

# **ARQUITETURA PARA NAVEGAÇÃO AUTÔNOMA UTILIZANDO DRONES**

Aluno(a): Matheus Mahnke

Orientador: Dalton Solano dos Reis

# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Requisitos
- Especificação
- Implementação
- Desafios
- Análise dos resultados
- Conclusões e sugestões

# Introdução

- Origem do drone
- Quadrotores
- Sensores
- Câmeras
- Processamento de imagem
- Dependência de servidor
- Plano de voo
- GPS

# Objetivos

- **Objetivo geral:**

Apresentar uma arquitetura de navegação baseada em GPS, aliado a imagens de câmera para o reconhecimento e desvio de obstáculos em voo, com processamento a bordo resultando em uma navegação totalmente autônoma.

# Objetivos

- **Objetivos específicos:**
  - I. Definir a arquitetura para a navegação independente de servidor
  - II. Detectar e desviar de obstáculos visíveis pela câmera
  - III. Executar um plano de voo baseado em GPS

# Fundamentação Teórica

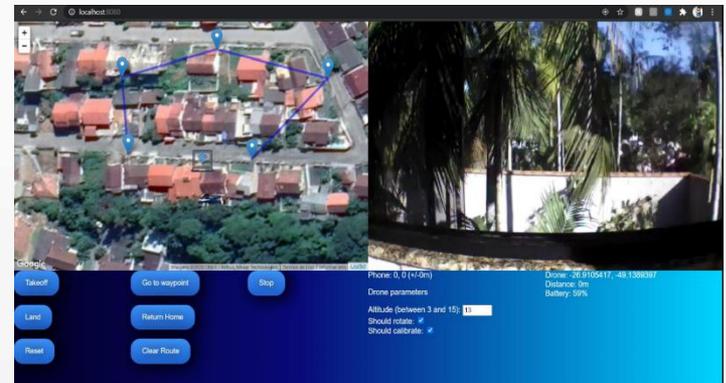
- Raspberry Pi 3 A+
- Pyparrot
- Lei dos cossenos
- MiDaS

# Trabalhos Correlatos

- **Título:** DRONE AUTÔNOMO: VIGILÂNCIA AÉREA DE ESPAÇOS EXTERNOS

Corrêa (2020)

Características	Corrêa (2020)
usa GPS	sim
processamento a bordo	não
desvio de obstáculos	não

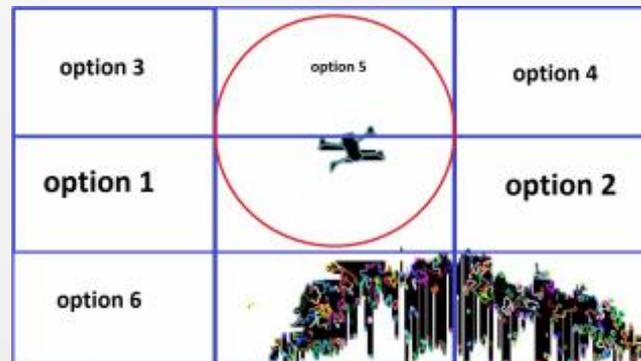


# Trabalhos Correlatos

- **Título:** A COMPUTER VISION BASED ALGORITHM FOR OBSTACLE AVOIDANCE (OUTDOOR FLIGHT)

Martins, Ramos e Mora-Camino (2018)

Características	Martins, Ramos e Mora-Camino (2018)
usa GPS	não
processamento a bordo	sim
desvio de obstáculos	sim



# Requisitos

## Requisitos funcionais:

- RF01: o drone deverá seguir um plano de voo baseado em GPS;
- RF02: o drone deverá possuir uma câmera frontal;
- RF03: o drone deverá possuir um sistema de estabilização com base em sensores;
- RF04: o processamento em voo deverá ser totalmente a bordo, sendo dependente apenas da conexão com GPS;
- RF05: o drone deverá desviar de objetos em voo.

