



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

ANTONIO CÉSAR DE ANDRADE JÚNIOR

SIMULAÇÃO DE MODELO DE COMBATE A INCÊNDIOS FLORESTAIS
UTILIZANDO ENXAME DE VANTS

QUIXADÁ

2023

ANTONIO CÉSAR DE ANDRADE JÚNIOR

SIMULAÇÃO DE MODELO DE COMBATE A INCÊNDIOS FLORESTAIS UTILIZANDO
ENXAME DE VANTS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Arthur de Castro Callado

Coorientador: Prof. Dr. Thiago Werley Bandeira da Silva

QUIXADÁ

2023

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

A565s Andrade Júnior, Antonio César de.
Simulação de modelo de combate a incêndios florestais utilizando enxame de VANTs / Antonio César de Andrade Júnior. – 2023.
87 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Computação, Quixadá, 2023.
Orientação: Prof. Dr. Arthur de Castro Callado.
Coorientação: Prof. Dr. Thiago Werley Bandeira da Silva.

1. Drone. 2. Incêndio florestal. 3. Prevenção de incêndios. 4. LoRa. I. Título.

CDD 621.39

ANTONIO CÉSAR DE ANDRADE JÚNIOR

SIMULAÇÃO DE MODELO DE COMBATE A INCÊNDIOS FLORESTAIS UTILIZANDO
ENXAME DE VANTS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: ___/___/___

BANCA EXAMINADORA

Prof. Dr. Arthur de Castro Callado (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Thiago Werlley Bandeira da
Silva (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. André Ribeiro Braga
Universidade Federal do Ceará (UFC)

Prof. Dr. Marcelo Gomes da Silva Bruno
Instituto Tecnológico de Aeronáutica (ITA)

À minha família e amigos, por sempre me apoiarem nas minhas escolhas do caminho que estou seguindo.

AGRADECIMENTOS

Deus, obrigado por tuas bênçãos e por permitir que eu siga este caminho.

Agradeço aos meus pais, Antonia Alves e Antonio César, e minha família, por sempre apoiarem às minhas escolhas e estarem comigo nos momentos de felicidade e tristeza.

Aos meus orientadores na trajetória acadêmica, os professores Antônio Joel, Arthur Callado e Thiago Werlley, por toda dedicação, disposição, auxílio e orientação realizados a mim durante anos.

Aos meus amigos da turma de eletrônica e laboratório de robótica, Anderson Silva, Samuel Henrique, Erick Silva, Gabriel Moreira, Larissa Matos, Imário Almeida, Elias Filho, Abdul Hamid, Pedro Odillon, Pedro Botelho, Rafael Gonçalves, José Batista, Ítalo Moraes e Pedro Ítalo pelas loucuras e brincadeiras proporcionadas durante a realização de trabalhos e jogos de sinuca.

Aos meus amigos, praticamente irmãos, David Machado, Paulo Aragão, Anderson Moura e Javel Queiroz por serem minha segunda família.

Aos meus amigos do ensino médio, Allan, Antonio Francisco, Magayver, Kaynan, Gustavo, Lailson, Jonatas, Kerlly, Giovanna, Renata e Talyvia.

A todos os professores da Universidade Federal do Ceará, Campus Quixadá, que colaboraram no processo de minha formação profissional e pessoal.

Agradeço a todos que fizeram parte desse ciclo vivido ao longo destes 5 anos de graduação e 3 do ensino médio, e contribuíram de alguma forma na minha formação profissional e pessoal.

“Quando tudo tiver parecendo ir contra você,
lembre-se que o avião decola contra o vento,
e não a favor dele.”

(Henry Ford)

RESUMO

Cerca de 31% da superfície do planeta é coberta por área florestal, sendo um importante recurso para um quarto da população mundial e para economia. Uma das principais ameaças a esse recurso são os incêndios florestais. Estimando-se que mesmo após 15 anos da queima, as florestas não demonstram sinais de recuperação. O presente trabalho, propõe um modelo de combate a incêndios florestais como um complemento às formas de combate já utilizadas. O modelo consiste no uso de um enxame de drones e uma base de controle. Um grupo de drones de vigilância percorre uma área em busca de detectar algum foco de incêndio. Caso a detecção ocorra, um sinal com dados do incêndio é enviado por uma rede *LoRaWAN* para a base de controle. A base, então, mandará drones de supressão para o local do incêndio. O trabalho foca na tarefa de supressão, estimando, por meio de simulações, o impacto dos drones em incêndios. Além disso, também foi montada uma rede *LoRaWAN* utilizando computadores de placa única como nós do modelo. Para que esses objetivos fossem alcançados, foram seguidas as seguintes etapas: pesquisa com brigadistas e análise das respostas obtidas, análise do fluxo de água necessário para extinção completa de um incêndio, simulação da supressão, desenvolvimento da rede *LoRaWAN* e análise dos resultados obtidos. A pesquisa com os brigadistas revelou que os mesmos preferem o uso de drones para tarefas passivas (monitoramento e transporte de carga). Os resultados obtidos nas análises e simulações demonstram proximidade com o que, provavelmente, ocorreria em cenários reais e a possibilidade do uso de enxames como ferramenta auxiliar a outros métodos de combate, principalmente em momentos iniciais do incêndio. Também, foi mostrado uma possível arquitetura para rede *LoRaWAN*. Como futuros trabalhos, é pretende-se a adição de novas variáveis nas simulações, desenvolvimento da tarefa de detecção de focos de incêndio e novos testes na rede *LoRaWAN*.

Palavras-chave: drones; incêndio florestal; prevenção de incêndios; LoRa

ABSTRACT

Approximately 31% of the planet's surface is covered by forest area, serving as an important resource for a quarter of the global population and the economy. One of the main threats to this resource are wildfires. It is estimated that even after 15 years from burning, forests show no signs of recovery. This work proposes a model for combating forest fires as a complement to the existing methods of combat. The model involves the use of a swarm of drones and a control base. A group of surveillance drones traverses an area in search of detecting any signs of a fire. If detection occurs, a signal with fire data is sent via a LoRaWAN network to the control base. The base then sends suppression drones to the fire location. The work focuses on the suppression task, estimating, through simulations, the impact of drones on fires. Additionally, a LoRaWAN network was also set up using single-board computers as nodes in the model. To achieve these objectives, the following steps were taken: research with firefighters and analysis of the responses obtained, analysis of the water flow necessary for the complete fire extinction, simulation of suppression, development of the LoRaWAN network, and analysis of the results obtained. The research with firefighters revealed that they prefer the use of drones for passive tasks (monitoring and cargo transport). The results obtained in the analyses and simulations demonstrate proximity to what would likely occur in real scenarios and the possibility of using swarms as an auxiliary tool to other combat methods, especially in the early stages of a fire. Also, a possible architecture for the LoRaWAN network was shown. As future work, it is intended to add new variables in simulations, development of the task of detecting fire foci, and new tests on the LoRaWAN network.

Keywords: drones; forest fire; fire prevention; LoRa.

LISTA DE FIGURAS

Figura 1 – Estrutura de navegação de um robô	24
Figura 2 – Grafo para exemplificar o A-star	26
Figura 3 – Arquitetura de rede <i>LoRaWAN</i>	35
Figura 4 – <i>BeagleBone Black</i>	36
Figura 5 – <i>Orange Pi 5</i>	37
Figura 6 – Comparação do crescimento da área do incêndio com e sem os VANTs	39
Figura 7 – Exemplo de simulação de combate a incêndio florestal	40
Figura 8 – Modelo da organização da proposta	42
Figura 9 – Fluxograma dos procedimentos metodológicos	44
Figura 10 – Ângulos das células no AC	57
Figura 11 – Arquitetura de comunicação	59
Figura 12 – Circuito LoRa STM32	60
Figura 13 – Porcentagem das respostas para a primeira pergunta	64
Figura 14 – Porcentagem das respostas para a segunda pergunta	65
Figura 15 – Porcentagem das respostas para a terceira pergunta	65
Figura 16 – Porcentagem das respostas para a quarta pergunta	66
Figura 17 – FC em função do vento, umidade e profundidade das chamas	67
Figura 18 – FC em função da umidade, do vento e da profundidade das chamas	67
Figura 19 – mn em função do vento, do número de drones e da profundidade das chamas	68
Figura 20 – mn em função da umidade, de RoS, da profundidade das chamas e do número de drones	68
Figura 21 – Simulação do incêndio com velocidade do vento em $4m/s$, $5m/s$ e $6m/s$, respectivamente	69
Figura 22 – Área queimada em função do tempo	70
Figura 23 – Simulação do incêndio, com drones, com velocidade do vento $4m/s$, $5m/s$ e $6m/s$, respectivamente	71
Figura 24 – Dados da simulação do incêndio com velocidade do vento em $4m/s$, $5m/s$ e $6m/s$, respectivamente	72
Figura 25 – Saída do <i>Sender</i> (base)	73
Figura 26 – Saída do <i>Receiver</i> (drones)	73

LISTA DE TABELAS

Tabela 1 – Comparação entre <i>LPWAN</i> e outra tecnologias	33
Tabela 2 – Tabela comparativa	41
Tabela 3 – Parâmetros de FC	48
Tabela 4 – Parâmetros de AC	56
Tabela 5 – Valores dos coeficientes a, b, c e d	70

LISTA DE ALGORITMOS

Algoritmo 1 – Script em MATLAB para análise de FC	49
Algoritmo 2 – Script em MATLAB para análise de estimativa de metros de incêndio apagados (em função da profundidade do incêndio e umidade)	50
Algoritmo 3 – Classe que descreve um nó	51
Algoritmo 4 – Cálculo do valor de $h(n)$ e verificação se o nó pertence ao mapa	51
Algoritmo 5 – Primeira parte da função principal de <i>A-star</i>	52
Algoritmo 6 – Segunda parte da função principal de <i>A-star</i>	52
Algoritmo 7 – Terceira parte da função principal de <i>A-star</i>	53
Algoritmo 8 – Classe para as células do AC	54
Algoritmo 9 – Funções para desenhar o mapa	55
Algoritmo 10 – Funções de definição do AC	56
Algoritmo 11 – Classe de drone	58
Algoritmo 12 – Configuração dos módulos LoRa	61
Algoritmo 13 – Função de envio do <i>Sender</i>	62
Algoritmo 14 – Tratamento de mensagem no <i>Receiver</i>	63
Algoritmo 15 – Código principal da simulação do incêndio	80
Algoritmo 16 – Código principal da simulação do incêndio	81
Algoritmo 17 – Código principal da simulação do incêndio	82
Algoritmo 18 – Código principal da simulação do incêndio	83
Algoritmo 19 – Código principal da simulação do incêndio	84
Algoritmo 20 – Código principal da simulação do incêndio	85
Algoritmo 21 – Código principal da simulação do incêndio	86
Algoritmo 22 – Código principal da simulação do incêndio	87
Algoritmo 23 – Código principal da simulação do incêndio	88

LISTA DE ABREVIATURAS E SIGLAS

VANTs	Veículos Aéreo Não Tripulado
LoRaWAN	<i>Long Range Wide Area Network</i>
SBC	<i>Single Board Computer</i>
AC	Autômato Celular
IoT	<i>Internet of Things</i>
WLANs	<i>Wireless Local Area Network</i>
LPWAN	Low Power Wide Area Network
LoRa	<i>Long Range</i>
MAC	<i>Media Access Control</i>
IP	<i>Internet Protocol</i>
ARM	<i>Advanced RISC Machine</i>
USB	<i>Universal Serial Bus</i>
SPI	<i>Serial Peripheral Interface</i>
I2C	<i>Inter-Integrated Circuit</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
BBB	<i>BeagleBone Black</i>
TI	<i>Texas Instruments</i>
DDR3 SDRAM	<i>Double Data Rate Synchronous Dynamic Random Access Memory</i>
eMMC	<i>Embedded Multimedia Card</i>
HDMI	<i>High-Definition Multimedia Interface</i>
SD Card	<i>Secure Digital Card</i>
SO	Sistema Operacional
SSD	<i>Solid State Drive</i>
CNN	<i>Convolutional Neural Networks</i>

LISTA DE SÍMBOLOS

p_h	Probabilidade de propagação de chamas, sem influência de vento e inclinação do terreno
p_v	Probabilidade de propagação de chamas, influenciada pela velocidade do vento
p_d	Probabilidade de propagação de chamas, influenciada pela inclinação do terreno
p_{veg}	Probabilidade do tipo de vegetação
p_{den}	Probabilidade da densidade da vegetação
$c_1 c_2$	Constantes de influência da velocidade do vento na propagação do incêndio
U	Velocidade do vento em metros por segundo
θ	Ângulo entre a direção de propagação do incêndio e direção do vento, em radianos
a	Constante de influência do declive do terreno
θ_d	Ângulo do declive do terreno
E	Altura de inclinação de uma célula, em metros
l	Comprimento de uma célula
f	Fração de liberação de calor transferida de volta para a superfície do combustível por convecção e radiação
ΔH_c	Calor efetivo de combustão, ou seja, quantidade de energia liberada durante a combustão, em kJ/kg
kJ	Quilojoule
kg	Quilograma
L_v	Calor de gaseificação do combustível, em kJ/kg
\dot{m}''	Taxa de queima de massa por unidade de área de combustível, em kg/m^2s
m	Metro
s	Segundo
\dot{q}''_E	Fluxo de calor externo, taxa de calor transferido de uma fonte externa, em kW/m^2

\dot{q}''_P	Taxa da perda de calor da superfície, em kW/m^2
\dot{q}''_{agua}	Taxa de perda de calor por vaporização da água, em kW/m^2
kW	QuiloWatts
h	Coeficiente de transferência de calor por convecção, em kW/m^2
Y_{O_2}	Fração de massa de oxigênio
$\Delta H_{R,O_2}$	Calor de combustão por unidade de massa de oxigênio consumido, em kJ/kg
ϕ	Fração de perda de calor por convecção da chama necessária para extinguir a chama
c_p	Calor específico do ar em pressão constante, em kJ/kgK
K	Kelvin
η_{agua}	Eficiência da aplicação da água
$L_{v,agua}$	Variação de entalpia da água em $283K$ e do vapor de água em $373K$
$\dot{m}''_{agua,cr}$	Taxa crítica de aplicação da água, em L/m^2s
r_c	Componente radiativo por unidade de comprimento da frente de fogo
I	Intensidade da linha de fogo, kW/m
C	Comprimento da chama, em metros
P	Profundidade da zona de combustão ativa, em metros
τ	Transmissividade atmosférica
$\dot{q}''_{P,rad}$	Taxa de perda de calor por radiação
$\dot{q}''_{P,conv}$	Taxa de perda de calor por convecção
ε	Mede a capacidade do combustível de emitir radiação térmica
σ	Constante de <i>Stefan-Boltzmann</i> , em W/m^2K^4
W	Watts
T_c	Temperatura do combustível
T_a	Temperatura do ambiente
m/s	Metros por segundo
l/min	Litros por minuto
mAh	mili ampére hora

$kbps$	Kilobits por segundo
$Mbps$	Megabits por segundo
km	Quilômetro
KHz	Frequência
GHz	Frequência
MB	Megabytes
\dot{m}_{cr}''	Taxa de queima de massa no ponto de extinção da chama
FC	Taxa de fluxo de água necessária para extinguir um incêndio, em L/m^2s
T_g	Temperatura do gás, em K
W	Carga total do combustível
RoS	Taxa de propagação da frente de fogo
a, b, c	Parâmetros obtidos experimentalmente por análise de regressão não linear
M_d	Porcentagem de teor de umidade do combustível
D	Comprimento da chama em metros
n_n	Número de drones necessários para suprimir um número específico de metros lineares de frente de fogo ativa
Δt	Tempo de um drone ir da base até o incêndio
m_n	Número de metros lineares de frente de fogo ativa
L_d	Carga total carregada por um drone
T_b	Taxa de bit
FE	Fator de espalhamento
TC	Taxa de codificação
LB	Largura de banda
T_p	Tempo de transmissão de um pacote
$T_{preamble}$	Comprimento de um preâmbulo
TP	Tamanho de um pacote
$n_{preamble}$	Comprimento de um preâmbulo programado
H	Indicação de cabeçalho de um pacote

DE	Indicação de otimização de baixa taxa de dados
VRC	Verificação de erro de uma mensagem
p_m	Probabilidade de propagação das chamadas, influenciada pela umidade do combustível
t/ha	Toneladas por hectare
S	Conjunto de nós do <i>A-star</i>
S'_n	Conjunto de vizinhos de um nó n
S''_n	Conjunto de vizinhos abertos de um nó n
S'''_n	Conjunto de vizinhos fechados de um nó n

SUMÁRIO

1	INTRODUÇÃO	20
1.1	Objetivos	21
<i>1.1.1</i>	<i>Objetivo geral</i>	<i>21</i>
<i>1.1.2</i>	<i>Objetivos específicos</i>	<i>22</i>
1.2	Organização do Trabalho	22
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Enxame de Drones	23
2.2	Simulação	23
<i>2.2.1</i>	<i>Path Planning</i>	<i>23</i>
<i>2.2.2</i>	<i>A-star</i>	<i>24</i>
<i>2.2.3</i>	<i>Modelo de Autômato Celular</i>	<i>26</i>
<i>2.2.4</i>	<i>Método de Hansen</i>	<i>28</i>
<i>2.2.5</i>	<i>Estimativa do número de drones</i>	<i>31</i>
<i>2.2.6</i>	<i>OpenCV</i>	<i>31</i>
<i>2.2.7</i>	<i>DJI AGRAS T40</i>	<i>32</i>
2.3	Rede LoRaWAN	32
<i>2.3.1</i>	<i>LPWAN</i>	<i>32</i>
<i>2.3.2</i>	<i>Long Range</i>	<i>33</i>
<i>2.3.3</i>	<i>LoRaWAN</i>	<i>34</i>
<i>2.3.4</i>	<i>Módulo sx1278</i>	<i>35</i>
2.4	Single Board Computer	35
<i>2.4.1</i>	<i>BeagleBone Black</i>	<i>36</i>
<i>2.4.2</i>	<i>Orange Pi 5</i>	<i>37</i>
2.5	STM32 Bluepill	37
3	TRABALHOS RELACIONADOS	38
3.1	<i>Drone Swarms in Fire Suppression Activities: A Conceptual Framework</i> (Ausonio et al., 2021)	38
3.2	<i>Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting</i> (Aydin et al., 2019)	39

3.3	<i>Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems (Innocente; Grasso, 2019)</i>	40
3.4	Análise Comparativa	41
4	PROPOSTA DO SISTEMA	42
4.1	Detecção	43
4.2	Outras considerações	43
5	PROCEDIMENTOS METODOLÓGICOS	44
5.1	Pesquisa com Brigadistas	44
5.2	Análise das Respostas Obtidas	45
5.3	Simulação	45
5.3.1	<i>Análise utilizando FC</i>	46
5.3.2	<i>Desenvolvimento do A-star</i>	46
5.3.3	<i>Modelagem do Incêndio</i>	46
5.3.4	<i>Tarefa de Supressão</i>	47
5.4	Desenvolver Rede <i>LoRaWAN</i>	47
6	MÉTODOS E MATERIAIS	48
6.1	Plotagem de FC	48
6.2	Estimativa de metros lineares apagados	50
6.3	Desenvolvimento do <i>A-star</i>	51
6.4	Modelagem do incêndio	54
6.4.1	<i>Efeito do vento</i>	57
6.4.2	<i>Plotagem dos gráficos de evolução do incêndio</i>	57
6.5	Modelagem do incêndio com a presença dos drones	58
6.6	Rede <i>LoRaWAN</i>	59
6.6.1	<i>Arquitetura</i>	59
6.6.2	<i>Circuito</i>	60
6.7	<i>Sender e Receiver</i>	61
7	RESULTADOS	64
7.1	Pesquisa com Brigadistas	64
7.2	Simulação	66
7.2.1	<i>Análise utilizando FC</i>	66

7.2.2	<i>Modelagem do incêndio</i>	69
7.2.3	<i>Tarefa de supressão</i>	71
7.3	Desenvolver rede <i>LoRaWAN</i>	73
8	CONCLUSÃO E TRABALHOS FUTUROS	74
8.0.1	<i>Trabalhos futuros</i>	74
	REFERÊNCIAS	75
	ANEXOS	79
	ANEXO A – CÓDIGO COMPLETO DE SIMULAÇÃO DO INCÊNDIO	80

1 INTRODUÇÃO

As florestas desempenham papéis indispensáveis para o equilíbrio ecológico no planeta como manutenção de solos, controle de gases de efeito estufa e vetores de doenças, regulação do número de espécies, seja animal ou vegetal, entre outras funções (Fearnside, 2003). Cerca de 31% da superfície do planeta é coberta por área florestal, sendo um recurso indispensável para a sobrevivência de um quarto da população mundial, garantindo segurança alimentar e meios de subsistência. Também é um importante meio econômico, já que é a origem de fibras, de fármacos e de alimentos (Vargas; Zakrzewski, 2020). Uma das principais ameaças a esse importante recurso são os incêndios florestais.

Esses incêndios podem ser causados por ação humana, com a finalidade de usar o espaço para atividades de agropecuária ou plantio, ou fenômenos naturais, como raios e mudanças bruscas de temperatura. Os últimos três anos apresentaram um alto número de queimadas nas regiões da Amazônia e do Cerrado nos primeiros 8 meses do ano. Supõe-se que haverá um aumento do risco de incêndios de 30% até o final de 2050 e de 50% até o final do século (PNUMA, 2022). Esse tipo de desastre implica em uma considerável perda para o ecossistema local. Estima-se que mesmo após 15 anos da queima, algumas florestas não demonstram sinais de recuperação (Bufacchi *et al.*, 2016). Tudo isso resulta em perdas humanas, ambientais e econômicas.

