

**FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO
“ADIB MOISÉS DIB”**

GUSTAVO SANTANA SAMPAIO
ISABELI GONÇALVES SILVA
JULIANO DE SOUZA SANTOS
WILLIAM GOMES DOS SANTOS

**SISTEMA PARA AUXÍLIO NO RESGATE EM ÁREA AFETADA POR
DESLIZAMENTO**

São Bernardo do Campo - SP
Novembro/2022

**GUSTAVO SANTANA SAMPAIO
ISABELI GONÇALVES SILVA
JULIANO DE SOUZA SANTOS
WILLIAM GOMES DOS SANTOS**

**SISTEMA PARA AUXÍLIO NO RESGATE EM ÁREA AFETADA POR
DESLIZAMENTO**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo (a) em Automação Industrial.

Orientador: Professor Dr. Claudio Rodrigo Torres

Trabalho de Conclusão de Curso apresentado e aprovado em:_____/_____/2022

Banca Examinadora:

Professor Dr. Claudio Rodrigo Torres, FATEC SBC - Orientador

Avaliador

Avaliador

Dedicamos esse trabalho aos nossos pais e a todos que nos ajudaram durante essa caminhada em busca de conhecimento. Agradecemos a FATEC por fazer a diferença em nossas vidas e proporcionar uma evolução pessoal e profissional.

Agradecemos ao Prof. Dr. Cláudio Rodrigo Torres pela orientação durante a elaboração deste trabalho.

“A verdadeira medida de um homem não é como ele se comporta em momentos de conforto e conveniência, mas como ele se mantém em tempos de controvérsia e desafio.”

MARTIN LUTHER KING JR.

RESUMO

Nos últimos anos a sociedade presenciou diversos acidentes causados por deslizamento de terra, que tiveram como consequência vítimas fatais. Pensando nisso esse TCC tem como objetivo utilizar a tecnologia aplicada a um sistema de gerenciamento, composto idealmente por um drone e um aplicativo para movimentação e gerenciamento de alarmes, que visa auxiliar os bombeiros no resgate e identificação das possíveis vítimas e objetos. As linhas de pesquisa utilizadas para a fundamentação teórica foram voltadas para causas, consequências e exemplos de deslizamentos, inteligência artificial, drones, sensoriamento e termologia. Durante o desenvolvimento, por questões de viabilidade econômica, houve a implementação de um robô móvel com sensores de distância e temperatura. O funcionamento do protótipo consiste no drone mapeando a área, e através do treinamento de imagem reconhecendo se existem pessoas sobre o solo, logo em seguida o carrinho, por meio do aplicativo, se movimenta sobre o local alertando sobre a distância dos destroços e se ocorre alteração de temperatura no solo, em caso de soterramento das vítimas. Por fim o projeto obteve êxito no reconhecimento de imagem e obtenção de temperatura do solo, no entanto foi necessário adaptar a leitura da imagem a uma webcam devido a capacidade de processamento do computador. As considerações finais apresentam melhorias futuras para o projeto.

Palavras-chave: Deslizamento. Drone. Robô. Sensor. YOLO.

ABSTRACT

In recent years, society has witnessed several accidents caused by landslides, which resulted in fatalities. With that in mind, this TCC presented to the faculty of Industrial Automation aims to use the technology applied to a management system, ideally composed of a drone and an application for movement and alarm management, which aims to assist firefighters in the rescue and identification of possible victims and objects. The lines of research used for the theoretical foundation were focused on causes, consequences and examples of landslides, artificial intelligence, drones, sensing and thermology. During the development, for reasons of economic viability, there was the implementation of a mobile robot with distance and temperature sensors. The functioning of the prototype consists of the drone mapping the area, and through image training recognizing if there are people on the ground, then the cart, through the application, moves over the place warning about the distance of the wreckage and if occurs alteration of temperature in the soil, in case of burial of the victims. Finally, the project was successful in image recognition and soil temperature acquisition, however it was necessary to adapt the image reading to a webcam due to the computer's processing capacity. The final considerations present future improvements for the project.

Keywords: Drone. Landslide. Robot. Sensor. YOLO.

LISTA DE FIGURAS

Figura 1.1 - Suscetibilidade de deslizamento do Brasil	14
Figura 1.2 - Drone Militar.....	17
Figura 1.3 - Drone utilizado para a agricultura	17
Figura 1.4 - Drone com câmera.....	18
Figura 1.5 - Áreas relacionadas com a Inteligência artificial	18
Figura 1.6 - Linha do tempo da inteligência artificial	20
Figura 1.7 - Aprendizagem de máquina	22
Figura 1.8 - Formas de energia de um sensor.	26
Figura 1.9 - Transmissor e receptor ultrassônico.	27
Figura 1.10 - Fotografia através de uma câmera térmica.....	29
Figura 1.11 - Arduino UNO.....	30
Figura 1.12 - Lei de Stefan-Boltzmann.....	32
Figura 2.1 - Desenho do robô móvel	34
Figura 2.2 - Drone utilizado	34
Figura 2.3 - Fluxograma de movimentação.....	35
Figura 3.1 - Projeto Final.....	39
Figura 3.2 - Drone com o suporte para o espelho.....	40
Figura 3.3 - Robô móvel desmontado.	40
Figura 3.4 - Placas de montagem do protótipo.....	41
Figura 3.5 - Ligações eletrônicas.	41
Figura 3.6 - Módulo bluetooth ligado ao Arduino.....	42
Figura 3.7- Sensores utilizados no projeto	43
Figura 3.8 - Sensor ultrassônico posicionado na base.....	43
Figura 3.9 - Sensor laser de temperatura.....	44
Figura 3.10 - Módulo bluetooth.....	44
Figura 3.11 - Tela de <i>Login</i>	46
Figura 3.12 - Diagrama de blocos, tela login.....	46
Figura 3.13 - Conexão do celular com o módulo HC-05	47
Figura 3.14 - Botão DESCONECTAR	47
Figura 3.15 - Identificação da movimentação.....	48
Figura 3.16 - Diagrama das teclas de movimentação	48
Figura 3.17 - Tela de configuração.....	49

Figura 3.18 - Programação da tela de configuração	49
Figura 3.19 - Tela de alteração de senha.....	50
Figura 3.20 - Diagrama da tela de alteração de senha.	51
Figura 3.21 - Tela de temperatura.....	52
Figura 3.22 - Programação da tela de temperatura.....	52
Figura 3.23 - Tela de movimentação	53
Figura 3.24 - Imagem do aplicativo utilizado	53
Figura 3.25 - Projeto Idealizado no <i>software SolidWorks</i>	55

SUMÁRIO

INTRODUÇÃO	12
1 FUNDAMENTAÇÃO TEÓRICA	14
1.1 Índice de deslizamentos e chuva.....	14
1.2 Acidentes	15
1.2.1 Brumadinho e Petrópolis	15
1.3 Drone.....	16
1.3.1 VANT militar.....	16
1.3.2 Drone agrícola	17
1.3.3 Drone para fotos e vídeos	17
1.4 Inteligência Artificial	18
1.4.1 <i>Machine Learning</i>	21
1.4.2 <i>Python e Open CV</i>	23
1.4.3 YOLO.....	25
1.5 Sensores	26
1.5.1 Sensor ultrassônico	26
1.5.2 Sensor infravermelho.....	27
1.6 Microcontrolador e Arduino UNO.....	29
1.7 Termologia.....	31
2 METODOLOGIA.....	33
2.1 O que é metodologia	33
2.2 O tema-problemática, justificativa e diagrama de funcionamento.	33
2.3 Etapas teóricas e práticas para o desenvolvimento do projeto	36
3 DESENVOLVIMENTO DO PROJETO	39
3.1 Montagem do espelho do drone e do robô móvel	39
3.2 Reconhecimento de imagem	53
3.3 Melhoria futura	55
CONSIDERAÇÕES FINAIS	56
REFERÊNCIAS.....	58
APÊNDICE A – PROGRAMAÇÃO DO ROBÔ MÓVEL	60
APÊNDICE B – PROGRAMAÇÃO Reconhecimento de imagem.....	63

INTRODUÇÃO

No decorrer dos anos a sociedade enfrentou diversos casos de deslizamento de terra e acidentes relacionados ao rompimento de barragens. Esses acontecimentos afetam diretamente a sociedade tendo como consequência muitas pessoas desabrigadas e diversas vítimas fatais

Apesar da movimentação de terra fazer parte da dinâmica de transformação da natureza, outros fatores influenciam nesse desastre como a construção de casas em lugares impróprios que ocasionam o aumento do peso da massa sedimentada já existente, suprimindo a vegetação natural do local causando a vulnerabilidade do solo e aumentando os casos de deslizamento.

De acordo com Marcelino (2007), as geotecnologias são as principais ferramentas para o gerenciamento de desastres naturais, representadas especialmente pelos Sistemas de Informação Geográfica (SIG) que é formado por um conjunto de ferramentas responsáveis por armazenar, recuperar, transformar informações espaciais e pelo sensoriamento remoto que são as informações coletadas de sensores e mecanismos digitais de entrada de dados. A SIG permite que esses dados espaciais georreferenciados possam ser aproveitados permitindo consulta a um banco de dados.

Segundo Aronoff (2007), dividiu o sistema SIG em 4 etapas: a primeira etapa trata-se do armazenamento provenientes de sistemas e sensoriamento remoto, dados cartográficos, imagens de satélite, modelos numéricos etc. A segunda etapa tem como função analisar as informações através de um algoritmo de manipulação para que possa se realizar um mapeamento das informações coletadas. A terceira etapa plota o conteúdo de dados já tratados. A quarta etapa armazena essas informações no banco de dados.

Visto que o corpo de bombeiros é o agente principal para ajudar em casos de deslizamento, faz-se necessário a criação de um projeto que auxilie as equipes de resgate.

Pensando na irregularidade do terreno coberto de destroços e terra, o maior obstáculo enfrentado está na localização dos corpos devido à dificuldade no reconhecimento da área, o projeto visa utilizar a tecnologia presente na inteligência artificial para melhorar a visualização do ambiente afetado, aumentando a velocidade de trabalho dos bombeiros e diminuindo o número de óbitos.

O trabalho que se intitula, Sistema para Auxílio no Resgate em Área Afetada por Deslizamento, tem como objetivo disponibilizar aos bombeiros um sistema composto por um drone que sobrevoa o local dos destroços e detecta pessoas por meio do reconhecimento de imagem e alteração na temperatura do solo.

As informações são enviadas por meio de um aplicativo, que possibilita o socorrista obter dados dos sensores de presença e temperatura auxiliando no resgate das vítimas.

O trabalho é dividido em três capítulos:

Capítulo 01 – Fundamentação teórica: encontram-se as pesquisas e análises que dão alicerce para a construção e desenvolvimento do projeto;

Capítulo 02 – Metodologia: são descritos métodos, técnicas e os procedimentos seguidos para chegar ao resultado esperado e no objetivo previamente traçado;

Capítulo 03 – Desenvolvimento do projeto: é demonstrado o passo a passo da construção do protótipo.

Considerações finais: Abordado os objetivos propostos e alcançados, melhorias futuras

1 FUNDAMENTAÇÃO TEÓRICA

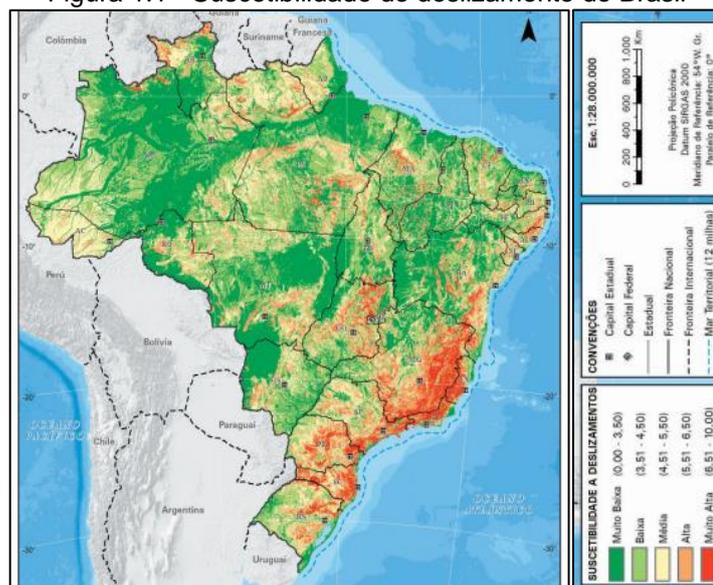
Neste capítulo são abordadas teorias de autores renomados que dão sustentação ao desenvolvimento e construção do projeto Sistema de reconhecimento de área afetada por deslizamentos.

1.1 Índice de deslizamentos e chuva

Segundo o *Global Facility For Disaster Reduction and Recovery (GRDRR)* (2008) Deslizamento é uma movimentação do solo em declive que acontece na ruptura de uma superfície, sendo que a maior parte do solo se move como uma massa coerente, com deformação interna.

De acordo com Azevedo (2019) as regiões Sul e Sudeste concentram as maiores áreas de suscetibilidade a deslizamentos por estar em regiões planálticas ou serranas. Ambas as regiões foram edificadas em terrenos de grande mobilidade e fragilidade do solo. A Figura 1.1 mostra detalhes suscetibilidade a deslizamentos do Brasil.

Figura 1.1 - Suscetibilidade de deslizamento do Brasil



Fonte: www.biblioteca.ibge.gov.br, 2019

Em contrapartida as regiões Norte e Centro-Oeste concentram as áreas de menor suscetibilidade a deslizamentos, pois se trata de um local com estabilidade do

solo. É possível notar que as áreas com suscetibilidade muito baixa correspondem a mais de 60% do território brasileiro.

