UNIVERSIDADE FEDERAL DE ITAJUBÁ

ARTHUR MORAES MARQUES DOS SANTOS
BRENO GABRIEL ALVES DE SOUZA
GUSTAVO FERNANDES GONÇALVES DE LIMA
MARCO ANTÔNIO CARNEIRO VILELA
VICTOR AUGUSTO DE AQUINO SILVÉRIO

Projeto Final - ECAE00

Projeto de controle de acesso, temperatura e iluminação de um ambiente interno, de forma automática e manual.

ITAJUBÁ/MG

2025.1

ARTHUR MORAES MARQUES DOS SANTOS - 2025001389
BRENO GABRIEL ALVES DE SOUZA - 2025013746
GUSTAVO FERNANDES GONÇALVES DE LIMA - 2025002985
MARCO ANTÔNIO CARNEIRO VILELA - 2025005299
VICTOR AUGUSTO DE AQUINO SILVÉRIO - 2025016677

Projeto Final - ECAE00

Projeto de controle de acesso, temperatura e iluminação de um ambiente interno, de forma automática e manual.

Projeto de Controle e Automação com ESP32, apresentado como requisito parcial para aprovação na disciplina *Introdução à Engenharia e ao Método Científico – ECAE00*, do curso de Engenharia de Controle e Automação da Universidade Federal de Itajubá (UNIFEI).

Trabalho desenvolvido sob a orientação dos professores Luiz Lenarth G. Vermaas e Jeremias Barbosa Machado.

ITAJUBÁ/MG

2025.1

SUMÁRIO

1 INTRODUÇÃO

Este trabalho tem como objetivo documentar o desenvolvimento de um projeto de automação residencial que integra controle de acesso físico, monitoramento ambiental e detecção de presença, utilizando o microcontrolador ESP32 como unidade central de processamento. A proposta surge diante da necessidade crescente por sistemas inteligentes capazes de gerenciar de forma eficiente o acesso a ambientes restritos, além de monitorar suas condições internas e otimizar o uso de recursos como iluminação e ventilação.

A solução proposta visa combinar segurança, conforto e eficiência energética, por meio de uma interface intuitiva para controle e supervisão dos elementos automatizados. O sistema será projetado para atuar de maneira autônoma e/ou manual, permitindo o acionamento remoto de mecanismos, conforme as condições detectadas no ambiente.

Para a prototipagem do sistema, os seguintes componentes eletrônicos foram selecionados:

- ESP32 (microcontrolador);
- Leitor RFID MFRC522;
- Sensor de temperatura e umidade DHT11;
- Sensor ultrassônico HC-SR04;
- Display LCD 16x2 com interface I2C;
- Servo motor;
- Buzzer;
- LED indicador;
- Ventoinha;
- Transistor TIP120;
- Diodos;
- Cabos de conexão (jumpers);

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um sistema de automação residencial inteligente que integre controle de acesso, monitoramento do ambiente (temperatura e umidade) e acionamento de dispositivos, visando proporcionar maior segurança, conforto e economia de energia, por meio de uma interface intuitiva que permita o gerenciamento remoto dos recursos do ambiente. Trabalhando não somente com comunicação local, integrando IoT e microssistemas de automação.

2.2 Objetivos Específicos

- Realizar o levantamento dos requisitos e planejamento das funcionalidades do sistema;
- Projetar e montar fisicamente o circuito eletrônico do protótipo;
- Projetar e construir maquete representativa;
- Desenvolver o código de controle para o ESP32, integrando os sensores e atuadores;
- Realizar testes funcionais e implementar ajustes para otimização do desempenho do sistema;

3 Pesquisa & Planejamento

O desenvolvimento do projeto teve início com a realização de um brainstorm em conjunto com os demais membros do grupo, com o objetivo de explorar possíveis ideias a partir dos componentes disponibilizados no kit inicial fornecido pela UNIFEI. Após uma análise crítica das possibilidades, concluiu-se que os componentes básicos disponíveis seriam insuficientes para a implementação de um projeto com o nível de complexidade e inovação esperado para um projeto final de disciplina.

Diante disso, optou-se pela utilização de componentes adicionais previamente adquiridos por um dos integrantes do grupo, bem como pela aquisição de novos dispositivos, com investimento pessoal do mesmo membro, visando complementar o escopo do projeto e permitir maior abrangência funcional.

