

CESAR SCHOOL

ARTHUR FREIRE, GABRIEL RODRIGUES, JOÃO MARCELO

SMART HOME:

Sistema de Monitoramento, Automação e Eficiência Energética com ESP32, MQTT e Web
Dashboard

Recife

2025

INTRODUÇÃO

O presente relatório apresenta o desenvolvimento do **Smart Home**, um sistema de automação residencial acessível, voltado para residências mais simples. O projeto integra tecnologias de IoT, comunicação em nuvem, detecção de presença e alertas remotos, com o objetivo de **aumentar a segurança domiciliar e reduzir o desperdício de energia elétrica**.

A motivação do projeto surgiu da necessidade de oferecer soluções inteligentes a um público que, tradicionalmente, não têm acesso aos altos custos de dispositivos comerciais de smart home. Dessa forma, buscou-se implementar um sistema de baixo custo, fácil replicação e alto impacto social, utilizando microcontroladores, sensores, protocolos de comunicação e um dashboard web responsivo.

Os principais objetivos deste trabalho são:

- Desenvolver um sistema de detecção de presença com acionamento automático de alarme sonoro e visual;
- Disponibilizar um painel web para monitoramento remoto em tempo real;
- Realizar comunicação em nuvem usando broker MQTT;
- Permitir controle de iluminação interna (sala e quarto) via interface gráfica;
- Implementar alertas por e-mail em caso de intrusão;
- Criar mecanismo de monitoramento de tempo das luzes acesas, contribuindo para **eficiência energética**.

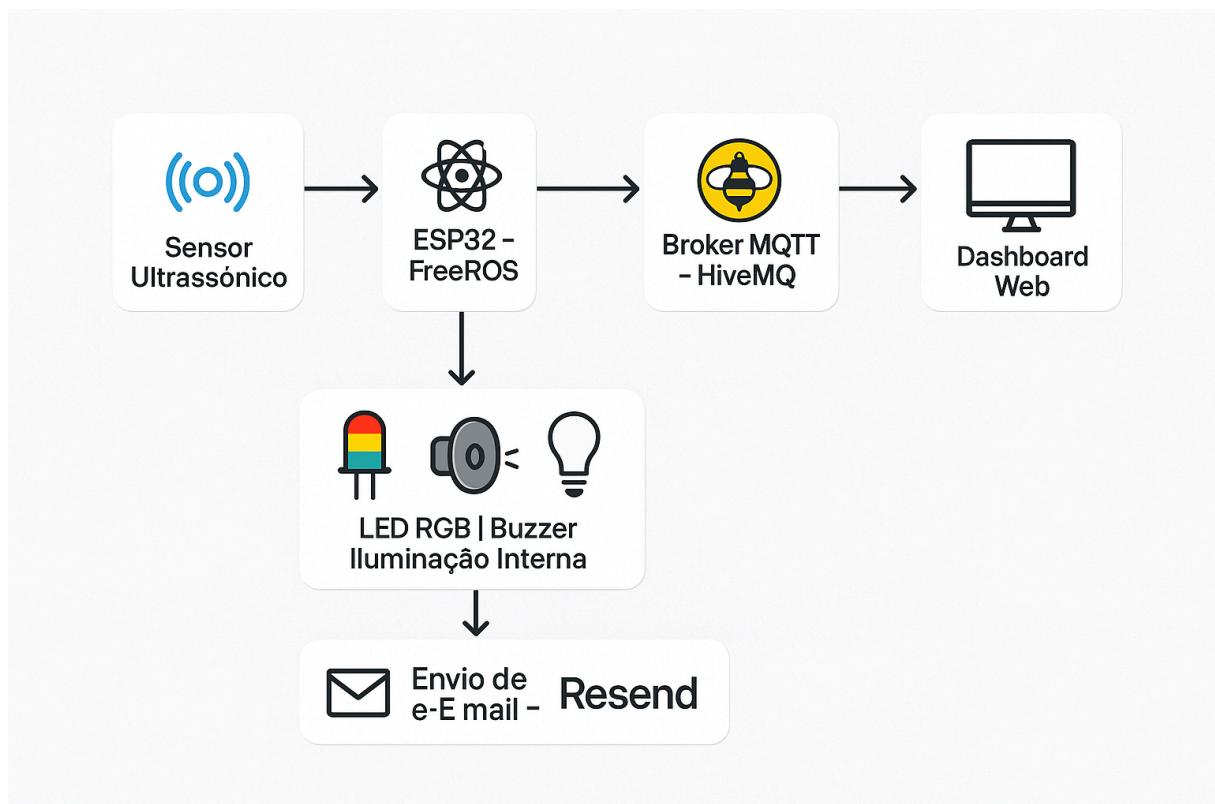
METODOLOGIA

1.1 Arquitetura Geral do Sistema

O Smart Home segue uma arquitetura IoT distribuída, formada pelos seguintes elementos:

