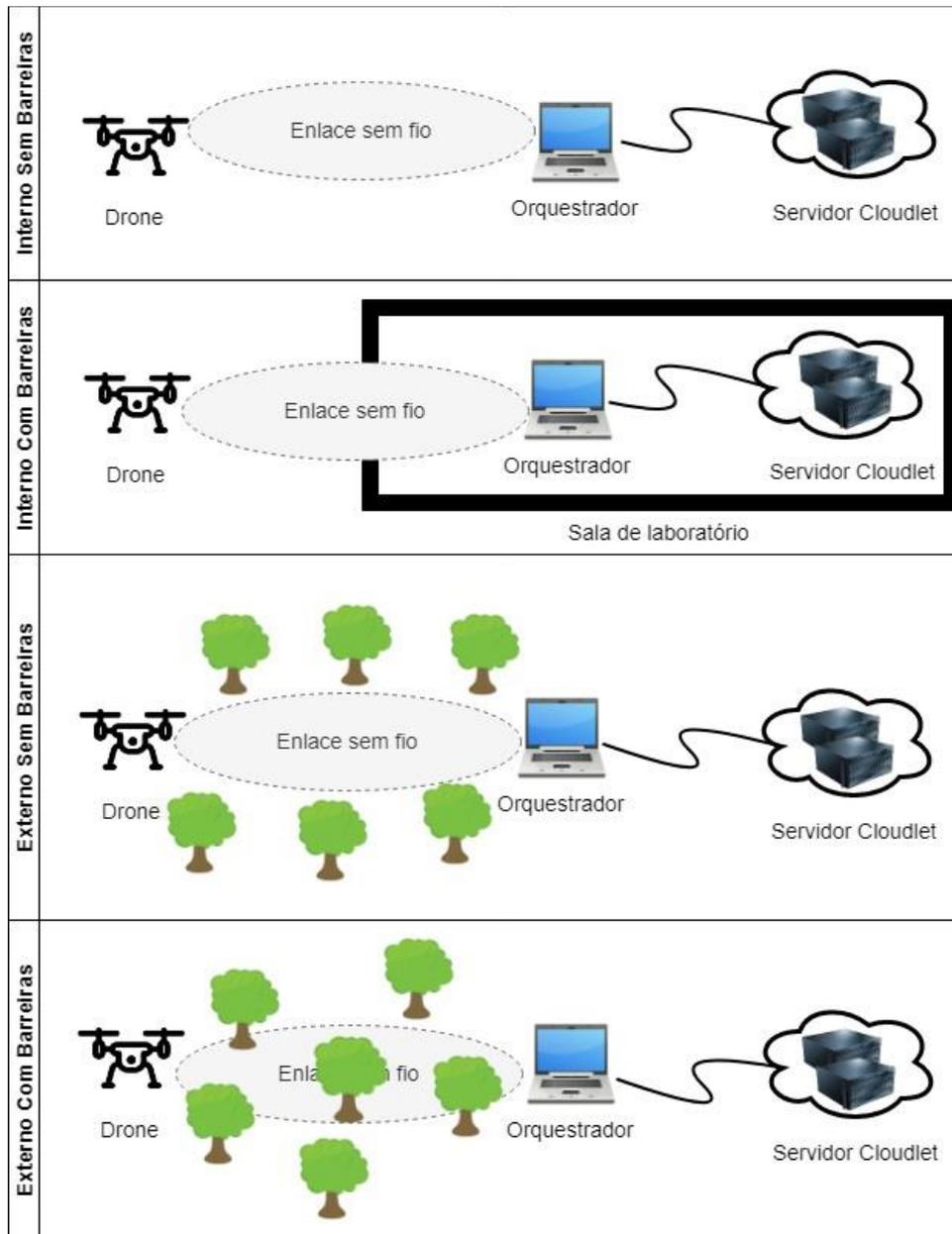
Tabela 2 – Parâmetros do Experimento.

Tamanho da imagem (MP)	Quantidade de faces	Algoritmo de detecção e reconhecimento	Número de execuções	Distância (m)
1,5	68	Haar Cascades	30	5
7				10
14				15
				20

## 5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

A Figura 5 apresenta os quatro cenários utilizados no experimento, na qual os dois primeiros caracterizam-se por um ambiente interno, realizado no interior do espaço físico do galpão de laboratórios da UNIFESSPA, um sem barreira entre o drone e o orquestrador, o outro apresentando uma parede como obstáculo entre o drone e o orquestrador. Os dois últimos cenários configuram-se por um ambiente externo, realizado nas áreas abertas da unidade II da UNIFESSPA, em Marabá-PA, na qual utilizou-se de áreas abertas sem obstáculos para o experimento sem barreira, porém com árvores e prédios em torno do local, e áreas com presença de árvores como obstáculos para o experimento com barreira.