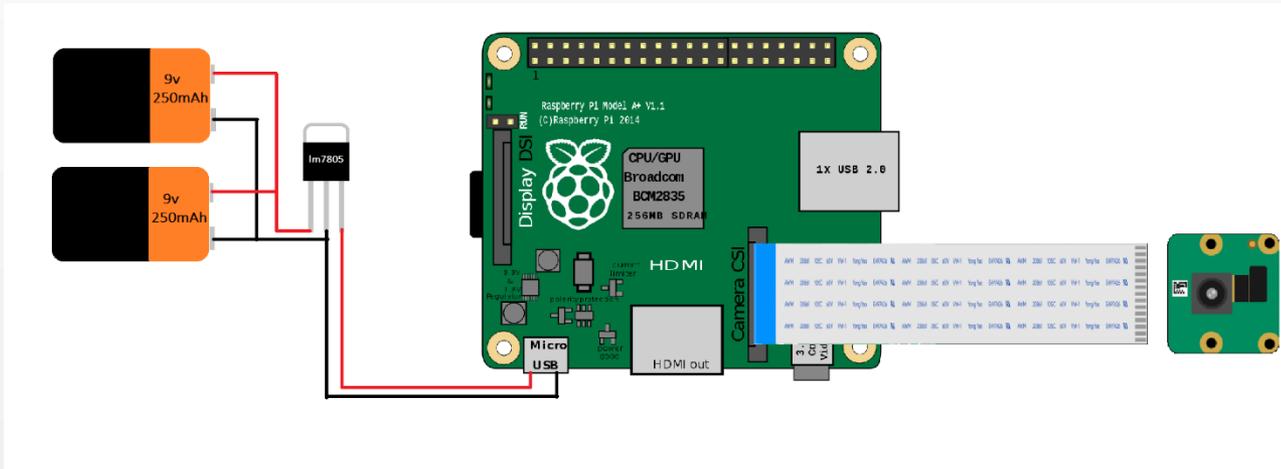
# Requisitos

## Requisitos não funcionais:

- RNF01: a detecção de obstáculos deve ser feita com o framework de aprendizado de máquina MiDaS;
- RNF02: o drone deverá possuir um Raspberry Pi III modelo A+ para processamento de imagem a bordo.

# Especificação

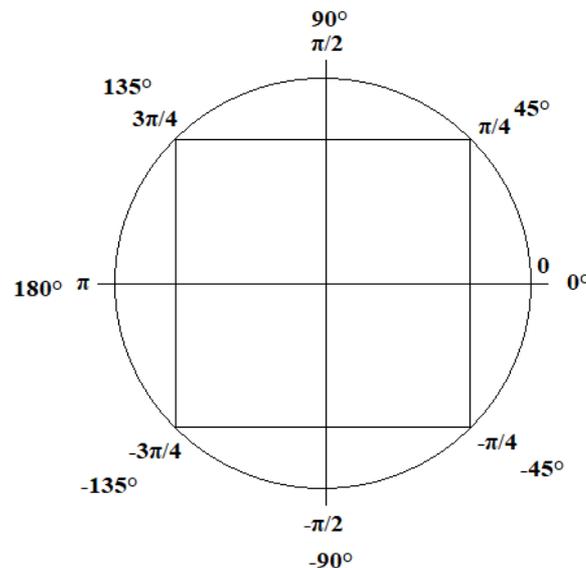
- Raspberry Pi 3 A+
- Baterias 9v 250mAh
- Câmera 5MP
- Regulador de tensão Lm7805



# Implementação

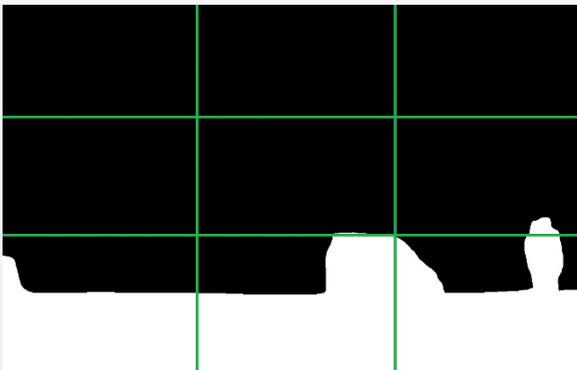
- Direcionamento para o destino

```
def get_yaw_angle_from_coordinates(latitude, longitude, latitude_dest, longitude_dest):  
    delta_x = latitude_dest - latitude  
    delta_y = longitude_dest - longitude  
    result = math.atan2(delta_y, delta_x)  
    return result
```