Com a ideia-base do projeto definida, foi solicitada uma orientação formal ao professor responsável, para validação e sugestões de melhorias e incrementos. Após a incorporação das recomendações recebidas, a equipe iniciou o desenvolvimento da lógica de funcionamento do sistema.

3.1 Lógica de funcionamento interno

A primeira etapa envolveu a implementação da lógica de controle interno, com foco nos sistemas de iluminação e ventilação automáticas. Inicialmente, a lógica foi implementada de forma local, executando-se apenas internamente no microcontrolador, sem interface de controle externo. Posteriormente, foi integrada uma interface remota por meio de um servidor web embarcado no próprio ESP32, permitindo o gerenciamento dos dispositivos via rede local.

Na lógica de funcionamento, foi definido que a iluminação só poderá ser ativada se for detectada a presença de um usuário no ambiente, com um tempo de permanência mínima de 5 segundos antes do acionamento automático da luz. A ventilação, por sua vez, pode ser acionada tanto manualmente, via interface web, quanto automaticamente, caso a temperatura do ambiente ultrapasse um valor pré-estabelecido. Para evitar oscilações constantes no acionamento da ventoinha devido a variações mínimas de temperatura, foi implementada uma zona morta, na qual a ventoinha permanece ligada até que a temperatura desça abaixo de um segundo valor limiar.

3.2 Lógica de Acesso à sala

Com a lógica de controle interno consolidada, a etapa seguinte foi o desenvolvimento do sistema de controle de acesso, utilizando um leitor RFID MFRC522 e tags RFID previamente cadastradas. A autenticação é feita com base no UID de cada tag, armazenado no código-fonte. O comportamento programado define que, ao apresentar um cartão autorizado pela primeira vez, o mecanismo de abertura da porta é ativado, mantendo-a aberta até que o mesmo cartão seja apresentado novamente, sinalizando o encerramento do uso do ambiente.

Para aumentar a segurança, foi implementada uma restrição adicional: caso outro cartão autorizado seja apresentado enquanto a porta estiver aberta por um usuário anterior, será exibida uma mensagem de erro informando que a sala já está em uso, e o acesso será negado. Essa medida visa impedir que um segundo usuário externo feche a porta inadvertidamente, dado que o leitor RFID está instalado do lado de fora da sala, impossibilitando o acesso direto por quem já está no interior do ambiente.

3.3 Finalização da lógica de funcionamento

Após a definição do escopo, validação com o professor e desenvolvimento incremental da solução, a lógica de funcionamento do sistema foi finalizada conforme descrito a seguir:

Controle de Acesso (RFID):

A entrada no ambiente é controlada por um leitor RFID. O sistema reconhece tags previamente cadastradas por meio de seus UIDs. Ao apresentar uma tag autorizada pela primeira vez, a porta é destravada via servo motor. A mesma tag deve ser apresentada novamente para que a porta seja travada. Caso outra tag autorizada seja apresentada enquanto o ambiente já estiver em uso, o sistema rejeita a solicitação e exibe uma mensagem de erro, indicando que a sala está ocupada, reforçando a segurança do acesso.

Detecção de Presença:

Um sensor ultrassônico é utilizado para identificar a presença de pessoas no ambiente. A presença contínua por mais de 5 segundos aciona automaticamente a iluminação,

garantindo que o sistema não seja ativado por passagens rápidas ou movimentos momentâneos.

Iluminação:

A luz pode ser ativada automaticamente com base na presença detectada ou manualmente via interface web hospedada no próprio ESP32. A iluminação é desativada automaticamente quando a ausência é detectada, respeitando um tempo mínimo de inatividade.

Ventilação:

A ventoinha pode ser acionada manualmente, por meio da interface web, ou de forma automática com base nas leituras do sensor de temperatura e umidade (DHT11). Quando a temperatura ultrapassa um valor pré-definido, o sistema ativa a ventilação.
Para evitar o acionamento/desligamento frequente causado por pequenas oscilações térmicas, foi implementada uma *zona morta*: a ventoinha permanece ligada até que a temperatura caia alguns graus abaixo do valor de ativação.