1.2 Acidentes

Ao longo dos anos o Brasil passou por diversas tragédias relacionadas a deslizamento de terra, causando diversas fatalidades, exemplo disso são as tragédias de Brumadinho e Petrópolis.

1.2.1 Brumadinho e Petrópolis

De acordo com o ANPEGE (Associação Nacional de Pós-Graduação e Pesquisa em Geografia) deslizamentos são o auge dos desastres que partem do licenciamento ambiental e a gestão ambiental ineficiente da reparação de danos. A ruptura da barragem de Brumadinho (MG), 2019, custou a vida de 270 pessoas (aproximadamente 258 trabalhadores). A lama tóxica se estendeu por mais de 300 quilômetros no rio Paraopeba.

Segundo a matéria redigida por Daré (2022) publicada no jornal da UNICAMP (Universidade Estadual de Campinas) a chuva em excesso não pode ser considerada a única causa do ocorrido. As chuvas intensas que ocorreram no mês de janeiro e fevereiro já encontravam um terreno muito úmido por precipitações anteriores, analisando no quesito técnico o solo encharcado oferece riscos para toda a população que vivem no contorno da área propícia a um deslizamento.

É notório que com a evolução da tecnologia os acidentes relacionados a chuva diminuirão por conta da análise do solo e de sua movimentação ao passar do tempo contendo sensores para detecção das mudanças indesejáveis.

De acordo com Daré é nítido que faltou políticas públicas efetivas em Petrópolis e que as pessoas que vivem no local estão por necessidade e não por opção o que aumenta ainda mais o número de óbitos.

1.3 Drone

De acordo com Soares (2017), “veículo aéreo não tripulado (VANT) ou Aeronave Remotamente Pilotada (ARP) é todo e qualquer tipo de aeronave que não necessita de pilotos embarcados para ser guiada”. Os Vant surgiram inicialmente para uso militar, cujo objetivo era realizar as missões militares e preservar a vida dos soldados, com a evolução da tecnologia e redução dos custos dos dispositivos, passaram a ficar mais populares e ganharam uma enorme gama de modelos e aplicações, o modelo mais comum é o quadricóptero conhecido como drone e é muito usado principalmente para a captação de fotos e vídeos.

A utilização do espaço aéreo brasileiro é regulamentada pela Agência Nacional de Aviação Civil (ANAC) através do Regulamento Brasileiro de Aviação Civil nº 94/2017, juntamente com as normas da Agência Nacional de Telecomunicações (Anatel) e o Departamento de Controle do Espaço Aéreo (Decea), criando assim as regras para a utilização das aeronaves remotamente pilotadas (RPA).

A operação dos drones seguem regras básicas de segurança definidas pela ANAC, sendo classificados de acordo com o seu peso máximo de decolagem, as aeronaves que possuem peso acima de 250g precisam ser cadastradas junto à ANAC, e enquadram-se nas seguintes classes:

Classe 1: peso máximo de decolagem acima de 150 kg;

Classe 2: peso máximo de decolagem acima de 25 kg e abaixo de 150 kg;

Classe 3: peso máximo de decolagem acima de 250 g e abaixo de 25 kg.

1.3.1 VANT militar

Os veículos aéreos não tripulados militares foram utilizados no início dos anos 80, a Força Aérea Israelense contra a Força Aérea da Síria, onde as aeronaves eram enviadas para realizar as missões, e os soldados eram poupados. A Figura 1.2 ilustra um drone militar.

Figura 1.2 - Drone Militar



Fonte: www.defesanet.com.br, 2022.

1.3.2 Drone agrícola

Na agricultura os drones ganharam destaque, sendo utilizados para a pulverização de defensivos agrícolas, para mapeamento geográfico e monitoramento de plantações e rebanhos. A Figura 1.3 ilustra um drone utilizado para a agricultura.

Figura 1.3 - Drone utilizado para a agricultura



Fonte: lorddrone.com.br, 2022.

1.3.3 Drone para fotos e vídeos

O mercado de conteúdo audiovisual tem se beneficiado da evolução da tecnologia e dos drones, devido ao seu tamanho compacto, o que permite voo em locais fechados ou restritos e risco de acidente reduzidos, além da excelente qualidade de fotos e vídeos que suas câmeras proporcionam, aliados ao excelente custo-benefício. A Figura 1.4 ilustra um drone com câmera para captação de fotos e vídeos.

Figura 1.4 - Drone com câmera

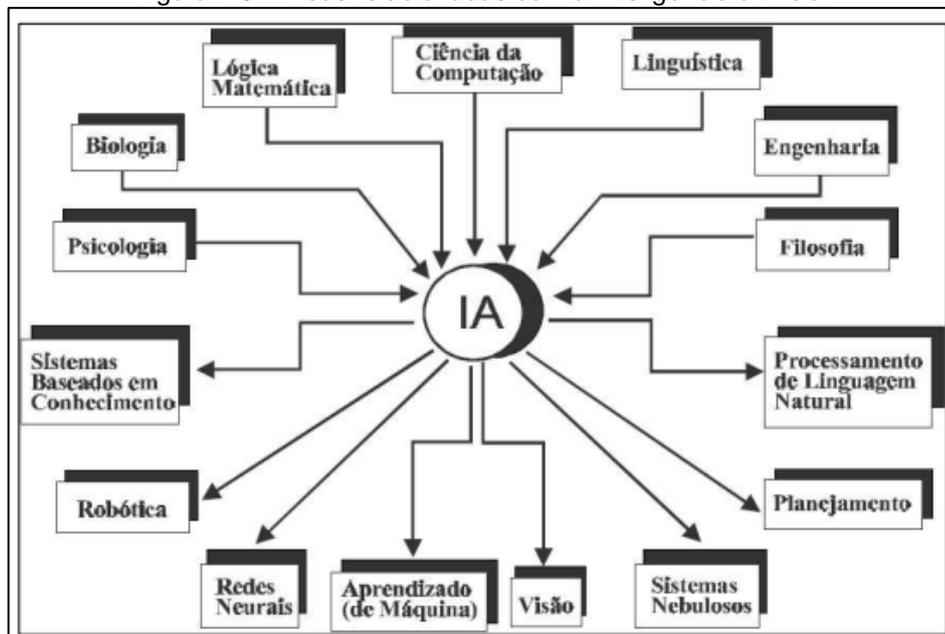


Fonte: www.cined.com, 2022

1.4 Inteligência Artificial

Segundo Honório (2010) a inteligência artificial é a característica atribuída a uma máquina ou equipamento que possibilita por meio da utilização de um algoritmo o funcionamento semelhante ao pensamento humano, tomando decisões sem interferência. Essa capacidade está ligada a diversos campos da ciência podendo atuar desde a criação de modelos matemáticos até auxiliar na biologia e medicina. Abaixo a Figura 1.5 ilustra as áreas que possuem relação com a inteligência artificial.

Figura 1.5 - Áreas relacionadas com a Inteligência artificial



Fonte: HONÓRIO, 2010.

Sendo assim, com a exploração da sua capacidade, ao longo dos anos podemos ver que a inteligência artificial possui diversas aplicações, como por exemplo:

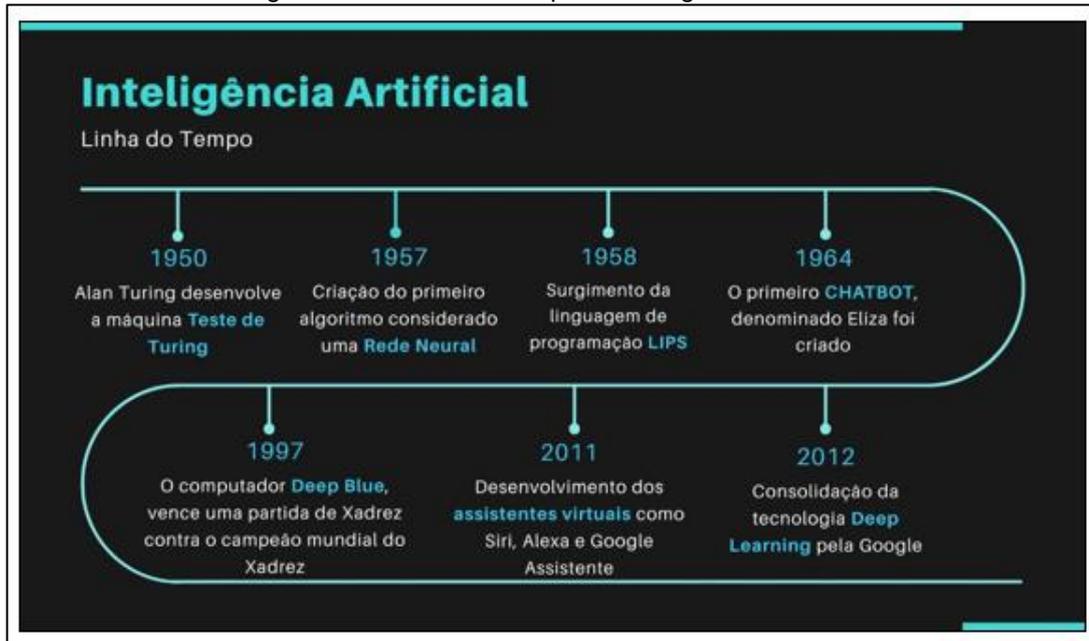
- Reconhecimento de imagem;
- Distribuição de conteúdo nas redes sociais;
- Detecção e classificação de objetos;
- Manutenção preditiva.

A IA (Inteligência Artificial) surgiu a partir da necessidade do homem de desenvolver tecnologias que simulassem as ações humanas, segundo Barbosa (2020) o período que marcou o início da inteligência artificial foi a Segunda Guerra Mundial quando os países começaram a estudar como elevar suas tecnologias e analisar o cérebro humano por meio da utilização de judeus presos em campos de concentração.

Já em 1950 o matemático Alan Turing desenvolveu uma máquina denominada Teste de Turing que simulava a comunicação escrita de um humano sem que o receptor percebesse que era uma máquina. No mesmo ano Turing publicou o Artigo *Computing Machinery and Intelligence*, que foi considerado um marco para a Inteligência Artificial e lhe conferiu o título de pai da computação.

Ainda segundo o artigo desenvolvido por Barbosa (2020) com o início das experiências a comunidade científica e órgãos privados governamentais começaram a investir na área, inclusive a Agência de pesquisa de projetos avançados (ARPA) mesma instituição que desenvolveu a internet. Essa tecnologia continuou avançando no decorrer do tempo, a Figura 1.6 ilustra a linha do tempo da inteligência artificial.

Figura 1.6 - Linha do tempo da inteligência artificial



Fonte: BARBOSA, 2020 (modificado).

Segundo a Fundação Instituto de Administração (2018) a inteligência artificial pode ser dividida em IA simbólica e IA conexionista. A IA simbólica refere-se a um programa com regras estabelecidas previamente que são interpretadas por meio de símbolos como letras e números, e não alteram a sua programação no decorrer do tempo.

Já a IA conexionista foi desenvolvida inspirada no funcionamento dos neurônios, sendo as informações disponibilizadas por meio de uma rede neural e os dados tratados através de um aprendizado profundo conhecido como *Deep Learning* (aprendizado profundo), esses dados são fornecidos a uma máquina que os interpreta e treina semelhante ao cérebro, esse treino é atribuído a uma tecnologia conhecida como *Machine Learning* (aprendizado de máquina).

Tecnologias que implementam *deep learning* e *machine learning* são classificadas como inteligência artificial forte, uma vez que possuem a capacidade de emular o raciocínio humano. O *machine learning* é amplamente utilizado para o treinamento e para o desenvolvimento do trabalho, esse termo será abordado mais profundamente a seguir.

1.4.1 *Machine Learning*

O termo *machine learning* foi criado por Arthur Lee Semuels em 1959, ele era engenheiro do MIT¹ e pesquisador da IBM. O aprendizado de máquina é uma das vertentes da inteligência artificial, que segundo Barbosa (2020) significa sistema que permite as máquinas a aprenderem determinada função sem um programa prévio, utilizando apenas dados que são inseridos no algoritmo para que a máquina aprenda a executar a tarefa automaticamente.

O seu funcionamento segundo a Fundação Instituto de Administração (2018) tem como base os algoritmos, que são um conjunto de regras e operações, que atrelado a um grande volume de dados (*Big Data*) geram um modelo automático.

Para que esse conjunto possa aprender, eles são submetidos a métodos de aprendizagem, que podem ser subdivididos em três metodologias: não supervisionada, supervisionada e aprendizagem por reforço.

O método supervisionado ocorre através da apresentação ao algoritmo de dados que contêm a resposta correta, recebendo padrões já identificados, por meio da ação humana, porém esse modelo também precisa ser exposto a dados não rotulados, ou seja, que não foram treinados anteriormente, para evitar que ele memorize os dados e não aprenda com eles, o que causaria o baixo desempenho quando apresentado a novas informações.

Já no modelo não supervisionado a própria máquina encontra os padrões desejados, não recebendo nenhum dado pré-organizado, essa abordagem é normalmente utilizada quando a quantidade de informações é excessiva e necessita ser aprimorada, essa metodologia é considerada mais avançada e possui maior autonomia do sistema.

O procedimento aprendizado por reforço, o algoritmo recebe um retorno do usuário toda vez que um resultado é apresentado, dessa forma o sistema aprende por

¹ **MIT** é a abreviação utilizada para denominar o Instituto de Tecnologia de Massachusetts (Massachusetts Institute of Technology).

meio dos erros e acertos e não por informações pré-estabelecidas. A Figura 1.7 demonstra essas formas de aprendizagem.

Figura 1.7 - Aprendizagem de máquina



Fonte: www.redesdesaude.com.br, 2019.