Figura 4 - Os quatro ambientes experimentais.



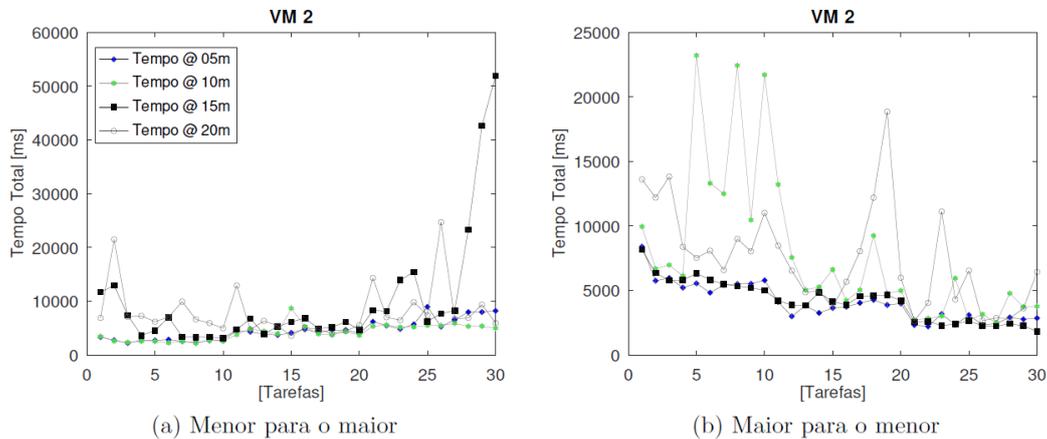
## 5.1 AMBIENTE INTERNO

### 5.1.1 Experimento 1 - Interno sem barreiras

A Figura 6a e 6b representam os resultados obtidos da métrica Tempo Total de Processamento (em milissegundos) na VM2 (resultados da VM1 não foram apresentados devido a semelhança com os da VM2), ou seja, é o tempo que correspondente ao envio do comando para o drone enviar a imagem com as faces, mais o tempo de processamento do drone, tempo de *download* da imagem do drone para o orquestrador e o tempo de processamento na Máquina Virtual. A primeira Figura apresenta sequência de mudança de imagens na execução das tarefas partindo da imagem de menor tamanho (1,5 MP) para a imagem de maior tamanho (14 MP), ou seja, no intervalo de 1 a 10 a imagem

de tamanho 1,5 MP foi executada 10 vezes, e assim sucessivamente para as demais imagens. Já na segunda Figura, a variação das imagens na sequência de execução das tarefas ocorreu iniciando com a maior imagem, de 14 MP, até a menor imagem de 1,5 MP.

**Figura 5 - Tempo total em ambiente interno sem barreiras.**



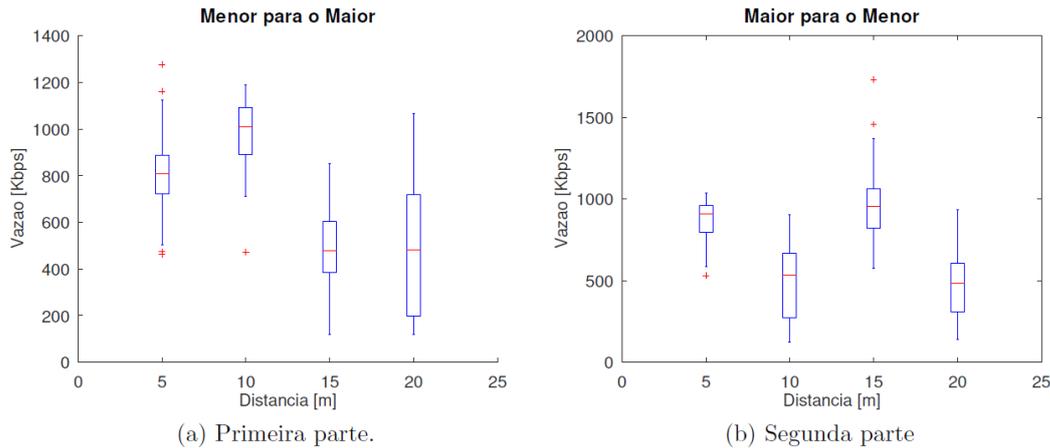
A Figura 6a possibilita entender a influência nos resultados coletados, ocasionada pela variação da distância entre o drone e o orquestrador, na qual houve um aumento acentuado no tempo total nas 3 últimas tarefas para a distância de 15 metros, diferentemente dos resultados para as distâncias de 05 e 10 metros, que apresentam comportamentos mais lineares ao longo da execução das 30 tarefas.