Para vigilância de florestas e detecção de incêndios são empregados meios como satélites e sensores em terra. Contudo, estes perdem eficiência devido a variáveis climáticas e espaciais, no caso de imagens obtidas por satélites, ou sofrem com o alcance de detecção, um problema encontrado no emprego de recursos em terra (Aydin *et al.*, 2019). Já em missões de extinção de incêndios são usadas aeronaves tripuladas, com altos custos de operação e manutenção. Além de recursos humanos, colocando em risco a vida de brigadistas (Afghah *et al.*, 2019).

Em decorrência do aumento no número de incêndios, das limitações dos recursos utilizados atualmente e dos riscos em que os bombeiros ficam expostos, como chamas intensas além do normal, novos meios de combate estão surgindo para complementar os já existentes. Uma das propostas é a utilização de *Drone Swarm*, ou Enxame de drones, organização em rede formada por vários Veículos Aéreo Não Tripulado (VANTs) (ou drones). As ideias vão desde o uso “simples” em tarefas de prevenção, por exemplo, vigilância (Saffre *et al.*, 2022), até sugestões de empregá-los em missões de supressão em incêndios de pequeno e médio porte

(Ausonio *et al.*, 2021). Isso ocorre porque drones podem ser equipados com diversos sensores ou câmeras e possuem custos relativamente baixos, comparados à operação de aeronaves tripuladas.

Em um enxame de drones, cada VANT, ou nó, colabora com informações objetivando a realização de tarefas em ampla escala de cobertura. São empregados na tentativa de atenuar as deficiências no uso de apenas um VANT, por exemplo, com baixa cobertura territorial e ainda com limitação de autonomia (Asaamong *et al.*, 2021).

Este trabalho objetiva propor um modelo de enxame de VANTs para atuar como recurso complementar aos já existentes no combate a incêndios florestais. No modelo proposto, essas aeronaves atuam em áreas de difícil acesso e em estações mais secas. Usando visão computacional, os VANTs de vigilância detectam focos de incêndio. Após a detecção, o nó que fez o reconhecimento envia uma mensagem à base contendo a localização e uma estimativa do tamanho do foco. A base, então, envia uma mensagem de ativação, com as coordenadas recebidas anteriormente, para um número de VANTs de supressão adequado ao tamanho do incêndio. O grupo de aeronaves ativadas vai até o local e despeja água sobre as chamas.

No escopo deste trabalho, primeiro foi realizada uma pesquisa, com bombeiros militares especializados em combate a incêndios florestais, sobre os atuais métodos utilizados, as dificuldades enfrentadas na utilização dos mesmos e a utilidade de VANTs para o propósito abordado. O trabalho segue focado na tarefa de supressão do modelo, com a simulação de um enxame de VANTs atuando sobre incêndios, mostrando que a utilização desses equipamentos pode auxiliar outros métodos de combate já utilizados. Em seguida, o trabalho seguiu com um teste de uma possível forma de comunicação entre os nós do modelo, utilizando uma rede *Long Range Wide Area Network (LoRaWAN)* e *Single Board Computer (SBC)*, já visando baixo consumo energético e a possibilidade de adição de algoritmos de visão computacional, em trabalhos futuros, para detecção de incêndios.

1.1 Objetivos

1.1.1 Objetivo geral

Portanto, o objetivo geral deste trabalho é desenvolver um modelo de enxame de drones para suprir algumas necessidades das formas de combate a incêndios já utilizadas.

1.1.2 *Objetivos específicos*

- Simular, em ambiente virtual, a evolução e supressão de incêndios florestais.
- Estimar o impacto de VANTs na supressão de incêndios.
- Desenvolver uma rede *LoRaWAN*, que será usada como ponte de comunicação entre os nós do enxame.

1.2 Organização do Trabalho

Esse trabalho está organizado da seguinte forma:

- Capítulo 2: trata a fundamentação teórica, com os conceitos fundamentais para a compreensão da proposta descrita.
- Capítulo 3: aborda os trabalhos relacionados, comparando os aspectos comuns ou divergentes entre eles e o trabalho aqui proposto.
- Capítulo 4: descreve a organização do modelo da proposta.
- Capítulo 5: descreve a metodologia que deve ser abordada para o desenvolvimento deste trabalho.
- Capítulo 6: mostra os métodos e materiais utilizados para o desenvolvimento deste trabalho.
- Capítulo 7: apresenta os resultados obtidos.
- Capítulo 8: descreve a conclusão deste trabalho e ideias para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo fundamenta os principais conceitos relacionados ao trabalho, trazendo a colaboração de cada conceito para o trabalho proposto. Na Seção 2.1, é mostrada uma visão geral do que é um enxame de drones. Na Seção 2.2, são apresentados os conceitos utilizados na simulação. Na Seção 2.3, são explicados os principais conceitos envolvendo uma rede *LoRaWAN*. E nas Seções 2.4 e 2.5, são apresentadas as placas, SBCs e microcontrolador, usadas nesse trabalho.

2.1 Enxame de Drones

Enxames são formados por uma cadeia de agentes para desempenhar determinadas tarefas coletivas. Há diversos exemplos de enxames na natureza como, por exemplo, pássaros, abelhas e formigas, onde todos os indivíduos realizam interações em conjunto para um propósito em comum (McCune; Madey, 2013).

As limitações de VANTs, atuando individualmente, são de carga útil e tempo de voo limitados. Coordenar vários VANTs para realizar tarefas em um ambiente de enxame se torna atraente a partir do fato de diminuir as limitações de um único VANT, além de adicionar mais funcionalidades (Campion *et al.*, 2019).

2.2 Simulação

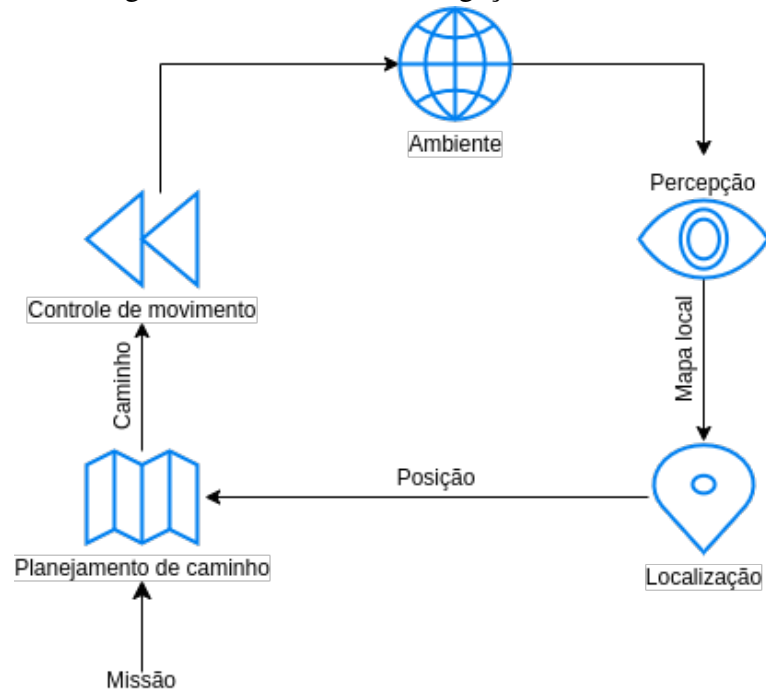
Nesta seção são apresentados os conceitos utilizados na simulação do enxame, por exemplo, o modelo de VANT usado como parâmetro na simulação, as técnicas para construção do mapa e algoritmo usado para o deslocamento até um ponto específico.

2.2.1 *Path Planning*

Na Figura 1, são mostrados os componentes presentes na navegação de um VANT. A partir da missão submetida, é feito o planejamento de caminho e o VANT ajusta seu movimento para realizar a trajetória que se deseja. Em seguida, o VANT extrai informações do ambiente e através dos sensores embarcados é determinada a localização do equipamento. Por fim, todo o ciclo é refeito até a tarefa estar concluída (Mac *et al.*, 2016).

O problema de *path planning*, ou planejamento de caminhos, é uma questão im-

Figura 1 – Estrutura de navegação de um robô



Fonte: Elaborado pelo próprio autor baseado em (Mac *et al.*, 2016)

portante na robótica móvel devido à necessidade da existência de algoritmos para haver uma conversão das especificações de alto nível, de tarefas desempenhadas por humanos, em descrições de baixo nível de como um VANT, ou robô, deve movimentar-se (Contreras-Cruz *et al.*, 2015). Um planejamento de caminho bem implementado pode economizar tempo, reduzir o desgaste e o investimento de capital no equipamento (Zhang *et al.*, 2018).

De modo geral, as abordagens de planejamento de caminhos podem ser divididas em planejamento de caminho global e planejamento de caminho local. O primeiro gera um caminho com base em um mapa conhecido ou em informações atuais e passadas do ambiente, obtidas na etapa de percepção. Esse método é ineficaz em ambientes desconhecidos ou dinâmicos. Como contraponto, algoritmos de planejamento de caminho local não precisam de informações do ambiente e traçam um caminho sobre um fragmento do caminho global, com a desvantagem de ser ineficiente quando o destino é muito distante (Mac *et al.*, 2016).

2.2.2 A-star

O algoritmo de *A-star* é uma heurística usada para pesquisa em grafo, classificado como uma abordagem global para o problema de *path planning*. Foi proposto por Hart *et al.* (1968), como uma possível solução para o problema de pesquisa em grafo, garantindo que irá

examinar o menor número de nós para se chegar em uma solução ótima.

Para tornar o algoritmo eficiente, em termos de avaliar os nós que devem ser observados, uma função de avaliação $f(n)$ foi proposta por Hart *et al.* (1968), como visto na Equação 2.1. Essa função pode ser calculada para qualquer nó, onde o nó com o menor valor de f é o nó a ser seguido. O termo $g(n)$ representa o custo de ir do nó inicial (chamado de s) até o nó intermediário n e $h(n)$ é o custo do caminho ótimo de n até o nó de destino (Karur *et al.*, 2021). Esse custo, é tratado como o menor caminho a ser percorrido do nó de início até o nó de destino.

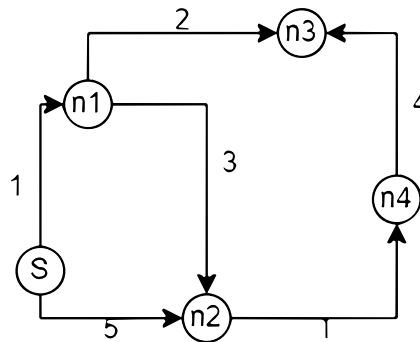
$$f(n) = g(n) + h(n) \quad (2.1)$$

O algoritmo segue os seguintes passos:

- 1 - Marca o nó inicial $s = n$, pertencente ao conjunto de nós S , como aberto e calcula $f(s)$;
- 2 - Calcula $f(n)$ para os nós vizinhos de n , pertencentes ao subconjunto S'_n , que estão abertos (subconjunto S''_n), e seleciona aquele com menor valor.
- 3 - Se n for o nó de destino, é marcado como fechado, subconjunto S'''_n , e o algoritmo termina.
- 4 - O nó atual é marcado como fechado e os nós vizinhos, que ainda não foram marcados como fechados, são marcados como abertos e colocados em S''_n . Se existir algum nó, marcado como fechado, onde $f(n)$ é menor do que valor calculado quando houve a marcação, esse nó é marcado como aberto. O algoritmo volta para o passo 2.

Na Figura 2, é exemplificado o funcionamento do *A-star*, considerando $n3$ como nó de destino. Primeiro, o nó s é marcado como aberto e $f(s)$ é calculado para o mesmo, cujo resultado será 3, já que s é o nó inicial e o caminho ótimo dele para o destino tem custo 3. Os nós vizinhos de s , $n1$ e $n2$, são marcados como abertos, s é marcado como fechado e $f(n)$ é calculado para cada um dos nós vizinhos. Para $n1$ é obtido 3 e para $n2$ 10. Então, $n1$ é escolhido e $n2$ é marcado como fechado. É verificado se existe algum vizinho de $n1$, marcado como fechado, com valor de $f(n)$ menor que o anterior, nesse caso, $n2$ é remarcado com aberto, pois o seu valor de $f(n)$ passa a ser 9. O nó $n3$ é marcado como aberto, já que é vizinho de $n1$, e seu valor de $f(n)$ é calculado como 3, por fim, $n1$ e $n2$ são marcados como fechados. O nó $n3$ é escolhido e o algoritmo termina, por ser o nó de destino.

Figura 2 – Grafo para exemplificar o A-star



Fonte: Elaborado pelo próprio autor inspirado em (Hart *et al.*, 1968)

O algoritmo *A-star* é inviável em ambientes dinâmicos, por avaliar o caminho mais curto com base em pares de nós conhecidos. Contudo, é computacionalmente eficiente em ambientes estáticos, já que não precisa avaliar todos os nós para alcançar o destino (Karur *et al.*, 2021). Considerando que o ambiente usado nas simulações é estático, essa é a abordagem de *path planning* utilizada neste trabalho, onde cada pixel é considerado como um nó.

2.2.3 Modelo de Autômato Celular

Para mostrar a eficácia da proposta do trabalho, é preciso modelar um incêndio e a evolução do mesmo. Será utilizado o modelo de Autômato Celular (AC) apresentado em Alexandridis *et al.* (2008) e adaptado por Ausonio *et al.* (2021), por ser mais simples em termos de implementação caso comparado a outros modelos, como o modelo de *Rothermel*, proposto em Rothermel (1972), onde são considerados diversos fatores físicos e químicos.

Um AC consiste em uma matriz, geralmente infinita em extensão, com uma variável discreta em cada célula, esta de tamanho atômico. O estado de um AC é especificado pelos valores das variáveis em cada célula. O AC evolui em passos discretos de tempo, com o valor da variável em uma célula sendo afetado pelos valores das variáveis nas células vizinhas. As células vizinhas são imediatamente adjacentes, diagonalmente, horizontalmente e verticalmente, a uma célula. As variáveis em cada célula são atualizadas simultaneamente, com base nos valores das variáveis nas células vizinhas no passo de tempo anterior e de acordo com um conjunto definido de regras (Wolfram, 1983).

O modelo de AC determina 4 estados:

- Estado 1: A célula não contém combustível florestal. Assume-se que as células neste estado não podem ser queimadas.
- Estado 2: Contém combustível ainda não queimado.

- Estado 3: Contém combustível que está queimando.
- Estado 4: O combustível foi totalmente queimado.

A cada passo de tempo t as células, representadas por (i, j) , podem mudar de estado com a aplicação das seguintes regras:

- Regra 1: Se $(i, j, t) = 1$, então $(i, j, t + 1) = 1$. Isso implica que a célula permanece sem mudança, pois não possui combustível.
- Regra 2: Se $(i, j, t) = 3$, então $(i, j, t + 1) = 4$. Se uma célula estiver queimando, ela será completamente queimada no próximo instante t .
- Regra 3: Se $(i, j, t) = 4$, então $(i, j, t + 1) = 4$. Se uma célula já foi completamente queimada, ela permanece no mesmo estado.
- Regra 4: Se $(i, j, t) = 3$, então $(i \pm 1, j \pm 1, t + 1) = 3$. Se uma célula está queimando, existe uma probabilidade p_{queima} de as chamas passarem para as células vizinhas.

A probabilidade p_{queima} é definida como

$$p_{queima} = p_h(1 + p_{veg})(1 + p_{den})p_v p_d. \quad (2.2)$$

O fator p_h representa a chance de uma célula adjacente a uma célula em chamas começar a queimar, sem considerar vento e com terreno plano.

Os fatores p_{veg} e p_{den} representam o tipo de vegetação e da densidade, respectivamente. Cada uma possui uma matriz, onde cada célula possui um valor representando a probabilidade para a respectiva célula no AC.

Já p_v considera a velocidade e direção do vento. É dada por

$$p_v = e^{c_1 U} e^{U c_2 (\cos \theta - 1)}. \quad (2.3)$$

Sendo U a velocidade do vento, c_1 e c_2 constantes e θ a angulação entre a direção de propagação das chamas e a direção do vento.

O fator p_d considera a inclinação do terreno. É calculada como

$$p_d = e^{a \theta_d}. \quad (2.4)$$

Sendo a uma constante e θ_d é dado por

$$\theta_d = \tan^{-1}\left(\frac{E_1 - E_2}{d}\right), \quad (2.5)$$

E_1 e E_2 são as alturas de inclinação de duas células vizinhas e d pode ser igual a l (comprimento da célula) ou a $\sqrt{2}l$, dependendo da posição das células vizinhas. Há casos onde p_{queima} passa de 1, isso é tratado em código considerando que nesses casos o valor é considerado 1.

2.2.4 Método de Hansen

A abordagem de *Hansen* (Hansen, 2012), determina a quantidade de água necessária para combater um incêndio. O método foi baseado na teoria do *Fire Point*, que descreve as condições necessárias para ignição e extinção de uma chama (Beyler, 1992), onde o balanço de energia na superfície do combustível, quando a água é aplicada, é

$$(f \times \Delta H_c - L_v) \times \dot{m}'' + \dot{q}''_E - \dot{q}''_P - \dot{q}''_{agua} = 0. \quad (2.6)$$

Onde:

- f é a fração de liberação de calor transferida de volta para a superfície do combustível por convecção e radiação;
- ΔH_c é o calor efetivo de combustão, ou seja, quantidade de energia liberada durante a combustão, dada em kJ/kg (quilojoule por quilograma);
- L_v é o calor de gaseificação do combustível, dado em kJ/kg ;
- \dot{m}'' é a taxa de queima de massa por unidade de área de combustível, dada em kg/m^2s (quilograma por metro quadrado por segundo);
- \dot{q}''_E é o fluxo de calor externo, taxa de calor transferido de uma fonte externa, dada em kW/m^2 (quilowatts por metro quadrado);
- \dot{q}''_P é a taxa da perda de calor da superfície, dada em kW/m^2 ;
- \dot{q}''_{agua} é a taxa de perda de calor por vaporização da água, dada em kW/m^2 .

Usando o número B de *Spalding*, uma medida adimensional que relaciona as taxas de transferência de calor por convecção e radiação com as características da chama (Spalding, 1953), a taxa de queima de massa no ponto crítico, ponto de extinção da chama, pode ser dada como

$$\dot{m}''_{cr} = \frac{h}{c_p} \times \ln(1 + B) = \frac{h}{c_p} \times \ln\left(1 + \frac{Y_{O_2} \times \Delta H_{R,O_2}}{\phi \times \Delta H_c}\right). \quad (2.7)$$

Sendo:

- h o coeficiente de transferência de calor por convecção, dado por $W/m^2/K$ (Watts por metro quadrado Kelvin);
- Y_{O_2} é a fração de massa de oxigênio presente;
- $\Delta H_{R,O_2}$ é o calor de combustão por unidade de massa de oxigênio consumido, dado em kJ/kg ;
- ϕ é a fração de perda de calor por convecção da chama necessária para extinguir a chama;
- c_p é o calor específico do ar em pressão constante, dado em kJ/kgK (quilojoule por quilograma por kelvin).

A perda de calor devido à vaporização da água é dada por

$$\dot{q}''_{agua} = \eta_{agua} \times \dot{m}''_{agua,cr} \times L_{v,agua}, \quad (2.8)$$

onde η_{agua} é a eficiência da aplicação da água, $L_{v,agua}$ é a variação de entalpia da água em $283K$ e do vapor de água em $373K$, dada em kJ/kg , e $\dot{m}''_{agua,cr}$ é a taxa crítica de aplicação da água, dada em kg/m^2s .

Substituindo a Equação 2.8 na Equação 2.6 e f por ϕ , a taxa crítica de fluxo de água FC, taxa necessária para extinguir um incêndio em L/m^2s (litros por metro quadrado por segundo), é

$$FC = \dot{m}''_{agua,cr} = \frac{(\phi \Delta H_c - L_v) \dot{m}''_{cr} + \dot{q}''_E - \dot{q}''_P}{\eta_{agua} L_{v,agua}}, \quad (2.9)$$

segundo Hansen (2012), assumindo não haver influência de vento e inclinação no terreno, o componente \dot{q}''_E pode ser assumido como

$$\dot{q}''_E = \left(\frac{r_c I}{2 \times C + P} \times \phi \times \tau \right) + (h \times (T_g - T_c)). \quad (2.10)$$

Onde:

- r_c é o componente de radiação por unidade de comprimento da frente de fogo, dado em porcentagem. Porcentagem de calor liberado por meio de radiação;
- I é a intensidade da linha de fogo em kW/m (quilowatt por metro);
- C é o comprimento da chama, em metros;
- P é a profundidade da zona de combustão ativa, em metros;

- τ é a transmissividade atmosférica;
- T_g é a temperatura do gás, dada em K ;
- T_c é a temperatura do combustível, dada em K ;

O componente \dot{q}_p'' pode ser dado como

$$\dot{q}_p'' = \dot{q}_{p,rad}'' + \dot{q}_{p,conv}'' \quad (2.11)$$

$$\dot{q}_{p,rad}'' = \varepsilon \sigma (T_c^4 - T_a^4), \quad (2.12)$$

$$\dot{q}_{p,conv}'' = h(T_c - T_a), \quad (2.13)$$

onde ε mede a capacidade do combustível de emitir radiação térmica, σ é a constante de *Stefan-Boltzmann*, dada em $W/m^2 K^4$ (watts por metro quadrado por kelvin a quarta) e T_a é a temperatura ambiente, dada em K .

A intensidade da linha de fogo, em função da velocidade do vento e do teor de umidade do combustível, pode ser expressa como

$$I = \frac{\Delta H_c \times W \times RoS}{36}, \quad (2.14)$$

onde W é a carga total do combustível, dado em t/ha (tonelada por hectare), e RoS é a taxa de propagação da frente de fogo, dada por

$$RoS = 0,06 \times a \times U^b \times e^{-c \times M_d}, \quad (2.15)$$

sendo U a velocidade do vento, em km/h , a 2 metros acima do solo, a , b e c são parâmetros obtidos experimentalmente por análise de regressão não linear em Martins Fernandes (2001) e M_d é porcentagem de teor de umidade do combustível.