Existem diferentes classificações de algoritmos e eles devem ser adotados de acordo com o resultado desejado. Segundo Mueller e Massaron (2021) algoritmos são conjuntos de dados que devem ser manipulados de forma simples ou complexa, por meio de uma linguagem de programação.

Ainda segundo Mueller e Massaron (2021) o *Machine Learning* segue um ciclo contínuo, devendo passar pelas seguintes etapas:

1ª etapa: Identificar os dados, localizar que fonte será utilizada;

2ª etapa: Preparar os dados, certificar-se de que os dados estão organizados e protegidos evitando falhas por imprecisão;

3ª etapa: Selecionar o algoritmo utilizado na aprendizagem de máquina, existem diversos algoritmos de aprendizado, cabe nessa etapa decidir qual será o mais útil visando a aplicação;

4ª etapa: Treinar, o algoritmo deve ser treinado para criação do modelo, nessa etapa é aplicado os tipos de aprendizagem que foram listados;

5ª etapa: Avaliar se o modelo gerado está de acordo com a necessidade;

6ª etapa: Avaliar as previsões, avaliar o desempenho do modelo, e as informações obtidas nesse processo são realimentadas no ciclo, ajudando a melhorar a precisão do aprendizado.

Essas etapas servem como referência para o desenvolvimento de programas que visam realizar o treinamento de imagem e podem ser realizados através de diversas plataformas e linguagens de programação.

Para o desenvolvimento desse projeto utilizamos a linguagem de programação *Python* com a biblioteca *Open CV* e o algoritmo YOLO² que irá servir como base para a identificação da imagem.

1.4.2 *Python* e *Open CV*

A linguagem de programação *Python* surgiu na década de 90 e foi desenvolvida por Guido Van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação, segundo (BARANAUSKAS; MONARD, 2000). Essa linguagem vem se destacando atualmente pela sua simplicidade e capacidade de desenvolver programas complexos apenas com a utilização correta das bibliotecas. Sendo utilizada por grandes empresas.

2 **YOLO** é uma abreviação do termo em inglês *you only look once* (Você só olha uma vez).

Esse sistema é classificado como *High Level Language* por ser dinâmico, modular e orientada a objetos, possuindo muitas bibliotecas que permitem a sua utilização em áreas de análise de dados e *machine learning*.

Algumas das suas aplicações são:

- *Scripting* e Automação;
- Desenvolvimento Web;
- *Big Data*;
- Computação Gráfica;
- Inteligência Artificial.

O *Python* tem entre suas bibliotecas uma que além de oferecer outras funções possui destaque na utilização voltada a treinamento de imagens, essa biblioteca é conhecida *Open CV* e foi explorada mais profundamente para auxiliar no desenvolvimento do nosso projeto.

A biblioteca *Open CV* é utilizada em visão computacional e processamento de imagem, sendo um projeto *open source*, criado pela Intel com o objetivo de deixar a visão computacional mais acessível. Esse *software* possui mais de 500 funções sendo utilizado principalmente em análise em imagens e vídeos, como detecção e reconhecimento facial, edição de fotos e vídeos, detecção e análise de textos etc.

Essa biblioteca utiliza de algoritmos para manipular as imagens tendo melhor desempenho em detecção de bordas, a borda é caracterizada por uma região que contém um alto gradiente. Os autores destacam que a detecção de bordas é uma das operações fundamentais quando realizamos o processamento de imagens. Isso ajuda a reduzir a quantidade de dados (*pixels*) para processamento e mantém o aspecto estrutural da imagem, além de extrair características dela.

Para o desenvolvimento do projeto o algoritmo de detecção que auxilia em conjunto com a biblioteca *Open CV* na detecção da imagem é YOLOV3 que é abordado a seguir.

1.4.3 YOLO

O YOLO é uma rede neural convolucional criada com a intenção de classificar objetos e reconhecer sua localização, esse algoritmo foi desenvolvido por Joseph Redmon e Ali Farhadi e teve sua primeira versão lançada em 2015.

Segundo Redmon et al. (2016), o YOLO, diferente dos algoritmos criados anteriormente predefine caixas delimitadoras na imagem e suas classes ao mesmo tempo detectando os objetos em uma única passagem (*single pass*) tornando a leitura mais rápida, além de reconhecer o contexto da imagem, que diminui a falsa detecção.

Para a sua execução o YOLO utiliza um modelo de aprendizagem profunda conhecido como *Darknet*, esse modelo assim como o YOLO possui código aberto permitindo com que qualquer pessoa treine a sua própria rede neural, classificando as imagens de acordo com a aplicação.

O seu funcionamento consiste em dividir a imagem em pequenas células e logo em seguida avaliar a possibilidade de conter um objeto dentro de cada uma delas, caso o objeto esteja no centro da célula a rede neural faz a classificação dessa célula, baseado no grau de confiança inserido no código, como um possível objeto, caso não seja identificado nenhum item, o código elimina as células com baixa probabilidade aplicando o algoritmo de identificação apenas nas que possuem alto grau de confiança.

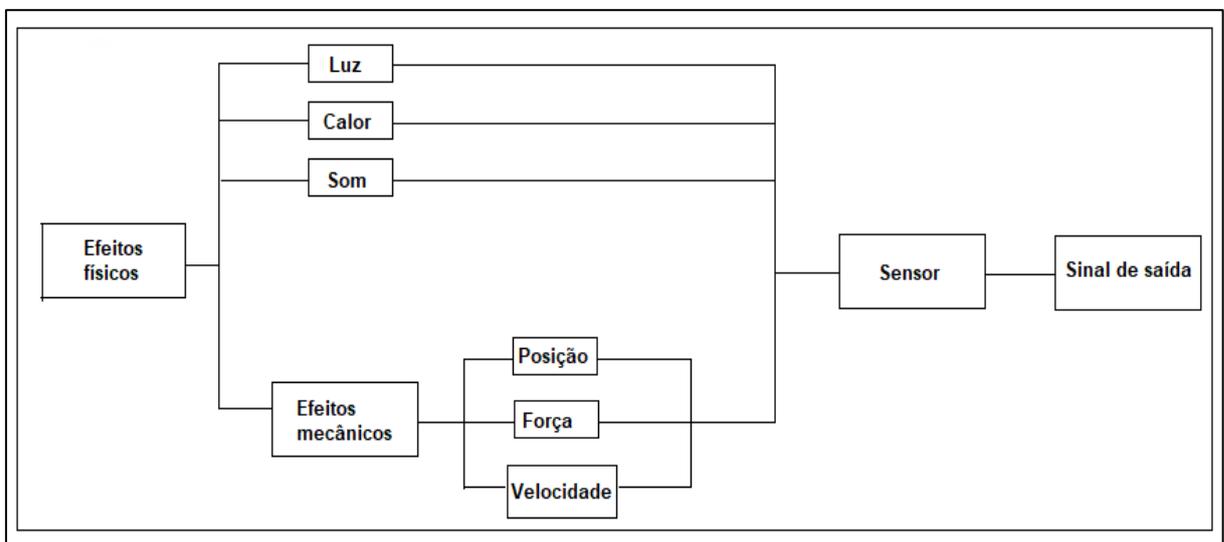
Após esse processo de separação o algoritmo *non-max-suppression* é aplicado definindo qual a classe do objeto e suas dimensões, finalizando assim o processo de reconhecimento de imagem.

Essa rede neural foi sendo atualizada com o tempo por seus criadores otimizando não só a velocidade como a precisão no reconhecimento do objeto, para o presente projeto utilizaremos a versão 3.

1.5 Sensores

Sistemas automatizados necessitam de sensores para se obter informações do processo. Segundo Thomazini e Albuquerque (2007) sensores é o termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica e cinética, relacionando informações sobre uma grandeza que precisa ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição etc. A Figura 1.8 ilustra as formas de energia de um sensor:

Figura 1.8 - Formas de energia de um sensor.



Fonte: THOMAZINI, D; ALBUQUERQUE, B., 2007.

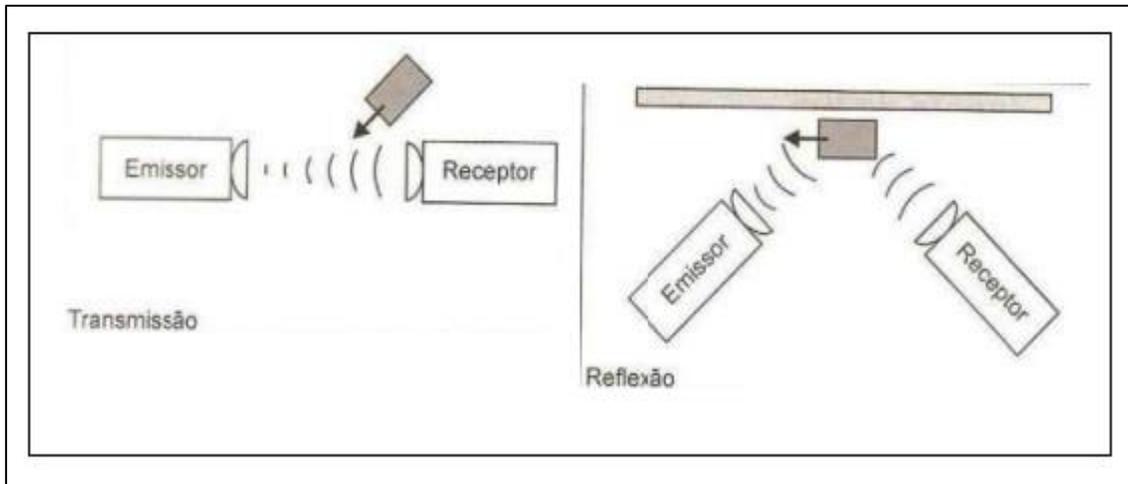
1.5.1 Sensor ultrassônico

De acordo com a empresa WEG (2019) os sensores ultrassônicos podem ser definidos como um conjunto de elementos onde um é de emissão e o outro de recepção, estão posicionados justapostos no mesmo *hardware*. O transmissor emite raios que refletem em uma determinada área do elemento que possa ser detectado e quando isso ocorre o raio retorna ao elemento receptor.

Segundo Thomazini e Albuquerque (2007) Sensores ultrassônicos executam sua função através de sinais que são como ondas de som audíveis, porém em uma frequência muito mais alta. Possuem cristais piezelétricos que ressonam a uma frequência desejada e convertem energia elétrica em energia acústica e vice-versa.

As ondas são transmitidas e refletidas na forma de um cone. Um sinal de saída do transdutor é gerado para fazer alguma indicação ou controle de função. A Figura 1.9 ilustra a transmissão e recepção do ultrassônico.

Figura 1.9 - Transmissor e receptor ultrassônico.



Fonte: THOMAZINI, D; ALBUQUERQUE, B., 2007.

É necessário utilizá-lo com uma distância mínima do sensor para conseguir executar a leitura de forma atrasada no tempo, sendo possível interpretar até mesmo sons de baixo volume. Diversas variáveis podem efetuar a manipulação de um sensor ultrassônico tais como: rugosidade da superfície, mudanças na temperatura ou umidade do ambiente, ângulo da superfície do objeto. Os objetos podem ter qualquer tipo de forma refletiva, até mesmo redondos, sendo essencial para uso em projetos que necessitam precisão no posicionamento desviando de objetos.

1.5.2 Sensor infravermelho

Um sensor pode ser classificado como um instrumento que responde a estímulos físico e/ou químicos tornando possível mensurá-lo obtendo um valor analogicamente e transformá-los em informação. De acordo com a fabricante de sensores OMEGA ENGINEERING (2022), para se obter um valor de radiação Infravermelha do objeto e depois transformá-lo em temperatura é preciso que o objeto

não esteja na temperatura de zero absoluto³, pois nessa condição não há movimentação das moléculas. Quando o objeto está na temperatura acima do zero absoluto os átomos se movem dentro dele, com isso a velocidade de movimento está relacionada diretamente com sua temperatura. Quanto maior a temperatura, mais rápido será o movimento das moléculas, pois, essas moléculas em movimento emitem energia na forma de radiação infravermelha.

Embora não seja possível ver a radiação infravermelha, ainda podemos senti-la na forma de calor. Como por exemplo o calor que sentimos quando estamos próximo ao fogão e até mesmo de uma panela quente, é esse calor que os termômetros infravermelhos detectam e com isso medem a temperatura dos objetos. Isso ocorre porque os termômetros infravermelhos funcionam de forma semelhante à luz visível, pois eles também conseguem ser focalizados, refletidos ou absorvidos.

Através das lentes empregadas nesse equipamento é possível focalizar a luz Infravermelha ou radiação infravermelho emitida do objeto e convertê-la em temperatura. Um equipamento usualmente utilizado que possui o mesmo princípio de funcionamento é o termopilha. A saída de tensão é proporcional a energia infravermelha incidente e o detector usa essa saída para determinar a temperatura.

Existem sensores que são capazes de detectar e indicar as partes de um corpo que esteja acima de uma dada temperatura como por exemplo as câmeras FLIR, ou câmeras térmicas. A Figura 1.10 ilustra uma fotografia de uma pessoa através de uma câmera térmica.

³ **O ZERO ABSOLUTO** é a temperatura mais baixa possível, pois nessa condição não há movimentação das moléculas e nem calor. O zero absoluto ocorre a uma temperatura de 0° graus Kelvin, ou -273,15° graus Celsius, ou a -460° graus Fahrenheit).

Figura 1.10 - Fotografia através de uma câmera térmica



Fonte: www.flir.com.br, 2022.

1.6 Microcontrolador e Arduino UNO

Segundo Souza (2005) microcontrolador é um componente eletrônico de pequeno porte com características para o usuário gravar uma programação na sua memória. A lógica possibilita que faça o controle das entradas e saídas podendo realizar processos que buscam controlar periféricos como botões, sensores, *displays*, entre outros.