Os resultados levantados para o experimento de mudança de imagens na sequência de execução de tarefas, do maior para o menor, ilustrados na Figura 6b, mostram que alguns dados obtiveram comportamentos distintos e não lineares, como a linha de resultado para a distância de 05 metros, na qual os tempos coletados no intervalo da 1ª a 10ª tarefa, são maiores em comparação aos valores obtidos para a distância de 20 metros, que podem ser explicados por fatores tais como movimentação não controlada do drone, que realiza quando tentar se manter plano em um local, podendo distanciar ou aproximar do orquestrador, interferência de sinais por diversos aparelhos eletrônicos, pois o local de realização do experimento apresenta, mesmo que pouco, a movimentação de docentes, discentes e outras pessoas, portando equipamentos como computadores e celulares, assim como a presença de roteadores e outros dispositivos da própria infraestrutura de rede da universidade.

As Figuras 7a e 7b representam os resultados obtidos da vazão de transmissão das imagens entre o drone e o orquestrados, com a mesma variação de tamanho das imagens na sequência de execução das tarefas que as figuras 6a e 6b, respectivamente. Dado os resultados mostrados acerca do valores coletados da comunicação entre o drone e o orquestrador, pode-se afirmar que houve uma

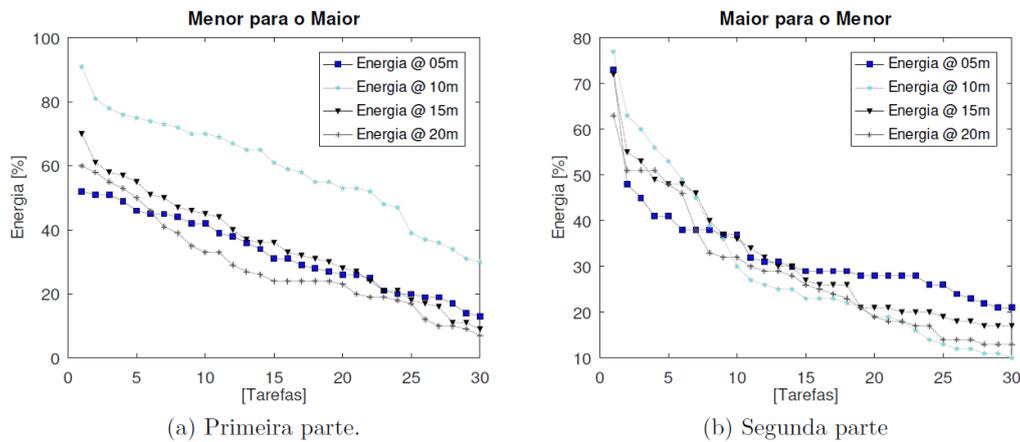
menor concentração de dados, para os resultados coletados na primeira imagem, na qual os valores para a distância de 5 metros, pertencem a uma faixa de variação entre 450 a 1300 Kbps, com o valor da média de 800 Kbps, que em comparação aos dados obtidos para a mesma distância, mas com sentido de mudança da imagem contrário, onde os valores pertencem a faixa de variação entre 500 a 1000 Kbps, com o valor da média de 900 Kbps.

**Figura 6 - Vazão em ambiente interno sem barreiras.**



As Figuras 8a e 8b representam os resultados obtidos para a métrica de consumo de energia da bateria do drone, para a variação das imagens na sequência de execução de tarefas, no sentido da menor imagem para o maior, no primeiro gráfico, e no sentido da maior imagem para o menor no segundo gráfico, ilustrando os valores coletados para as distâncias de 05, 10, 15 e 20 metros entre o drone e o orquestrador. Para os resultados coletados no primeiro gráfico é perceptivo um padrão de comportamento semelhante para as quatro variações de distância, apresentando, em determinados intervalos entre as tarefas, consumo energético uniforme, já os resultados obtidos da Figura 8b apresenta um maior valor de consumo energético para as quatro variações de distância no intervalo de tarefas de 0 a 10, na qual se utiliza a figura de maior tamanho, e sucessivamente a figura mostra uma maior uniformidade no comportamento dos resultado para as demais tarefas.

Figura 17 - Consumo energético em ambiente interno sem barreiras.



Com os resultados de tempo de processamento, consumo energético e vazão, é possível compreender alguns pontos, como a influência causada pelo aumento da distância entre o drone e orquestrador nas métricas de interesse, na qual ocasiona impacto direto no na intensidade de sinal da conexão entre drone e orquestrador, no tempo de processamento relacionado à comunicação entre as duas entidades, e no consumo energético da bateria do drone, pois quando uma tarefa consome mais tempo para finalizar, ocasiona uma acentuação para baixo na curva dos resultados de consumo energético, uma vez que o drone permanece em estado do voo, ou seja, esta sempre consumindo bateria para manter-se planando. Outro ponto a se destacar no experimento é a identificação de uma maior dificuldade de comunicação entre drone e orquestrador nas primeiras tarefas em todos os testes, quando o sentido de mudança da imagem é da maior para a menor imagem, evidenciado principalmente nos resultados obtidos para consumo energético e tempo de processamento, na qual a faixa de intervalo da 1<sup>a</sup> a 10<sup>a</sup> tarefa, apresenta dados piores que os demais.