Recomenda-se que esses limiares sejam ajustados conforme o clima do dia. Para testes em ambiente controlado, sugere-se configurar o limiar superior em cerca de 5 a 6 °C acima da temperatura ambiente. Já em uso real, recomenda-se um valor entre 8 e 12 °C, proporcionando maior tolerância térmica. O limiar inferior, por sua vez, deve ser ajustado para cerca de 2 a 3 °C acima da temperatura ambiente, garantindo que a ventilação só seja acionada quando houver um aumento real da temperatura do ambiente.

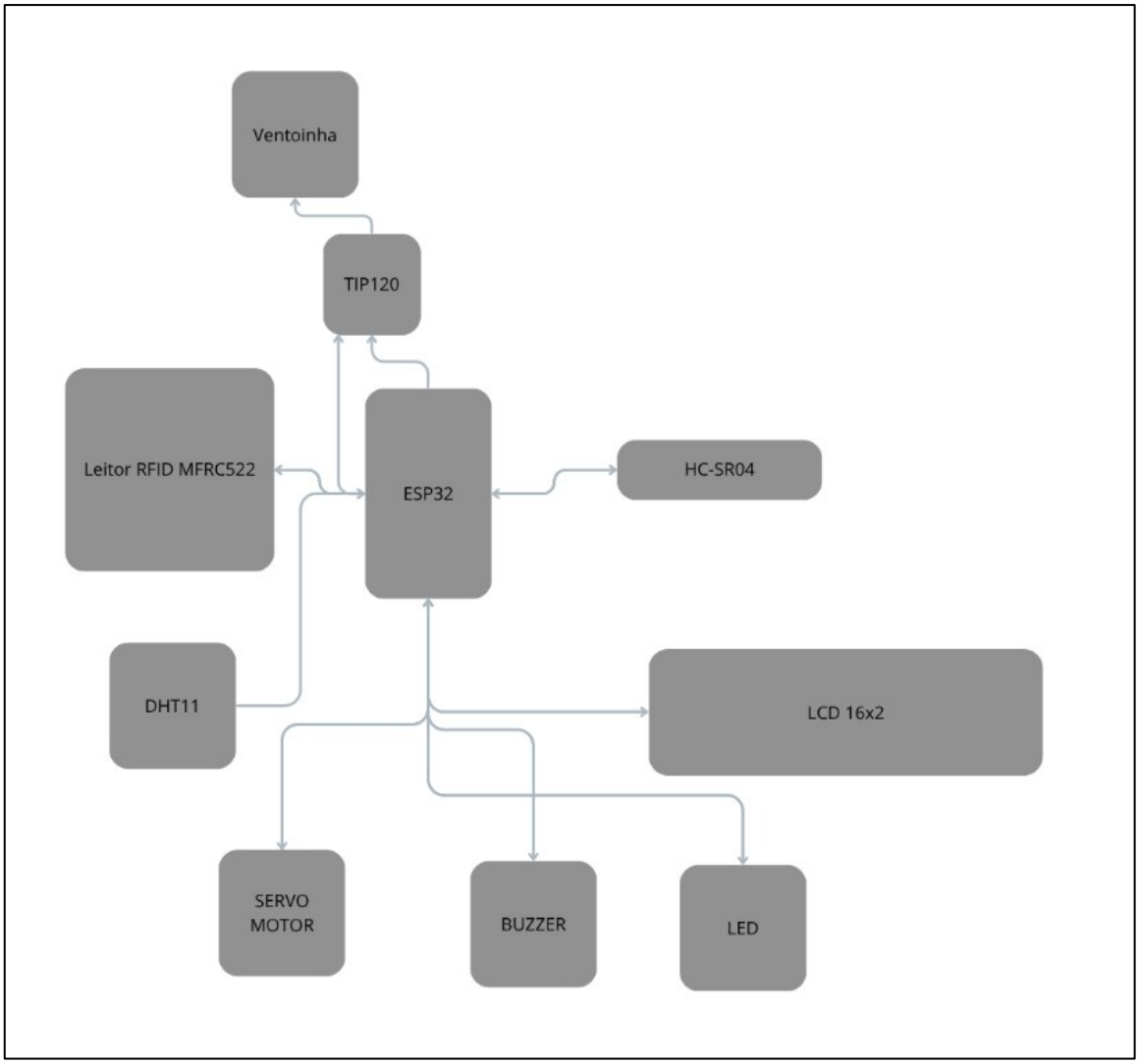
Interface Web (Servidor ESP32):

Um servidor web embarcado no ESP32 permite o controle remoto da iluminação e da ventilação. A interface apresenta o estado atual dos dispositivos e permite a interação do usuário em tempo real, mantendo a usabilidade e acessibilidade do sistema.