Sabendo como chegar no valor de RoS , é possível encontrar o valor de C , segundo (Penney *et al.*, 2019), com

$$C = \frac{13RoS + 0.24W}{2}. \quad (2.16)$$

Na equação original de Martins Fernandes (2001), a mesma é dada em m/min (metros por minuto). Por esse motivo, é feito um produto por 0,06 em RoS , para converter em km/h . Além disso, a Equação 2.9 é multiplicada por $60D$, para que o valor de retorno esteja em $L/mmin$ (litros por metro por minuto) e seja referente a metros lineares, D sendo o comprimento da chama em metros.

$$FC = 60D \frac{(\phi \Delta H_c - L_v) \dot{m}_{cr}'' + \dot{q}_E'' - \dot{q}_P''}{\eta_{agua} L_{v,agua}} \quad (2.17)$$

2.2.5 Estimativa do número de drones

Com o número aproximado do fluxo de água necessário para suprimir um incêndio, é possível chegar a uma aproximação da quantidade de drones necessários para essa tarefa, sabendo sua velocidade, carga máxima e vazão de água de cada equipamento. O número de drones necessários para suprimir um número específico de metros lineares de frente de fogo ativa é dado por

$$n_n = \frac{FC \times \Delta t \times m_n}{L_d}, \quad (2.18)$$

com Δt representando o tempo de um drone ir da base até o incêndio, despejar todo o líquido da carga e voltar para base. A variável m_n é o número, em metros lineares, de frente de fogo ativa e L_d é carga total, em litros, carregada por um drone.

2.2.6 OpenCV

Open Source Computer Vision Library (OpenCV) é uma biblioteca de código aberto, desenvolvida pela Intel, que disponibiliza centenas de algoritmos de visão computacional. Embora sua primeira versão tenha sido baseada em *C*, hoje está disponível para diversas linguagens, incluindo *Python*, *C++* e *Ruby*. Possui uma estrutura completamente modular, ou seja, inclui várias bibliotecas compartilhadas (OpenCV, 2023).

O OpenCV visa fornecer uma infraestrutura de visão computacional fácil de usar que auxilie desenvolvedores a criarem, com facilidade, ferramentas de visão sofisticadas. A biblioteca possui mais de 500 funções que podem ser usadas em áreas como inspeção de fábrica de produtos, imagens médicas, segurança, interface do usuário, calibração de câmera, visão estéreo e robótica (Bradski, 2008).

2.2.7 DJI AGRAS T40

Para se ter uma estimativa de quantos litros de água cada VANT de supressão simulado carregará e sua duração de bateria, será considerado o modelo *DJI AGRAS T40* (DJI, 2022), uma aeronave, desenvolvida pela *DJI*, utilizada para pulverização e mapeamento de grandes plantações.

Esse VANT possui um tanque para carga com volume de 40 litros ou 40 quilos. Essa carga pode ser liberada em uma taxa de fluxo de até $12L/min$ (12 litros por minuto), em uma velocidade de voo de $7m/s$ (7 metros por segundo). Com relação à bateria, tem uma capacidade de até $30000mAh$ (30000 mili ampéres hora), isso o possibilita a um voo estacionário, com carga máxima, de 7 minutos (DJI, 2022).

2.3 Rede LoRaWAN

Nesta seção são apresentados os conceitos que serão usados no desenvolvimento da rede *LoRaWAN*, como *LPWAN*, *LoRa* e as placas de desenvolvimento que serão utilizadas.

2.3.1 LPWAN

Com o crescimento da *Internet of Things* (IoT), ou Internet das Coisas (Al-Sarawi *et al.*, 2017), e da comunicação entre dispositivos, a curto prazo, espera-se um crescimento massivo na utilização de sensores. Atualmente, grande parte desses dispositivos efetuam a comunicação entre si utilizando tecnologias de rede sem fio voltadas para aplicações de grande cobertura, como *Bluetooth* (Bisdikian, 2001) e *Wireless Local Area Network (WLANs)* (Khalajmehrabadi *et al.*, 2017), ou rede local sem fio. Entretanto, essas tecnologias não cumprem os requisitos de longo alcance e de baixo consumo energético das redes de sensores (Chaudhari *et al.*, 2020).

Esses requisitos proporcionaram o surgimento de uma nova tecnologia de comunicação sem fio, *Low Power Wide Area Network (LPWAN)*, em português Rede de Área Ampla de Baixo Consumo de Energia (Mekki *et al.*, 2019). As características dessa tecnologia são:

- Longo alcance;
- Baixo consumo de energia;
- Baixo custo monetário.

Na Tabela 1 é apresentada uma comparação entre *Zigbee* (Ramya *et al.*, 2011), uma tecnologia utilizada em pequenos projetos com baixo consumo energético, *Wi-Fi* (Deng *et al.*,

2020), que possui uma alta taxa de envios de dados, mas é energeticamente ineficiente, *Bluetooth*, apresentando uma boa taxa de envio, contudo, tem um raio de comunicação ainda muito limitado, e LPWAN, possuindo uma taxa de envio intermediária e um alto alcance de comunicação mesmo em ambiente urbano.

Tabela 1 – Comparação entre LPWAN e outra tecnologias

Tecnologia	Distância de comunicação	Transmissão de dados
<i>ZigBee</i>	10-75m	171 kbps
<i>Wi-Fi</i>	100m-	>10 Mbps
<i>Bluetooth</i>	40-400m	50 Mbps
<i>LPWAN</i>	3km em ambiente urbano	0,1-250 kbps

Fonte: Elaborado pelo próprio autor baseado em (Song *et al.*, 2017)

2.3.2 Long Range

Long Range (LoRa) é uma tecnologia de comunicação que segue os requisitos das redes LPWANs. Então, pode ser utilizada em aplicações de casas inteligentes, monitoramento de pacientes, industriais e comunicação em locais remotos (Gkotsiopoulos *et al.*, 2021). *LoRa* é a camada física do modelo *Long Range (LoRa)*, foi patenteada como um esquema de modulação pela *Semtech Corporation* (SELLER, 2016).

LoRa faz uso da técnica de modulação de espectro espalhado por *chirp*. Essa técnica já é utilizada há algum tempo em comunicações militares e espaciais devido à sua natureza robusta e capacidade de longo alcance. O espectro espalhado por *chirp* tem baixa exigência de potência de transmissão. *Chirp* é um sinal cuja frequência aumenta ou diminui ao longo do tempo, ou seja, pode ser de subida ou descida. Nesse modelo de modulação, o sinal de dados desejado é multiplicado pelo sinal de *chirp*. Isso expande a largura de banda além da original do sinal de dados. No receptor, o sinal recebido é multiplicado novamente com a cópia, localmente gerada, do sinal de *chirp*, comprimindo o sinal modulado de volta à largura de banda original, reduzindo ruído e interferência. (Devalal; Karthikeyan, 2018).

A taxa de bit da modulação é calculada como (Corp, 2015a)

$$T_b = FE * \frac{4}{\frac{4+TC}{2^{FE}} \frac{LB}{}} \text{ bit/seg.} \quad (2.19)$$

Onde:

- T_b é a taxa de bit;

- TC é a taxa de codificação, varia entre 1 e 4;
- FE é o fator de espalhamento, pode variar de 7 a 12;
- LB é a largura de banda, pode variar nos valores de $125kHz$ (quilohertz), $250kHz$ e $500kHz$.

A taxa de codificação é uma técnica de correção de erro, que melhora a robustez do sinal transmitido em detrimento da taxa efetiva de bits. O fator de espalhamento representa o número bits codificados em um símbolo. Um símbolo é a menor unidade de dados em um sinal digital, nesse caso, é representado pelo sinal de *chirp*.

O tempo de transmissão de um pacote é dado por (Corp, 2015b)

$$T_p = T_{preambulo} + 8 + \max\left(\left\lceil \frac{(8TP - 4FE + 28 + 16VRC - 20H)}{4(FE - 2DE)} \right\rceil (TC + 4), 0\right) \left(\frac{2^{FE}}{LB}\right), \quad (2.20)$$

$$T_{preambulo} = (n_{preambulo} + 4.25) \left(\frac{2^{FE}}{LB}\right). \quad (2.21)$$

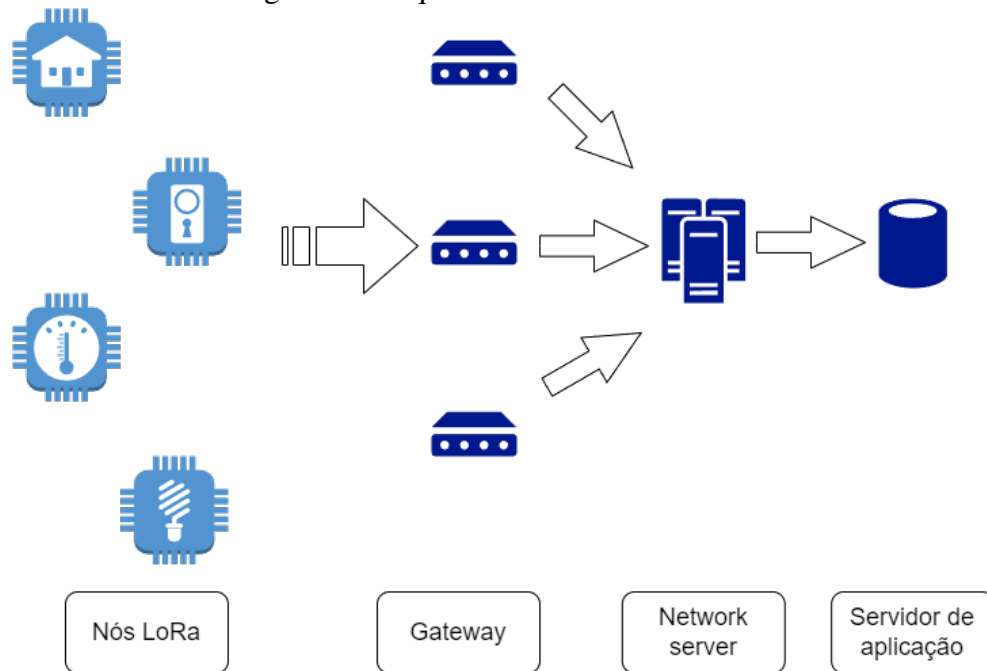
Sendo:

- TP é o tamanho do pacote a ser enviado;
- $T_{preambulo}$ é o comprimento do preâmbulo, sinal de sincronização entre emissor e receptor;
- $n_{preambulo}$ é o comprimento do preâmbulo programado, configurado antes da transmissão;
- H indica se há cabeçalho no pacote, 1 para existência e 0 se não houver;
- DE indica se a otimização de baixa taxa de dados está habilitada, 1 se estiver e 0 se não estiver;
- VRC variável utilizada para verificação de erros na mensagem.

2.3.3 LoRaWAN

LoRaWAN (*Long Range Wide Area Network*) é a camada *Media Access Control* (MAC) do modelo *LoRa*, que define a arquitetura da rede e o protocolo de comunicação (Devalal; Karthikeyan, 2018). A especificação *LoRaWAN* é padronizada e disponibilizada pela LoRa Alliance (Sornin; Yegin, 2017).

Na Figura 3, é ilustrada a arquitetura de rede presente em uma rede *LoRaWAN*. Os nós *LoRa* são conectados, via camada física, a um ou mais *gateways*. O *gateway* é então conectado a um servidor de rede. O mesmo é conectado a um servidor de aplicativos via *Internet Protocol* (IP) (Mahmood *et al.*, 2019).

Figura 3 – Arquitetura de rede *LoRaWAN*

Fonte: Elaborado pelo próprio autor inspirado em (Devalal; Karthikeyan, 2018)

- Os nós *LoRa* representam sensores ou aplicativos onde ocorrem o monitoramento e controle.
- O *gateway* é o principal elemento de uma rede *LoRa*. Os dados transmitidos pelo nó são enviados a algum *gateway*, esse transmite a mensagem para um servidor de rede.
- O servidor de rede identifica se os dados enviados pelos *gateways* são destinados a algum servidor de aplicativos, para enviar os dados ao servidor adequado.
- Destino dos dados enviados dos nós *LoRa*.

2.3.4 Módulo *sx1278*

Além das SBCs, para montar a rede *LoRaWAN*, será necessário um módulo *sx1278*, que servirá tanto para transmitir como para receber dados. É fabricado e comercializado pela *Semtech Corporation*, mesma detentora da patente da modulação *LoRa*, esse módulo trabalha em uma banda de frequência de 433Mhz (Corp, 2015b).

2.4 *Single Board Computer*

Um *Single Board Computer* (SBC) é um computador completo que possui memória, portas de entrada e saída, microprocessador e outras funcionalidades em um único dispositivo. (Matthews *et al.*, 2016). Dentre eles, destacam-se, tamanho compacto, baixo custo e

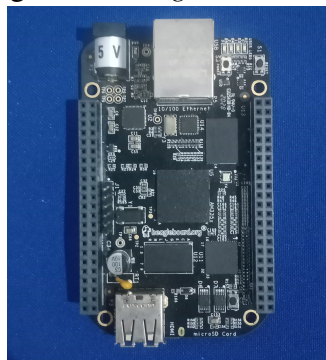
complexidade.

Alguns exemplos de SBCs comumente utilizados são *BeagleBone Black* (COLEY, 2014) e *Raspberry Pi* (MCMANUS, 2021). Todas possuem arquitetura *Advanced RISC Machine* (ARM), têm conectividade *Ethernet* e *Universal Serial Bus* (USB), além de possuírem suporte a comunicação de baixo nível como *Serial Peripheral Interface* (SPI), *Inter-Integrated Circuit* (I2C) e *Universal Asynchronous Receiver/Transmitter* (UART). De forma geral, esses computadores são adequados para projetos com requisitos que não podem ser cumpridos por microcontroladores.

2.4.1 *BeagleBone Black*

A placa *BeagleBone Black* (BBB), mostrada na Figura 4, é uma SBC lançada pela *Texas Instruments* (TI) em 2013 e revisada pela última vez em 2014, recebendo algumas melhorias (COLEY, 2014). Possui um processador TI Sitara™ ARM Cortex™-A8, que opera a 1 GHz (gigahertz), 512 MB (megabytes) de *Double Data Rate Synchronous Dynamic Random Access Memory* (DDR3 SDRAM), memória *flash Embedded Multimedia Card* (eMMC) *onboard* de 4GB, segundo a última revisão. Com relação à conectividade, inclui 46 pinos, portas USB, um conector micro *High-Definition Multimedia Interface* (HDMI), uma porta *Ethernet*, além de um conector para *Secure Digital Card* (SD Card), caso o usuário queira executar um Sistema Operacional (SO) diferente do que está embarcado na BBB (He *et al.*, 2016).

Figura 4 – *BeagleBone Black*

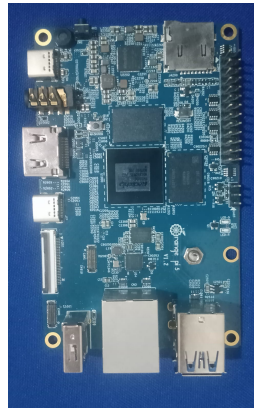


Fonte: Elaborado pelo próprio autor

2.4.2 Orange Pi 5

A *Orange Pi 5*, presente na Figura 5, foi lançada comercialmente em 2022 pela *Orange Pi* como uma concorrente a *Raspberry Pi 4 model B*, contudo, possui vantagens em questão de *hardware*. Tem um processador de arquitetura ARM 64-bit *octa-core*, com 4 núcleos *ARM Cortex-A76* e 4 *ARM Cortex-A55*, de 2.4 GHz. Outras melhorias, em relação a SBC concorrente, são o limite de até 32GB de SDRAM e suporte a armazenamento *Solid State Drive* (SSD) M.2. Por fim, suporta, nativamente, as mesmas tecnologias de rede sem fio e portas de conexão para periféricos que a *Raspberry* (Pi, 2022).

Figura 5 – *Orange Pi 5*



Fonte: Elaborado pelo próprio autor

2.5 STM32 Bluepill

A STM32F103C8T6 (*Bluepill*), é uma plataforma microcontroladora, de médio desempenho, fabricado pela *STMicroelectronics*. É amplamente utilizado para aplicações IOT, por possuir um microcontrolador ARM Cortex-M3 e vários pinos para periféricos. Também possui a vantagem de ser compatível com plataformas disponibilizadas pela *STMicroelectronics* e a *framework Arduino* (STMicroelectronics, 2023).

3 TRABALHOS RELACIONADOS

Neste capítulo, serão retratados e comparados alguns trabalhos da literatura relacionados ao uso de drones no contexto de incêndios florestais com o trabalho proposto.

3.1 *Drone Swarms in Fire Suppression Activities: A Conceptual Framework (Ausonio et al., 2021)*

Ausonio *et al.* (2021) propõem um sistema de enxame de VANTs para o combate de incêndios em áreas limitadas e apoio a outras técnicas de combate. Nesse sistema, centenas de drones expõem água para simular o efeito de chuva. Ainda é proposto um reabastecimento automático, que representa um diferencial significativo em relação a outros trabalhos nessa área, já que pode diminuir os custos de operação.

Os autores estimaram o impacto do uso de uma plataforma que gerencia um número variável de drones capazes de espalhar água ou outro líquido extintor em um incêndio florestal, estimando também seus efeitos na propagação de um incêndio florestal considerando vários fatores, como intensidade do fogo, comprimento da chama, vegetação, umidade, velocidade e direção do vento.

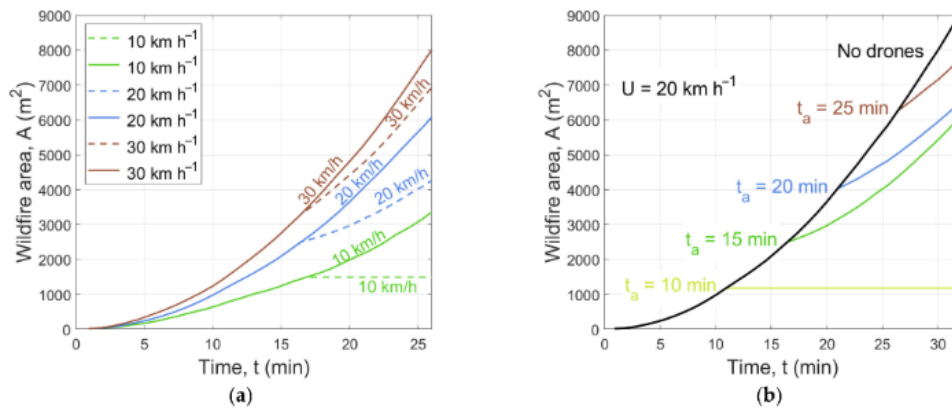
Para validar a proposta, os autores calculam a taxa de fluxo de água, tanto em função do comprimento do incêndio como em função da velocidade do vento e da umidade. Ainda é estimado o número de VANTs necessários para mitigar, ou extinguir, um número específico de metros lineares de frente de fogo ativa e como esses drones impactam a evolução do incêndio.

Os experimentos mostram que a taxa crítica de vazão é diretamente proporcional ao tamanho do incêndio e à velocidade do vento, enquanto é inversamente proporcional à umidade do ambiente. Já a frente de fogo combatida, medida em metros, é proporcional ao número de drones usados e à umidade do ar, porém, decai com os efeitos da velocidade do vento e o aumento da taxa de espalhamento das chamas.

Concluindo os resultados obtidos pelos autores, é exibido na Figura 6 o efeito que o método de extinção causa na variação da área do incêndio, em diferentes velocidades de vento, efetuando uma comparação entre cenários com (linhas pontilhadas) e sem método de supressão (linhas contínuas), no primeiro gráfico, e demonstrando o comportamento da variação da área, adicionando VANTs em momentos específicos no segundo gráfico. Em ambas as figuras, é possível perceber que o crescimento da área queimada diminui com a adição de métodos de

supressão.

Figura 6 – Comparação do crescimento da área do incêndio com e sem os VANTs



Fonte: (Ausonio *et al.*, 2021)

Os autores esclarecem que os efeitos do vento e da temperatura das chamas nos VANTs serão alvos de trabalhos futuros. Também será estudada a possibilidade de utilização de VANTs mais pesados, com carga maior.

3.2 Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting (Aydin *et al.*, 2019)

Aydin *et al.* (2019) propõem e fazem um estudo do uso de drones com bolas extintoras para o combate de incêndios. Além disso, neste trabalho, também é proposto o uso de VANTs para monitoramento de riscos de incêndios próximo a áreas habitadas.

Os autores demonstram os resultados de experimentos controlados para testar a eficiência de bolas extintoras para combate a incêndios. Esses experimentos foram realizados como parte de um projeto de pesquisa em andamento cujo objetivo é projetar um sistema de VANTs incorporando sensoriamento remoto e bolas extintoras para controlar incêndios florestais. Os resultados mostram que as bolas de extinção de incêndio podem ser uma ferramenta eficaz e eficiente, considerando que as usadas foram as menores disponíveis no mercado, para auxiliar no combate a incêndios florestais se um enxame de drones puder jogá-las em pontos ideais, em números ideais, a tempo. A tecnologia de sensoriamento remoto será necessária para determinar os pontos de ataque.

O trabalho futuro dos autores envolve o desenvolvimento de plataformas de enxame para os componentes de sensoriamento remoto e combate a incêndios, atualização do mecanismo de queda e desenvolvimento de modelos de trajetória do vento para experimentar métodos ideais

de queda sobre vegetação. Existem algumas limitações neste estudo, principalmente devido à disponibilidade de recursos no momento dos experimentos. A equipe de pesquisa usou o modelo de bola de extinção de incêndio mais barato nos experimentos.