Segundo Cocian (2004), a linguagem C é utilizada com sucesso em diversos locais, desde sistemas operacionais até sistemas inteligentes. Atualmente grande parte das linguagens de programação conseguem chegar nos mesmos resultados, algumas mais eficientes do que outras sempre dependendo da aplicação e para o que se destina. A linguagem C foi projetada para a construção de sistemas operacionais com controle de *hardware*. Para Cocian a linguagem C++ é uma alternativa para grandes sistemas, já que ela consegue interpretar linhas do código escritas em C.

Fonseca e Beppu (2010) argumentam que o conceito Arduino iniciou-se em 2005 na Itália com o objetivo de utilização em projetos e protótipos de forma mais acessível comparado com os sistemas disponíveis no mercado. O Arduino é um dispositivo que possui uma plataforma de computação física, baseada numa placa de entrada e saída microcontrolada respeitando uma biblioteca que traduz a linguagem de programação.

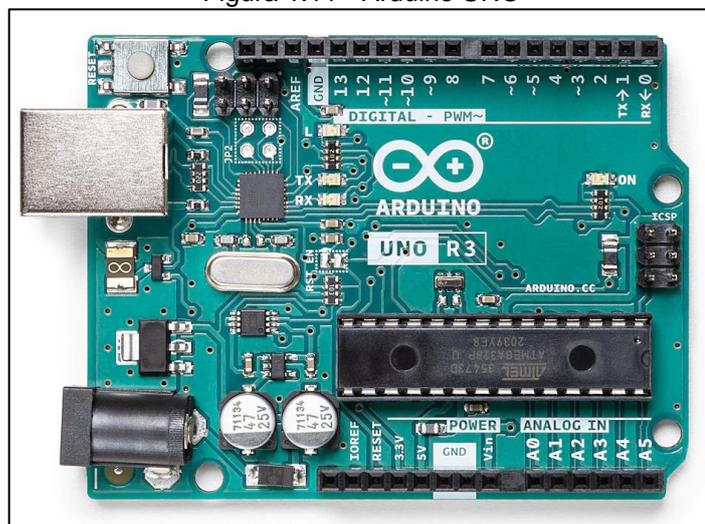
Utiliza como base o microcontrolador ATmega328 na sua versão de entrada (Arduino UNO) que é programável, possibilitando a criação de programas que utilizam uma linguagem própria, semelhante ao C++. Quando a programação é aplicada fazem com que a parte física realize ações previamente configuradas, nessa etapa que ocorre o processamento.

O Arduino é capaz de obter dados e processar informações, transformando em sinais elétricos através de componentes e sensores ligados em seus terminais, ou seja, é um elemento de controle de entrada e saída de dados, executando ações conforme a programação.

A principal vantagem é que foi desenvolvido no modo *open-source* (código aberto), dessa forma, qualquer pessoa que possua conhecimento em programação pode alterar e aperfeiçoar a programação base conforme sua necessidade, causando assim uma maior aplicabilidade e versatilidade do dispositivo.

Para o desenvolvimento do projeto, foi utilizado o Arduino UNO que possui o microcontrolador ATmega328, são 14 portas digitais, 6 portas analógicas, sua tensão de operação é 5 V, sua tensão de entrada deve ser de 7 a 12 V, a velocidade clock é de 16 MHz e a corrente nos pinos I/O é de 40 mA. A Figura 1.11 ilustra a imagem do Arduino UNO.

Figura 1.11 - Arduino UNO



Fonte: www.arduino.cc, 2022.

1.7 Termologia

Quando ocorre um deslizamento de terra, algumas vítimas são soterradas impedindo a localização direta pelos bombeiros, visando auxiliar também pessoas nessas condições, o projeto utiliza um sensor de temperatura, para identificar variações de calor no solo, visando minimizar o número de desaparecidos. Porém para identificar como o calor do corpo humano se comporta nessas condições fez-se necessário buscar informações do campo Termológico da física para fundação do projeto.

A palavra termologia é deriva das palavras em grego “Termo” que significa calor e “Logia” estudo, sendo esse o ramo da física voltado aos fenômenos relacionados ao calor e à temperatura, como transferência de calor, equilíbrio térmico, transformações sofridas por gases etc. Para esse projeto será aprofundado a matéria de transferência de calor, mais especificamente o processo de radiação térmica.

Visando identificar como a temperatura do corpo da vítima irá se comportar na superfície do solo, é necessário se aprofundar nos processos de transferência de calor, os processos de transferência de calor podem ser divididos em três partes (INCROPERA et al., 2007):

- Condução é quando ocorre a transferência de energia das partículas mais enérgicas para as menos enérgicas, havendo transmissão de calor através do meio;
- Convecção é quando o modo de transferência de calor ocorre devido movimentação das partículas e movimentação global do fluido, ou seja, transmissão de calor por fluido em movimento;
- Radiação Térmica é energia emitida por uma matéria que se encontra em uma temperatura diferente de zero absoluto (Kelvin), emitindo ondas eletromagnéticas que transmitem calor sem a necessidade de um meio.

Considerando a necessidade de identificar um corpo humano, através da temperatura, foi levado em consideração o processo de transmissão por meio da radiação térmica.

Por volta dos anos de 1879 e 1874, os físicos Stefan e Boltzmann desenvolveram uma lei que demonstra através da expressão matemática a relação entre ondas eletromagnéticas e a temperatura de um corpo, através dessa lei é possível identificar qual a potência total irradiada pela superfície externa de um corpo que se encontra a uma temperatura absoluta. A figura 1.12 a seguir demonstra como ocorre essa relação:

Figura 1.12 - Lei de Stefan-Boltzmann

$$\left(\frac{\Delta Q}{\Delta t} \right) = \varepsilon A \sigma T^4$$

Fonte: <https://www.ifsc.usp.br>

Onde:

- $\frac{\Delta Q}{\Delta t}$ = potência total irradiada pela superfície externa de um corpo (Watts)
- ε = Emissividade térmica, que define o quanto um corpo tem capacidade de emitir calor, segundo o laboratório de biomecânica da UFSC (2022) a emissividade do corpo humano é próxima de 0,98;
- A = Representa a área que o corpo ocupa, segundo o professor SILVA (2022), para o corpo humano essa medida é aproximadamente 1,6 m² para mulheres e 1,9 m² para homens;
- σ = constante de Boltzmann, cujo valor é $\sigma = 5,67 \times 10^{-8} \text{ W/m}^2\text{K}^4$
- T = Temperatura da superfície em Kelvin.

Sendo assim, uma pessoa em seu estado natural irradia uma potência que varia de acordo com sua temperatura, área corporal e temperatura do ambiente onde está exposta. Para fins de aplicação no projeto a potência vinculada a emissão de ondas

eletromagnéticas, produz espectro infravermelho que pode ser identificado por câmeras térmicas, podendo assim, com utilização desse equipamento, auxiliar as equipes de resgate na identificação das vítimas.

2 METODOLOGIA

Encontra-se nesta parte do trabalho todo o seu percurso até o seu desenvolvimento, como também as partes que compõem a construção física do projeto denominado: Sistema para auxílio no resgate em área afetada por deslizamento.

2.1 O que é metodologia

De acordo com Prodanov e Freitas (2013) pode-se compreender metodologia como uma disciplina onde estudar os diversos métodos idealizados até o momento, compreender seus conceitos para a montagem do conhecimento validando-o, avaliá-los quanto as aplicações e seguir seus preceitos são algumas das definições para metodologia.

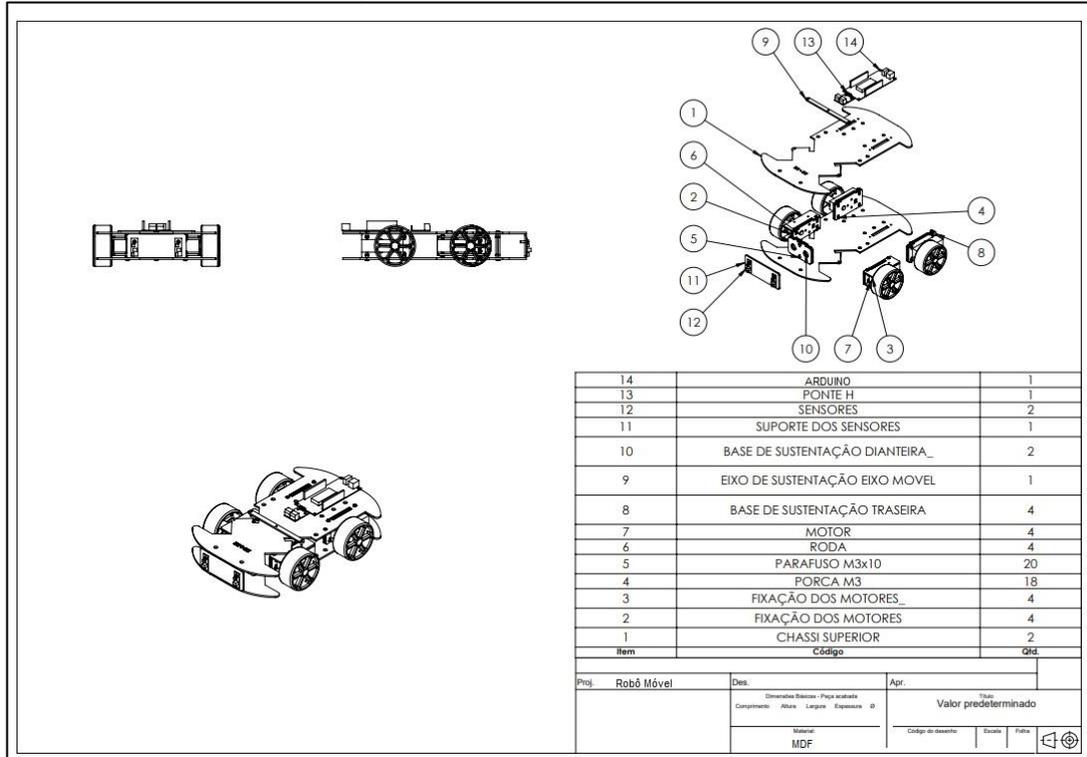
A palavra metodologia vem do grego “meta” que significa ao largo; “odos” que significa caminho e por fim “logos” que significa discurso e/ou estudo.

2.2 O tema-problemática, justificativa e diagrama de funcionamento

Para a idealização da ideia foi levantado diversos questionamentos a respeito das condições iniciais que se encontram um ambiente após deslizamento. O transporte aéreo será o meio utilizado nesse projeto, entretanto para o desenvolvimento do protótipo, por questões de viabilidade econômica foi necessário adaptar as funções do sensor de temperatura e do sensor ultrassônico para um robô móvel, mantendo somente a captação e tratamento de imagem no drone.

O deslocamento do robô móvel e do drone será feito através do controle de movimentação manual que determina o caminho a ser percorrido. A Figura 2.1 ilustra o desenho do robô móvel.

Figura 2.1 - Desenho do robô móvel



Fonte: Autoria própria, 2022.

A escolha do drone que será utilizado foi realizada considerando alguns aspectos como legislações vigentes e código *open source*. A Figura 2.2 mostra o modelo do drone que será utilizado no projeto.

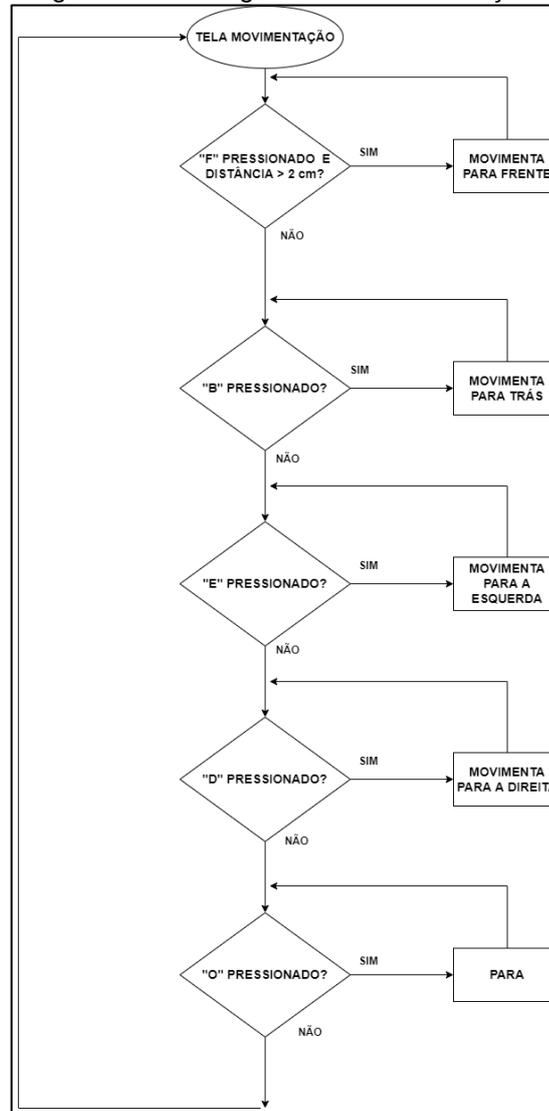
Figura 2.2 - Drone utilizado



Fonte: Autoria própria, 2022.

Para o desenvolvimento do projeto foi construído um fluxograma de movimentação do robô. A Figura 2.3 ilustra o fluxograma de movimentação do robô.

