

# Sistema de Alarme Assistido com Robot

Grupo 26:

- 93143 - Miguel Nabais
- 93168 - Ricardo Febra

# Breve Introdução do Projeto

- Objetivo
- Funcionalidades

Este projeto tem o propósito de permitir o controlo de um robot num dado espaço quando há deteção de presença numa dada divisão. Desta maneira, numa app, o utilizador poderá confirmar a identidade do intruso, já que terá acesso a uma câmara presente no robot, bem como a possibilidade de alertar o intruso produzindo som.

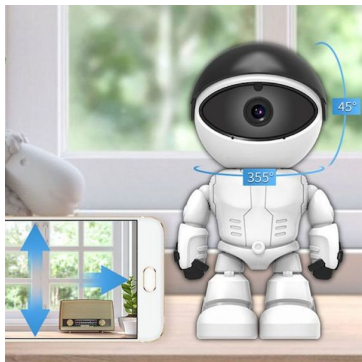
A deteção de movimento realiza-se através de um sensor de ultrasom conectado a uma base de dados em tempo real, Firebase. Comunicação base: LoRa.

O robot, quando solicitado pelo utilizador, pode ser ativado através da aplicação e, por sua vez, controlado. Comunicação base: WiFi.

# O que já existe



ESCAM PT205-R 2MP Wifi



Robot estático com o objetivo de passar despercebido num ambiente de casa. Através de uma aplicação é possível controlar a posição da cabeça, onde se encontra incorporada uma câmara.



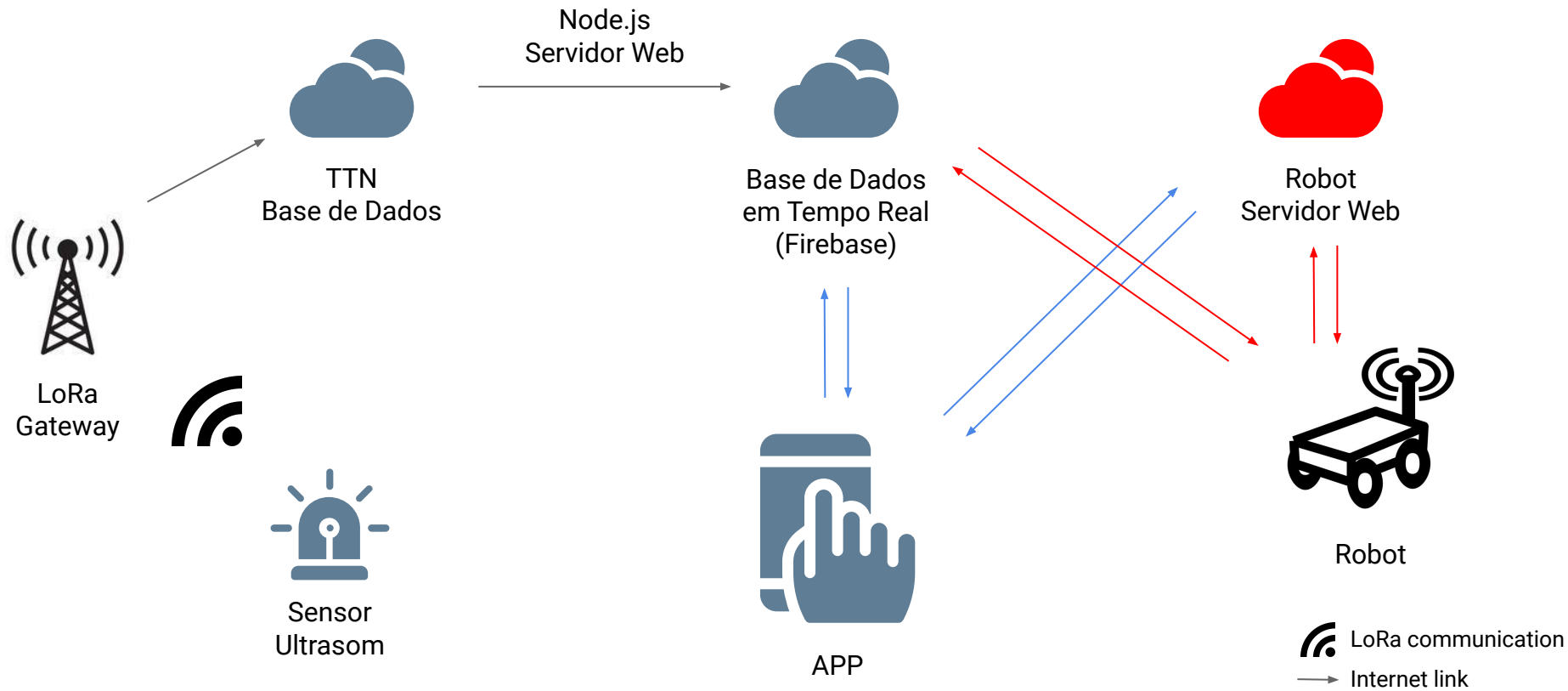
Robot de vigilância ligado a rede privada 5G. Responsável por complementar tarefas de vigilância física do campus da Universidade de Vigo, Espanha. O robot é controlado remotamente através de uma sala de controlo montada dentro da universidade, com capacidade para visão térmica, vídeo de alta definição e visão panorâmica 360°.