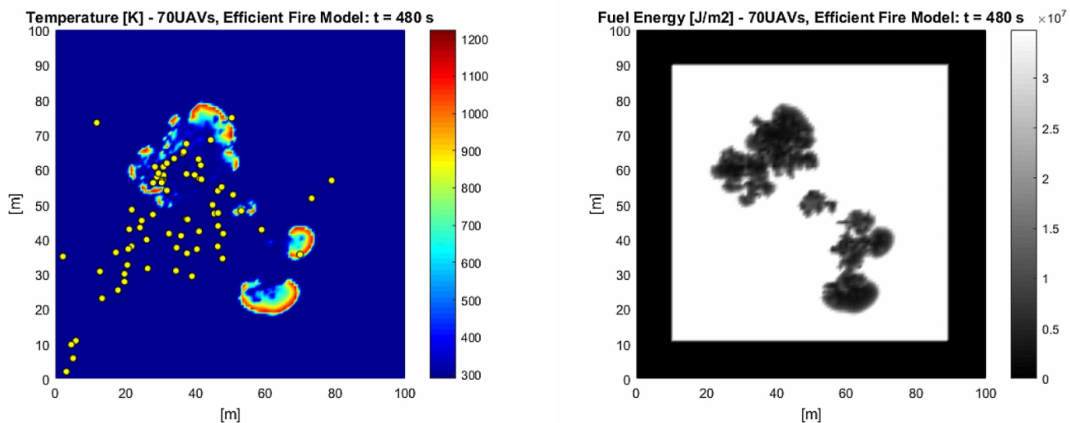
3.3 *Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems (Innocente; Grasso, 2019)*

Já Innocente e Grasso (2019) desenvolvem um modelo contra a propagação de incêndio baseado em um algoritmo de enxame de partículas adaptado para indivíduos que operam em ambientes com alta dinâmica e em frequente mudança. Ainda são feitos testes simulados e análises para validar a eficácia do modelo.

Este trabalho investigou a viabilidade e o potencial do uso de sistemas auto-organizados em enxames de drones para combater a propagação de incêndios de forma autônoma e colaborativa. O foco do trabalho não estava no design dos robôs físicos, mas em seus mecanismos de autocoordenação para que surgisse o comportamento desejado de combate a incêndios. Assim, um modelo de propagação do fogo e um algoritmo de auto-organização para enxames de drones de combate a incêndios foram desenvolvidos e acoplados.

Na Figura 7, é mostrado um dos resultados obtidos durante os experimentos dos autores. Cerca de 70 VANTs foram liberados para combater um incêndio de 4 fontes diferentes, mas, como é ilustrado no gráfico da direita (área sem combustível, ou área já queimada, no final da simulação), não foram bem sucedidos. Esse cenário é revertido com o aumento do enxame para 110 VANTs, obtendo 100% de sucesso.

Figura 7 – Exemplo de simulação de combate a incêndio florestal



Fonte: (Innocente; Grasso, 2019)

Os experimentos realizados pelos autores demonstraram que o modelo de propagação de fogo baseado em física poderia fazer previsões realistas e confirmou a escalabilidade esperada e a tolerância a falhas do enxame de VANTs como um sistema autônomo de combate a incêndios.

Em pesquisas futuras, os autores planejam utilizar outros materiais de supressão, como espuma química. Também serão estudados o uso de algoritmos de dinâmica de voo e de prevenção de colisão, por exemplo, campos potenciais.

3.4 Análise Comparativa

O foco de Ausonio *et al.* (2021) é demonstrar que um enxame de drones pode auxiliar outros métodos de supressão no combate a incêndios, então, não foi usado qualquer algoritmo de *path planning*, ou apresentada uma proposta de comunicação entre os agentes atuantes. Neste trabalho em comparação, foi utilizado o algoritmo de *A-star* para traçar o caminho percorrido pelos VANTs, além de propor a utilização de uma rede *LoRaWAN* para comunicação.

Em Aydin *et al.* (2019), embora proponham todos os principais pontos que serão abordados nesse trabalho, mostrou a eficácia de apenas um drone carregado com material de supressão. Foi utilizada espuma química como material, diferente deste trabalho, que simulará o efeito da água.

Já Innocente e Grasso (2019) desenvolveram o próprio algoritmo de modelo de incêndio que foi acoplado ao modelo do enxame. Resultados positivos foram obtidos com a utilização de vários VANTs. Neste trabalho foi utilizado um AC para modelar o incêndio, uma abordagem mais simples de implementar em comparação da utilizada no trabalho citado.

Na Tabela 2, é apresentado um resumo das características encontradas nos trabalhos relacionados descritos anteriormente, e a comparação com o projeto apresentado neste trabalho.

Tabela 2 – Tabela comparativa

Trabalho	Estimativa do número de drones	Tarefa de supressão	Enxame	Proposta de modelo de comunicação	<i>Path Planning</i>
Ausonio <i>et al.</i> (2021)	Sim	Sim	Sim	Não	Não
Aydin <i>et al.</i> (2019)	Não	Sim	Trabalho futuro	Trabalho futuro	Não
Innocente e Grasso (2019)	Não	Sim	Sim	Sim	Sim
Proposta	Sim	Sim	Sim	Sim	Sim

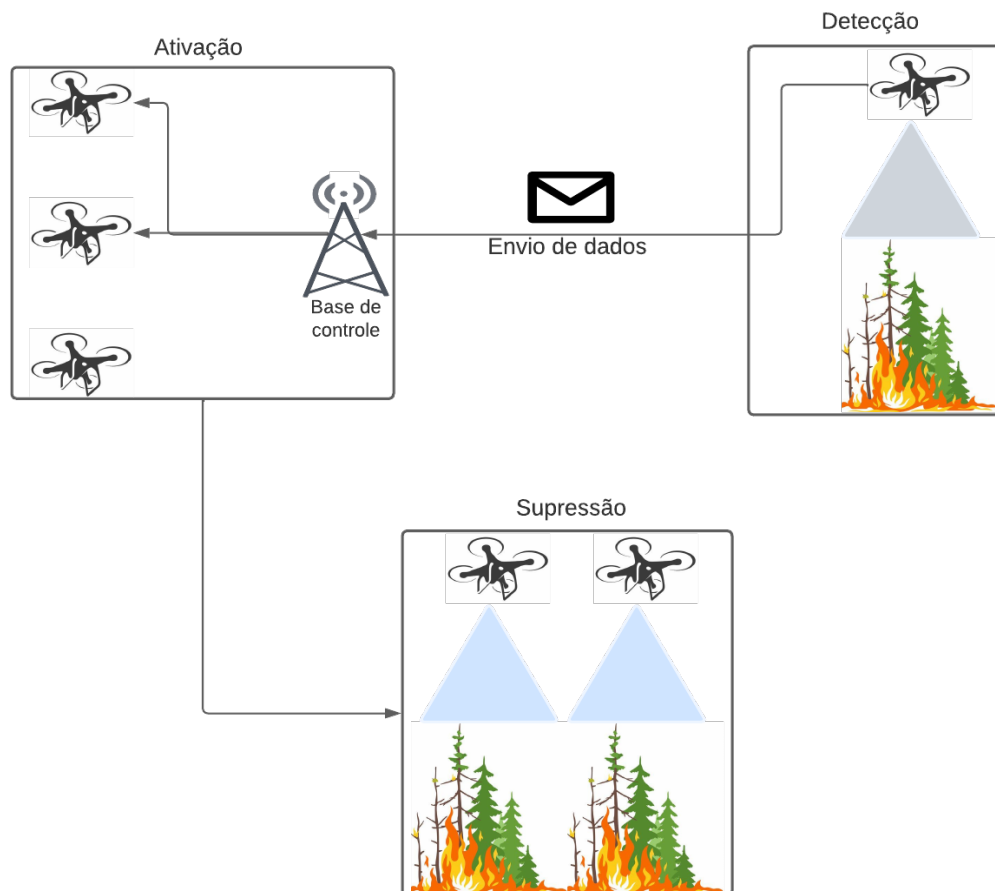
Fonte: Elaborado pelo próprio autor

4 PROPOSTA DO SISTEMA

Neste capítulo, é mostrada a arquitetura da proposta do sistema e como cada tarefa funciona.

Na Figura 8, está ilustrado o modelo geral da proposta. VANTs de vigilância irão realizar missões de detecção em certos intervalos de tempo. Caso ocorra a identificação de algum foco de incêndio, o drone de vigilância enviará um pacote de dados, por meio de uma rede *LoRaWAN*, contendo a localização e a proporção do incêndio, utilizada para calcular FC, para a base de controle. Então, a base mandará um sinal de ativação para um número de VANTs de supressão adequado para a proporção do incêndio. A localização, recebida anteriormente, será enviada junto ao sinal. Na fase de supressão, os VANTs serão carregados com uma carga de água, que será dispersada em um efeito de chuva sobre as chamas. O foco do presente trabalho será na fase de supressão e em testes na forma de comunicação.

Figura 8 – Modelo da organização da proposta



Fonte: Elaborado pelo próprio autor

4.1 Detecção

Como o trabalho atual não considera a fase de detecção da proposta, a tarefa fica aberta para a consideração de diferentes métodos, um deles sendo o usado em Aydin *et al.* (2019), um modelo de visão computacional. Considerando o uso de SBCs, a utilização de modelos de aprendizado de máquina mais complexos, como *Convolutional Neural Networks* (CNN), não é descartado (Kattenborn *et al.*, 2021). Essa tarefa é feita apenas por drones de detecção.

4.2 Outras considerações

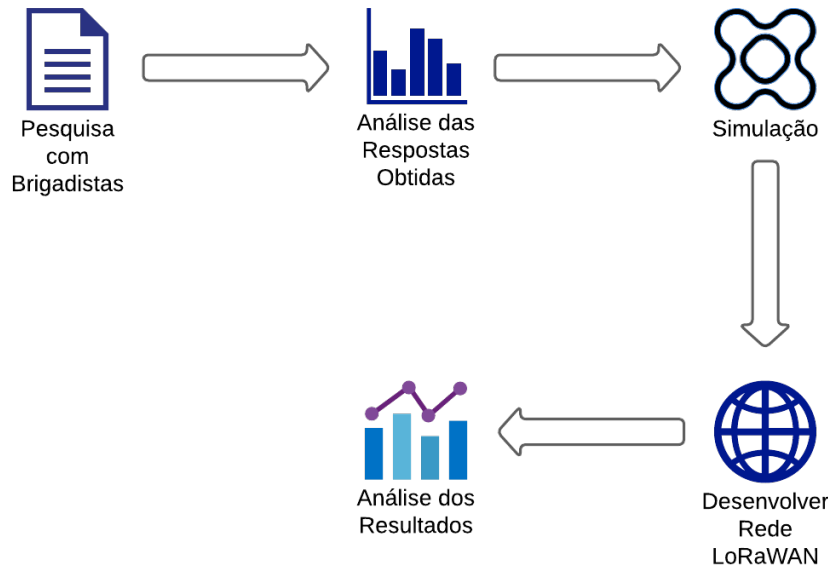
Em um cenário real, diversas variáveis do incêndio e do ambiente devem ser consideradas no voo dos drones. Porém, para simplificar as simulações, apenas a velocidade do vento, o tipo de vegetação e a sua umidade foram consideradas. Também é considerado apenas um foco de incêndio, ao contrário de um cenário real, onde o enxame de supressão precisaria ser dividido na proporção do tamanho dos focos.

Foi utilizado o *A-star* como algoritmo de *path planning*, por ser uma abordagem simples, sendo executado em um único computador. Porém, em um cenário real, cada drone executaria seu próprio algoritmo *A-star*, reduzindo custo de processamento. Contudo, algoritmos mais complexos podem ser utilizados em trabalhos futuros.

5 PROCEDIMENTOS METODOLÓGICOS

Para alcançar os objetivos propostos neste trabalho, faz-se necessário um conjunto de etapas a serem seguidas, conforme pode ser observado no fluxograma apresentado na Figura 9.

Figura 9 – Fluxograma dos procedimentos metodológicos



Fonte: Elaborado pelo próprio autor

5.1 Pesquisa com Brigadistas

Como primeira parte do projeto, foi realizada uma pesquisa com vários bombeiros militares de diversos estados, visando saber quais as formas de combate usadas atualmente em incêndios florestais, as principais dificuldades e a opinião desses profissionais sobre o uso de drones nesse cenário. As perguntas foram as seguintes:

- O quão útil você acha que seria o uso de drones, no contexto de combate a incêndios florestais, como vetor complementar aos métodos de combate atuais, principalmente a locais de difícil acesso? (Opções: escala de 1, inútil, até 5, útil)
- Quais dessas opções você acha que seria mais útil como a tarefa de drones no combate a incêndios florestais? (Opções: Nenhum; Combate direto (drones carregados com água para despejar no fogo); Combate indireto (Vigilância, transporte de carga))
- Que recursos são usados no seu batalhão para o combate a incêndios em locais de difícil

acesso? (Opções de múltipla escolha: Helicóptero ou aviões tanques; Imagens de satélite; Detectores de fumaça florestal; Robôs terrestres; Drones; Outros, os cite)

- Quais os maiores desafios no combate a incêndios florestais? (Opções de múltipla escolha: Acesso aos locais; Identificação dos focos de incêndio; Recursos apropriados; Manutenção do equipamento; Outros, os cite)

As respostas auxiliaram a se ter uma melhor compreensão de como a proposta desse trabalho pode ajudar no cenário de incêndios florestais.

5.2 Análise das Respostas Obtidas

As respostas obtidas foram lidas e analisadas por meio de gráficos, mostrando a distribuição das respostas das perguntas de múltipla escolha. As respostas complementares foram divididas em tópicos. Assim, obteve-se um maior conhecimento sobre a opinião dos profissionais da área sobre o uso de VANTs em incêndios florestais e quais as principais dificuldades encontradas nesse cenário.

5.3 Simulação

Para realizar as simulações do enxame, foi criado um ambiente virtual 2D como a representação de uma floresta, onde um foco de incêndio surge no centro do mapa. Para a construção de tal cenário, foi utilizado o OpenCV para gerar um ambiente 800x800 com proporção pixel para metro de 4x1, resultando em um tamanho de 40000 metros quadrados, a mesma dimensão usado por Ausonio *et al.* (2021). Considerando esses fatores, foi escolhida a linguagem de programação *Python* para montar as simulações, por sua simplicidade em codificação e grande gama de ferramentas.

Como o foco do trabalho é analisar os efeitos de um enxame de drones no combate a incêndio florestais, em termos de supressão, alguns parâmetros externos serão desconsiderados: inclinação do terreno, fumaça e efeitos da temperatura das chamas no voo dos VANTs. Contudo, sabe-se que há implicações dessas variáveis no cenário proposto, por exemplo, causar turbulência nos VANTs e afetar a taxa de espalhamento do incêndio.

Foram utilizados dados baseados nas características do modelo *DJI AGRAS T40*, especificamente, capacidade de carga, velocidade e taxa de fluxo de liberação de água, já que é um VANT com um grande tanque de carga. Por esse motivo, nas simulações de incêndio, foi

considerada uma velocidade máxima, para o vento, de $6m/s$. Pois, essa é a velocidade limite que não afeta o drone usado como parâmetro. Além disso, na etapa de simulação, foi considerada uma comunicação direta entre os VANTs, sem mediação da base de controle.

5.3.1 *Análise utilizando FC*

Antes de começar a etapa da simulação, foi realizada uma análise de como diferentes valores para as variáveis nas Equações 2.17 e 2.18 afetam os resultados. Foram obtidos resultados para as seguintes questões:

- FC em função da velocidade do vento e da umidade do combustível;
- FC em função da velocidade do vento e da profundidade das chamas P ;
- FC em função da umidade do combustível e de P ;
- m_n em função da velocidade do vento e da quantidade de drones;
- m_n em função da velocidade do vento e de P ;
- m_n em função da umidade do combustível e de P ;
- m_n em função da taxa de espalhamento RoS e da quantidade de drones.

5.3.2 *Desenvolvimento do A-star*

Como primeiro passo para simular o combate ao incêndio, foi desenvolvido um algoritmo de *A-star* para levar os drones até o incêndio. O algoritmo considera que cada pixel do mapa é um nó sujeito a testagem de melhor caminho do *A-star*.

5.3.3 *Modelagem do Incêndio*

Com o propósito de modelar os efeitos de um incêndio em uma área de vegetação, foi utilizado o modelo de AC apresentado na Seção 2.2.3, com cada célula sendo um quadrado de lado igual a 2 metros. Um tamanho escolhido por ser intermediário, considerando as proporções, citadas em etapas anteriores, do mapa a ser criado. Porém, como citado antes, o efeito de elevação do solo foi desconsiderado.

Nessa etapa, foram realizadas simulações, considerando a velocidade do vento em $4m/s$, $5m/s$ e $6m/s$, com o intuito de analisar a intensidade do incêndio nessas diferentes condições. Cada iteração do AC equivale a 1 minuto, cada teste foi realizado com 18 iterações. A cada minuto, a quantidade de células incendiadas foi guardada em um arquivo de texto.

Em seguida, os arquivos com a quantidade de células incendiadas foram lidos por um script em MATLAB, para a análise da área total perdida em função do tempo em forma de gráfico. Também, para cada valor de velocidade do vento, realizou-se uma dedução da equação que descreve o crescimento do incêndio.

5.3.4 Tarefa de Supressão

Nessa etapa, foi simulado o incêndio evoluindo até certa quantidade de iterações. Nesse ponto, cada drone do enxame, estacionado um ponto distante do incêndio, segue até a parede de chamas ativas. Cada drone fica responsável por uma célula em chamas e, em cada iteração do incêndio, eles avançam para a célula vizinha que está em chamas ou que está em um estado de completamente queimada, mas, em um caso real, poderia ainda haver focos de incêndio que poderiam ser espalhados.

Durante as simulações, a quantidade de células perdidas é guardada em um arquivo de texto, mesmo processo realizado na Seção 5.3.3, possibilitando a análise de como o incêndio se comporta com a presença dos drones. Por fim, esses dados foram plotados em forma de gráfico contínuo utilizando um script em MATLAB.

Os seguintes testes foram realizados com os drones entrando em ação em 5, 10 e 15 iterações do incêndio, totalizando 9 simulações:

- 27 drones atuando em um incêndio com velocidade do vento em $4m/s$
- 27 drones atuando em um incêndio com velocidade do vento em $5m/s$
- 40 drones atuando em um incêndio com velocidade do vento em $6m/s$

5.4 Desenvolver Rede LoRaWAN

Para montar a rede LoRaWAN foram usadas as SBCs, *Beagle Bone Black* e *Orange PI 5*, e um computador pessoal, como nós do enxame, cada um conectado a uma STM32 *bluepill*, por possuir uma compatibilidade com a tecnologia

6 MÉTODOS E MATERIAIS

Neste capítulo, será detalhado o desenvolvimento do ambiente de simulação. Cada uma das subseções seguintes explica sobre as implementações e métodos utilizados para simulação e análise dos resultados.

6.1 Plotagem de FC

Para plotar os gráficos utilizando a equação FC, é necessário o conhecimento dos valores de seus parâmetros. Na Tabela 3, são mostrados a maioria dos parâmetros utilizados para a análise de FC, os valores e as referências dos mesmos.

Tabela 3 – Parâmetros de FC

Símbolo	Valor	Referência
a	3,258	(Martins Fernandes, 2001)
b	0,958	(Martins Fernandes, 2001)
c	0,111	(Martins Fernandes, 2001)
ΔH_c	19500 kJ/kg	(Santoni <i>et al.</i> , 2006)
W	15 t/ha	(Penney <i>et al.</i> , 2019)
r_c	0,27	(Hansen, 2012)
ϕ	1	(Penney <i>et al.</i> , 2019)
τ	1	(Hansen, 2012)
T_g	800K	(Padhi <i>et al.</i> , 2016)
T_c	693K	(Tihay-Felicelli <i>et al.</i> , 2016)
η_{agua}	0,7	(Hansen, 2012)
$L_{v,agua}$	2640 kJ/kg	(Hansen, 2012)

Fonte: Elaborado pelo próprio autor

O valor padrão do parâmetro P é $2m$, de M_d é 14% e de $6m/s$ para U . Contudo, quando FC é calculado em função da profundidade das chamas ativas, umidade do combustível ou velocidade do vento, esses parâmetros variam entre $2 - 8m$, $14 - 42\%$ e $2 - 10m/s$, respectivamente.

Segundo Silvani e Morandini (2009), h pode variar de $24W/m^2/K$, com vento em $1m/s$, até $77W/m^2/K$, com vento em $10m/s$. Com esses valores, foi possível chegar na equação de primeiro grau que descreve h

$$h = \frac{(5,89U + 18,11)}{1000}. \quad (6.1)$$

Hansen (2012) fala que, considerando não haver influência externa de calor,

$$\frac{(\phi \Delta H_c - L_v) \dot{m}_{cr}'' - \dot{q}_p''}{\eta_{agua} L_{v,agua}} \quad (6.2)$$

é equivalente ao valor $12,9 \times 10^{-3}$

No Algoritmo 1, é apresentado um exemplo de script para plotagem de FC em função de seus parâmetros, velocidade do vento e umidade do combustível, nesse caso. Todo o processo do cálculo é feito dentro de dois laços aninhados, já que o valor de FC depende de dois vetores que representam a velocidade do vento e a umidade. Para as demais análises, considerando que outros parâmetros variam, é necessário apenas mudar os vetores e adaptar a variável equivalente no cálculo para considerar vetor.