- **ESP32** como controlador principal (firmware executado com FreeRTOS);
- **Sensor ultrassônico** para detecção de objetos ou pessoas próximas da residência;
- **LED RGB e buzzer** para sinalização local do estado do alarme;
- **Lâmpadas inteligentes (representadas por LEDs RGB)** nos ambientes interno (sala e quarto);
- **Comunicação via Wi-Fi**, integrando o dispositivo ao broker MQTT (HiveMQ Cloud);
- **Dashboard Web** desenvolvido em Next.js para monitoramento e comandos;
- **Serviço de e-mail** (Resend) para envio de alertas e notificações ao usuário.

1.2 Diagrama do Sistema



1.3 Lista de Hardware

- **ESP32** (modelo ESP-WROOM-32);
- Sensor Ultrassônico **HC-SR04**;
- LED RGB principal (estado do alarme);
- Buzzer PWM;
- Botão físico para acionamento manual;
- LEDs RGB adicionais (representando cômodos);
- Protoboard, jumpers e fonte de alimentação 5V.

1.4 Lista de Software

- Firmware em C++/Arduino com FreeRTOS;
- Biblioteca PubSubClient para MQTT;
- Plataforma HiveMQ Cloud como broker;
- Dashboard em Next.js + TailwindCSS;
- API de alertas usando Resend;
- Pacotes adicionais: WiFi.h, WiFiClientSecure.h, ledc PWM driver.

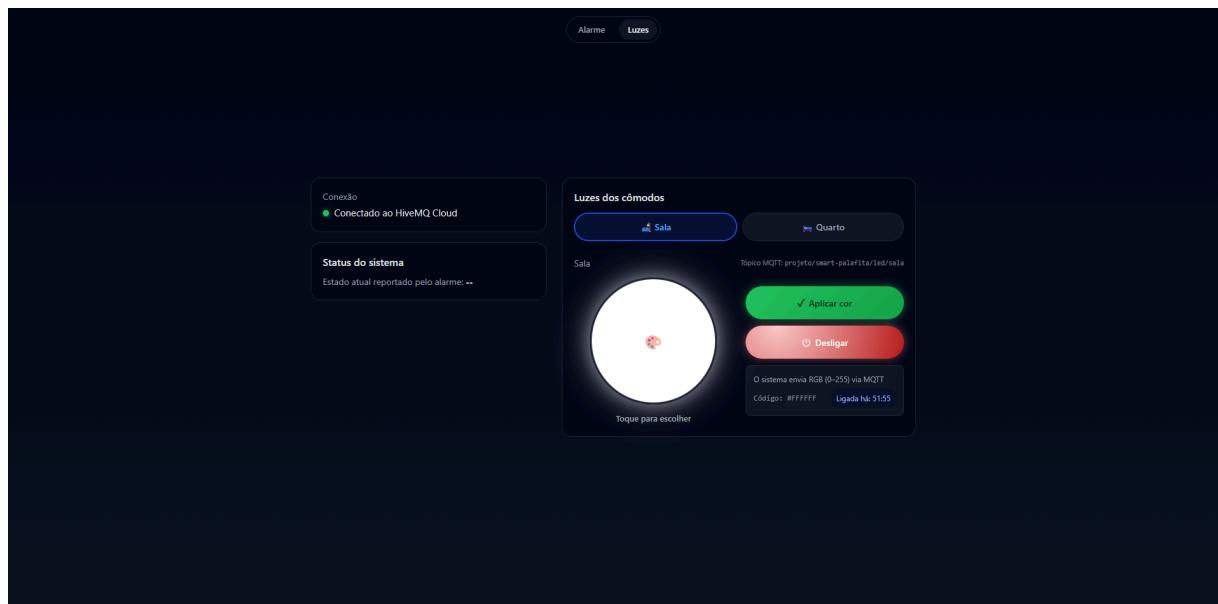
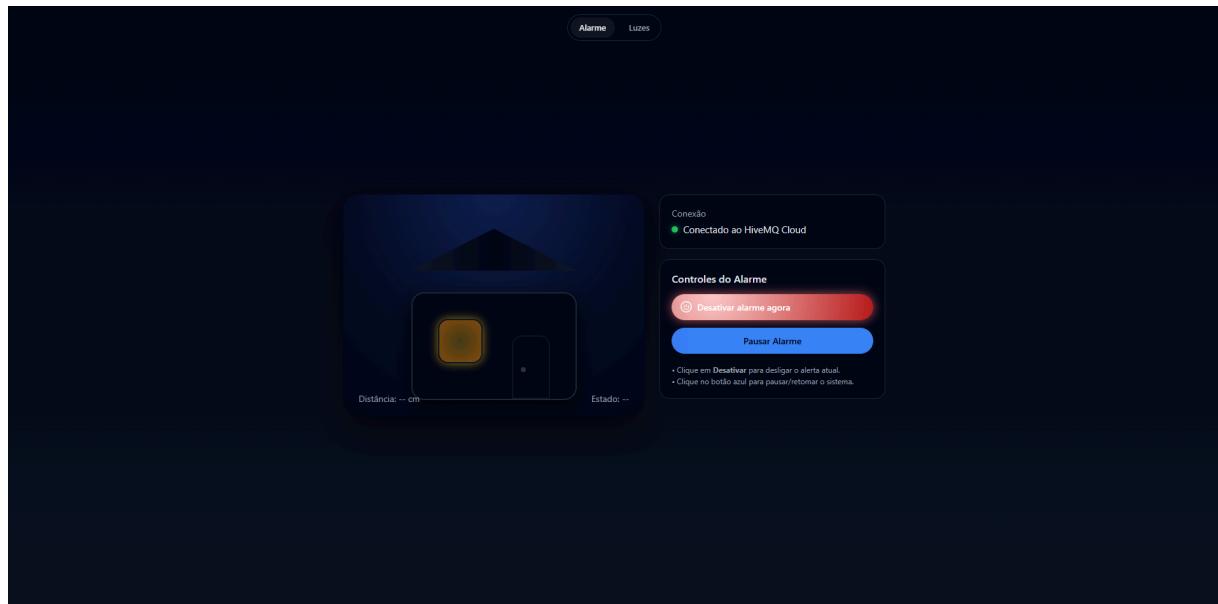
1.5 Fluxo de Comunicação MQTT + Wi-Fi