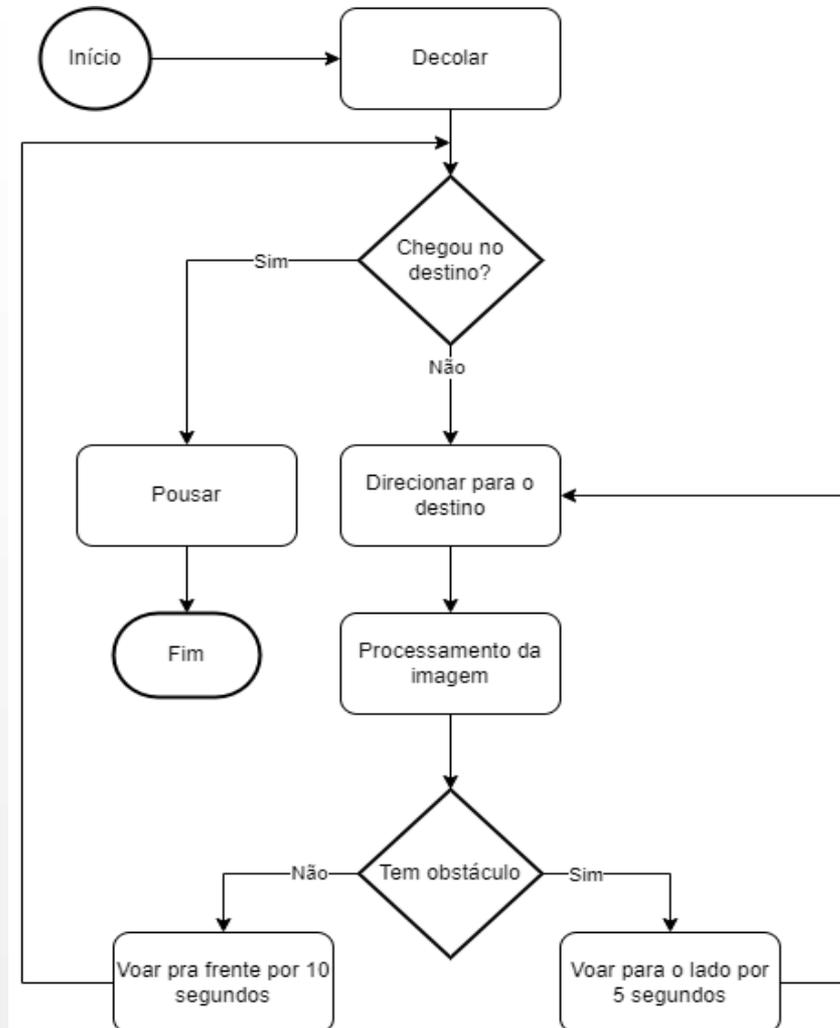


# Implementação

- Detecção e desvio de obstáculos



# Implementação



# Desafios

- Biblioteca para os comandos de voo
- Execução do algoritmo MiDaS no sistema operacional Raspibian
- Obter imagens da câmera em tempo real