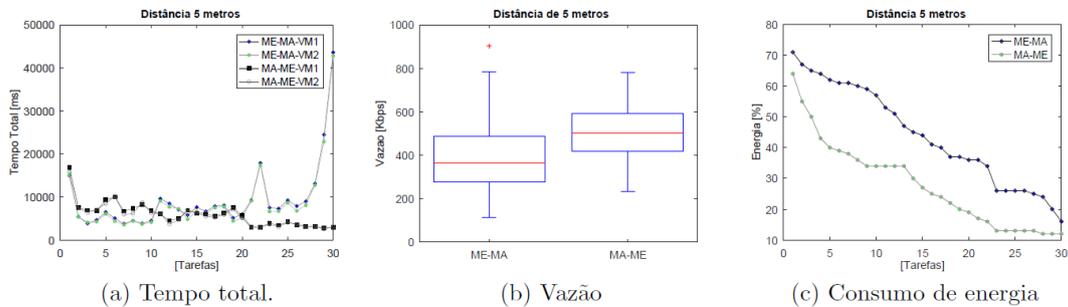
### 5.1.2 Experimento 2 - Interno com barreiras

As Figuras 9a, 9b e 9c representam os resultados obtidos para as métricas de tempo de processamento, vazão e consumo energético, respectivamente, para o ambiente interno com barreiras, com a distância entre os dispositivos drone e orquestrador em 5 metros. Não foram obtidos resultados para as distâncias de 10, 15 e 20 metros entre o drone e o orquestrador, pois os dois dispositivos não conseguiam realizar conexão entre eles.

A Figura 9a apresenta os resultados das VMs 1 e 2, com as mudanças de imagens na sequência de execução de tarefas, do maior para o menor e no sentido contrário, é possível compreender uma maior dificuldade de transmissão entre o drone e o orquestrador para a imagem de maior tamanho, evidenciado nos intervalos das tarefas de 0 a 10 para as sequência de mudança de imagens de maior para menor, nas VM01 e VM02, na qual possui resultados maiores de tempo de processamento em

comparação ao sentido de mudança contrário, e nos intervalos de 20 a 30 para as sequência de mudança de imagens de menor para maior, nas VMs 1 e 2, que também apresenta resultados expressivamente maiores em relação ao sentido contrário de mudança de imagens.

**Figura 8** - Tempo, vazão e energia em ambiente interno com barreiras.



A Figura 9b ilustra os resultados obtidos da vazão (taxa de transferência) entre o drone e o orquestrador, para as duas sequências de mudança de imagem, na qual a sequência de mudança da menor imagem para a maior, obteve resultados inferiores em comparação a sequência do sentido contrário, de maior imagem para menor, uma vez que a primeira apresenta dados mais dispersos em relação a segunda, inseridos em um intervalo de valores de aproximadamente 100 a 800 Kbps, com média de aproximadamente 400 Kbps, já a segunda mostra resultados mais concentrados e contidos em uma faixa de valores maiores, com intervalo de aproximadamente 200 a 800 Kbps, com sua média em 500 Kbps.

A Figura 9c mostra os resultados coletados do consumo energético da bateria do drone em relação as tarefas realizadas, para as sequências de mudança de imagens, no sentido de maior para menor e de menor para maior. O gráfico apresenta uma consumação acentuada no intervalo de tarefas de 0 a 5 no sentido de mudança da imagem de maior para menor, na qual está correlacionado com os resultados obtidos do tempo de processamento, ilustrados na Figura 9a, uma vez que quanto maior o tempo de processamento em determinada tarefa, seja o tempo da VM01, VM02 ou do drone, implica diretamente na diferença de valor do percentual da bateria do drone coletado de uma tarefa com relação a seguinte.