O fluxo de dados segue os seguintes passos:

1. O **ESP32** conecta-se à rede Wi-Fi e estabelece sessão segura com o **broker MQTT**;
2. O sensor ultrassônico realiza leituras periódicas e publica no tópico:
 - projeto/home-security/sensor/medida;
3. O estado atual do sistema é publicado em:
 - projeto/home-security/sensor/estado;
4. O dashboard envia comandos através dos tópicos:
 - projeto/home-security/comandos;
 - projeto/home-security/led/sala;
 - projeto/home-security/led/quarto;
5. Caso o **alarme seja disparado**, o ESP32 envia atualização imediata ao broker e aciona envio de e-mail pelo servidor web;
6. A interface web recebe os dados publicados e atualiza o painel instantaneamente.

RESULTADOS

1.1 Prints do Dashboard



The screenshot shows the HiveMQ Cloud Web Client interface. On the left, there's a sidebar with 'ORGANIZATIONS' (CESAR), 'CLUSTERS OVERVIEW' (Free #1 selected), and a 'Documentation' link. The main area has tabs for 'Overview', 'Access Management', 'Web Client' (selected), and 'Getting Started'. Under 'Web Client', the 'Connection Settings' section shows a form with 'Username' (admin) and 'Password' (redacted). Below it is a 'Topic Subscriptions' section with a table for setting QoS levels (QoS: 0 or 1). The right side displays a scrollable list of messages from the 'projeto/smart-palafita/sensor/*' topic, categorized by type (ALERTA, OK, PAUSADO) and value (e.g., '41.0', '129.0').

Evidência de Funcionamento: https://youtube.com/shorts/B_cLTMwnSl8?feature=share

CONCLUSÃO

A implementação do **Smart Home** permitiu validar a proposta de um sistema de automação residencial acessível, combinando tecnologia, segurança e eficiência energética. Entre os principais desafios enfrentados, destacam-se:

Ajustes de sincronização entre tarefas utilizando **FreeRTOS**;

Estabilidade da comunicação **MQTT** em ambiente real;

Criação de um **dashboard intuitivo e responsivo**;

Integração entre **firmware, backend e serviço de e-mail**.

Domínio de multitarefas em microcontroladores;

Integração com serviços externos (**MQTT, Resend**);

Aplicação de conceitos de IoT e arquitetura distribuída;

Desenvolvimento de sistemas com impacto social real.

Monitoramento automático do tempo em que luzes permanecem acesas;

Inclusão de sensores adicionais (**futuro**, ex: fumaça, gás, inundação, temperatura);

Dessa forma, conclui-se que o Smart Home **alcança seus objetivos**, oferecendo segurança acessível e economia de energia, podendo ser replicado em diferentes comunidades e contextos sociais.

APÊNDICES

Os apêndices deste projeto, incluindo todos os códigos-fonte, configurações do broker MQTT, rotas da API, diagramas e demais materiais complementares, encontram-se disponíveis no repositório oficial do projeto no GitHub. Para acesso completo, basta consultar o link:
<https://github.com/gabrielgrm/smart-home>