ROBOTVIGIL: Robot Vigilante

Robot orientado para espaços interiores, capaz de recolher e enviar informação de sensores e vídeo de várias câmaras em tempo real. Oferece a possibilidade de ser operado remotamente ou instruído para realizar rondas de forma autónoma. O sistema é capaz de cooperar com outros robots e humanos.

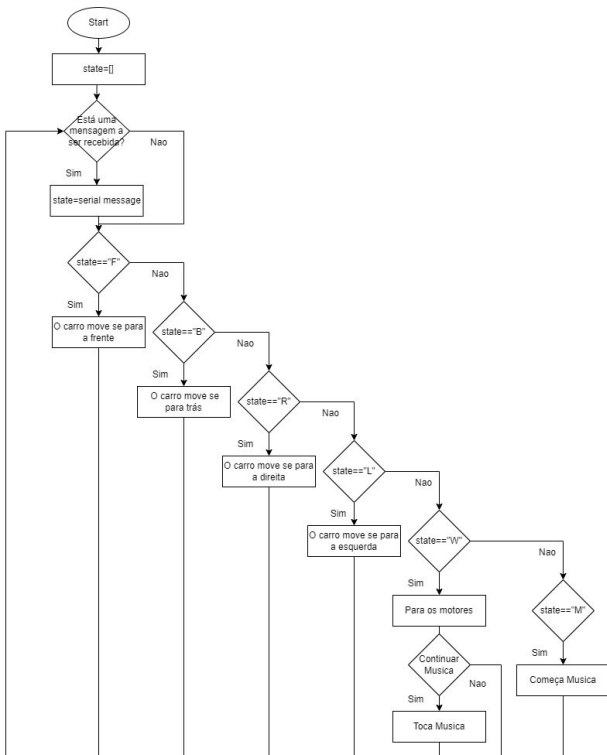


# Arquitetura do Projeto

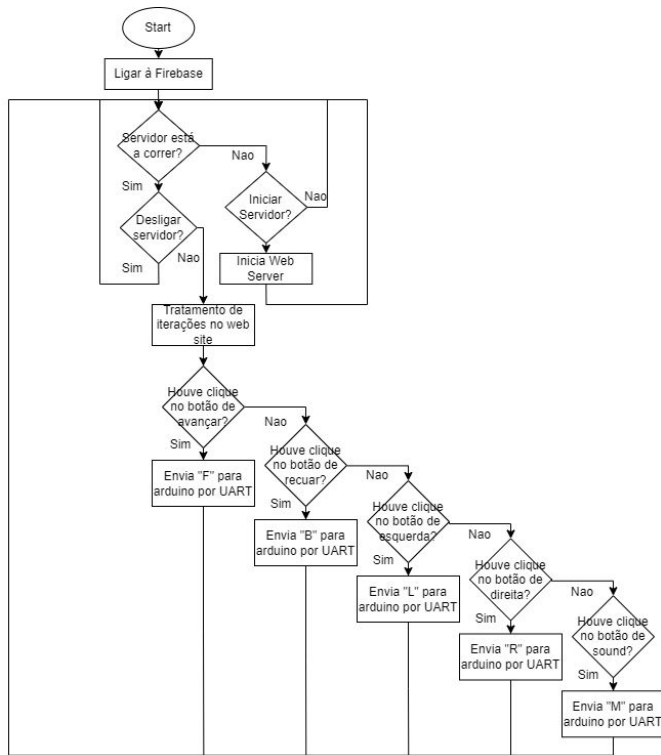


# Algoritmos Implementados

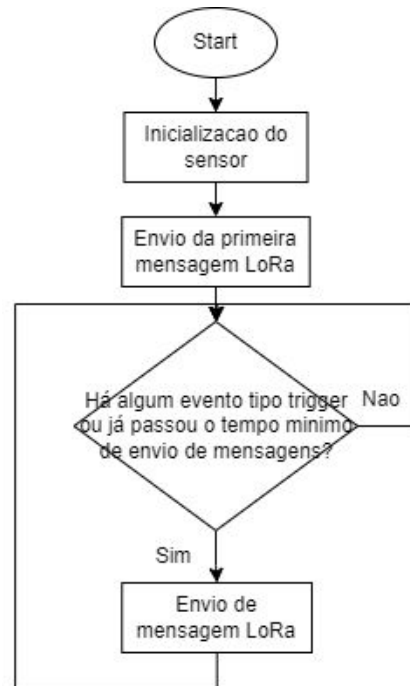
## Arduino Uno



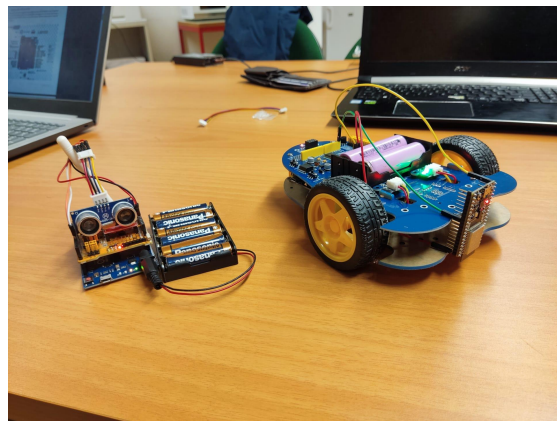
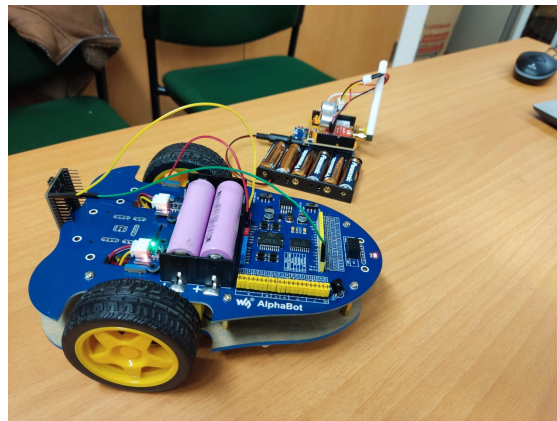
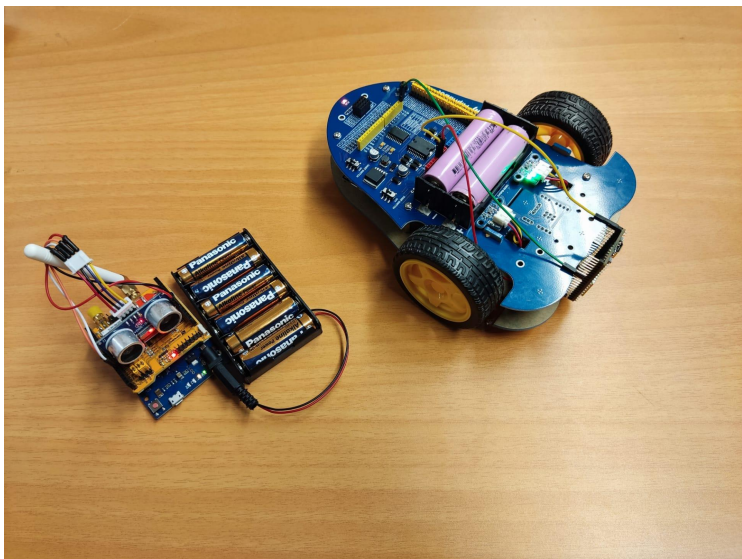
## ESP32



## Arduino Leonardo



# Montagem



# Bases de Dados

TTN (The Things Network):

- Base de dados (temporário) para dispositivos que comunicam via LoRa. Utilizado como intermediário entre o dispositivo LoRa e a base de dados Firebase. Recebe dados vindos do sensor de ultrasom - Variável: alarm\_status.

Firebase:

- Base de dados que resume todas as variáveis relacionadas com o projeto, que necessitem de ser conhecidas entre dispositivos (e.g., app <-> robot) - Variáveis: alarm\_status, shut\_down, start\_server.

# APP Descrição

A aplicação desenvolvida comprime-se em duas páginas/atividades. Primeira página:

- Dar a conhecer ao utilizador o status do alarme - sensor de ultrasom. Se for detetado movimento a aplicação exibe “ALERT!ALERT!”, e, caso contrário, exibe “Nothing to Report”.
- Oferecer ao utilizador a possibilidade de ligar o servidor do robot. Assim, sempre que pretender é possível ligar-se ao robot, visualizar a stream e controlá-lo.