Para os resultados obtidos no experimento com cenário de teste interno com barreiras, fora observado um aumento significativo na dificuldade do funcionamento da comunicação e transmissão de dados entre os dispositivos da arquitetura proposta, evidenciado principalmente pelo fato de não haver resultados para as distâncias de 10, 15 e 20 metros entre o drone e o orquestrador, onde os mesmos, não conseguiam estabelecer conexão entre eles. Outro fator identificado no experimento se dar pelos resultados negativos obtidos nas métricas de tempo de processamento e consumo energético, quando se utiliza a imagem de maior tamanho, apresentados por exemplo na Figura 9a, na qual o

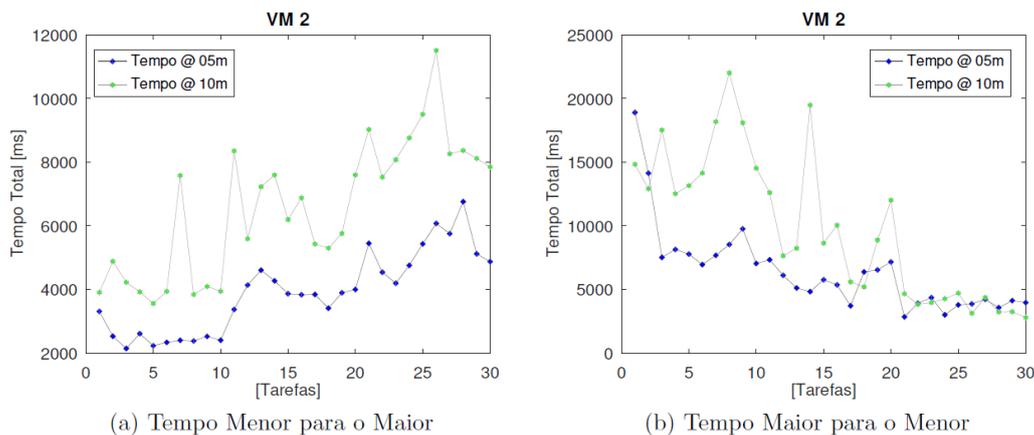
intervalo de tarefas entre 20 a 30 para os resultados de tempo de processamento, obtido das VMs 1 e 2, para a sequência de mudanças de imagens de menor para maior, mostra um aumento significativo nos valores colhidos, alterando de maneira visível o comportamento do gráfico, ocasionando, nesta faixa de intervalo, uma curva acentuada para cima.

## 5.2 AMBIENTE EXTERNO

### 5.2.1 Experimento 3 - Externo sem barreiras

As Figuras 10a e 10b representam os valores coletados para a métrica de tempo de processamento da VM2, para os testes com distância entre o drone e o orquestrado de 05 e 10 metros, para os sentidos de mudança da menor imagem para maior, e no sentido contrário, respectivamente. Pode-se perceber uma maior variação dos dados, se comparado com os resultado obtidos tanto no ambiente interno com barreiras e o interno sem barreiras, pois os gráficos apresenta muitas variações abruptas entre os valores na sequência de execução das tarefas, configurando um comportamento gráfico das linhas de resultados mais disperso e não linear. Também é perceptivo a obtenção de resultados com tempos maiores de processamento, quando a imagem utilizada fora a de maior tamanho, ou seja, nos intervalos das tarefas em execução de 20 a 30, para a Figura 10a, e no intervalo das tarefas em execução de 1 a 10, para a Figura 10b.

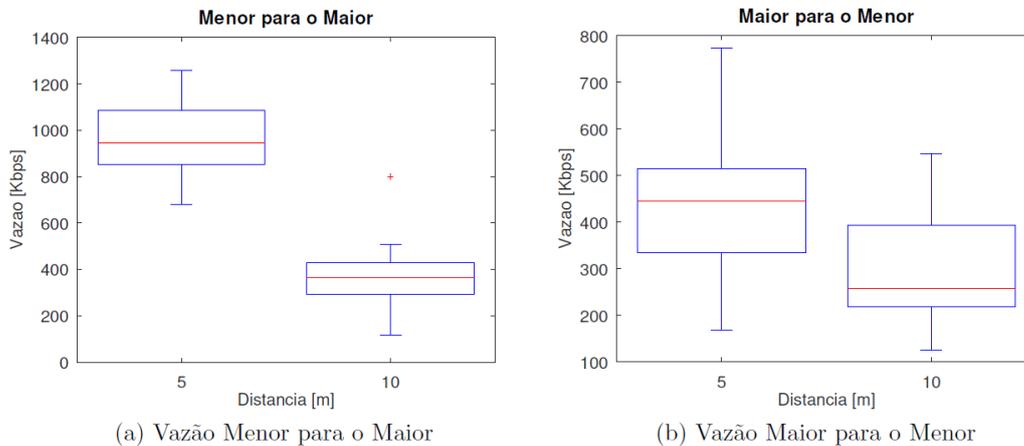
**Figura 9** - Resultados referentes ao tempo em ambiente externo sem barreiras.