Essa lógica garante que o sistema opere de forma eficiente, segura e autônoma, com possibilidade de intervenção manual quando necessário, proporcionando conforto e economia de energia no ambiente monitorado.

3.4 Diagrama de Blocos

Para a demonstração simplificada das conexões, um diagrama de blocos baseado na lógica do projeto foi realizado, exibido abaixo:



4 Desenvolvimento

O desenvolvimento prático do projeto teve início com a seleção criteriosa dos componentes eletrônicos, considerando suas características técnicas, compatibilidade com o microcontrolador e a quantidade de portas disponíveis na placa ESP32. A alocação das portas foi feita de forma estratégica, priorizando os dispositivos que requerem conexões específicas para seu correto funcionamento, como o leitor RFID MFRC522 e o display LCD 16x2 com interface I2C. Após essa etapa, foram atribuídas as demais portas para os periféricos restantes, como o buzzer, LED, servo motor, sensor ultrassônico HC-SR04 e os circuitos de acionamento controlados pelo transistor TIP120.

A seguir, são detalhados os componentes utilizados no projeto, bem como os critérios para sua escolha:

4.1 Escolha dos componentes

- ESP32 (microcontrolador):
    - O microcontrolador ESP32 foi escolhido por atender plenamente às necessidades do projeto, especialmente pela conectividade Wi-Fi integrada, essencial para a implementação do servidor web. Além disso, sua arquitetura permite o controle de múltiplos periféricos, desempenhando também as funções de um Arduino convencional.

- Leitor RFID MFRC522:
    - Utilizado para o controle de acesso ao ambiente, o MFRC522 é um leitor de baixo custo e alta versatilidade. Embora no projeto tenha sido empregada apenas a leitura do UID das tags, o módulo possui capacidade para leitura e gravação de dados nas tags RFID, o que o torna útil para aplicações futuras mais avançadas.

- Sensor de temperatura e umidade DHT11:
    - Com aquisição prévia por um dos membros da equipe, o DHT11 foi selecionado por fornecer medições básicas de temperatura e umidade com precisão suficiente para a proposta do projeto (±1°C e ±5% UR). Sua principal

limitação é a sensibilidade ao calor gerado pelo toque humano, o que foi solucionado com sua instalação em local isolado.

- Sensor ultrassônico HC-SR04:
    - Entregue no kit básico, este sensor foi escolhido por sua eficiência na detecção de presença em curtas distâncias. Seu baixo custo e confiabilidade o tornam ideal para monitoramento de ocupação do ambiente

- Display LCD 16x2 com interface I2C:
    - Utilizado para exibição local de mensagens e estados do sistema, o display foi integrado via módulo I2C para reduzir a quantidade de portas necessárias no ESP32, otimizando a alocação de recursos da placa.

- Servo motor:
    - Empregado como atuador do mecanismo de trava da porta durante a prototipagem. Embora não seja adequado para sistemas reais de segurança, cumpre bem sua função em um ambiente de demonstração e testes.

- Buzzer:
    - Responsável por fornecer feedback sonoro ao usuário, emitindo tons distintos para aceitação (tons suaves) e rejeição (tons graves) de tags RFID. Foi implementado utilizando a função tone () nativa da plataforma ESP32.

- LED indicador:
    - Utilizado para simular o sistema de iluminação do ambiente. Seu acionamento está vinculado à detecção de presença ou ao comando via interface web.

- Ventoinha:
    - Representa um sistema de ventilação. Embora seja um componente demonstrativo, sua posição estratégica próxima ao sensor DHT11 permite observar uma redução prática na temperatura medida, simulando com fidelidade o funcionamento de um sistema de climatização.