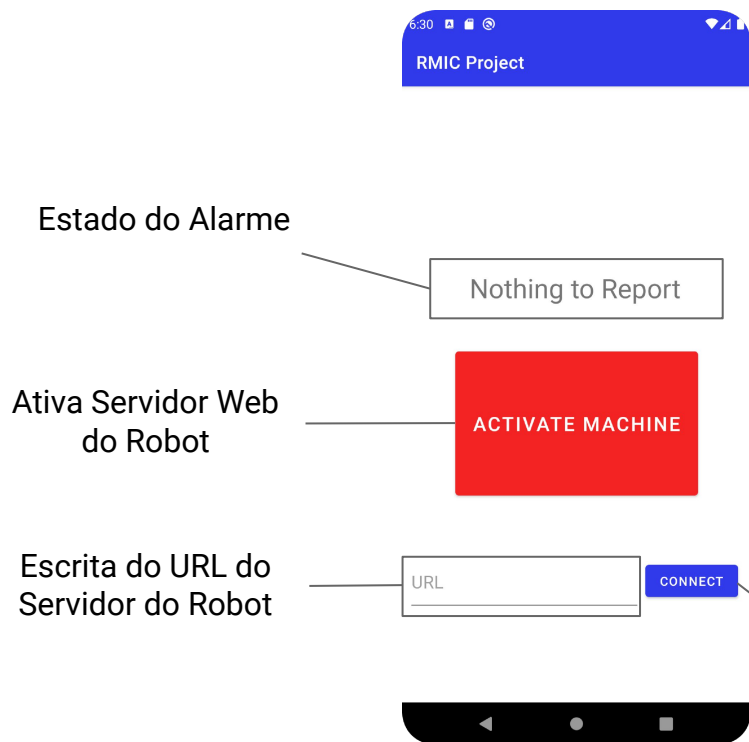
Segunda página:

- Exibe o servidor Web disponibilizado pelo robot. Desta forma é possível não só visualizar o que o robot dispõe à sua frente como controlá-lo. Também é oferecido ao utilizador a possibilidade de ligar/desligar o *buzzer* incorporado no robot.
- Permite ao utilizador controlar o servidor Web, na medida em que este o pode desligar, colocando o robot outra vez num estado em que apenas escuta a Firebase até receber instruções que digam o contrário, nomeadamente a nova ativação do servidor por pedido do utilizador (na primeira página).
- Realizar o *refresh* da página Web onde o servidor Web se encontra aberto, não sendo necessário retornar à primeira página para o fazer.



# Composição da APP

## Primeira Página:



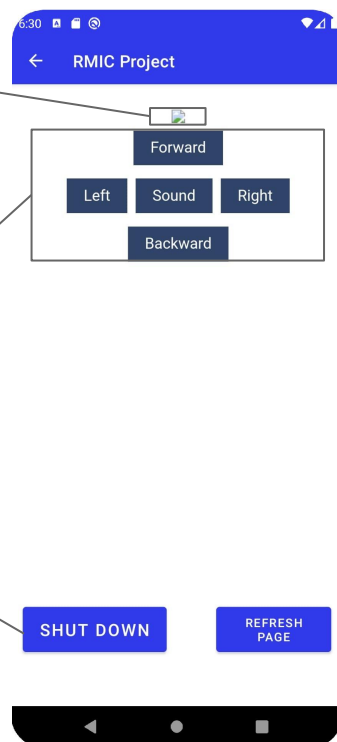
## Segunda Página:

**Stream da Câmera a Bordo do Robot**  
(Devido a problemas na placa do ESP32-CAM não foi possível simular o vídeo. Os passos finais do projeto foram realizados usando um segundo ESP32, mas sem possibilidade de incorporação de câmera. É de notar que todo o código se encontra válido para apresentação com o ESP32-CAM.)

Controlos do Robot

Desativa Servidor Web do Robot

Abre ligação ao servidor e encaminha utilizador para a segunda página



# Conclusão

O sistema implementado teve bastantes dificuldade na sua implementação, quer devido ao seu grande consumo de energia quer na complexidade do mesmo originada pela grande quantidade de ligações presentes no mesmo.

O código implementado no ESP32-cam permitiu-nos retirar um sensor do circuito, já que o mesmo hospeda um web server. Sendo o controlo do carro feito a partir do web site hospedado no web server, a comunicação com o Arduino Uno (de maneira a ser feito o controlo dos motores) pode ser feita diretamente entre o ESP32-cam e o arduino uno.

Apesar disto, ainda há algumas funcionalidades e melhorias que podem ser adicionadas ao sistema exposto:

- Regular a velocidade do carro a partir da aplicação
- Controlo do carro fora da rede local (fazer o encaminhamento de porta através do router da rede local, por forma a abrir o servidor criado pelo ESP32 num IP público)