As Figuras 11a e 11b apresentam os resultados obtidos para a vazão da comunicação entre drone e orquestrador nas distancia de 5 e 10 metros, com a sequência de mudança da imagem da menor para a maior, na primeira Figura, e da maior para a menor, na segunda Figura. Os resultados mostram dados melhores para a primeira Figura, quando a sequência de mudança da imagem é de menor para maior, em relação a segunda. Quando a alteração das imagens se dá da maior para a menor, tendo os resultados da Figura 11b menores valores e maior dispersão dos dados, com uma

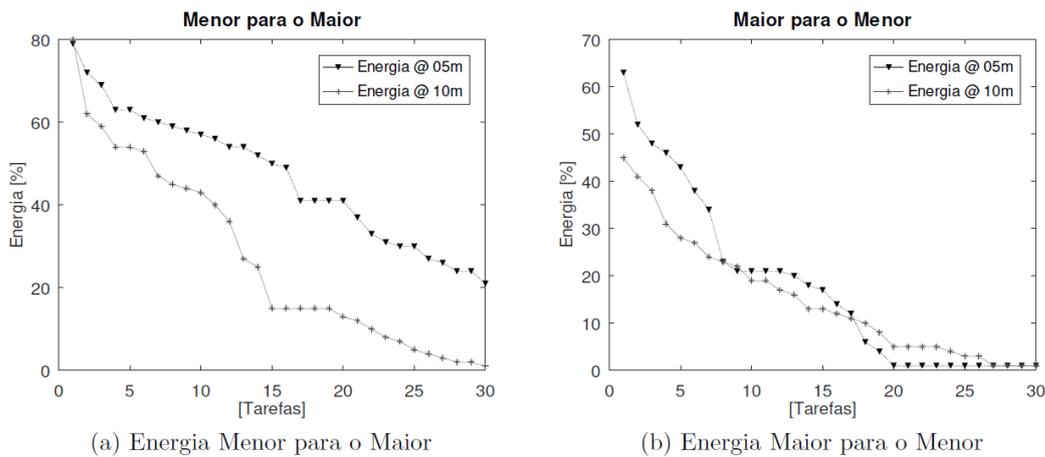
faixa de intervalo de aproximadamente 200 a 800 Kbps, e média de 450 Kbps, para a distância de 5 metros entre o drone e o orquestrador, em relação a Figura 11a, que para a mesma distância, possui uma faixa de intervalo de aproximadamente 700 a 1300 Kbps com média de 1000 Kbps.

**Figura 102** - Resultados referentes a vazão em ambiente externo sem barreiras.



As Figuras 12a e 12b mostram os resultados coletados para a métrica de consumo energético para as distâncias de 05 e 10 metros entre o posicionamento do drone em relação ao do orquestrador, ilustrando os valores obtidos para a sequência de mudança da imagem da menor para a maior, na Figura 12a e a mudança da maior imagem para a menor na Figura 22b. Para os resultados ilustrados na primeira Figura, observa-se uma diferença considerável da consumação da bateria do drone entre a distância de 5 metros com a de 10 metros, na qual a primeira finaliza a sequência de execução das tarefas com aproximadamente 20% de bateria restante, já a segunda praticamente termina a sequência com a bateria do drone esgotada em 0%. Para resultados obtidos na segunda Figura, obteve-se dados piores que a primeira, na qual a acentuada curva no comportamento dos dados nas primeiras execuções da tarefas, no intervalo de 1 a 10, quando utilizava-se a imagem de maior tamanho, ilustra uma elevada consumação energética da bateria do drone, restando pouca carga energética para finalizar a sequência de execução, evidenciado nos resultados de 5 metros, na qual a carga da bateria do drone esgota-se ainda na tarefa 20.

Figura 11 - Resultados referente a energia em ambiente externo sem barreiras.



O experimento realizado em ambiente externo sem barreira, mostra uma maior dificuldade de comunicação, transmissão e voo por parte do drone, ocasionado principalmente pelos fatores climáticos adversos, que dependendo das circunstâncias impedem a qualquer tipo de atividade com UAVs, esta dificuldade fica mais visível após análise comparativa dos resultados para o ambiente externo em relação ao interno, pois para os teste sem barreiras de cada, os realizados em ambiente interno alcançaram a mediação de 20 metros de distância entre o drone e o orquestrador, já para o ambiente externo, só foi possível chegar a 10 metros de distância, tendo a bateria chegado ao limite durante a execução das tarefas.