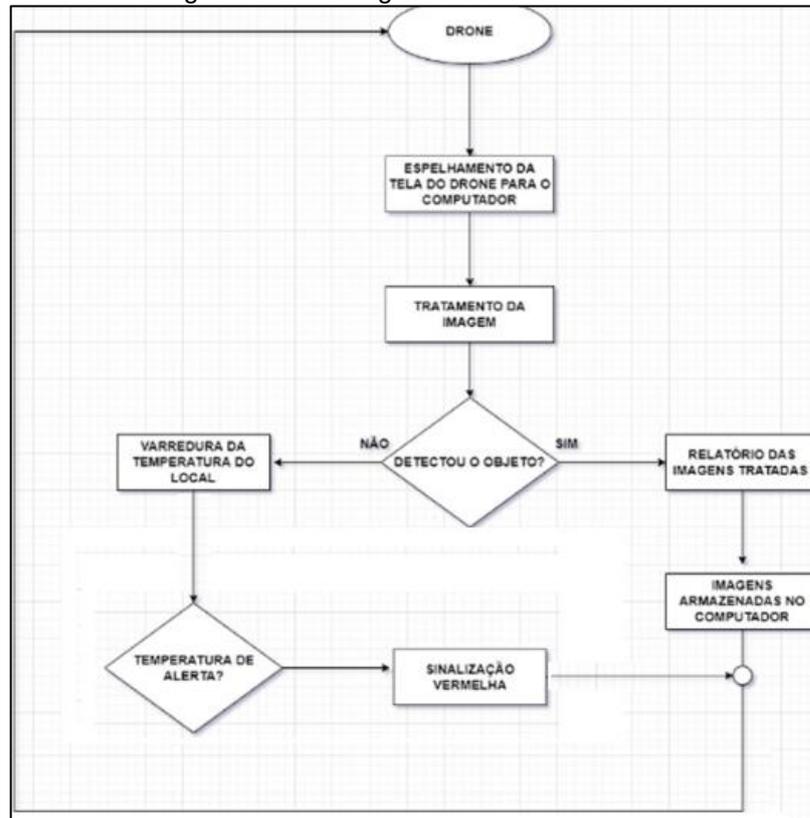
Figura 2.3 - Fluxograma de movimentação.



Fonte: Autoria própria, 2022.

O conjunto do sistema utiliza o drone e o robô móvel para captação de temperatura e tratamento de imagens, a Figura 2.4 ilustra o fluxograma de funcionamento do sistema.

Figura 2.4 – Fluxograma de funcionamento



Fonte: Autoria própria, 2022.

2.3 Etapas teóricas e práticas para o desenvolvimento do projeto

Depois de determinadas o tema-problema, justificativa e descrição do projeto, parte-se para as seguintes etapas:

Primeira etapa: reunião dos componentes do grupo, para escolha do orientador. Após o convite, o orientador se colocou à disposição do grupo e concordou nos orientar com o tema.

Segunda etapa: reunião dos integrantes do grupo com o orientador para traçar as diretrizes de como efetuar a pesquisa teórica e sua seleção. O orientador fez uma breve explanação sobre o tema e colocou-se à disposição para atender

Figura 2.6 - Levantamento de custos do projeto

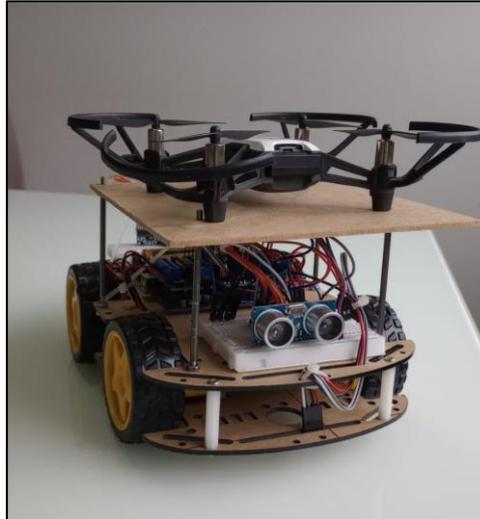
Descrição	Especificação	Quantidade	Preço
Arduino UNO R3	UNO R3	1 peça	R\$ 132,90
Módulo Bluetooth	RS232 HC-06	1 peça	R\$ 41,44
Jumpers	Macho/macho e Macho/Fêmea	2 kits (40peças cada)	R\$ 17,49
Sensor Ultrassônico	HC-SR04	1 peça	R\$ 14,25
Sensor temperatura	MLX 90614	1 peça	R\$ 142,49
Ponte H	L293D	1 peça	R\$ 31,72
Robô 4 rodas- 4WD	4WD/ 120mA/ 22x14,7 cm	1 peça	R\$ 101,78
Chapa de acrílico	100cmx200cmx2mm	1 peça	R\$ 291,00
Protoboard	170 pontos/branca	2 peças	R\$ 10,44
Fonte de alimentação	5v e 1A	1 peça	R\$ 16,50
Webcam	720p/logitech	1 peça	R\$ 156,54
Módulo USB	CH375	1 peça	R\$ 51,51
Suporte para drone	MDF e Barra roscada	1 kit	R\$ 20,00
Bateria	18650	4 peças	R\$ 62,00
Drone	Dji Tello	1 peça	R\$ 949,00
Buzzer	5V/ 12mm	1 peça	R\$ 2,54
Total			R\$ 2.041,60

Fonte: Autoria própria, 2022.

3 DESENVOLVIMENTO DO PROJETO

Neste capítulo encontra-se passo a passo o desenvolvimento e construção do projeto que se intitula: Sistema para auxílio no resgate em área afetada por deslizamento. A Figura 3.1 ilustra o projeto final.

Figura 3.1 - Projeto Final



Fonte: Autoria própria, 2022.

3.1 Montagem do espelho do drone e do robô móvel

O drone utilizado no projeto é o DJI Tello, que possui uma câmera frontal com resolução de 5 megapixels para fotos e 720p(HD) para vídeo. Como sua câmera se encontra apontada para frente o grupo optou por colocar um espelho na parte da frontal do drone, fazendo com que a imagem captada pelo drone passe a ser a que está abaixo dele. A Figura 3.2 ilustra o drone com o suporte para o espelho refletor.

Figura 3.2 - Drone com o suporte para o espelho.



Fonte: Autoria própria, 2022.

Após a montagem do suporte para espelho no drone, foram iniciadas as montagens do robô móvel. A Figura 3.3 ilustra o robô móvel desmontado, na embalagem comercial.

Figura 3.3 – Robô móvel desmontado.

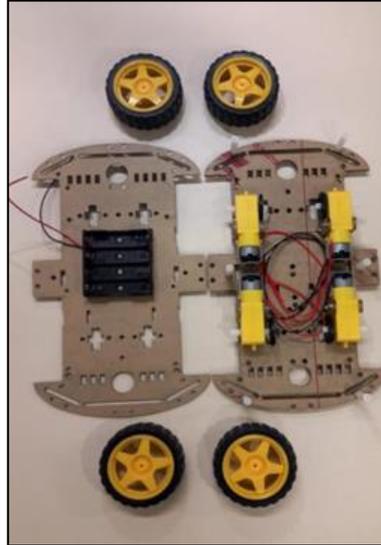


Fonte: Autoria própria, 2022.

Contidos na embalagem se encontra duas placas de montagem já furadas, um denominado chassi, onde se encontram acoplados os 4 motores contidos na embalagem, e a outra onde irão ficar os circuitos de controle que são: *drives* para os motores, placa do Arduino, módulo *bluetooth*, sensor infravermelho de temperatura e

sensor ultrassônico, como também o suporte de pilhas. A Figura 3.4 ilustra as placas de montagem do protótipo.

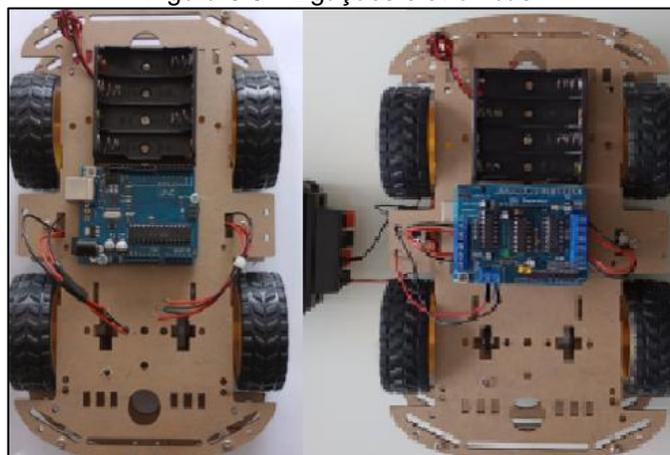
Figura 3.4 - Placas de montagem do protótipo.



Fonte: Autoria própria, 2022.

Após finalizada a estrutura, foi posicionado o motor *Shield* L293D, que fará o controle do sentido de rotação dos motores do robô móvel. Esse motor *Shield* executa a função de uma ponte H, fazendo o controle de sentido de rotação de até 4 motores DC de até 24 V, tendo que utilizar uma fonte externa para tensões acima de 5 V. A placa do motor *Shield* L293D possui também mais 2 entradas de controle analógico para servo motores, e sua placa, é alimentada pelo próprio controlador Arduino. A Figura 3.5 ilustra as ligações eletrônicas que foram feitas no *drive* motor *Shield* L293D para o Arduino, assim como do *drive* para os motores.

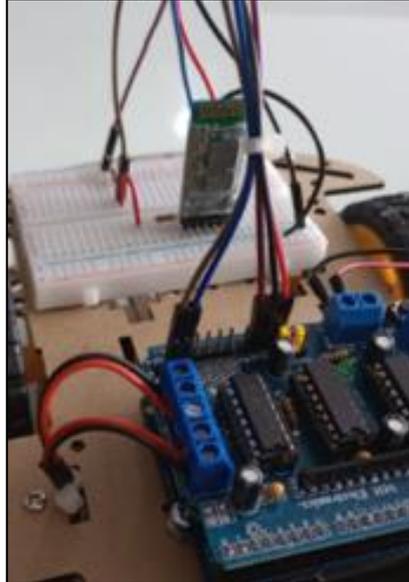
Figura 3.5 - Ligações eletrônicas.



Fonte: Autoria própria, 2022.

Após a conclusão da montagem mecânica e disposição dos componentes eletrônicos, iniciou-se os testes de comunicação com o módulo *bluetooth* para que fosse feito o controle da direção através do aplicativo. A Figura 3.6 ilustra o módulo HC-05 conectado na entrada A0 e A1 do Arduino.

Figura 3.6 - Módulo bluetooth ligado ao Arduino



Fonte: Autoria própria, 2022

Após os testes de comunicação por *bluetooth* entre o protótipo e o *smartphone*, iniciou-se o teste de comunicação com o aplicativo, cujo objetivo era realizar a movimentação do protótipo através do *smartphone*, onde obtivemos êxito e assim pudemos seguir com o desenvolvimento do protótipo.

Com a realização das etapas anteriores e resultados satisfatórios, iniciamos a escolha dos sensores que se adequassem ao projeto, onde pudemos escolher um modelo de sensor de distância e outro de temperatura que fosse agradável ao orçamento e em proporção de protótipo para teste. A Figura 3.7 ilustra os sensores escolhidos para o projeto.

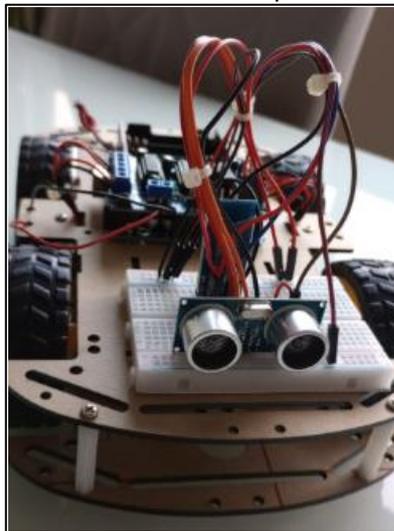
Figura 3.7- Sensores utilizados no projeto



Fonte: Autoria própria, 2022.

Iniciou-se assim os testes de comunicação dos sensores e o seu posicionamento na base do robô móvel de forma onde seja possível obter seus valores de temperatura em graus Celsius ($^{\circ}\text{C}$) e sua distância em milímetros (mm). A Figura 3.8 mostra o sensor ultrassônico já posicionado na base do robô móvel.

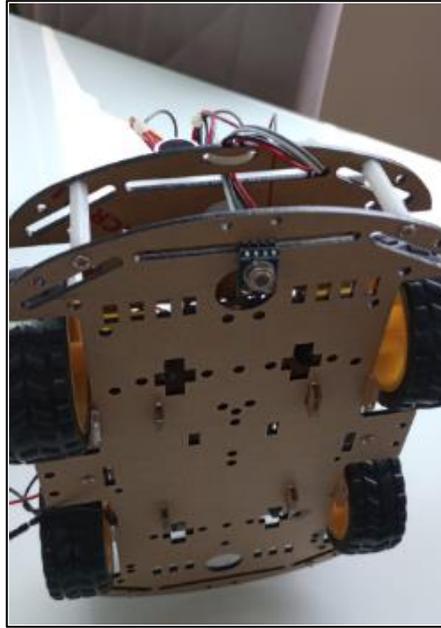
Figura 3.8 - Sensor ultrassônico posicionado na base



Fonte: Autoria própria, 2022.

Após realizado os testes de funcionamento do sensor de temperatura através do monitor serial do Arduino, o sensor foi posicionado na base do carrinho o mais próximo do solo. Como o sensor MLX90614 tem um alcance máximo de mensuração da temperatura de objetos de até 10 mm, em condições reais um sensor infravermelho de alcance superior a 5 metros seria empregado ao drone, e a temperatura do solo seria medida do alto sem a necessidade do uso do robô móvel. A Figura 3.9 ilustra o posicionamento do sensor de temperatura.