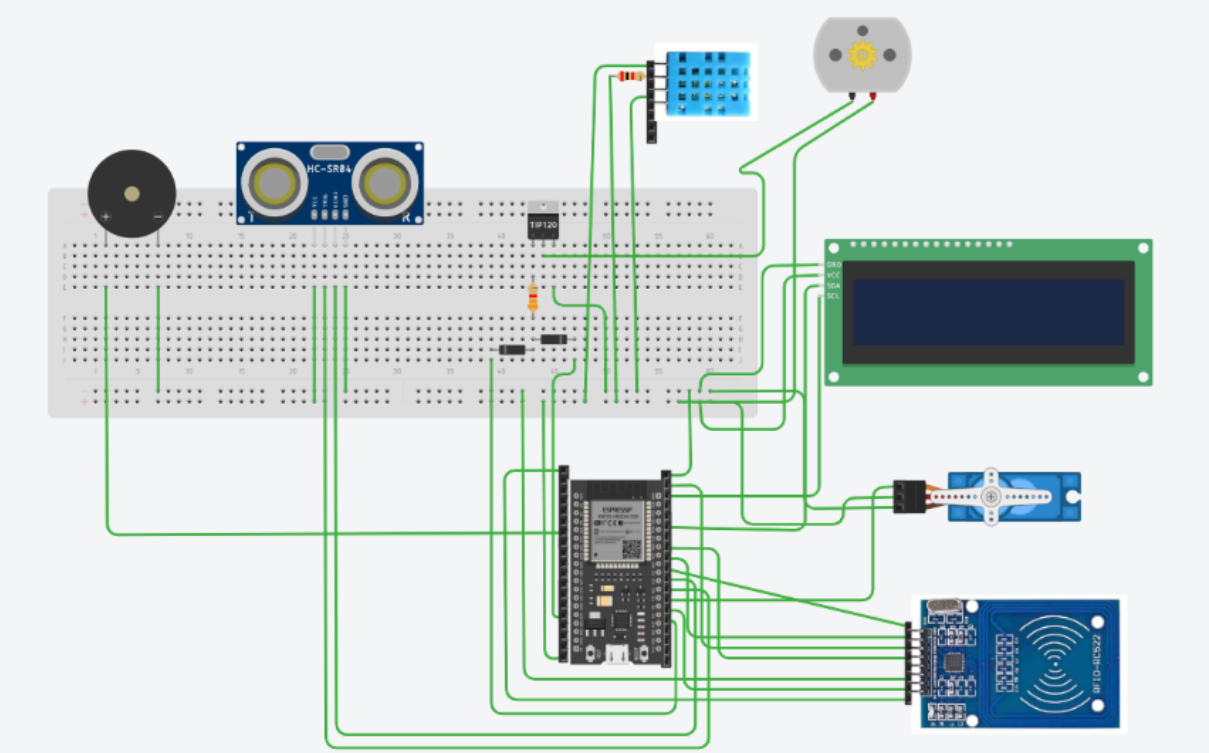
- Transistor TIP120:
    - Utilizado para chaveamento da ventoinha, uma vez que o ESP32 não fornece corrente suficiente para acioná-la diretamente. O TIP120 permite controlar cargas maiores com segurança e eficiência.

- Diodos:
    - Implementados para evitar retorno de corrente às portas digitais da ESP32, visto que o controle da ventoinha pode ser realizado por dois canais distintos. O uso dos diodos impede que a tensão de uma porta se propague pela outra, garantindo a integridade do circuito.

- Cabos de conexão:
    - Utilizados para realizar todas as interligações entre os componentes e o microcontrolador, compondo a estrutura física do protótipo.

4.2 Montagem Física

Durante a etapa de simulação e representação visual do circuito, identificou-se que diversas plataformas, como o Tinkercad e o Wokwi, não oferecem suporte simultâneo a todos os componentes utilizados no projeto. Para fins de exibição nos slides e na documentação técnica, foi realizada uma montagem demonstrativa no Tinkercad, utilizando barramentos de 8 pinos para representar os pontos de conexão dos componentes ausentes. Posteriormente, a composição visual final foi complementada na plataforma Canva, onde foram adicionadas imagens ilustrativas do ESP32, do leitor RFID MFRC522 e do sensor DHT11, que não estão disponíveis nativamente na simulação.



É importante destacar que, embora essa montagem tenha sido elaborada com fins exclusivamente ilustrativos e didáticos, todos os testes funcionais foram realizados com o circuito montado fisicamente, utilizando os componentes reais conectados ao microcontrolador ESP32 em protoboard. Essa abordagem garantiu resultados confiáveis e condizentes com o comportamento esperado do sistema em condições reais de operação.