Algoritmo 1: Script em MATLAB para análise de FC

```

1 for j = 1:length(md)
2     for i = 1:length(U)
3
4         %calculo de FC
5
6         h = (5.89*U(i)+18.11)/1000;
7         vento = U(i)*3.6;
8         RoS = 0.06*a*vento^b * exp(-c*md(j));
9         I = (deltaH_c*W*RoS)/36;
10        C = (13*RoS+0.24*W)/2;
11
12        q_e = ((r_c*I)/(2*C+P))*phi*tau+h*(T_g-T_c);
13
14        FC = (60*P*(0.0129 + q_e))/(eta_agua*L_vagua);
15
16        result = [result, FC];
17
18    end
19 plot(U, result, 'DisplayName', strcat(num2str(md(j)), '%'));
20 result = [];
21 end
22
23 %Processo de plotagem
24
25 xlabel('Velocidade m/s');
26 ylabel('FC em L/min*m');
27 title('FC em fun ao do vento e da umidade');
28 legend
29 grid on;
30 hold off;

```

6.2 Estimativa de metros lineares apagados

Para fazer a estimativa de metros lineares de incêndio apagados, é necessário adicionar a função 2.18, com m_n isolado, ao final do cálculo de FC, além de adaptar o código, nos laços e variáveis, para os vetores de valores necessários. Um exemplo de script para esse caso é mostrado no Algoritmo 2.

Algoritmo 2: Script em MATLAB para análise de estimativa de metros de incêndio apagados (em função da profundidade do incêndio e umidade)

```

1 for j = 1:length(P)
2     for i = 1:length(md)
3         %calculo de FC
4
5         h = (5.89*U+18.11)/1000;
6         vento = U*3.6;
7         RoS = 0.06*a*vento^b * exp(-c*md(i));
8         I = (deltaH_c*W*RoS)/36;
9         C = (13*RoS+0.24*W)/2;
10
11         q_e = ((r_c*I)/(2*C+P(j)))*phi*tau+h*(T_g-T_fuel);
12
13         FC = (60*P(j)*(0.0129 + q_e))/(eta_agua*L_vagua);
14
15         mn = (Ld*nn)/(8*FC);
16
17         result = [result, mn];
18
19     end
20     plot(md, result, 'DisplayName', strcat(num2str(P(j)), 'm'));
21     result = [];
22 end
23
24 %Processo de plotagem
25
26 xlabel('Umidade (m/s)');
27 ylabel('Frente de fogo ativa apagada (mn) em metros');
28 title('mn em fun  o da umidade e da profundidade do incendio');
29 legend;
30 grid on;
31 hold off;
32 result = [];

```

Fonte: Elaborado pelo autor.

O valor definido para Δt foi de 8 minutos. Pois, considerando que o incêndio está a 1km da base dos drones e os parâmetros de velocidade e vazão de água do drone:

- O drone leva 2,4 minutos para ir até o incêndio e o mesmo valor para voltar a base, pois a

velocidade do drone é $7m/s$ ou, aproximadamente, $420m/min$;

- O drone leva 3,33 minutos para descarregar toda a água de seu tanque, ao ter uma vazão de $12L/min$ e carregar até 40 litros.

6.3 Desenvolvimento do *A-star*

Primeiro, foi definida uma classe que descreve um nó, com cada pixel sendo considerado um nó. No Algoritmo 3, é mostrada a estrutura dessa classe:

- x e y : as coordenadas do nó;
- g , h e f : os valores da Equação 2.1;
- $parent$: o nó anterior

Algoritmo 3: Classe que descreve um nó

```

1 class Node:
2     def __init__(self, x, y):
3         self.x = x
4         self.y = y
5         self.g = 0
6         self.h = 0
7         self.f = 0
8         self.parent = None

```

Fonte: Elaborado pelo autor.

O código principal do *A-star* possui três funções, as duas primeiras são mostradas no Algoritmo 4. A primeira, calcula a distância do nó atual até o nó de destino ($h(n)$). Já a segunda, verifica se as coordenadas do nó estão dentro das dimensões do mapa.

Algoritmo 4: Cálculo do valor de $h(n)$ e verificação se o nó pertence ao mapa

```

1 #calcula por distancia do no atual ate o no de destino (h(n))
2 def heuristic(node, target):
3     return abs(node.x - target.x) + abs(node.y - target.y)
4
5
6 #Valida que um ponto esta dentro do mapa
7 def is_valid_point(x, y):
8     return 0 <= x < map_width and 0 <= y < map_height

```

Fonte: Elaborado pelo autor.

No Algoritmo 5, é mostrada a primeira parte da função principal do *A-star*. Nessa parte, há apenas a inicialização das listas de nós abertos e fechados, além da atribuição de valores para os componentes g , h e f do nó inicial.

Algoritmo 5: Primeira parte da função principal de *A-star*

```

1 #A*
2 def astar(start, target):
3     open_list = []      #lista de nos abertos
4     closed_list = []   #lista de nos fechados
5
6     #Inicializa o no inicial
7     start.g = 0
8     start.h = heuristic(start, target)
9     start.f = start.h
10    heapq.heappush(open_list, start)

```

Fonte: Elaborado pelo autor.

Na segunda parte da função, apresentada no Algoritmo 6, inicia o laço principal do algoritmo, ou seja, esse laço irá continuar executando até que a lista de nós abertos esteja vazia. Nesse trecho de código, o nó com menor valor de $f(n)$ é retornado como o nó atual. Também é verificado se o nó atual é o nó de destino.

Algoritmo 6: Segunda parte da função principal de *A-star*

```

1 #procura o melhor caminho enquanto existir algo na lista de abertos
2 while len(open_list) > 0:
3     #retorna o no com menor valor de f e coloca o no atual na lista
4     #de fechados
5     current_node = heapq.heappop(open_list)
6     closed_list.append(current_node)
7
8     #Verifica se chegou no ponto de destino
9     if current_node == target:
10        path = []
11        while current_node is not None:
12            path.append((current_node.x, current_node.y))
13            current_node = current_node.parent
14        return path[::-1]

```

Fonte: Elaborado pelo autor.

Na última parte da função, dois laços aninhados são usados para gerar os vizinhos do nó atual, calcular os valores para a Função 2.1 desses vizinhos, verificar se eles já foram considerados pelo algoritmo anteriormente e, caso já tenham sido, analisar sua função $f(n)$ tem um valor menor que o valor anterior. No final, todos os vizinhos gerados são adicionados na lista de abertos.

Algoritmo 7: Terceira parte da função principal de *A-star*

```

1 #Gera os vizinhos do no atual
2 neighbors = []
3 for dx in [-1, 0, 1]:
4     for dy in [-1, 0, 1]:
5         if dx == 0 and dy == 0:
6             continue
7         new_x = current_node.x + dx
8         new_y = current_node.y + dy
9         if is_valid_point(new_x, new_y):
10
11             #calcula o custo para o novo no
12             neighbor = Node(new_x, new_y)
13             neighbor.g = current_node.g + 1
14             neighbor.h = heuristic(neighbor, target)
15             neighbor.f = neighbor.g + neighbor.h
16             neighbor.parent = current_node
17
18             #Verifica se o vizinho ja esta na lista aberta ou
19             #Verifica se o vizinho ja foi considerado anteriormente
20             if neighbor in closed_list:
21                 continue
22             if neighbor in open_list:
23                 existing_neighbor = open_list[open_list.index(
24                     neighbor)]
25
26                 #verifica se os valores atuais (f,g) do vizinho sao
27                 #os anteriores
28                 if neighbor.g < existing_neighbor.g:
29                     existing_neighbor.g = neighbor.g
30                     existing_neighbor.f = neighbor.f
31                     existing_neighbor.parent = neighbor.parent
32             else:
33                 neighbors.append(neighbor)
34
35 #Adiciona os vizinhos na lista aberta
36 for neighbor in neighbors:
37     heapq.heappush(open_list, neighbor)

```

6.4 Modelagem do incêndio

A mesma estratégia, de criar uma classe para descrever um nó na seção passada, é utilizada para representar uma célula para o AC. Essa classe é representada no Algoritmo 8.

As componentes x e y , representam as coordenadas da célula em relação ao mapeamento do AC. *Color* determina qual cor a célula recebe em função de seu estado. Já *value* e *value2*, determinam, respectivamente, o estado de AC que a célula se encontra (com combustível, queimando ou totalmente queimada) e se um drone já está naquela célula.

As componentes p_{veg} , p_m , p_{den} e p_w determinam, respectivamente, os valores p_{veg} , p_m , p_{den} e p_w da Equação 2.2. Foi necessário mudar p_d para p_m , representado pela Equação 6.3, pois foi considerado terreno plano e umidade da vegetação.

$$p_m = e^{-bM_d} \quad (6.3)$$

Algoritmo 8: Classe para as células do AC

```

1 class Cell:
2     def __init__(self, x, y):
3         self.x = x
4         self.y = y
5         self.color = None
6         self.value = 2
7         self.p_veg = 0
8         self.p_m = 0
9         self.p_den = 0
10        self.p_w = 0
11        self.value2 = 0

```

Fonte: Elaborado pelo autor.

Para desenhar o mapa dinâmico, foram usadas duas funções apresentadas no Algoritmo 9. A primeira, cria um mapa de células para o AC, utilizando um tamanho pré-definido de célula, 8x8 nesse caso, e as dimensões do mapa de pixels.

A segunda, basicamente, atualiza o mapa pintando as células a partir do estado no qual a mesma se encontra. Também, desenha os drones no mapa em cada iteração, realizando um efeito de movimento deles.

A cor base das células é verde, passa a ser vermelho quando estão em chamas, cinza para totalmente queimadas e verde mais claro quando as chamas são apagadas. Ou seja, se uma célula estiver em chamas ou no estado de queimada e um drone estiver nela, na próxima

interação a célula passar, totalmente, para o estado de apagada e não mais em combustão. Os valores que representam cada estado citado são, respectivamente, 2, 3, 4 e 1. Os drones, são representados por um círculo azul.

Algoritmo 9: Funções para desenhar o mapa

```

1  #Cria o mapa de celula usando o tamanho de uma celula e as
    dimensoes do mapa de pixels
2  def create_cell_map():
3      cells_x = map_width // cell_size
4      cells_y = map_height // cell_size
5      return [[Cell(x, y) for x in range(cells_x)] for y in range(
        cells_y)]
6
7
8  #Desenha o mapa
9  def draw_burn(map_data, drones):
10     map_img = np.zeros((map_height, map_width, 3), dtype=np.uint8)
11     map_img[:] = color_green
12
13     #Pinta as celulas de acordo com o seu estado de AC
14     for row in map_data:
15         for cell in row:
16             x1 = cell.x * cell_size
17             y1 = cell.y * cell_size
18             x2 = (cell.x + 1) * cell_size - 1
19             y2 = (cell.y + 1) * cell_size - 1
20
21             if cell.value == 1:
22                 cv2.rectangle(map_img, (x1, y1), (x2, y2),
23                               color_green2, -1)
24             if cell.value == 3:
25                 cv2.rectangle(map_img, (x1, y1), (x2, y2),
26                               color_red, -1)
27             if cell.value == 4:
28                 cv2.rectangle(map_img, (x1, y1), (x2, y2),
29                               color_gray, -1)
30
31     #Faz o efeito de movimento do drone
32     for drone in drones:
33         cv2.circle(map_img, (drone.current_node.x, drone.
34                       current_node.y), 3, color_blue, -1)
35
36     cv2.imshow("Map", map_img)
37     cv2.waitKey(1)

```

Fonte: Elaborado pelo autor.

A vegetação, escolhida para a simulação, é composta por grama e arbustos. Cada

célula do AC tem uma probabilidade de 50% de ser ocupada com grama ou arbusto, e uma probabilidade de 33% de possuir uma densidade de vegetação espalhada, normal ou densa.

Os valores escolhidos para compor a Equação 2.2, foram retirados de Ausonio *et al.* (2021) e mostrados na Tabela 4

Tabela 4 – Parâmetros de AC

p_{veg}	valores
Grama	0,4
Arbusto	0,4
p_m	valores
Grama	0,18
Arbusto	0,24
p_{den}	valores
Espalhada	-0,4
Normal	0
Densa	0,3
Outros parâmetros	valores
p_h	0,6
c_1	0,045
c_2	0,131
b	0,111

Fonte: Elaborado pelo próprio autor

No Algoritmo 10, são mostradas as funções usadas para calcular as componentes da Equação 2.2. O θ , dado em radianos, depende da posição que uma célula não queimada está de uma célula em chamas. Os ângulos dessas posições são apresentados na Figura 10. Já map , é a célula que está sendo processada.

Algoritmo 10: Funções de definição do AC

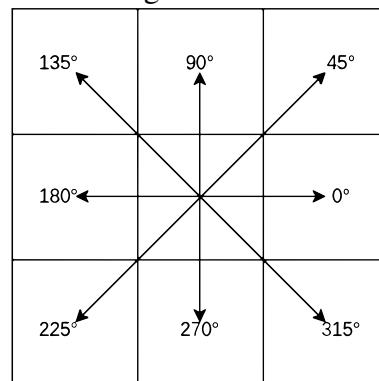
```

1 #Calcula pm
2 def prop_pm(mb):
3     return mt.exp((-b)*mb)
4
5 #calcula pv
6 def prob_pw(theta):
7     return mt.exp(c1*U)*mt.exp(U*c2*(mt.cos(theta)-1))
8
9 #calcula p_queima
10 def pburn(map, theta):
11     return PROB_PH*(1+map.p_veg)*(1+map.p_den)*map.p_m*(prob_pw(
        theta))

```

Fonte: Elaborado pelo autor.

Figura 10 – Ângulos das células no AC



Fonte: Elaborado pelo próprio autor

6.4.1 Efeito do vento

Para ter um efeito de vento, mais realista, nas chamas, foi adaptada a mesma estratégia de espalhamento usada em Freire e DaCamara (2019). Esse método consiste em espalhar as chamas para mais de uma célula, dependendo da velocidade do vento.

No caso deste trabalho, caso o vento esteja a uma velocidade de $5m/s$, as chamas vão se espalhar para 2 células que estiverem na mesma direção do vento. Caso o vento esteja a $6m/s$, as chamas vão ser espalhadas para 3 células que estiverem na mesma direção do vento e para as duas células nas diagonais da célula que as chamas se originaram (90° e 315°).

6.4.2 Plotagem dos gráficos de evolução do incêndio

Foi utilizado um script simples de plotagem, usando MATLAB, para plotar os dados coletados durante as simulações. E para a dedução das fórmulas, que descrevem o crescimento do incêndio para cada velocidade do vento, foi usada a ferramenta *curveFitter*, presente no MATLAB.

6.5 Modelagem do incêndio com a presença dos drones

Seguindo a mesma estratégia das células e nós do *A-star*, foi criada uma classe para descrever um drone, esta classe é apresentada no Algoritmo 11.

As componentes *current_node* e *target_node* representam, respectivamente, o nó atual do drone e o nó de destino. Já *cell_x* e *cell_y*, são as coordenadas da célula que o drone está no momento. Por último, *state* diz se o drone já se moveu em determinada iteração.

Algoritmo 11: Classe de drone

```

1 class Drone:
2     def __init__(self, start_node):
3         self.current_node = start_node
4         self.target_node = None
5         self.cell_x = 0
6         self.cell_y = 0
7         self.state = 0

```

Fonte: Elaborado pelo autor.

Considerando que 4 pixels equivalem a 1 metro, o lado de um pixel tem o tamanho de 0,25cm e diagonal de 0,35cm. Simplificando essas medidas, um pixel equivale a 0,3cm.

Cada drone é criado em um ponto fixo no mapa, nesse caso a célula de coordenada (15,15). Considerando que cada drone percorre, aproximadamente, 420 metros em 1 minuto, então ele passa por 1400 pixels nesse período (supondo o valor de 0,3cm). Isso é verificado em cada iteração da simulação, para ser considerado que o incêndio continua a evoluir em cada minuto. Já que não é possível ter o controle das coordenadas da parede de fogo na iteração onde os drones são liberados, apenas nessa parte da simulação, foi considerada a distância da célula inicial dos drones e foco inicial do incêndio.

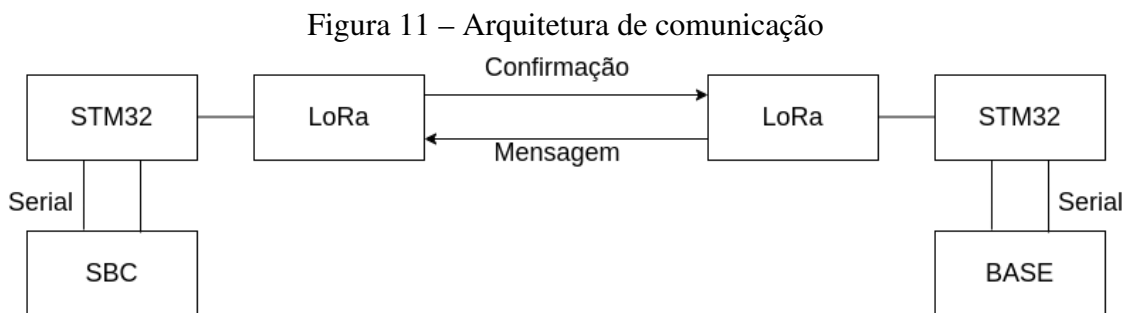
O código principal, utilizado na simulação, está no Anexo A.

6.6 Rede LoRaWAN

Nessa seção, são mostrados os métodos utilizados para montar a rede *LoRaWAN*.

6.6.1 Arquitetura

Na Figura 11, está representada a arquitetura de comunicação usada na rede *LoRaWAN*. A base envia, via serial, uma mensagem para sua STM32, posteriormente, o microcontrolador manda a mensagem recebida para o nó *LoRa* de destino da mensagem. A STM do destino, envia, via serial, a mensagem para a SBC de seu nó. Caso a mensagem seja recebida com sucesso, o mesmo processo de envio é feito, em sentido contrário, para confirmação de recebimento. A comunicação entre a STM32 e o módulo é feita via SPI.

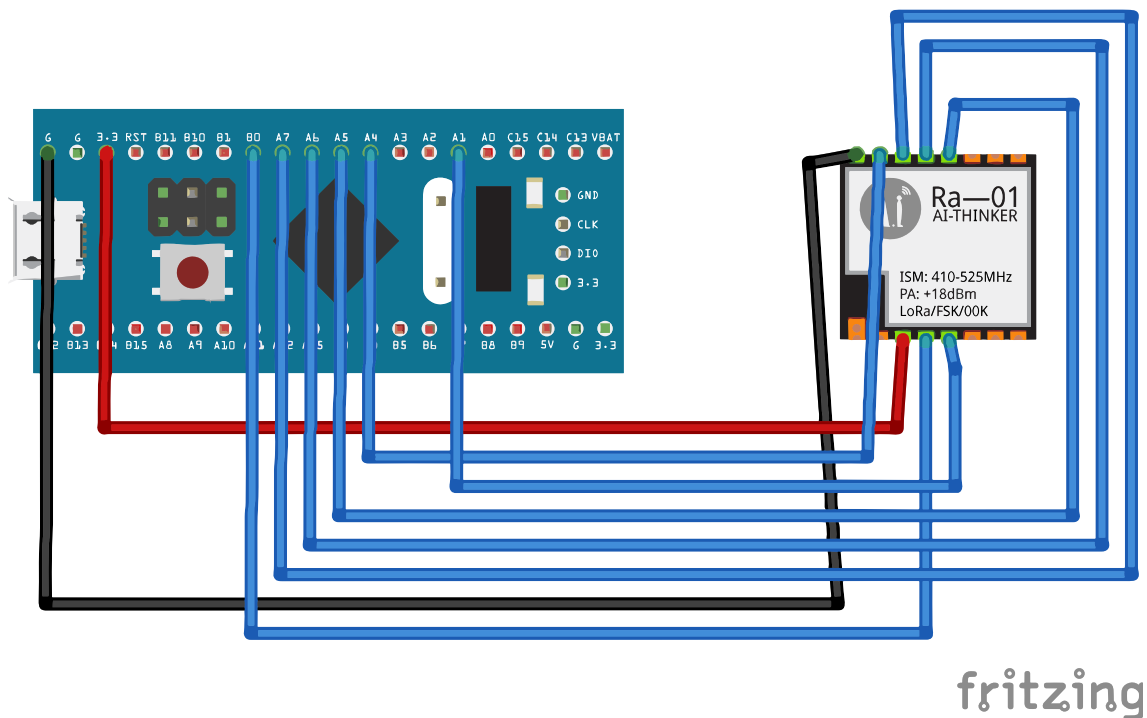


Fonte: Elaborado pelo próprio autor

6.6.2 Circuito

O circuito que conecta o módulo *LoRa* na STM32, é mostrado na Figura 12. Os pinos A4, A5, A6 e A7, estão conectados, respectivamente, aos pinos NSS, SCK, MISO e MOSI do módulo, estes usados para comunicação SPI. O pino A1 está sendo utilizado como um GPIO auxiliar em DIO0 e B0 como *RESET* do módulo.

Figura 12 – Circuito LoRa STM32



Fonte: Elaborado pelo próprio autor

6.7 *Sender e Receiver*

No Algoritmo 12, é mostrada a configuração padrão utilizada para o nó base e o nós que representam os drones. Cada módulo foi configurado com uma potência de sinal de 20dBm (decibel miliwatt) e fator de espalhamento 12.

Algoritmo 12: Configuração dos módulos LoRa

```
1 void setup() {
2   Serial.begin(115200);
3   while (!Serial);
4
5   Serial.println("LoRa Sender");
6   LoRa.setTxPower(TX_P); //Potencia do sinal em 20dBm
7   LoRa.setSpreadingFactor(12); //Fator de espalhamento
8   LoRa.setSyncWord(ENCRYPT); //Palavra de encriptacao
9
10  LoRa.setPins(SS, RST, DIO); //config dos pinos
11
12  //Verifica se o LoRa inicializou corretamente
13  if (!LoRa.begin(BAND))
14  {
15    Serial.println("Starting LoRa failed!");
16    while (1);
17  }
18 }
```

Fonte: Elaborado pelo autor.