### 5.2.2 Experimento 4 - Externo com barreiras

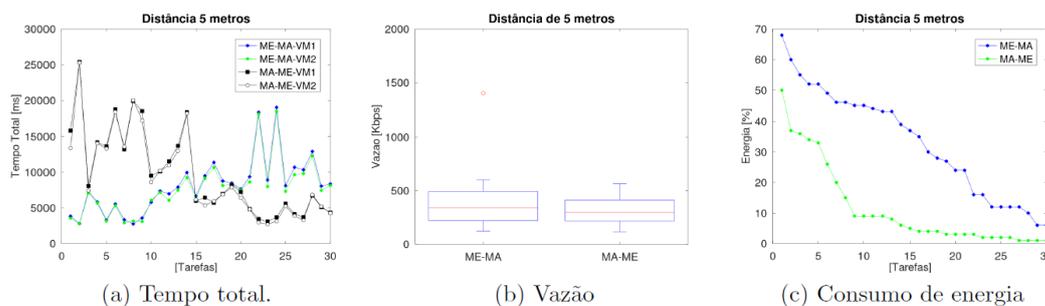
As Figuras 13a, 13b e 13c representam os resultados obtidos para as métricas de tempo de processamento, vazão e consumo energético, respectivamente, para o ambiente externo com barreiras, com a distância entre as entidades drone e orquestrador em 5 metros. Assim como no experimento interno, não foram obtido resultados para as distâncias de 10, 15 e 20 metros entre o drone e o orquestrador, pois os dois dispositivos não conseguiam realizar conexão entre eles.

A Figura 13a apresenta os resultados coletados para de tempo de processamento, em milissegundos com relação a sequência de execução das tarefas, de 1 a 30, para as VM1 e VM2, com as sequência de mudança da imagem da menor para a maior, e no sentido contrário. Os resultados ilustram um comportamento contrario dos dados para os resultados de mudanÇa de imagem da maior para a menor, em relação ao sentido contrário, na qual o primeiro apresenta comportamento pior no intervalo da sequência de execução das tarefas de 1 a 10, mas apresenta resultados melhores no intervalo de 20 a 30.

A Figura 13b mostra os resultados levantados da vazão de conexão entre o drone e o orquestrador, para as variações de imagens, da maior para a menor e no sentido contrário. Os resultados apresentam uma maior semelhança para os dois sentidos de mudança da imagem, se comparados aos resultados dos experimentos anteriores para a mesma métrica, na qual a faixa de variação dos dados são aproximadamente a mesma, de 100 a 600 Kbps.

A Figura 13c ilustra os resultados coletados do consumo energético da bateria do drone em relação as tarefas realizadas, para as sequências de mudança de imagens, nos sentidos de maior para menor e no sentido de menor para maior. Os resultados evidenciam uma maior consumação energética do drone nas tarefas em que se utiliza a imagem de maior tamanho, assim como em outros resultados já apresentado nesta monografia, tendo a bateria do drone chegado ao seu limite no intervalo entre as tarefas 20 e 25, para a sequência de mudança da imagem da maior para a menor.

**Figura 12 - Tempo, vazão e energia em ambiente externo com barreiras.**



O experimento ratifica, através de seus resultados, a maior dificuldade do drone em termos de processamento e transmissão, quando utiliza-se a imagem de maior tamanho, e possui os piores resultados de tempo de processamento, vazão e consumo energético, obtidos em todos os cenários de testes, o que também mostra a dificuldade do drone de comunica-se com o orquestrador, quando o mesmo está sobre condições climáticas adversas e quando existem obstruções entre o drone e o orquestrador.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um estudo sobre o desenvolvimento e implementação de um sistema de monitoramento de métricas para aplicação da técnica de *offloading* com a utilização de drone e servidores *Cloudlet*, em ambientes de computação em névoa para a transmissão e processamento de três imagens de tamanhos distintos com o algoritmo de detecção e reconhecimento facial, inserindo a arquitetura em quatro ambientes diferentes: ambientes internos, com e sem barreiras entre o drone e o orquestrador, e ambiente externo, com e sem barreiras, para experimentação e coleta dos resultados.

O trabalho mostrou a eficiência da aplicação da técnica de *offloading* para melhoramento da consumação energética de dispositivos inseridos em contexto de sistemas inteligentes. Com os resultados obtidos no trabalho, é possível entender os benefícios ocasionados pela transferência do processamento das imagens para as máquinas virtuais, porém sua utilização limita-se a condição de conexão com a rede utilizada para os experimentos, a rede da UNIFESSPA.

As diferentes configurações de cenários impactaram diretamente as métricas de interesse da proposta, tempo de processamento, vazão e consumo energético, onde os experimentos internos mostraram-se mais controlados e com menos interferências externas, logo obtiveram melhores resultados em relação aos experimentos em ambiente aberto, para os experimentos que utilizaram objetos como obstáculo entre o drone e o orquestrador, fora obtidos dados que demonstram a influência dos obstáculos com relação aos resultados, onde os valores da intensidade de sinal e da largura de banda da conexão entre o drone e o orquestrador, são piores em relação aos mesmos experimentos, porém em ambiente sem barreiras.