Figura 3.9 - Sensor laser de temperatura

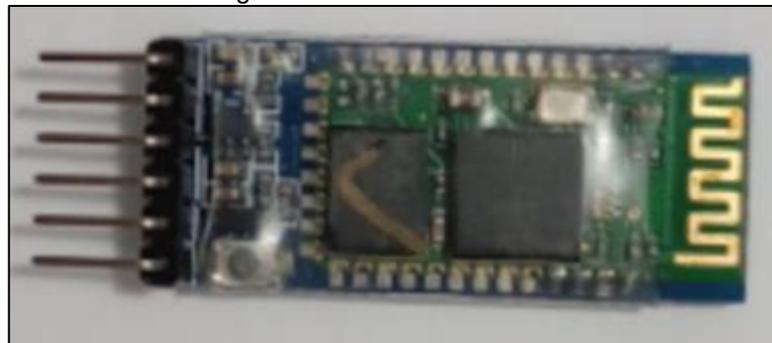


Fonte: Autoria própria, 2022.

3.2 Sensoriamento e módulo *bluetooth*

Para realizar a comunicação entre o Arduino e o celular faz-se uso do módulo *bluetooth* HC-05, que diferente do modelo HC-06, pode ser tanto mestre quanto escravo. Em sua placa existe um regulador de tensão que possibilita a alimentação com 3,3 a 5 V, seu alcance é de 10 metros, sua temperatura de trabalho é de $-40\text{ }^{\circ}\text{C}$ até $120\text{ }^{\circ}\text{C}$, a corrente parado é de 35 mA e a corrente conectado é de 8 mA, sua velocidade síncrona é de 1 Mbps para envio e coleta de dados. A Figura 3.10 ilustra o módulo *bluetooth* HC-05.

Figura 3.10 - Módulo bluetooth



Fonte: Autoria própria, 2022.

Para comunicar o drone com o celular, faz-se uso da comunicação *Wi-Fi* e as imagens do drone são espelhadas para o computador e tratadas

através da linguagem de programação *Python*. A comunicação ideal seria por rádio frequência pelo fato de que existem locais afastados que não temos rede, impossibilitando a comunicação via *Wi-Fi*.

Por fim, no Apêndice A – Programação do Robô Móvel é possível visualizar a programação completa do funcionamento do Robô, e analisá-la por meio dos comentários.

3.3 Aplicativo

A primeira etapa do aplicativo refere-se ao desenvolvimento das telas e configuração dos parâmetros de *layout*. Utilizamos o software APP Inventor do MIT (*Massachusetts of Institute Technology*) que possui uma linguagem de programação em blocos lógicos.

No controle do robô móvel utiliza-se o aplicativo com comunicação via *bluetooth*, que possibilita a comunicação do Arduino com o módulo HC-05, que por sua vez recebe dados do celular para efetuar a movimentação do robô e recebe dados dos sensores que são encaminhados para o aplicativo.

A tela inicial do aplicativo, denominada de *Login* tem como objetivo verificar o acesso através de um usuário padrão, chamado de “Adm” e a senha padrão “123”; podendo ser alterado nas telas subsequentes. A Figura 3.11 ilustra a tela de *login* do aplicativo.

Figura 3.11 - Tela de *Login*

Fonte: Autoria própria, 2022

Para o funcionamento da tela de *login* foi necessário efetuar comparações referentes ao valor escrito nas caixas de texto ao clicar no botão *Login*. É comparado se o texto escrito na caixa Usuário é condizente com o usuário correto e a mesma comparação se repete para a senha. Após a comparação ser efetuada, o aplicativo retorna um *feedback*, caso o *login* estiver correto a mensagem “Senha correta” é mostrada na tela e logo em seguida a tela de Menu é aberta, caso estiver incorreta a mensagem “ERRO!!!! Senha incorreta” é mostrada na tela e o usuário permanece na mesma tela. Se o usuário alterar a senha é necessário que o aplicativo seja encerrado e iniciado novamente para que a senha atualize. A Figura 3.12 ilustra o diagrama de blocos desenvolvido para o funcionamento da tela.

Figura 3.12 - Diagrama de blocos, tela login

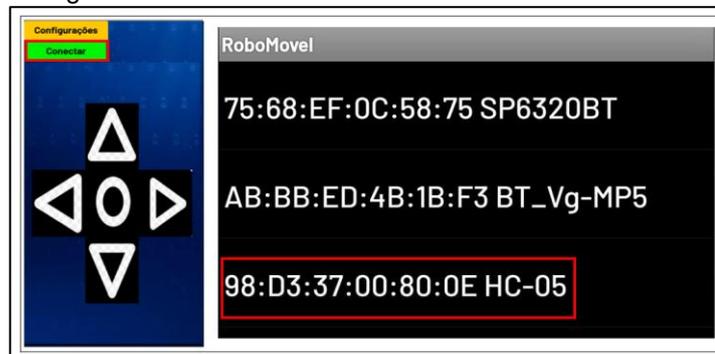


Fonte: Autoria própria, 2022.

Após o acesso a conta, é necessário que efetue a conexão com o módulo *bluetooth* HC-05 para enviar e receber dados, caso o dispositivo não seja encontrado é recomendado verificar se o *bluetooth* do celular está ativado. Pressionando o botão

“CONNECTAR”, uma lista de dispositivos disponíveis é aberta e o módulo deve ser escolhido, a Figura 3.13 ilustra a conexão que deve ser feita.

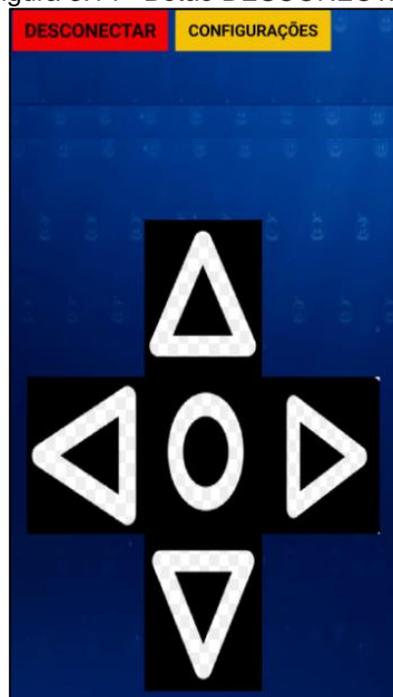
Figura 3.13 - Conexão do celular com o módulo HC-05



Fonte: Autoria própria, 2022.

Para o botão executar a função de buscar a lista de dispositivos disponíveis para conexão foi necessário usar a opção de “Cliente Bluetooth” e assim que o dispositivo é pareado o botão “CONNECTAR” na cor verde muda para “DESCONECTAR” na cor vermelha possibilitando o usuário desfazer a comunicação. A Figura 3.14 ilustra o botão “DESCONECTAR” em vermelho após a comunicação ser configurada com sucesso.

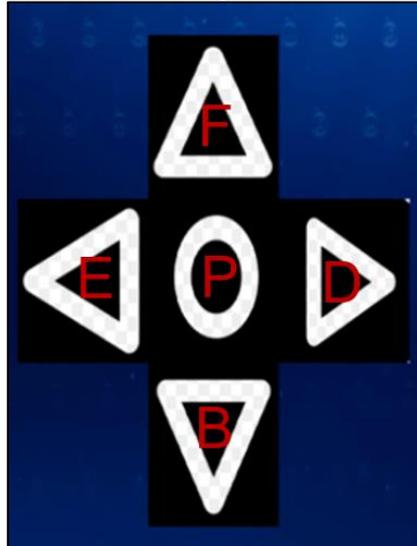
Figura 3.14 - Botão DESCONECTAR



Fonte: Autoria própria, 2022.

Assim que conectado, o robô móvel pode ser controlado pelo aplicativo através das teclas de indicação. Cada tecla possui uma identificação específica para envio de dados via *bluetooth*, movimentando o robô para a posição desejada. A Figura 3.15 ilustra o caractere que é enviado para o Arduino e a sua respectiva seta.

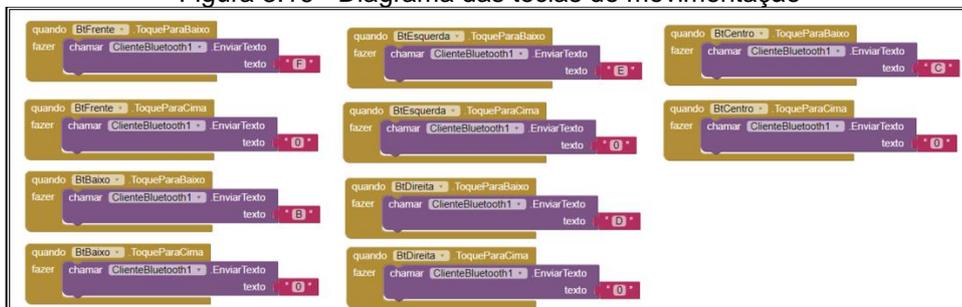
Figura 3.15 - Identificação da movimentação



Fonte: Autoria própria, 2022.

Para indicar o caractere específico de cada tecla faz-se uso da atribuição de valores para cada botão pressionado, ou seja, cada seta é um botão individual com identificação. Quando o botão é pressionado o valor mostrado na imagem anteriormente é enviado, caso não pressione nenhum botão é enviado o valor “0” indicando que o robô deve permanecer parado. A Figura 3.16 indica o diagrama utilizado para especificar cada tecla.

Figura 3.16 - Diagrama das teclas de movimentação



Fonte: Autoria própria, 2022.

Após a finalização da tela de movimentação foi necessário criar a tela de configuração e parametrização, nessa tela é possível alterar a temperatura e alterar

a senha do aplicativo. Utiliza-se dois botões que direcionam para outras telas. Na programação o botão “ALTERAR TEMPERATURA” está associado ao “BtTemperatura” e o “ALTERAR SENHA” está associado ao “BtSenha”. A Figura 3.17 ilustra a tela de configuração.

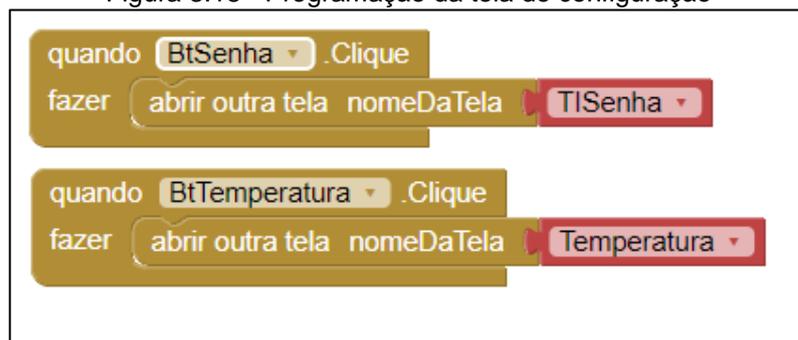
Figura 3.17 - Tela de configuração



Fonte: Autoria própria, 2022.

Faz-se uso de dois botões para o direcionamento de telas, quando o usuário escolher a opção “ALTERAR TEMPERATURA” ocorre o direcionamento para a tela “Temperatura” e quando escolher a opção “ALTERAR SENHA” seu direcionamento será para a tela “TISENHA”. A Figura 3.18 ilustra a programação da tela de configuração.

Figura 3.18 - Programação da tela de configuração



Fonte: Autoria própria, 2022.

Ao entrar na tela de alteração de senha o usuário tem a possibilidade de modificar os padrões de *login*, porém é necessário que a senha de confirmação seja igual a senha inserida na caixa de texto “Senha”, caso contrário o aplicativo emite um alerta visual informando que ambas as senhas devem ser iguais. A Figura 3.19 ilustra a tela de alteração de senha.

Figura 3.19 - Tela de alteração de senha

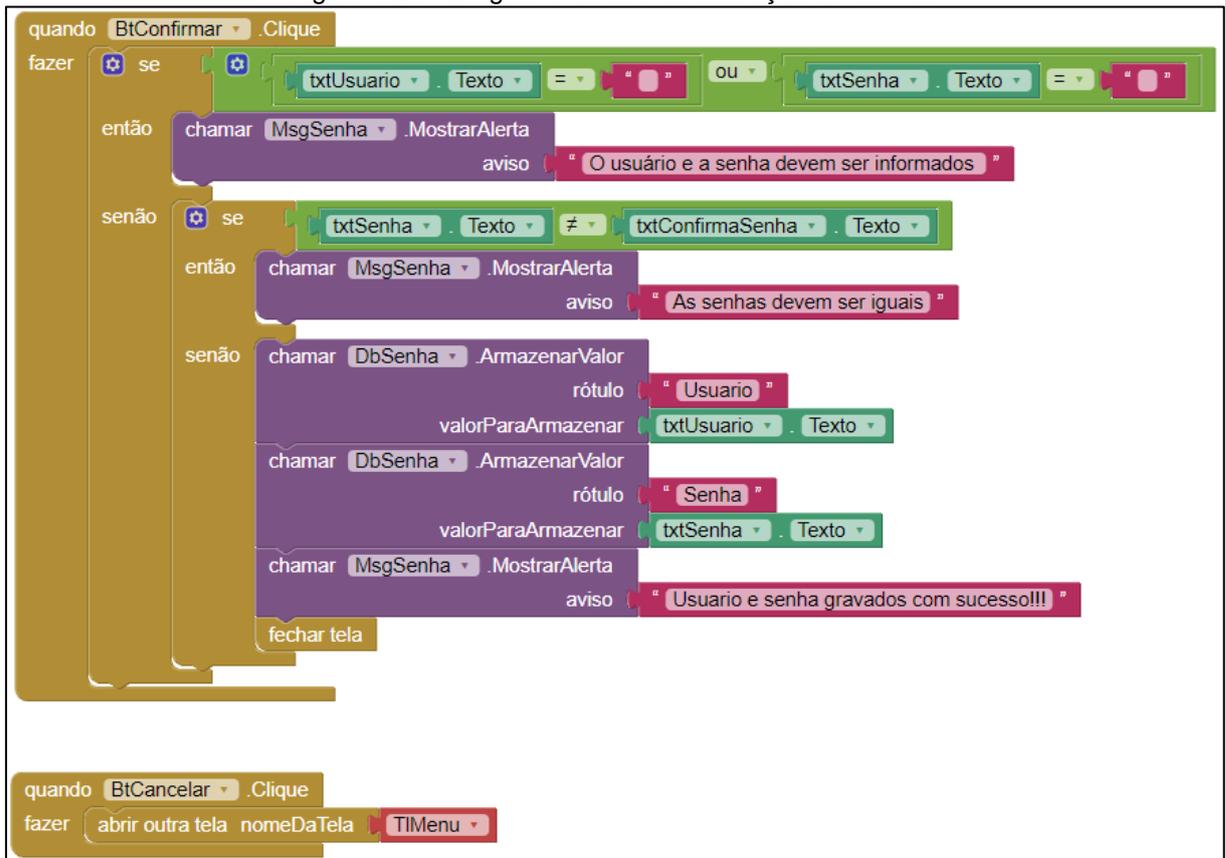


A captura de tela mostra uma interface de usuário para a alteração de senha. O título da tela é "Alteração de Senha". O formulário contém três campos de entrada de texto: "Usuário", "Senha" e "Confirme a senha". Abaixo dos campos, há dois botões: "Confirmar" em verde e "Cancelar" em vermelho.