A função de envio de dados do *Sender* (base), apresentado no Algoritmo 13, para cada nó envia um pacote contendo seu endereço, o endereço de destino e os dados a serem enviados. Logo depois, ocorre uma espera de 10 segundos por um sinal de confirmação de recebimento.

Algoritmo 13: Função de envio do *Sender*

```

1 void enviarDados() {
2     for(int i = 1; i<=2; i++){
3         //config do pacote para envio
4         Mensagem msg;
5         msg.enderecoDestino = i;
6         msg.enderecoOrigem = MEU_ENDERECO;
7         strncpy(msg.dados, "hello ", sizeof(msg.dados));
8
9         //envio do pacote
10        LoRa.beginPacket();
11        LoRa.write((uint8_t*)&msg, sizeof(msg));
12        LoRa.endPacket();
13
14        String received = "";
15
16        //espera por uma mensagem de confirmacao
17        unsigned long startWaiting = millis();
18        while(millis() - startWaiting < 10000) {
19            int packetSize = LoRa.parsePacket();
20            if(packetSize) {
21                while(LoRa.available()){
22                    received += (char)LoRa.read();
23                }
24                if(received){
25                    break;
26                }
27            }
28        }
29        Serial.println(received);
30    }
31 }

```

Cada *Receiver* (nó que representa um drone), trata uma mensagem recebida, primeiro verificando se o pacote tem como destino aquele endereço e, depois, enviando uma mensagem, para a base, que confirma o recebimento. Isso é apresentado no Algoritmo 14.

Algoritmo 14: Tratamento de mensagem no *Receiver*

```

1 void loop() {
2     //Testa se recebeu algum pacote
3     int tamanhoPacote = LoRa.parsePacket();
4     if (tamanhoPacote) {
5         //trata o pacote
6         Mensagem msg;
7         LoRa.readBytes((uint8_t*)&msg, sizeof(msg));
8
9         //Verifica de a mensagem e para o endereco deste no
10        if (msg.enderecoDestino == MEU_ENDERECO) {
11            //Imprime a mensagem
12            Serial.println(msg.dados);
13            //Manda um sinal de confirmacao de recebimento
14            LoRa.beginPacket();
15            LoRa.print("ack ");
16            LoRa.print(MEU_ENDERECO);
17            LoRa.endPacket();
18        }
19    }
20 }

```

Fonte: Elaborado pelo autor.

Para simplificar os testes, os dados são enviados e tratados diretamente pelas STM32, onde os códigos dessa seção são executados. As SBCs apenas recebem os dados para visualização.

7 RESULTADOS

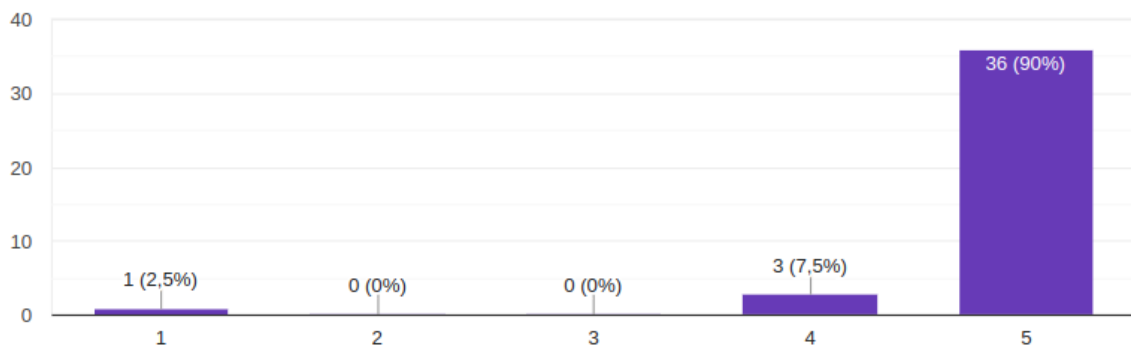
Neste capítulo, são apresentados os resultados obtidos na pesquisa realizada pelos brigadistas, nas análises de FC, nas simulações de incêndio, em diferentes velocidades de vento, com e sem drones.

7.1 Pesquisa com Brigadistas

Foi feito o contato com a central de vários corpos de bombeiros estaduais e solicitado que o formulário de pesquisa fosse repassado para as unidades de combate a incêndios florestais. Ao todo, foram obtidas 40 respostas, a maioria de comandantes de batalhões, representando os demais membros da unidade, especializados em incêndios florestais.

Para a primeira pergunta, cerca de 90% dos entrevistados marcaram 5, em uma escala de 1-5 o quão útil os drones são no cenário de incêndios florestais, 7,5% marcaram 4 e 2,5% assinalaram 1. Resultados mostrados na Figura 13.

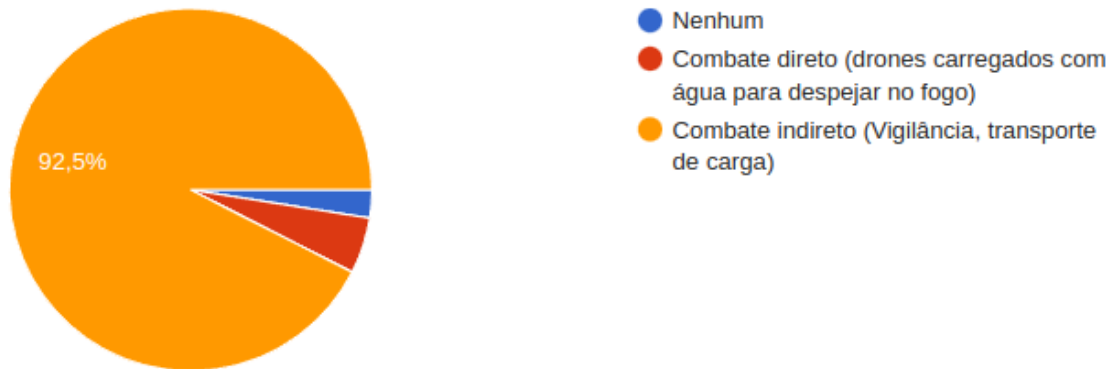
Figura 13 – Porcentagem das respostas para a primeira pergunta



Fonte: Elaborado pelo próprio autor

As respostas para a segunda pergunta mostram que a maioria dos brigadistas acham que os VANTs são mais úteis em tarefas de vigilância e transporte de carga (92,5% das respostas), enquanto apenas 5% dos entrevistados acham que os VANTs podem ser utilizados em tarefas de supressão. Resultados mostrados na Figura 14.

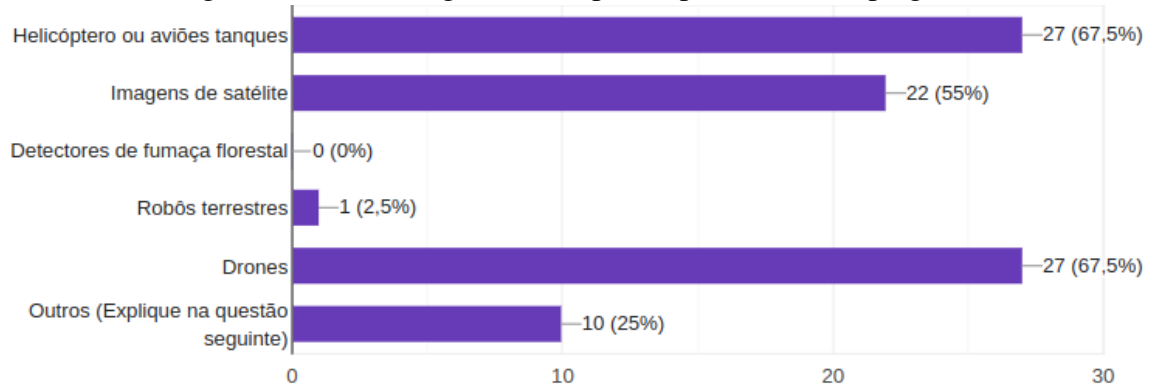
Figura 14 – Porcentagem das respostas para a segunda pergunta



Fonte: Elaborado pelo próprio autor

Na terceira pergunta, segundo os entrevistados, os equipamentos mais utilizados são drones para vigilância, aeronaves tripuladas e equipadas com tanque de água e imagens de satélites, as porcentagens foram, respectivamente, 67,5%, 67,5% e 55%. Vale ressaltar que os drones são operados por humanos e utilizados, basicamente, para vigilância e mapeamento. Outros recursos utilizados são ferramentas físicas, por exemplo, machados e picaretas, aceiros (eliminação de faixa de vegetação para conter a propagação do fogo), bombas costais e veículos terrestres para transporte. Porcentagens mostradas na Figura 15.

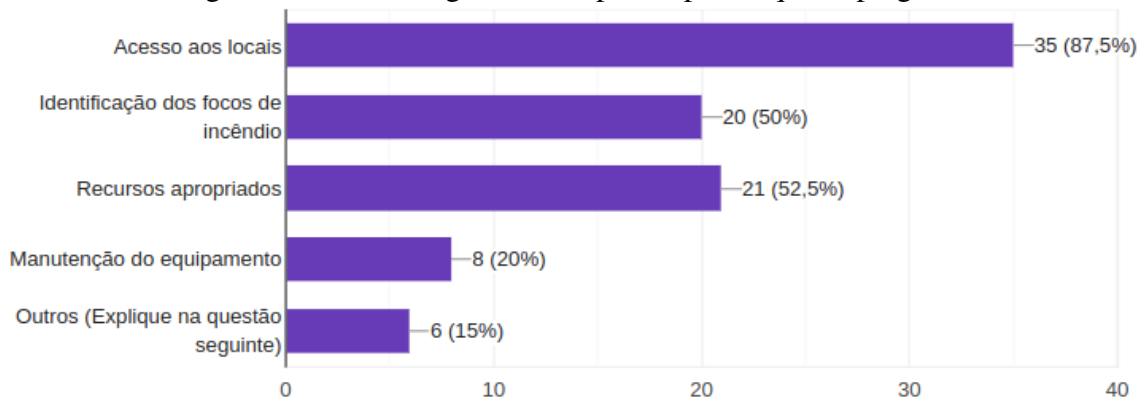
Figura 15 – Porcentagem das respostas para a terceira pergunta



Fonte: Elaborado pelo próprio autor

As respostas da quarta pergunta, mostradas na Figura 16, apresentaram que o maior problema enfrentado pelos brigadistas é a dificuldade de acesso aos locais de incêndio, com 87,5%, seguido da falta de recursos apropriados, com 52,5%, e por último, a dificuldade de detecção de focos, com 50%. Outros desafios citados foram a falta de recursos humanos, a manutenção de equipamentos, a falta de capacitação técnica e a conscientização da população.

Figura 16 – Porcentagem das respostas para a quarta pergunta



Fonte: Elaborado pelo próprio autor

No contexto geral, os brigadistas preferem o uso de VANTs para tarefas de vigilância de incêndios e mapeamento. Contudo, analisando as dificuldades citadas e pelos resultados obtidos nesse trabalho, enxames de VANTs podem atenuar dificuldades como falta de recursos humanos e acesso aos locais de incêndio em missões de supressão.

7.2 Simulação

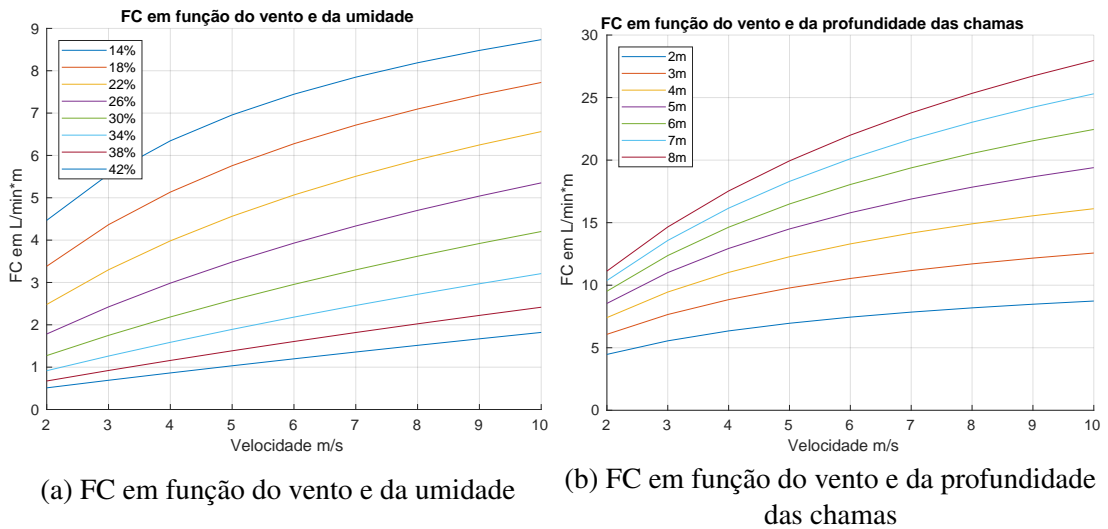
Nessa seção, são apresentados os resultados obtidos da análise de como as mudanças nos parâmetros de FC impacta na evolução de um incêndio e no processo de combate do mesmo.

Quando as variáveis U , M_d , P , L_d e n_n não variam, as mesmas assumem, respectivamente, os valores $6m/s$, 14% , 2 , 40 e 120 .

7.2.1 Análise utilizando FC

Nos gráficos, mostrados na Figura 17, é perceptível que a quantidade de FC, necessário para combater um incêndio, é diretamente proporcional a velocidade do vento. Porém, a taxa de crescimento de FC diminui na medida que a umidade do combustível aumenta, no Gráfico 17a, e que a profundidade das chamas fica menor, no Gráfico 17b.

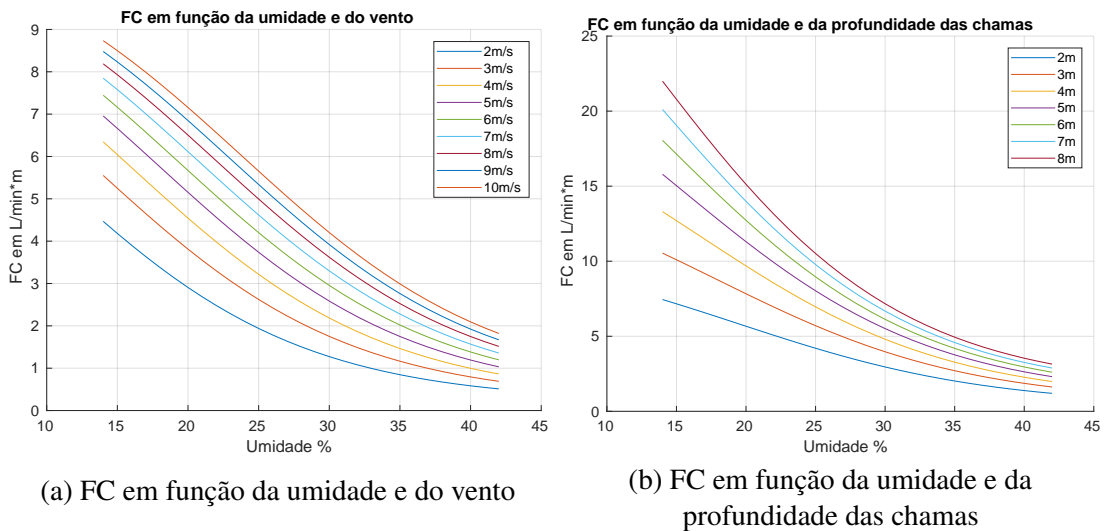
Figura 17 – FC em função do vento, umidade e profundidade das chamas



Fonte: Elaborado pelo próprio autor

Já na Figura 18, são apresentados gráficos que mostram como FC diminui com o crescimento da taxa de umidade do combustível. Contudo, quanto mais alta a velocidade do vento, Gráfico 18a, ou a profundidade das chamas, Gráfico 18b, mais suavizada a curva de decrescimento fica.

Figura 18 – FC em função da umidade, do vento e da profundidade das chamas

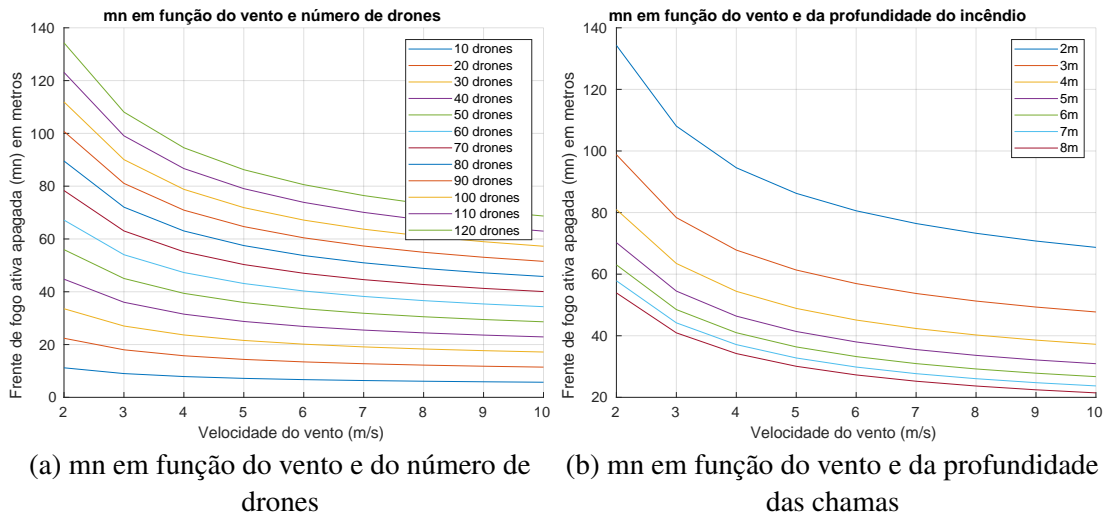


Fonte: Elaborado pelo próprio autor

Na Figura 19, é mostrado nos gráficos o impacto negativo que a velocidade do vento tem na supressão do incêndio, já que a quantidade de metros de fogo apagados diminui à medida que a velocidade do vento aumenta. Porém, ocorre uma queda no efeito do vento com o aumento do número de drones de supressão, Gráfico 19a, e diminuição da profundidade das chamas,

Gráfico 19b.

Figura 19 – mn em função do vento, do número de drones e da profundidade das chamas



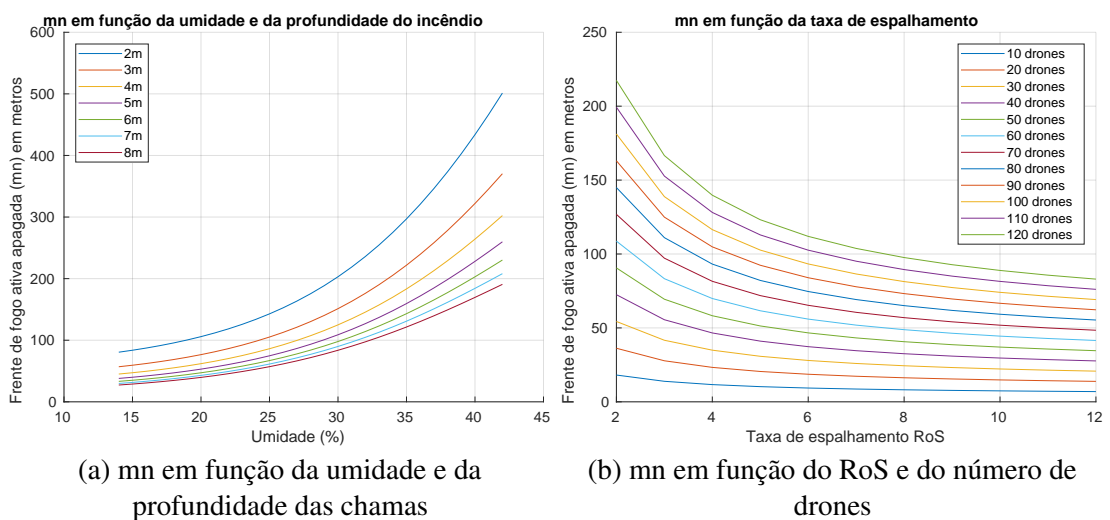
(a) mn em função do vento e do número de drones

(b) mn em função do vento e da profundidade das chamas

Fonte: Elaborado pelo próprio autor

No Gráfico 20a da Figura 20, é possível perceber que o número de metros de incêndio apagado é diretamente proporcional a umidade do combustível, mas seu efeito diminui com o crescimento da profundidade das chamas. Já no Gráfico 20b, é mostrado que a taxa de espalhamento do incêndio impacta negativamente em sua supressão, com essa influência negativa diminuindo com o aumento do número de drones.