4.3 Pinagem do Sistema (Pinout)

Para possibilitar a replicação precisa do projeto físico, segue abaixo a descrição detalhada das conexões entre os componentes e o microcontrolador ESP32:

Servo Motor

- GND → GND
- VCC → 5V
- Sinal/Controle → GPIO4

Display LCD 16x2 com Interface I2C

- GND → GND
- VCC → 5V
- SDA → GPIO21
- SCL → GPIO22

Leitor RFID MFRC522

- SDA → GPIO5
- SCK → GPIO18
- MOSI → GPIO23
- MISO → GPIO19
- IRQ → Não utilizado
- GND → GND
- RST → GPIO0
- 3.3V → VCC (3.3V)

Sensor Ultrassônico HC-SR04

- VCC → 5V
- TRIG → GPIO16
- ECHO → GPIO17
- GND → GND

Buzzer

- GND → GND

- Sinal/Acionamento → GPIO32

Sensor de Temperatura e Umidade DHT11

- GND → GND
- VCC → 5V (via resistor de 1kΩ)
- Sinal → GPIO15 (conectado ao barramento após o resistor de 1kΩ em série com o VCC)

LED Indicador

- Ânodo (positivo) → GPIO14 (via resistor de 330Ω)
- Cátodo (negativo) → GND

Ventoinha (via Transistor TIP120)

TIP120:

- Base →
    - GPIO2 (via diodo e resistor de 3.3kΩ)
    - GPIO13 (via diodo e resistor de 3.3kΩ)
- Coletor → GND da ventoinha
    - VCC da ventoinha → VCC 5V
- Emissor → GND

Nota: O uso dos diodos evita que a corrente de uma porta de acionamento retorne pela outra, protegendo os pinos do ESP32.

4.4 Desenvolvimento do software

O software foi desenvolvido utilizando a plataforma Arduino IDE, adotada por sua compatibilidade com o microcontrolador ESP32 e ampla disponibilidade de bibliotecas. A seguir, apresenta-se o resumo funcional do código, seguido da versão completa do programa, devidamente comentada para facilitar sua compreensão e manutenção.

4.4.1 Bibliotecas Utilizadas

O código desenvolvido faz uso de nove bibliotecas externas, essenciais para o funcionamento de diferentes módulos do sistema. Sendo elas:

- SPI.h (Comunicação RFID)
- MFRC522.h (Comunicação RFID)
- Wire.h (Comunicação I2C)
- LiquidCrystal_I2C.h (Comunicação I2C – Display LCD)
- DHT.h (Comunicação DHT11)
- WiFi.h (Comunicação WiFi)
- WebServer.h (Web Server)
- Ultrasonic.h (Comunicação HC-SR04)
- ESP32Servo.h (Comunicação Servo)

Essas bibliotecas possibilitam a comunicação com os sensores, o controle dos atuadores, a exibição de dados, o acesso remoto via rede Wi-Fi e a interface com o display.

4.4.2 Funções Implementadas

Foram implementadas 11 funções no total, sendo:

- 2 funções principais:
    - setup ()
    - loop ()

- 9 funções auxiliares, utilizadas para modularização e melhor organização do código:

    - verificarDesligamentoPorAusencia ()
    - lerRfid ()
    - atualizarEstadoOcupacao ()
    - atualizarDisplayTempUmi ()
    - controleAutomaticoVentoinha ()
    - controleLuz ()
    - controleVentilacao ()
    - handleRoot ()

- redirectToRoot ()

Observação: Os nomes das funções foram definidos de forma clara e autoexplicativa, de modo que não se fez necessário incluir comentários adicionais para descrever suas funcionalidades específicas.

4.4.3 Pinos Utilizados

No código-fonte, são declarados 10 pinos explicitamente. No entanto, o hardware físico utiliza 15 conexões (comunicação, excluindo GND, 5V e 3.3V), devido a particularidades de alguns componentes, como o leitor RFID MFRC522, cujos outros pinos são definidos internamente pela biblioteca MFRC522.h, além da comunicação I2C que só é possível nas portas 21 e 22 da ESP32.