Fonte: Autoria própria, 2022.

Para o funcionamento da tela de alteração de senha faz-se uso da lógica ao apertar o botão confirmar. Quando o usuário apertar “CONFIRMAR” o programa verifica se as caixas de texto não estão em branco e se as senhas não são diferentes, caso os valores inseridos atendam os atributos, o usuário e a senha são armazenados no “DbSenha”, cuja sua finalidade é guardar os valores de *login* para verificação e validação de usuário. A Figura 3.20 ilustra o diagrama de funcionamento da tela de alteração de senha.

Figura 3.20 - Diagrama da tela de alteração de senha.



Fonte: Autoria própria, 2022.

Ao entrar na tela de alteração da temperatura é possível visualizar duas caixas de texto, uma de temperatura mínima e outra de temperatura máxima. São ajustáveis o valor mínimo aceitável e o valor máximo aceitável em graus Celsius e após a parametrização é necessário pressionar o botão “SALVAR” para definir os valores como padrão. A Figura 3.21 ilustra a tela de alteração da temperatura.

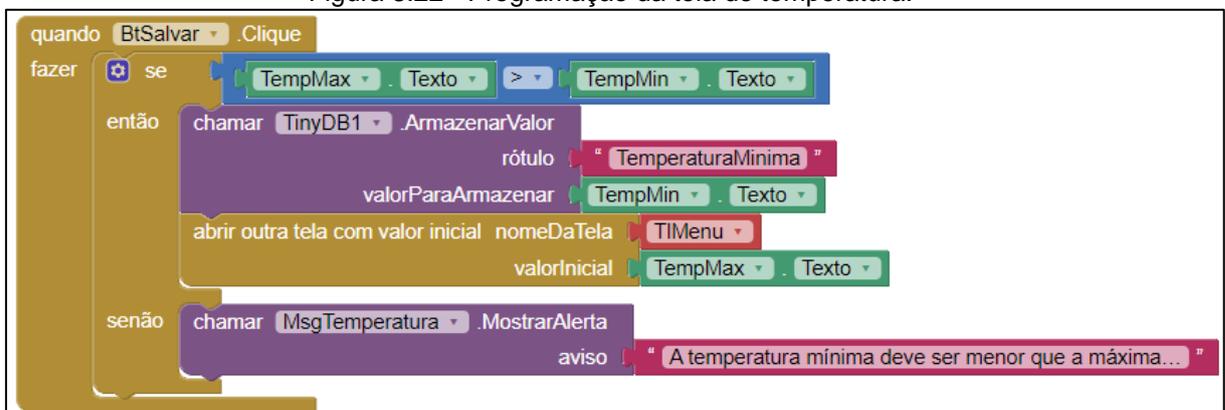
Figura 3.21 - Tela de temperatura



Fonte: Autoria própria, 2022.

A programação da tela de temperatura tem como início o botão “SALVAR”, pois é ele que faz com que o ciclo de varredura aconteça. Obrigatoriamente a temperatura mínima ajustável inserida pelo usuário deve ser menor que a máxima, caso contrário, é emitido um alerta visual informando que a temperatura mínima deve ser menor que a máxima possibilitando o usuário alterar os valores digitados. Caso o valor mínimo seja menor que o máximo o usuário é enviado para a tela de movimentação e os valores digitados são armazenados no banco de dados “TinyDB”. A Figura 3.22 ilustra a programação da tela de temperatura.

Figura 3.22 - Programação da tela de temperatura.

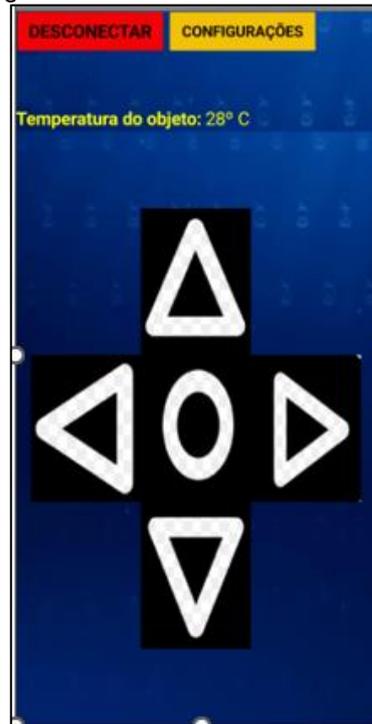


Fonte: Autoria própria, 2022.

Retornando para a tela de movimentação o valor de temperatura é visível na tela e caso o valor lido no sensor de temperatura ultrapasse os limites previamente

estabelecidos, o aplicativo emite um alarme sonoro informando que a temperatura daquele local está fora dos padrões possibilitando o corpo de bombeiros agir diretamente na anormalidade. A Figura 3.23 ilustra a tela de movimentação e o valor da temperatura visível.

Figura 3.23 - Tela de movimentação

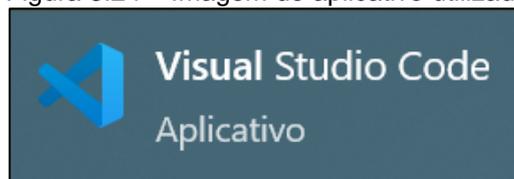


Fonte: Autoria própria, 2022

3.2 Reconhecimento de imagem

Para o desenvolvimento da programação de reconhecimento de imagem a linguagem de programação escolhida foi o Python e o aplicativo utilizado foi o Visual Studio Code demonstrado na Figura 3.24.

Figura 3.24 – Imagem do aplicativo utilizado



Fonte: Autoria própria, 2022

De acordo com o programa disponível no APENDICE B a primeira etapa da programação, comentário “PARTE 1”, foi dedicada a importação das bibliotecas, do

drone que tem como nome “djitellopy”, a biblioteca Open CV chamada “cv2” que responsável por habilitar todos os recursos de imagem dentro do programa, como por exemplo a abertura da câmera, pôr fim a biblioteca “numpy” utilizada para os recursos numéricos, como operações em matrizes.

Logo em seguida, comentário “PARTE 2”, foi necessário abrir os modelos de treinamento que serão usados no programa para a classificação da imagem, o *yolo weights* representa os modelos treinados e o coco names é um *dataset* utilizado para nomear as imagens classificadas.

A terceira etapa do programa, comentário “PARTE 3”, foi adicionado o Class Tello 2 com o intuito de conectar o IP da câmera do drone no computador, permitindo realizar o *streaming* entre esses dispositivos, dessa forma tudo que está sendo visualizado pela câmera do drone é transmitido para o computador. Caso seja utilizado uma webcam ao invés de habilitar a conexão com o IP, a programação é substituída por VideoCapture (1), reconhecendo a câmera conectada pela porta USB como o local por onde o vídeo será transmitido.

Logo em seguida, comentário “PARTE 4”, é realizado o início do processo de detecção dos objetos, o programa é colocado dentro de um *loop* para que visualize a imagem transmitida repetidas vezes identificando os itens mesmo com a câmera em movimento. Para o seu funcionamento os *frames* são captados e analisados pelos modelos adicionados anteriormente, na sequência seus recursos são extraídos de forma que o treinamento seja executado através das camadas, classificando assim os objetos.

Por fim, após a imagem ser identificada essa, comentário “PARTE 5”, verifica a precisão do que foi reconhecido e mostra visualmente, por meio de um retângulo colorido, o nome, a porcentagem de confiança e a localização dele na imagem capturada, finalizando assim o programa.

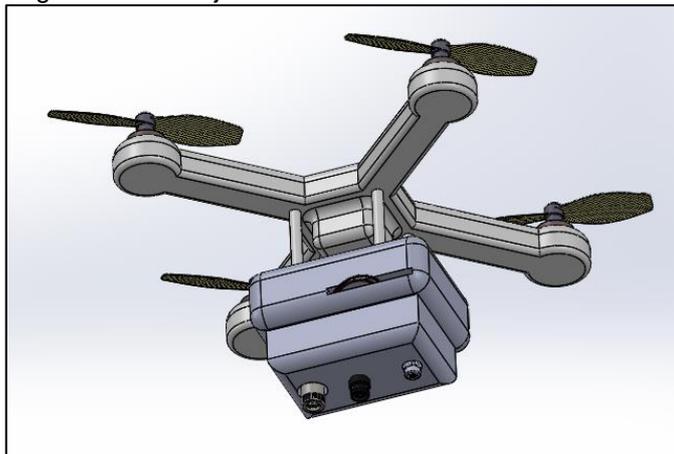
3.3 Melhoria futura

Como melhoria futura idealizamos um projeto em que as funções do protótipo sejam incorporadas em um único hardware, ou seja, através da utilização de uma microcâmera cujo modelo é “WOLFWHOOP WT03” com transmissão de imagem via rádio. O sinal seria capturado através do receptor de sinal “ROTG01 5.8G” permitindo assim a captura da imagem de vídeo da câmera. A partir daí inicia-se um estudo de comunicação para que a imagem capturada pela câmera seja trabalhada pelo software YOLO, e assim obter os vídeos e imagem tratadas.

Ainda partindo da ideia de acoplar todo o projeto em um mesmo *hardware*, para obtenção do valor de temperatura de um objeto a uma distância de até mais ou menos 2 metros indica-se o medidor de temperatura infravermelho compacto OS-AE11-201 da empresa OMEGA. Entretanto o custo-benefício deste produto é alto. Além de acoplá-lo ao dispositivo é necessário um estudo para que ele seja alimentado com uma tensão de 24 V.

Para que o sinal seja transmitido do *hardware* para um controle, foi pensado no ESP-32 com transmissão via *wi-fi*. A Figura 3.25 ilustra um possível *hardware* para acoplar esses componentes:

Figura 3.25 - Projeto Idealizado no software SolidWorks



Fonte: <https://grabcad.com/library/drone-quadcopter-19,2022> (modificado)

CONSIDERAÇÕES FINAIS

O trabalho teve como objetivo desenvolver um sistema integrado composto por um drone e um aplicativo que auxiliasse os bombeiros no resgate das vítimas de deslizamento de terra, visando diminuir os impactos desses acidentes na sociedade.

Para o seu desenvolvimento foi necessário a utilização dos seguintes *softwares*: Visual Code, App Inventor e Arduino IDE.

O Visual Code foi a plataforma utilizada para o reconhecimento de imagem através do drone. A programação foi realizada na linguagem *Python*, com a adição da biblioteca *Open CV*, *YOLO* e *Coco Databaset*.

O AppInventor foi utilizado em conjunto com o Arduino IDE para a programação do aplicativo e a movimentação do robô, realizando uma integração que gerencia e capta as informações referente a temperatura e distância e emite alarmes caso esses padrões estejam fora dos valores pré-estabelecidos.

Para a evolução e desenvolvimento do projeto as pesquisas realizadas na fundamentação teórica foram essenciais.

O projeto se destaca por se tratar de um sistema que age quando o desastre não foi evitado, visando diminuir o número de vítimas fatais e auxiliar as equipes de resgate, utilizando a tecnologia para o bem da sociedade.

Durante o desenvolvimento tivemos dificuldades na interrupção da movimentação do robô móvel quando havia obstrução do caminho, na coleta de dados do sensor de temperatura, no tempo de leitura e reconhecimento da imagem e transferência do treinamento da webcam para a câmera do drone.

No entanto, esses obstáculos foram solucionados após verificar que o tempo de resposta dos sensores estava divergente do tempo de captação do Arduino e que a capacidade de processamento do computador impedia que a imagem fosse fluida.

Foram realizados ajustes na programação, sincronizando o *baud rate* dos dispositivos e o treinamento de imagem foi adaptado para um dispositivo com maior tempo de resposta, solucionando assim os problemas.

Propõem como ações de melhorias futuras o desenvolvimento de um drone completo com uma placa de controle para captação da temperatura e de distância dos objetos, utilizar a tecnologia do *Big Data* para envio dos dados a nuvem com a finalidade de possuir maior gerenciamento dos valores captados, aplicar mecanismos de pré-carregamento dos dados (buffering) para melhorar a velocidade do reconhecimento de imagem em relação a transmissão da câmera e por fim, obter através de pesquisas realizadas por órgãos públicos dados referente a população do local afetado, visando ter uma estimativa da quantidade de vítimas.