Figura 20 – mn em função da umidade, de RoS, da profundidade das chamas e do número de drones



(a) mn em função da umidade e da profundidade das chamas

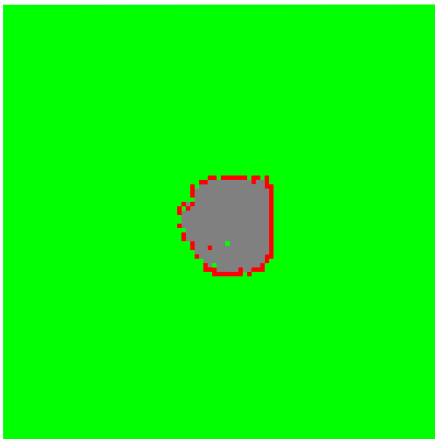
(b) mn em função do RoS e do número de drones

Fonte: Elaborado pelo próprio autor

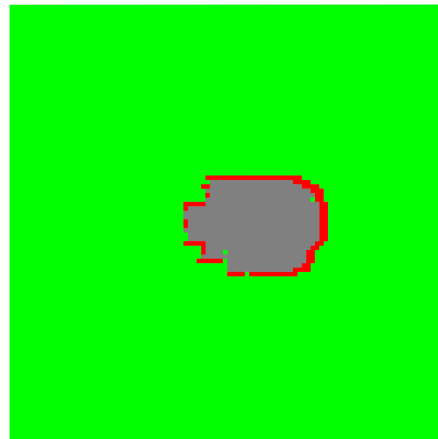
7.2.2 Modelagem do incêndio

Nos três mapas na Figura 21, é mostrado o estado do incêndio sobre a influência da velocidade do vento, respectivamente, $4m/s$, $5m/s$ e $6m/s$, depois de 10 iterações (minutos). É perceptível que as chamas evoluem mais rápido para a direção do vento, leste, a medida que a sua velocidade aumenta.

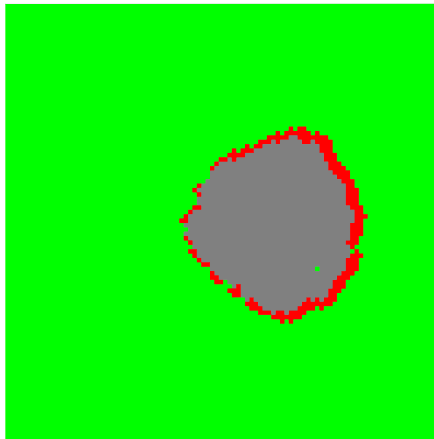
Figura 21 – Simulação do incêndio com velocidade do vento em $4m/s$, $5m/s$ e $6m/s$, respectivamente



(a) Incêndio com velocidade do vento em $4m/s$



(b) Incêndio com velocidade do vento em $5m/s$

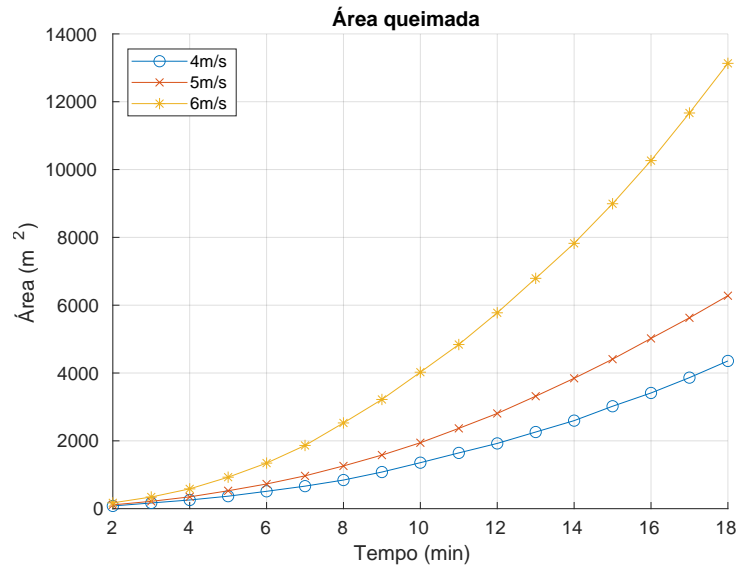


(c) Incêndio com velocidade do vento em $6m/s$

Fonte: Elaborado pelo próprio autor

Na Figura 22, são apresentadas as curvas de crescimento de um incêndio para cada velocidade de vento utilizada na simulação. As curvas que representam $4m/s$ e $5m/s$, possuem crescimentos aproximados, diferente da curva de $6m/s$, que demonstra uma taxa de crescimento maior.

Figura 22 – Área queimada em função do tempo



Fonte: Elaborado pelo próprio autor

Com esses dados da evolução de um incêndio, foi possível deduzir uma fórmula que descreve a área total queimada em função do tempo. Essa fórmula é escrita como

$$a \times e^{bx} + c \times e^{dx}. \quad (7.1)$$

Na Tabela 5, é mostrado os valores dos coeficientes, para cada valor da velocidade do vento, da Equação 7.1.

Tabela 5 – Valores dos coeficientes a, b, c e d

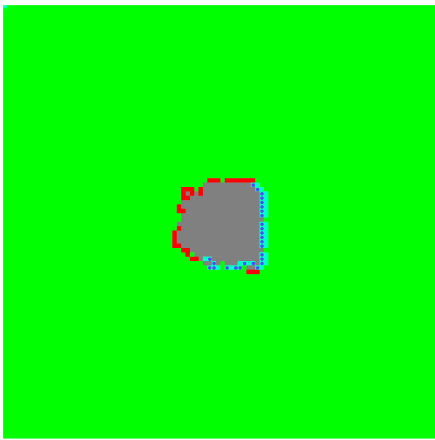
	a	b	c	d
$4m/s$	$6,092 \times 10^7$	0,05952	$-6,092 \times 10^7$	0,05952
$5m/s$	$-1,711 \times 10^6$	0,0637	$1,711 \times 10^6$	0,0637
$6m/s$	$1,223 \times 10^8$	0,05684	$-1,223 \times 10^8$	0,05684

Fonte: Elaborado pelo próprio autor

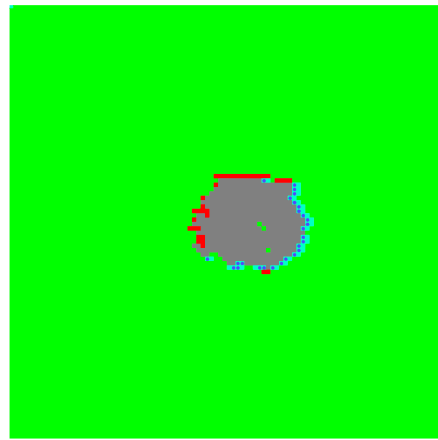
7.2.3 Tarefa de supressão

Como mostrado na Figura 23, foram liberados 27 drones para as simulações de incêndio com $4m/s$ e $5m/s$ de velocidade de vento, Gráficos 23a e 23b, e 40 drones para o cenário com $6m/s$, Gráfico 23c. O verde claro representa as células que já foram apagadas. Os drones não extinguem o incêndio, porém eles retardam a taxa de espalhamento das chamas, já que atuam diretamente na parede de incêndio mais ativa.

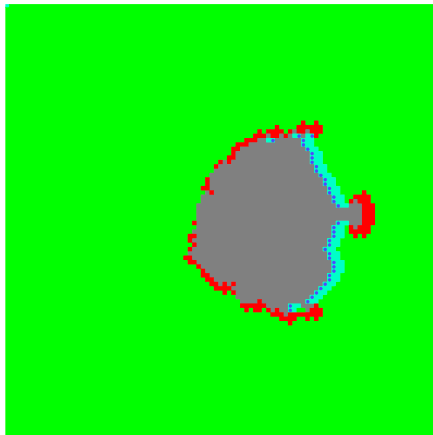
Figura 23 – Simulação do incêndio, com drones, com velocidade do vento $4m/s$, $5m/s$ e $6m/s$, respectivamente



(a) Incêndio, com drones, com velocidade do vento $4m/s$



(b) Incêndio, com drones, com velocidade do vento $5m/s$

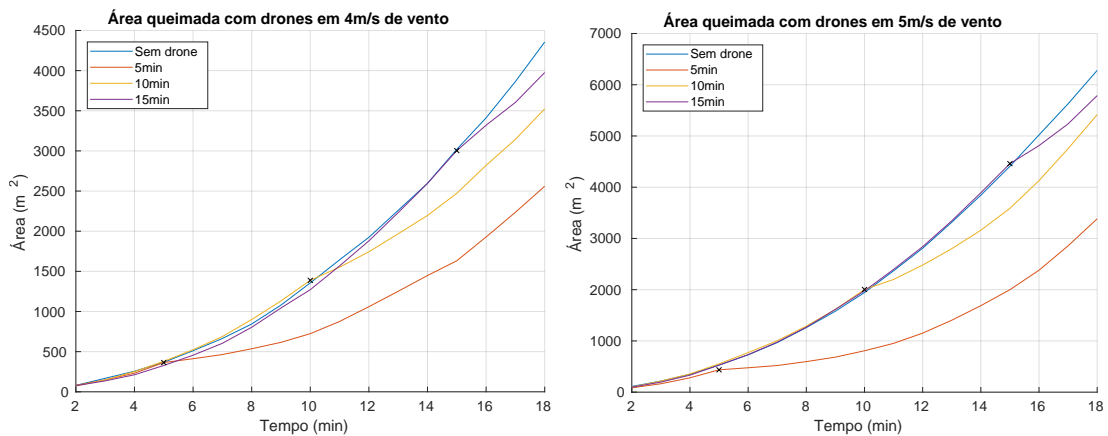


(c) Incêndio, com drones, com velocidade do vento $6m/s$

Fonte: Elaborado pelo próprio autor

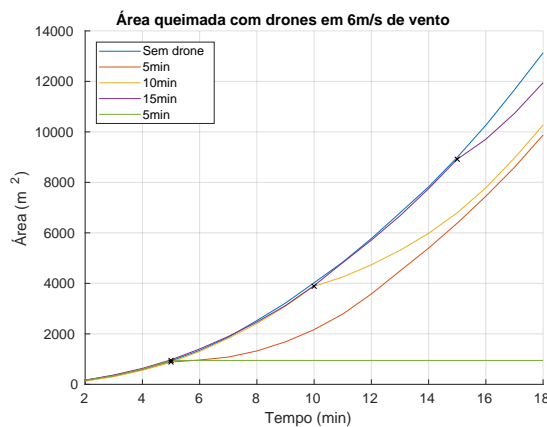
Nos gráficos da Figura 24, é mostrada uma comparação entre o crescimento de um incêndio sem drones e quando os drones são liberados 5, 10 e 15 minutos após o início do incêndio. Percebe-se que quando os drones são liberados após 5 minutos, nos cenários de $4m/s$ e $5m/s$, a taxa de espalhamento das chamas cai drasticamente, Gráficos 24a e 24b. Esse efeito é menor no cenário com velocidade do vento em $6m/s$, Gráfico 24c. Nesse último cenário, foi feita uma simulação considerando a disponibilidade de inúmeros drones, como resultado, é possível notar que a taxa de crescimento ficou constante, ou seja, o incêndio foi totalmente extinto. Esse último caso é representado pela reta verde.

Figura 24 – Dados da simulação do incêndio com velocidade do vento em $4m/s$, $5m/s$ e $6m/s$, respectivamente



(a) Dados da simulação do incêndio com velocidade do vento em $4m/s$

(b) Dados da simulação do incêndio com velocidade do vento em $5m/s$



(c) Dados da simulação do incêndio com velocidade do vento em $6m/s$

Fonte: Elaborado pelo próprio autor

7.3 Desenvolver rede *LoRaWAN*

Na saída mostrada na Figura 25, é possível perceber que o algoritmo de *Sender* está com o comportamento esperado para a arquitetura descrita anteriormente. Para cada pacote, a base faz um envio para todos os nós. Contudo, só envia o pacote para um novo nó, caso tenha recebido o sinal de confirmação do anterior, isso garante que não haverá colisão ou perda de pacotes.

Figura 25 – Saída do *Sender* (base)

```
LoRa Sender
Sending packet: 0
ack 2
ack 3
Sending packet: 1
ack 2
ack 3
Sending packet: 2
ack 2
ack 3
Sending packet: 3
ack 2
ack 3
Sending packet: 4
ack 2
ack 3
```

Fonte: Elaborado pelo próprio autor

Já na saída mostrada na Figura 26, percebe-se que mesmo o *Receiver* não especificando o endereço do *Sender*, quando envia a confirmação de recebimento, os outros nós não conseguem ler a mensagem. Pois, ela é enviada em um padrão diferente do pacote gerado pela base.

Figura 26 – Saída do *Receiver* (drones)

```
LoRa Receiver
hello
hello
hello
hello
█
```

Fonte: Elaborado pelo próprio autor

8 CONCLUSÃO E TRABALHOS FUTUROS

Enxame de VANTs podem ser utilizados como uma forma de apoio aos meios de combate convencionais, pois não precisa de intervenção direta de um humano, diminuindo os riscos de danos aos brigadistas. Também pode ser utilizado para atrasar o espalhamento de um incêndio até que outros meios possam ser usados no local, já que podem ser empregados em locais isolados. Além de ser um método que pode ser usado para detecção de focos de incêndio.

Este trabalho propôs análises e simulações para testar a viabilidade do uso de enxame de VANTs no combate a incêndio florestais. Foram feitas análises, em cenários com diferentes variáveis, do fluxo de água necessário para um incêndio ser extinto e de quantos metros de incêndio seriam suprimidos com certo número de drones. Depois, simulações de incêndios, com vários drones de supressão presentes, foram feitas com diferentes intensidades de vento. Por fim, foram realizados testes de sincronização de mensagens em uma rede *LoRaWAN*, como provável meio de comunicação entre os nós de um enxame.

As análises feitas, considerando o fluxo de água necessário para combater um incêndio, demonstraram resultados compatíveis com o que provavelmente ocorreria em um cenário real com diferentes variáveis. Já as simulações de incêndio com drones presentes, mostraram que um enxame pode ser utilizado como meio auxiliar na supressão de chamas, principalmente se for empregado em momentos iniciais do incêndio. Por fim, foi possível montar uma rede *LoRaWAN* estável em curta distância.

8.0.1 *Trabalhos futuros*

Como trabalhos futuros, pretende-se realizar novas simulações do comportamento de um incêndio com a presença de um enxame. Porém, considerando mais variáveis, como o efeito de fumaça nos drones e inclinação do terreno. Também é pretendido desenvolver algoritmos de visão computacional para detecção de incêndios. Além de desenvolver e fazer novos testes na rede *LoRaWAN*, por exemplo, teste de distância, comunicação entre os drones e adicionar novos nós.

Também pretende-se resolver o problema de localização dos drones com algum método já existente. Por exemplo, fusão de sensores e visão computacional.

REFERÊNCIAS

AFGHAH, F.; RAZI, A.; CHAKARESKI, J.; ASHDOWN, J. Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles. In: **IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. [S. l.: s. n.], 2019. p. 835–840.

AL-SARAWI, S.; ANBAR, M.; ALIEYAN, K.; ALZUBAIDI, M. Internet of things (iot) communication protocols: Review. In: **2017 8th International Conference on Information Technology (ICIT)**. [S. n.], 2017. p. 685–690. Disponível em: [10.1109/ICITECH.2017.8079928](https://doi.org/10.1109/ICITECH.2017.8079928). Acesso em: 10 out. 2023.

ALEXANDRIDIS, A.; VAKALIS, D.; SIETTOS, C.; BAFAS, G. A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through spetses island in 1990. In: **Applied Mathematics and Computation**. [S. n.], 2008. v. 204, n. 1, p. 191–201. ISSN 0096-3003. Disponível em: <https://doi.org/10.1016/j.amc.2008.06.046>. Acesso em: 10 out. 2023.

ASAAMONING, G.; MENDES, P.; ROSÁRIO, D.; CERQUEIRA, E. Drone swarms as networked control systems by integration of networking and computing. In: **Sensors**. [S. n.], 2021. v. 21, n. 8. ISSN 1424-8220. Disponível em: [10.3390/s21082642](https://doi.org/10.3390/s21082642). Acesso em: 10 out. 2023.

AUSONIO, E.; BAGNERINI, P.; GHIO, M. Drone swarms in fire suppression activities: A conceptual framework. In: **Drones**. [S. n.], 2021. v. 5, n. 1. ISSN 2504-446X. Disponível em: [10.3390/drones5010017](https://doi.org/10.3390/drones5010017). Acesso em: 10 out. 2023.

AYDIN, B.; SELVI, E.; TAO, J.; STAREK, M. J. Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting. In: **Drones**. [S. n.], 2019. v. 3, n. 1. ISSN 2504-446X. Disponível em: [10.3390/drones3010017](https://doi.org/10.3390/drones3010017). Acesso em: 10 out. 2023.

BEYLER, C. A unified model of fire suppression by. In: **Journal of Fire Protection Engineering**. [S. n.], 1992. v. 4, n. 1, p. 5–16. Disponível em: <https://doi.org/10.1177/104239159200400102>. Acesso em: 10 out. 2023.

BISDIKIAN, C. An overview of the bluetooth wireless technology. In: **IEEE Communications Magazine**. [S. n.], 2001. v. 39, n. 12, p. 86–94. Disponível em: [10.1109/35.968817](https://doi.org/10.1109/35.968817). Acesso em: 10 out. 2023.

BRADSKI, A. **Computer Vision with OpenCV Library ; software that sees**. [S. l.]: O'Reilly Media, 2008.

BUFACCHI, P.; KRIEGER, G. C.; MELL, W.; ALVARADO, E.; SANTOS, J. C.; CARVALHO, J. A. Numerical simulation of surface forest fire in brazilian amazon. In: **Fire Safety Journal**. [S. n.], 2016. v. 79, p. 44–56. ISSN 0379-7112. Disponível em: <https://doi.org/10.1016/j.firesaf.2015.11.014>. Acesso em: 10 out. 2023.

CAMPION, M.; RANGANATHAN, P.; FARUQUE, S. Uav swarm communication and control architectures: a review. In: **Journal of Unmanned Vehicle Systems**. [S. n.], 2019. v. 7, n. 2, p. 93–106. Disponível em: <https://doi.org/10.1139/juvs-2018-0009>. Acesso em: 10 out. 2023.

CHAUDHARI, B. S.; ZENNARO, M.; BORKAR, S. Lpwan technologies: Emerging application characteristics, requirements, and design considerations. In: **Future Internet**. [S. n.], 2020. v. 12, n. 3. ISSN 1999-5903. Disponível em: 10.3390/fi12030046. Acesso em: 10 out. 2023.

COLEY, G. **BeagleBone Black System Reference Manual**. [S. l.]: Texas Instruments, 2014.

CONTRERAS-CRUZ, M. A.; AYALA-RAMIREZ, V.; HERNANDEZ-BELMONTE, U. H. Mobile robot path planning using artificial bee colony and evolutionary programming. In: **Applied Soft Computing**. [S. n.], 2015. v. 30, p. 319–328. ISSN 1568-4946. Disponível em: <https://doi.org/10.1016/j.asoc.2015.01.067>. Acesso em: 10 out. 2023.

CORP, S. **LoRa ModulationBasics**. [S. l.]: Semtech Corp, 2015.

CORP, S. **SX1276/77/78/79 datasheet**. [S. l.]: Semtech Corp, 2015.

DENG, C.; FANG, X.; HAN, X.; WANG, X.; YAN, L.; HE, R.; LONG, Y.; GUO, Y. Ieee 802.11be wi-fi 7: New challenges and opportunities. In: **IEEE Communications Surveys Tutorials**. [S. n.], 2020. v. 22, n. 4, p. 2136–2166. Disponível em: 10.1109/COMST.2020.3012715. Acesso em: 10 out. 2023.

DEVALAL, S.; KARTHIKEYAN, A. Lora technology - an overview. In: **2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)**. [S. n.], 2018. p. 284–290. Disponível em: 10.1109/ICECA.2018.8474715. Acesso em: 10 out. 2023.

DJI. **T40_T20P User Manual v1.2 EN**. [S. l.]: DJI, 2022.

FEARNSIDE, P. M. **A FLORESTA AMAZÔNICA NAS MUDANÇAS GLOBAIS**. [S. l.]: Editora do Instituto Nacional de Pesquisas da Amazônia (INPA), 2003.

FREIRE, J. G.; DACAMARA, C. C. Using cellular automata to simulate wildfire propagation and to assist in fire management. In: **Natural Hazards and Earth System Sciences**. [S. n.], 2019. v. 19, n. 1, p. 169–179. Disponível em: <https://nhess.copernicus.org/articles/19/169/2019/>. Acesso em: 10 out. 2023.

GKOTSIPOULOS, P.; ZORBAS, D.; DOULIGERIS, C. Performance determinants in lora networks: A literature review. In: **IEEE Communications Surveys Tutorials**. [S. n.], 2021. v. 23, n. 3, p. 1721–1758. Disponível em: 10.1109/COMST.2021.3090409. Acesso em: 10 out. 2023.

HANSEN, R. Estimating the amount of water required to extinguish wildfires under different conditions and in various fuel types. In: **International Journal of Wildland Fire**. [S. n.], 2012. v. 21, p. 525–536. Disponível em: 10.1071/WF11022_CO. Acesso em: 10 out. 2023.

HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. In: **IEEE Transactions on Systems Science and Cybernetics**. [S. n.], 1968. v. 4, n. 2, p. 100–107. Disponível em: 10.1109/TSSC.1968.300136. Acesso em: 10 out. 2023.

HE, N.; QIAN, Y.; HUANG, H.-w. Experience of teaching embedded systems design with beaglebone black board. In: **2016 IEEE International Conference on Electro Information Technology (EIT)**. [S. n.], 2016. p. 0217–0220. Disponível em: 10.1109/EIT.2016.7535243. Acesso em: 10 out. 2023.

INNOCENTE, M. S.; GRASSO, P. Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems. In: **Journal of Computational Science**. [S. n.], 2019. v. 34, p. 80–101. ISSN 1877-7503. Disponível em: <https://doi.org/10.1016/j.jocs.2019.04.009>. Acesso em: 10 out. 2023.

KARUR, K.; SHARMA, N.; DHARMATTI, C.; SIEGEL, J. E. A survey of path planning algorithms for mobile robots. In: **Vehicles**. [S. n.], 2021. v. 3, n. 3, p. 448–468. ISSN 2624-8921. Disponível em: 10.3390/vehicles3030027. Acesso em: 10 out. 2023.