# Análise dos Resultados

- Cenário de teste 1



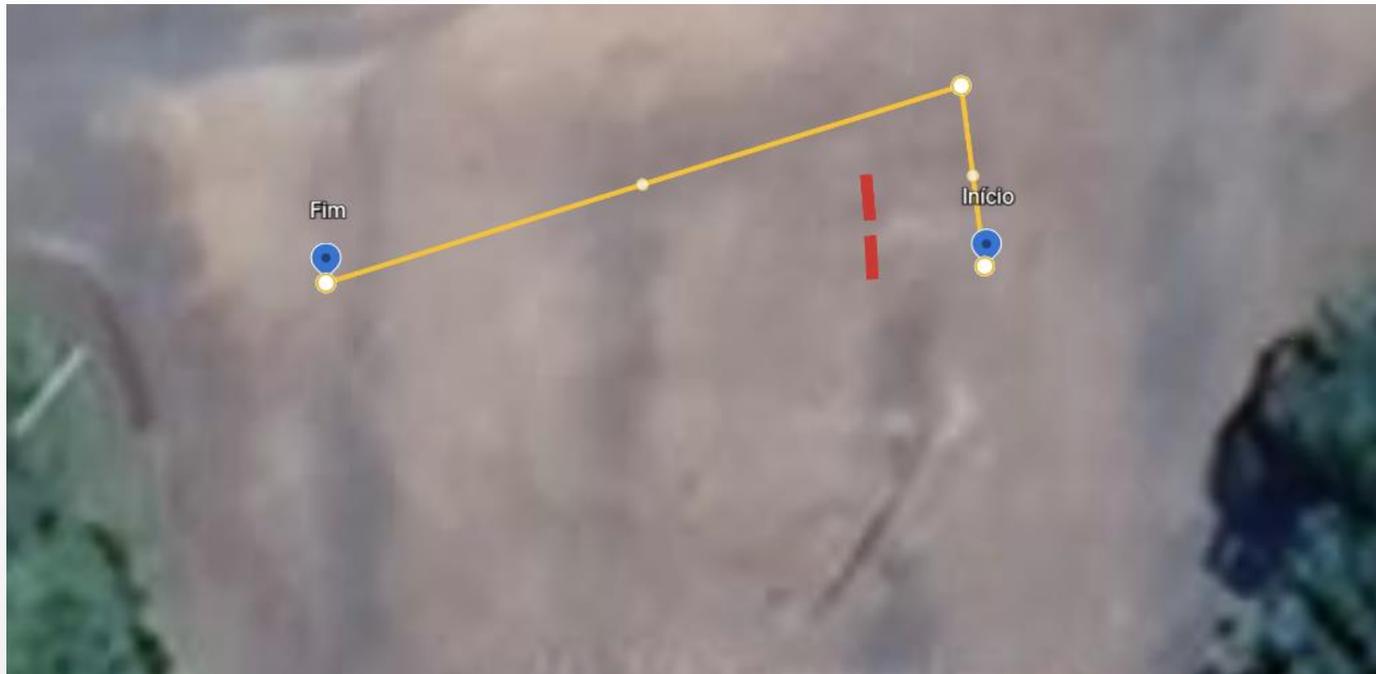
# Análise dos Resultados

- Cenário de teste 2



# Análise dos Resultados

- Cenário de teste 3



# Análise dos Resultados

- Cenário de teste 4



# Conclusões e Sugestões

- O drone conseguiu voar autonomamente em ambiente de testes.
- As baterias de 9v foram suficientes para fazer os cenários de testes.
- O algoritmo elaborado foi capaz de direcionar e pousar nas coordenadas definidas com um certo nível de arredondamento.
- A arquitetura foi capaz de desviar dos obstáculos mais próximos detectados.

# Conclusões e Sugestões

- Vibrações em voo.
- A detecção de objetos se mostrou sensível a iluminação e vibrações.
- O tempo de processamento do MiDaS no Raspberry Pi 3 A+ inviabiliza a utilização em cenários reais.

# Conclusões e Sugestões

## Sugestões de extensão ao trabalho:

- Pesquisar um algoritmo de detecção de obstáculos com baixo custo de processamento, para o reconhecimento de obstáculos em tempo real.
- Encontrar novas alternativas de acoplar hardware adicional ao drone, para que tenha baixo peso e pouca oscilação.
- Criar uma interface para permitir alterar parâmetros internos das rotinas.

# Demonstração

## Demonstração prática:

- Direcionamento para as coordenadas de destino.
- Algoritmo de Identificação e desvio de obstáculo no Raspberry Pi.