Conclui-se que o projeto obteve êxito em relação aos objetivos traçados, possibilitando que o grupo aplicasse os conhecimentos obtidos durante o curso e novos conhecimentos adquiridos no decorrer do desenvolvimento.

REFERÊNCIAS

ANPEGE. Análise de atributos morfométricos de vertentes em relação às ocorrências de deslizamento de terra e inundações, na bacia hidrográfica do Rio Sorocaba, São Paulo. Disponível em: <http://www.enanpege.ggf.br>. Acesso em 8 de mar. 2022.

ARONOFF, S. **Geographical Information Systems: A Management Perspective**. Ottawa, WDI Publications, 1989.

AZEVEDO, João. **Suscetibilidade a Deslizamentos do Brasil: Primeira aproximação**. ed. 1, p. 1-60, 20 fev. 2019. Disponível em: <https://biblioteca.ibge.gov.br/index.php/bibliotecacatalogo?view=detalhes&id=2101684>. Acesso em: 2 de mai. 2022.

BARBOSA, Xênia. Breve introdução à história da inteligência artificial. **Jamaxi**, [S. l.], v. 4, n. 2, 2020.

BARANAUSKAS, J; MONARD, M; **Aplicações de Inteligência artificial: Uma Visão Geral**. 2000. 02p. Dissertação – Instituto de Ciências Matemáticas e de Computação de São Carlos, 2000.

COCIAN, Luis Fernando Espinosa. **Manual da linguagem**. 1. ed. Universidade Luterana do Brasil: ULBRA, 2004. 493 p. v. 1.

DARÉ, Eliane. **Petrópolis: não foi desastre natural!** Disponível em: www.unicamp.br/unicamp/ju/noticias/2022/02/22/petropolis-nao-foi-um-desastre-natural. Acesso em 08 mai. 2022.

FONSECA, E. G. P.; BEPPU, M. M. **Apostila Arduino**. Rio de Janeiro: Universidade Federal Fluminense, 2010.

FUNDAÇÃO INSTITUTO DE ADMINISTRAÇÃO. Inteligência Artificial: o que é, como funciona e exemplos. *In: Inteligência Artificial: o que é, como funciona e exemplos*. [S. l.]: FIA, 13 out. 2021. Disponível em: <https://fia.com.br/blog/inteligencia-artificial/>. Acesso em: 26 jun. 2022.

GLOBAL FACILITY FOR DISASTER REDUCTION AND RECOVERY (GRDRR) (Estados Unidos). Serviço Geológico dos Estados Unidos (USGS). **O Manual de deslizamentos - Um guia para a compreensão de deslizamentos**. Science for a changing world (USGS), Denver-colorado, ano 2008, p. 1 - 176, 2 fev. 2008. Disponível em: https://www.sindsegsc.org.br/arquivo/biblioteca/obra/6949_97222754. Acesso em: 02 de maio 2022.

HONÓRIO, Luiz Fernando. Inteligência Artificial: Conceitos e Aplicações. **Inteligência Artificial: Conceitos e Aplicações**, Rondônia, v. 1, n. 1, p. 234-246, 2 ago. 2010.

INCROPERA, Frank et al. **Fundamentos de Transferência de Calor e de Massa**. 6. ed. LTC: 2007.

MARCELINO, E. V. **Desastres Naturais e Geotecnologias: Conceitos Básicos**. Versão Preliminar. Santa Maria: INPE, 2007.

MANUAL DE NORMALIZAÇÃO DE PROJETOS DE TRABALHO DE GRADUAÇÃO FATEC SÃO BERNARDO DO CAMPO. **Material didático para utilização nos projetos de trabalho de graduação dos cursos de tecnologia em automação industrial e informática**. São Bernardo do Campo:Fatec, 2017.

MUELLER, John; MASSARON, Luca. **Aprendizado de Máquina Para Leigos**. 2. ed. Alta Books, 2021.

OMEGA ENGINEERING (Reino Unido). **How Do Infrared Thermometers Work?**. [S. l.], 2022. Disponível em: <https://www.omega.com/en-us/resources/infrared-thermometer-how-work>. Acesso em: 5 set. 2022.

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho científico**. 2. ed. Rio Grande do Sul: Universidade Feevale, 2013.

REDMON, J. et al. **You only look once**: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 779–788.

SOARES, Douglas Aparecido. **Desenvolvimento de um sistema portátil e de baixo custo de estação de controle em terra para VANTs**. 2017. 64 f. Dissertação (Mestrado em Ciência e Tecnologia da Computação) – Universidade Federal de Itajubá, Itajubá, 2017.

SOUZA, David José de. **Desbravando o PIC**: ampliado e atualizado para PIC16f628a. 8. ed. São Paulo: Érica, 2005.

SWIEŻEWSKI, Jędrzej. **YOLO Algorithm and YOLO Object Detection**. [S. l.]: AppSilon, 22 maio 2020. Disponível em: <https://appsilon.com/object-detection-yolo-algorithm/>. Acesso em: 28 jun. 2022.

THOMAZINI, Daniel; ALBUQUERQUE, B. U. Pedro. **Sensores Industriais: Fundamentos e Aplicações**. 4. ed. Érica: 2007.

REDMON, J. et al. **You only look once**: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2016. p. 779–788.

SILVA, Marcos. **Área da Superfície de um Corpo Humano**. Brasil Escola. Disponível em:<https://brasilecola.uol.com.br/matematica/area-superficie-um-corpo-humano.htm>. Acesso em: 28 mai. de 2022.

UFSC.LABORATÓRIO DE BIOMECÂNICA. **Termografia**. Disponível em: https://biomec.paginas.ufsc.br/?page_id=166#:~:text=O%20corpo%20humano%20tem%20uma,CHRISTIANSEN%3B%20GEROW%2C%201990. Acesso em: 5 nov. de 2022.

APÊNDICE A – PROGRAMAÇÃO DO ROBÔ MÓVEL

```
// Bibliotecas
#include <AFMotor.h> //Biblioteca da motor shield
#include <SoftwareSerial.h> //Inclui serial do bluetooth
#include <SparkFunMLX90614.h> // Biblioteca do sensor de temperatura

// Parametrização do módulo bluetooth e configuração da ponte H
SoftwareSerial bluetooth( A0, A1 ); //RX(Tx do módulo bluetooth), TX(RX do módulo
bluetooth)
AF_DCMotor motor1(1);
AF_DCMotor motor1(2);
AF_DCMotor motor1(3);
AF_DCMotor motor3(4);

//char comando;
#define FORWARD 2 //Valor da biblioteca 1
#define BACKWARD 1 //Valor da biblioteca 2

// sensor ultrassônico
const int trigger1 = 8;
const int echo1 = 2;
float d2;
float d1;
char movimentacao; //string para movimentação do robô
IRTherm therm;
String temperatura = "";
int valor_int_temp_1;
void setup() {

// modulo bluetooth
bluetooth.begin(9600); //Inicia a comunicação serial do bluetooth
Serial.begin(9600); //inicializa serial em 9600bps

// Configuração do sensor de temperatura
therm.begin();
therm.setUnit(TEMP_C);

// sensor ultrassônico
delay(200);
pinMode(trigger2,OUTPUT);
pinMode(echo2,INPUT);
digitalWrite(trigger2,LOW);
```

```

// velocidade dos motores
motor1.setSpeed(100); //0 a 255
motor2.setSpeed(100); //0 a 255
motor3.setSpeed(100); //0 a 255
motor4.setSpeed(100); //0 a 255
//motores iniciam parados
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}
void loop() {

    //sensor de temperatura
    therm.read();
    temperatura = (String(therm.object(), 2));

    valor_int_temp_1 = atoi(temperatura.c_str());

    //APP
    bluetooth.print (valor_int_temp_1);
    delay(2000);

    //sensor ultrassônica frente
    long t2;
    long d2;
    digitalWrite(trigger2,HIGH);
    delay(20);
    digitalWrite(trigger2,LOW);
    t2 = pulseIn(echo2,HIGH);
    d2 = t2/59;
    Serial.print("Distancia parte de frente:");
    Serial.print(d2);
    Serial.println("cm");
    while (bluetooth.available()) {
        movimentacao = bluetooth.read(); // armazena na variável comando a leitura do
bluetooth

//Movimentação
        if((movimentacao == 'F')){ //Movimentação para frente
            Frente();
        }
        else if(movimentacao == 'B'){ //Movimentação para trás
            Tras();
        }
        else if(movimentacao == 'E'){ //Movimentação para esquerda
            Esquerda();
        }
        else if(movimentacao == 'D'){ //Movimentação para direita
            Direita();
        }
    }
}

```

```

    }
    else {
        Parado();
    }
}
}
void Frente() {
    while(d2 >=20 || d2 ==0 || movimentacao==0){
        if (movimentacao == 'F') {
            motor1.run(FORWARD);
            motor2.run(FORWARD);
            motor3.run(FORWARD);
            motor4.run(FORWARD);
            delay(20);
            Serial.println(d1);
            Serial.println(movimentacao);
            break;
            delay(20);
        }
    }
    void Tras() {
        motor1.run(BACKWARD);
        motor2.run(BACKWARD);
        motor3.run(BACKWARD);
        motor4.run(BACKWARD);
        delay(100);
        Serial.println(d1);
        Serial.println(movimentacao);
    }

    void Esquerda() {
        motor1.run(FORWARD);
        motor2.run(FORWARD);
        motor3.run(BACKWARD);
        motor4.run(BACKWARD);
        delay(100);
        Serial.println(d1);
        Serial.println(movimentacao);
    }
    void Direita() {
        motor1.run(BACKWARD);
        motor2.run(BACKWARD);
        motor3.run(FORWARD);
        motor4.run(FORWARD);
        delay(100);
        Serial.println(d1);
        Serial.println(movimentacao);
    }
}
void Parado() {
    motor1.run(RELEASE);
    motor2.run(RELEASE);

```

```

motor3.run(RELEASE);
motor4.run(RELEASE);
delay(100);
Serial.println(d1);
Serial.println(movimentacao); }

```

APÊNDICE B – PROGRAMAÇÃO RECONHECIMENTO DE IMAGEM

```
# PARTE 1
```

```

#bibliotecas
from djitellopy import Tello
import numpy as np
import cv2
#import cv2.aruco as aruco
import time, os, pickle
import threading
""" Python program to Scan and Read a QR code"""
#from pyzbar.pyzbar import decode
"""Para realizar o Json"""
import requests
import json

```

```
#PARTE 2
```

```

#Importação dos modelos
net = cv2.dnn.readNet ("yolov3.weights","yolov3.cfg")

```

```
classes = []
```

```

with open ("coco.names", "r") as f: #verifica todos os nomes do arquivo
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
output_layers = [layer_names [i[0] - 1] for i in net.getUnconnectedOutLayers()]
colors = np.random.uniform (0,255,size = (len(classes), 3))

```

```
# PARTE 3
```

```
#conexão com o IP do drone
```

```
class Tello_2:
```

```

    def __init__(self):
        self._running = True
        #self.video = cv2.VideoCapture("udp://192.168.10.1:11111")
        self.video = cv2.VideoCapture(0) # webcam
        self.video.set(cv2.CAP_PROP_FRAME_WIDTH, 560)
        self.video.set(cv2.CAP_PROP_FRAME_HEIGHT, 420)

```

```

    def terminate(self):
        self._running = False
        self.video.release()
        cv2.destroyAllWindows()

```

```
#me.streamoff()
#me.end()
```

```
def infos(self):
    while self._running:
        global frame, key
        ret, frame = self.video.read()
        height, width, _ = frame.shape
```

```
# PARTE 4
```

```
#Detectando objetos
```

```
# O Blob é usado para extrair recursos da imagem e redimensioná-los.
```

```
blob = cv2.dnn.blobFromImage(frame,0.00392,(416,416),(0,0,0),True, crop =
False)
```

```
net.setInput (blob)
```

```
outs = net.forward (output_layers) #é uma matriz que contém todas as
informações sobre os objetos detectados, sua posição e a confiança na detecção.
```

```
#mostrando as informações na tela
```

```
class_ids = []
confidences = []
boxes = []
```

```
# PARTE 5
```

```
#Verificando o nível de confiança da detecção
```

```
for out in outs:
```

```
    for detection in out:
```

```
        scores = detection [5:]
        class_id = np.argmax(scores)
        confidence = scores [class_id]
```

```
        if confidence > 0.5: #quanto mais próximo de 1 maior a precisão da
detecção
```

```
            #posição dos objetos detectados
            center_x = int (detection [0] * width)
            center_y = int (detection [1] * height)
            w = int (detection [2] * width)
            h = int (detection [3] * height)
```

```
            # Coordenadas retangulares
```

```
            x =int (center_x - w / 2)
            y = int (center_y - h / 2)
```

```
            boxes.append ([x,y,w,h])
            confidences. append (float(confidence))
```

```

class_ids.append (class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# box: contém as coordenadas do retângulo ao redor do objeto detectado.
# label: é o nome do objeto detectado

font = cv2.FONT_HERSHEY_PLAIN
for i in range (len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str (classes[class_ids[i]])
        color = colors [i]

        cv2.rectangle(frame, (x,y), (x+w, y+h),color,3)
        cv2.putText(frame,label, (x,y+30), font, 3, color,3)

cv2.imshow("Camera", frame)

key = cv2.waitKey(1) & 0xFF
if key == 27:

    self.video.release()
    cv2.destroyAllWindows()

    rcvThread2 = t.terminate()
    rcvThread2.start()
    break

"INICIA DRONE"

#Inicia drone em modo command e liga câmera
#me = Tello()
#me.connect()
#time.sleep(1)
#me.streamon()
#inicia thread de leitura da câmera

t = Tello_2()
rcvThread = threading.Thread(target=t.infos)
rcvThread.start()

# Main loop
#while True:
#    # use 'q' to quit
#    #me.end()

```