KATTENBORN, T.; LEITLOFF, J.; SCHIEFER, F.; HINZ, S. Review on convolutional neural networks (cnn) in vegetation remote sensing. In: **ISPRS Journal of Photogrammetry and Remote Sensing**. [S. n.], 2021. v. 173, p. 24–49. ISSN 0924-2716. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0924271620303488>. Acesso em: 10 out. 2023.

KHALAJMEHRABADI, A.; GATSIS, N.; AKOPIAN, D. Modern wlan fingerprinting indoor positioning methods and deployment challenges. In: **IEEE Communications Surveys Tutorials**. [S. n.], 2017. v. 19, n. 3, p. 1974–2002. Disponível em: 10.1109/COMST.2017.2671454. Acesso em: 10 out. 2023.

MAC, T. T.; COPOT, C.; TRAN, D. T.; De Keyser, R. Heuristic approaches in robot path planning: A survey. In: **Robotics and Autonomous Systems**. [S. n.], 2016. v. 86, p. 13–28. ISSN 0921-8890. Disponível em: <https://doi.org/10.1016/j.robot.2016.08.001>. Acesso em: 10 out. 2023.

MAHMOOD, A.; SISINNI, E.; GUNTUPALLI, L.; RONDÓN, R.; HASSAN, S. A.; GIDLUND, M. Scalability analysis of a lora network under imperfect orthogonality. In: **IEEE Transactions on Industrial Informatics**. [S. n.], 2019. v. 15, n. 3, p. 1425–1436. Disponível em: 10.1109/TII.2018.2864681. Acesso em: 10 out. 2023.

Martins Fernandes, P. A. Fire spread prediction in shrub fuels in portugal. In: **Forest Ecology and Management**. [S. n.], 2001. v. 144, n. 1, p. 67–74. ISSN 0378-1127. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378112700003637>. Acesso em: 10 out. 2023.

MATTHEWS, S. J.; BLAINE, R. W.; BRANTLY, A. F. Evaluating single board computer clusters for cyber operations. In: **2016 International Conference on Cyber Conflict (CyCon U.S.)**. [S. n.], 2016. p. 1–8. Disponível em: 10.1109/CYCONUS.2016.7836622. Acesso em: 10 out. 2023.

MCCUNE, R. R.; MADEY, G. R. Swarm control of uavs for cooperative hunting with dddas. In: **Procedia Computer Science**. [S. n.], 2013. v. 18, p. 2537–2544. ISSN 1877-0509. 2013 International Conference on Computational Science. Disponível em: <https://doi.org/10.1016/j.procs.2013.05.436>. Acesso em: 10 out. 2023.

MCMANUS, M. C. S. **Raspberry Pi For Dummies**. [S. l.]: TJohn Wiley Sons, 2021.

MEKKI, K.; BAJIC, E.; CHAXEL, F.; MEYER, F. A comparative study of lpwan technologies for large-scale iot deployment. In: **ICT Express**. [S. n.], 2019. v. 5, n. 1, p. 1–7. ISSN 2405-9595. Disponível em: <https://doi.org/10.1016/j.ict.2017.12.005>. Acesso em: 10 out. 2023.

OPENCV. **OpenCV: Introduction**. [S. l.]: Disponível em: <https://docs.opencv.org/4.x/d1/dfb/intro.html>. Acesso em 5 de abril de 2023, 2023.

PADHI, S.; SHOTORBAN, B.; MAHALINGAM, S. Computational investigation of flame characteristics of a non-propagating shrub fire. In: **Fire Safety Journal**. [S. n.], 2016. v. 81, p. 64–73. ISSN 0379-7112. Disponível em: <https://www.sciencedirect.com/science/article/pii/S037971121630011X>. Acesso em: 10 out. 2023.

PENNEY, G.; HABIBI, D.; CATTANI, M.; CARTER, M. Calculation of critical water flow rates for wildfire suppression. In: **Fire**. [S. n.], 2019. v. 2, n. 1. ISSN 2571-6255. Disponível em: <https://www.mdpi.com/2571-6255/2/1/3>. Acesso em: 10 out. 2023.

PI, O. **Orange Pi 5 User Manual**. [S. l.]: Shenzhen Xunlong Software Co., 2022.

PNUMA, P. das Nações Unidas para o M. A. **SPREADING LIKE WILDFIRE THE RISING THREAT OF EXTRAORDINARY LANDSCAPE FIRES**. [S. l.]: UNEP, 2022.

RAMYA, C. M.; SHANMUGARAJ, M.; PRABAKARAN, R. Study on zigbee technology. In: **2011 3rd International Conference on Electronics Computer Technology**. [S. n.], 2011. v. 6, p. 297–301. Disponível em: 10.1109/ICECTECH.2011.5942102. Acesso em: 10 out. 2023.

ROTHERMEL, R. C. **A Mathematical Model for Predicting Fire Spread in Wildland Fuels**. [S. l.]: USDA Forest Service, 1972.

SAFFRE, F.; HILDMANN, H.; KARVONEN, H.; LIND, T. Monitoring and cordoning wildfires with an autonomous swarm of unmanned aerial vehicles. In: **Drones**. [S. n.], 2022. v. 6, n. 10. ISSN 2504-446X. Disponível em: 10.3390/drones6100301. Acesso em: 10 out. 2023.

SANTONI, P.; SIMEONI, A.; ROSSI, J.; BOSSEUR, F.; MORANDINI, F.; SILVANI, X.; BALBI, J.; CANCELLIERI, D.; ROSSI, L. Instrumentation of wildland fire: Characterisation of a fire spreading through a mediterranean shrub. In: **Fire Safety Journal**. [S. n.], 2006. v. 41, n. 3, p. 171–184. ISSN 0379-7112. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0379711205001396>. Acesso em: 10 out. 2023.

SELLER, N. S. O. B. A. **Low power long range transmitter**. [S. l.]: Semtech Corp, 2016.

SILVANI, X.; MORANDINI, F. Fire spread experiments in the field: Temperature and heat fluxes measurements. In: **Fire Safety Journal**. [S. n.], 2009. v. 44, n. 2, p. 279–285. ISSN 0379-7112. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0379711208000817>. Acesso em: 10 out. 2023.

SONG, Y.; LIN, J.; TANG, M.; DONG, S. An internet of energy things based on wireless lpwan. In: **Engineering**. [S. n.], 2017. v. 3, n. 4, p. 460–466. ISSN 2095-8099. Disponível em: <https://doi.org/10.1016/J.ENG.2017.04.011>. Acesso em: 10 out. 2023.

SORNIN, N.; YEGIN, A. **LoRaWAN 1.1 Specification**. [S. l.]: LoRa Alliance, 2017.

SPALDING, D. The combustion of liquid fuels. In: **Symposium (International) on Combustion**. [S. n.], 1953. v. 4, n. 1, p. 847–864. ISSN 0082-0784. Disponível em: [https://doi.org/10.1016/S0082-0784\(53\)80110-4](https://doi.org/10.1016/S0082-0784(53)80110-4). Acesso em: 10 out. 2023.

STMICROELECTRONICS. **Medium-density performance line Arm®-based 32-bit MCU with 64 or 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 com. interfaces**. [S. l.]: STMicroelectronics, 2023.

TIHAY-FELICELLI, S.; SANTONI, P.-A.; BARBONI, T.; LEONELLI, L. Autoignition of dead shrub twigs: Influence of diameter on ignition. In: **Fire Technol.** [S. l.: s. n.], 2016. v. 52, p. 897–929.

VARGAS, C. V.; ZAKRZEWSKI, S. B. B. Importância das florestas e principais fontes de informação para agricultores do norte do rio grande do sul. In: **Revista Brasileira de Educação Ambiental (RevBEA)**. [S. n.], 2020. v. 15, n. 7, p. 259–276. Disponível em: 10.34024/revbea.2020.v15.10231. Acesso em: 10 out. 2023.

WOLFRAM, S. Statistical mechanics of cellular automata. In: **Reviews of Modern Physics**. [S. n.], 1983. v. 55, p. 601–644. Disponível em: <https://doi.org/10.1103/RevModPhys.55.601>. Acesso em: 10 out. 2023.

ZHANG, H.-y.; LIN, W.-m.; CHEN, A.-x. Path planning for the mobile robot: A review. In: **Symmetry**. [S. n.], 2018. v. 10, n. 10. ISSN 2073-8994. Disponível em: <https://doi.org/10.3390/sym10100450>. Acesso em: 10 out. 2023.

ANEXO A – CÓDIGO COMPLETO DE SIMULAÇÃO DO INCÊNDIO

Algoritmo 15: Código principal da simulação do incêndio

```

1 #Causa o efeito do vento
2 def efWind(map,x,y):
3     if(U >= 5 and map[x][y+1].value == 2):
4         map[x][y+1].value = 3
5         vx1.append(x)
6         vy1.append(y+1)
7
8     if(U >= 6 and map[x][y+2].value == 2):
9         map[x][y+2].value = 3
10        vx1.append(x)
11        vy1.append(y+2)
12
13       if(U >= 6 and map[x-1][y+1].value == 2):
14           map[x-1][y+1].value = 3
15           vx1.append(x-1)
16           vy1.append(y+1)
17
18       if(U >= 6 and map[x+1][y+1].value == 2):
19           map[x+1][y+1].value = 3
20           vx1.append(x+1)
21           vy1.append(y+1)
22
23 map_data = create_cell_map()
24
25 #Para cada celula, colocar um valor de umidade, densidade e o tipo
   de vegeta ao
26 for x1 in range(100):
27     for y1 in range (100):
28         map_data[x1][y1].p_veg = 0.4
29         random.seed(random.randint(1,10000))
30         if(random.random() <= PROB_UMI):
31             map_data[x1][y1].p_m = prop_pm(md_grama)
32         else:
33             map_data[x1][y1].p_m = prop_pm(md_arbusto)
34
35         rd = random.random()
36         if(rd > PROB_DEN+0.33):
37             map_data[x1][y1].p_den = pden_denso
38         elif(rd > 0.33):
39             map_data[x1][y1].p_den = pden_normal
40         else:
41             map_data[x1][y1].p_den = pden_espalhado

```

Algoritmo 16: Código principal da simulação do incêndio

```

1  #Cria o foco de incêndio inicial
2  vx1.append(x)
3  vy1.append(y)
4  map_data[x][y].value = 3
5  draw_burn(map_data, drones)
6
7  #Espalha o incêndio pelo mapa de acordo com p_queima de cada
   célula e o
8  #estado atual da mesma, presente em cell.value
9  while True:
10     random.seed(random.randint(1,1000))
11     vx2 = vx1[:]
12     vy2 = vy1[:]
13     for i in range(len(vx2)):
14         x = vx2[i]
15         y = vy2[i]
16
17         if(map_data[x+1][y].value == 2 and map_data[x][y].value ==
           3 and random.random() <= pburn(map_data[x+1][y],
           4.71239)):
18             map_data[x+1][y].value = 3
19             vx1.append(x+1)
20             vy1.append(y)
21             efWind(map_data, x+1,y)
22
23         if(map_data[x-1][y].value == 2 and map_data[x][y].value ==
           3 and random.random() <= pburn(map_data[x-1][y], 1.5708)
           ):
24             map_data[x-1][y].value = 3
25             vx1.append(x-1)
26             vy1.append(y)
27             efWind(map_data, x-1,y)
28
29         if(map_data[x][y+1].value == 2 and map_data[x][y].value ==
           3 and random.random() <= pburn(map_data[x][y+1], 0)):
30             map_data[x][y+1].value = 3
31             vx1.append(x)
32             vy1.append(y+1)
33             efWind(map_data, x,y+1)
34
35         if(map_data[x][y-1].value == 2 and map_data[x][y].value ==
           3 and random.random() <= pburn(map_data[x][y-1],
           3.14159)):
36             map_data[x][y-1].value = 3
37             vx1.append(x)
38             vy1.append(y-1)
39             efWind(map_data, x,y-1)

```

Algoritmo 17: Código principal da simulação do incêndio

```

1
2     if (map_data[x+1][y+1].value == 2 and map_data[x][y].value
      == 3 and random.random() <= pburn(map_data[x+1][y+1],
      5.49779)):
3         map_data[x+1][y+1].value = 3
4         vx1.append(x+1)
5         vy1.append(y+1)
6         efWind(map_data, x+1, y+1)
7
8     if (map_data[x+1][y-1].value == 2 and map_data[x][y].value
      == 3 and random.random() <= pburn(map_data[x+1][y-1],
      3.92699)):
9         map_data[x+1][y-1].value = 3
10        vx1.append(x+1)
11        vy1.append(y-1)
12        efWind(map_data, x+1, y-1)
13
14    if (map_data[x-1][y+1].value == 2 and map_data[x][y].value
      == 3 and random.random() <= pburn(map_data[x-1][y+1],
      0.785398)):
15        map_data[x-1][y+1].value = 3
16        vx1.append(x-1)
17        vy1.append(y+1)
18        efWind(map_data, x-1, y+1)
19
20    if (map_data[x-1][y-1].value == 2 and map_data[x][y].value
      == 3 and random.random() <= pburn(map_data[x-1][y-1],
      2.35619)):
21        map_data[x-1][y-1].value = 3
22        vx1.append(x-1)
23        vy1.append(y-1)
24        efWind(map_data, x-1, y-1)
25
26    if (map_data[x][y].value == 3):
27        map_data[x][y].value = 4
28
29
30    #Escreve os dados do incendio em um arquivo .txt
31    with open("wind9", 'a') as file:
32        file.write(f"{cont}\t{len(vx1)}\n")
33    cont+=1
34
35    draw_burn(map_data, drones)
36
37    #Atualiza o mapa a cada 60 segundos
38    if cv2.waitKey(60000) & 0xFF==ord('d'):
39        break

```

Algoritmo 18: Código principal da simulação do incêndio

```

1
2     #Se o incêndio chegou a um tempo pré-determinado de
3     iterações (cont_int),
4     #os drones são liberados para o combate
5     cont_int+=1
6     if cont_state == 0:
7         if(cont_int == 2):
8
9             #inicializa os drones
10            for i in range(len(vx1)):
11                x = vx1[i]
12                y = vy1[i]
13
14                if(map_data[x][y].value == 3):
15                    cont_drone+=1
16
17            drones += [Drone(Node(15, 15)) for _ in range(
18                cont_drone)]
19
20            draw_burn(map_data, drones)
21
22            #Para cada drone, se o mesmo ainda não foi designado
23            # para uma nova célula
24            # o mesmo recebe um alvo
25            for drone in drones:
26                if drone.state == 0:
27                    for i in range(len(vx1)):
28                        x = vx1[i]
29                        y = vy1[i]
30                        print(f"cell1: {map_data[x][y].value},
31                            cell2: {map_data[x][y+1].value}")
32                        #Testa se a célula designada como alvo está
33                        # queimando e se ainda não possui drone
34                        if((map_data[x][y].value == 3 and map_data[
35                            x][y].value2 != 1)):
36                            x1 = x * 8
37                            y1 = y * 8
38                            drone.cell_x = x
39                            drone.cell_y = y
40                            map_data[x][y].value2 = 1
41                            cx = x1 + 8 // 2
42                            cy = y1 + 8 // 2
43
44                            drone.state = 1
45                            break

```

Algoritmo 19: Código principal da simulação do incêndio

```

1
2         #A cada itera o , testa se o movimento do
3         drone n o passa da dist ncia max
4         #que ele consegue atingir em 1min
5         while cont_pixel < 1400:
6             drone.target_node = Node(cy,cx)
7             #Obt m o pr ximo passo de movimento
8             path = astar(drone.current_node, drone.
9                 target_node)
10
11             if path is None:
12                 print("Erro.")
13                 break
14
15             #Move o ponto atual para o pr ximo n no
16             caminho
17             drone.current_node = Node(path[1][0], path
18                 [1][1])
19             cont_pixel+=1
20
21             #Atualiza o mapa
22             draw_burn(map_data, drones)
23             if drone.current_node == drone.target_node:
24                 cont_state = 1
25                 break
26
27     for drone in drones:
28         if drone.state == 0:
29             for i in range(len(vx1)):
30                 x = vx1[i]
31                 y = vy1[i]
32                 if((map_data[x][y].value == 3 and map_data[
33                     x-1][y].value == 4 and
34                     map_data[x][y].value2 != 1) or (
35                         map_data[x][y].value == 3 and
36                         map_data[x+1][y].value == 4 and
37                         map_data[x][y].value2 != 1)):
38                     x1 = x * 8
39                     y1 = y * 8
40                     drone.cell_x = x
41                     drone.cell_y = y
42                     map_data[x][y].value2 = 1
43                     cx = x1 + 8 // 2
44                     cy = y1 + 8 // 2
45                     drone.state = 1
46                     break

```

Algoritmo 20: Código principal da simulação do incêndio

```

1         while cont_pixel < 1400:
2             drone.target_node = Node(cy,cx)
3             path = astar(drone.current_node, drone.
4                 target_node)
5             if path is None:
6                 print("Erro.")
7                 break
8
9             drone.current_node = Node(path[1][0], path
10                [1][1])
11            cont_pixel+=1
12            draw_burn(map_data, drones)
13            if drone.current_node == drone.target_node:
14                cont_state = 1
15                break
16
17        for drone in drones:
18            if drone.state == 0:
19                for i in range(len(vx1)):
20                    x = vx1[len(vx1)-1-i]
21                    y = vy1[len(vx1)-1-i]
22                    if((map_data[x][y].value == 3 and map_data[
23                        x][y].value2 != 1)):
24                        x1 = x * 8
25                        y1 = y * 8
26                        drone.cell_x = x
27                        drone.cell_y = y
28                        map_data[x][y].value2 = 1
29                        cx = x1 + 8 // 2
30                        cy = y1 + 8 // 2
31                        drone.state = 1
32                        break
33
34        while cont_pixel < 1400:
35            drone.target_node = Node(cy,cx)
36            path = astar(drone.current_node, drone.
37                target_node)
38            if path is None:
39                print("Erro.")
40                break
41
42            drone.current_node = Node(path[1][0], path
43                [1][1])
44            cont_pixel+=1
45            draw_burn(map_data, drones)
46            if drone.current_node == drone.target_node:
47                cont_state = 1
48                break

```

Algoritmo 21: Código principal da simulação do incêndio

```

1     for drone in drones:
2         if drone.state == 0:
3             for i in range(len(vx1)):
4                 x = vx1[len(vx1)-1-i]
5                 y = vy1[len(vx1)-1-i]
6
7                 if((map_data[x][y-1].value == 4 and
8                     map_data[x][y].value == 3 and
9                     map_data[x][y+1].value == 3 and
10                    map_data[x][y+2].value == 2 and
11                    map_data[x][y].value2 != 1)):
12                        x1 = x * 8
13                        y1 = y * 8
14                        drone.cell_x = x
15                        drone.cell_y = y
16                        map_data[x][y].value2 = 1
17                        cx = x1 + 8 // 2
18                        cy = y1 + 8 // 2
19
20                        drone.state = 1
21                        break
22
23                    while cont_pixel < 1400:
24                        drone.target_node = Node(cy,cx)
25                        path = astar(drone.current_node, drone.
26                            target_node)
27
28                        if path is None:
29                            print("Erro.")
30                            break
31
32                        drone.current_node = Node(path[1][0], path
33                            [1][1])
34                        cont_pixel+=1
35                        draw_burn(map_data, drones)
36                        if drone.current_node == drone.target_node:
37                            cont_state = 1
38                            break
39
40                for drone in drones:
41                    drone.state = 0
42                cont_pixel = 0

```

Algoritmo 22: Código principal da simulação do incêndio

```

1   #Define um novo alvo, ou celula, para cada drone
2   if cont_state == 1:
3       for drone in drones:
4           #Testa todas as celulas vizinhas da atual
5           for dx in [0, -1, 1]:
6               for dy in [1, -1, 0]:
7                   if((map_data[drone.cell_x+dx][drone.cell_y+dy].
8                       value == 3 or map_data[drone.cell_x+dx][
9                           drone.cell_y+dy].value == 4)
10                      and map_data[drone.cell_x+dx][drone.cell_y+
11                          dy].value != 1):
12
13                       x1 = drone.cell_x * 8
14                       y1 = drone.cell_y * 8
15
16                       cx = x1 + 8 // 2
17                       cy = y1 + 8 // 2
18                       drone.current_node = Node(cy,cx)
19                       map_data[drone.cell_x][drone.cell_y].value
20                           = 1
21
22                       x1 = (drone.cell_x+dx) * 8
23                       y1 = (drone.cell_y+dy) * 8
24
25                       x_prox = drone.cell_x+dx
26                       y_prox = drone.cell_y+dy
27
28                       map_data[x_prox][y_prox].value = 1
29
30                       cx = x1 + 8 // 2
31                       cy = y1 + 8 // 2
32                       drone.target_node = Node(cy,cx)
33                       drone.state = 1
34                       break
35
36   if drone.state == 1:
37       drone.state = 0
38       break

```

Algoritmo 23: Código principal da simulação do incêndio

```
1     print(drone)
2     print(f"x1: {x_prox}, y1: {y_prox}")
3     #Executa, novamente, o A* para cada drone
4     while drone.current_node != drone.target_node:
5         path = astar(drone.current_node, drone.
6                     target_node)
7
8         if path is None:
9             print("Erro.")
10            break
11
12            drone.current_node = Node(path[1][0], path
13                                     [1][1])
14
15            map_data[x_atual][y_atual].value = 1
16            draw_burn(map_data, drones)
17
18            if drone.current_node == drone.target_node:
19                map_data[drone.cell_x][drone.cell_y].value
20                    = 1
21                drone.cell_x = x_prox
22                drone.cell_y = y_prox
23                break
```

Fonte: Elaborado pelo autor.

Código presente em <https://github.com/Ceand1/Simulacoes-Exame-de-VANTs>.