Como trabalhos futuros pretende-se melhorar a aplicação de auxílio do orquestrador com a alimentação de um banco de dados contextual para auxílio em tomadas de decisão de onde executar o algoritmo de detecção, baseados em informações das máquinas virtuais como porcentagem de CPU, memória e RTT (*round-trip time*), e inserir novos dispositivos com capacidade computacional (por exemplo, uma *microfog*) no contexto da arquitetura proposta, para maiores opções de tomada de decisão.

## REFERÊNCIAS

BAHTOVSKI, A.; GUSEV, M. *Cloudlet* challenges. **Procedia Engineering**, Elsevier, v. 69, p. 704–711, 2014.

BARBOSA, J. et al. Análise de desempenho da técnica de *offloading* de imagens com drone em ambiente de computação em névoa. In: SBC. **Anais do XVIII Workshop em Desempenho de Sistemas Computacionais e de Comunicação**.

BONOMI, F. et al. *Fog Computing* and its role in the internet of things. In: ACM. **Proceedings of the first edition of the MCC workshop on Mobile Cloud computing**. [S.l.], 2012. p. 13–16.

HAYAT, S.; YANMAZ, E.; MUZAFFAR, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 4, p. 2624–2661, 2016.

HU, P. et al. Survey on *Fog Computing*: architecture, key technologies, applications and open issues. **Journal of network and computer applications**, Elsevier, v. 98, p. 27–42, 2017.

JUNIOR, W.; SILVA, B.; DIAS, K. A systematic mapping study on mobility mechanisms for *cloud* service provisioning in mobile *cloud* ecosystems. **Computers & Electrical Engineering**, Elsevier Ltd, v. 0, p. 1–18, feb 2018. ISSN 00457906.

LIMA, C. M. d. A.; SILVA, E. A. da; VELLOSO, P. B. Avaliação de desempenho do ieee 802.11 para coleta de grande volume de dados em florestas com auxílio de drones. In: SBC. **17º Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance 2018)**. [S.l.], 2018. v. 17, n. 1/2018.

LUO, C. et al. A *uav-cloud* system for disaster sensing applications. In: IEEE. **2015 IEEE 81st Vehicular Technology Conference (VTC Spring)**. [S.l.], 2015. p. 1–5.

MACH, P.; BECVAR, Z. *Mobile Edge Computing*: A survey on architecture and computation offloading. **IEEE Communications Surveys & Tutorials**, IEEE, v. 19, n. 3, p. 1628–1656, 2017.

MOHAMED, N. et al. *Uavfog*: A *uav-based Fog Computing* for internet of things. In: IEEE. **2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)**. [S.l.], 2017. p. 1–8.

MOHAMED, N. et al. Unmanned aerial vehicles applications in future *smart cities*. **Technological Forecasting and Social Change**, Elsevier, 2018.

MOTLAGH, N. H.; BAGAA, M.; TALEB, T. *Uav-based iot platform*: A crowd surveillance use case. **IEEE Communications Magazine**, IEEE, v. 55, n. 2, p. 128–134, 2017.

OPENFOGCONSORTIUM. **Openfog reference architecture for Fog Computing**. 2017

PAN, J.; MCELHANNON, J. Future edge *cloud* and *Edge computing* for internet of things applications. **IEEE Internet of Things Journal**, IEEE, v. 5, n. 1, p. 439–449, 2017.

ROMAN, R.; LOPEZ, J.; MAMBO, M. *Mobile Edge Computing*, fog et al.: A survey and analysis of security threats and challenges. **Future Generation Computer Systems**, Elsevier, v. 78, p. 680–698, 2018.

SARA, M.; JAWHAR, I.; NADER, M. A softwarization architecture for uavs and wsns as part of the *cloud* environment. In: IEEE. **2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)**. [S.l.], 2016. p. 13–18

SHAKHATREH, H. et al. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. **IEEE Access**, IEEE, v. 7, p. 48572–48634, 2019.

VALENTINO, R.; JUNG, W.-S.; KO, Y.-B. Opportunistic computational *offloading* system for *clusters* of drones. In: IEEE. **2018 20th International Conference on Advanced Communication Technology (ICACT)**. [S.l.], 2018. p. 303–306.

YI, S.; LI, C.; LI, Q. A survey of *Fog Computing*: concepts, applications and issues. In: ACM. **Proceedings of the 2015 workshop on mobile big data**. [S.l.], 2015. p. 37–42.

YOUSEFPOUR, A. et al. All one needs to know about *Fog Computing* and related *Edge computing* paradigms: A complete survey. **Journal of Systems Architecture**, Elsevier B.V., n. February, 2019.

YU, W. et al. A survey on the *Edge computing* for the internet of things. **IEEE access**, IEEE, v. 6, p. 6900–6919, 2017.