4.4.4 Variáveis

O código-fonte desenvolvido para o projeto utiliza um total de 31 variáveis, organizadas entre tipos primitivos, constantes, estruturas compostas, macros e instâncias de classes. Esta estrutura foi projetada com o objetivo de garantir clareza, modularidade e eficiência durante a execução do sistema. A seguir, apresenta-se uma análise detalhada sobre cada tipo de variável utilizada e sua finalidade dentro da lógica do programa:

Tipos Primitivos, Constantes e Strings:

- bool — 6 variáveis
  Utilizadas para controle de estados lógicos do sistema, como condições de presença, ativação de luzes, ventilação e controle de permissões.

- const int — 4 variáveis
  Constantes inteiras fixas, utilizadas principalmente para definição de pinos e parâmetros de configuração que não variam durante a execução.

- int — 3 variáveis
  Armazenam valores modificáveis, geralmente utilizados para capturar leituras de sensores ou estados temporários (como temperaturas medidas ou contadores simples).

- unsigned long — 2 variáveis
  Empregadas para controle de tempo não-bloqueante utilizando a função millis (), essenciais para o monitoramento de intervalos sem interromper o funcionamento do código.

- const long — 2 variáveis
  Constantes representando intervalos fixos de tempo, como o tempo de espera para desligamento automático da luz por ausência.

- const char* — 2 variáveis
  Ponteiros para cadeias de caracteres constantes, utilizados para armazenar as credenciais da rede Wi-Fi (SSID e senha) que o ESP32 utilizará para conexão à rede.

- String — 1 variável
  Utilizada para construir dinamicamente as mensagens que serão exibidas na interface web acessada via navegador, permitindo interação amigável com o usuário.

Arrays e Estruturas:

- byte [] — 1 variável
  Array que armazena o UID (identificador único) do último cartão RFID lido. Essa informação é comparada com a lista de cartões autorizados para liberar ou negar o acesso.

- const usuário [] — 1 variável
  Array de estruturas contendo os dados dos usuários autorizados no sistema. Cada elemento armazena um nome e o UID correspondente da tag RFID. Atualmente, estão cadastrados dois usuários.

Macros de Pré-processamento:

- #define — 3 diretivas

  Macros utilizadas para definir parâmetros do display LCD antes da compilação, como:
  - Endereço I2C do display;
  - Número de colunas (16);
  - Número de linhas (2).

    Essas definições otimizam o código e evitam repetição de valores constantes.

Instâncias de Objetos (Classes)

- Web Server

  Responsável por hospedar e gerenciar a interface web que permite o controle remoto do sistema (ligar luz, ventoinha etc.).

- LiquidCrystal_I2C LCD

  Controla o display LCD 16x2 com interface I2C, utilizado para exibir informações locais como temperatura, umidade e status de ocupação da sala.

- MFRC522 rfid

  Interface com o leitor RFID MFRC522, utilizado para autenticar o acesso de usuários via cartões com UID previamente autorizados.

- DHT

  Realiza a leitura de temperatura e umidade do ambiente através do sensor DHT11.

- Ultrasonic ultrasonic1

  Responsável pela medição da distância para detectar a presença de uma pessoa na sala, auxiliando no controle automático de iluminação.

- Servo Porta

  Controla o servo motor responsável pela abertura e fechamento da trava da porta, conforme o status de autenticação do cartão RFID.

4.4.5 Código Final

O código-fonte do projeto possui aproximadamente 470 linhas, o que inviabiliza sua inserção direta nesta documentação, por questões de legibilidade e organização. Assim, para garantir o fácil acesso ao material completo, foram disponibilizados dois QR Codes distintos, com finalidades complementares:

Leitura do Código (.DOCX):
O QR Code abaixo direciona para uma versão formatada e comentada do código-fonte, disponível em documento .docx. Esta versão é ideal para fins de consulta, estudo e análise, especialmente em ambientes que não oferecem suporte nativo à visualização de arquivos de código (.ino).

Link: https://docs.google.com/document/d/1iqVQgqkNS-4Y4tgL_c1GxrtuvKzYiygYLiJQXVtRkqM/edit?usp=drive_link

Download do Código-Fonte (.INO):

O QR Code a seguir permite o acesso direto ao arquivo original do código, no formato. ino, compatível com a plataforma Arduino IDE. Esta versão é recomendada para testes, execução prática e eventuais modificações no ambiente de desenvolvimento.

Link: https://drive.google.com/file/d/1DMb6zXAGRqVdXpo9pLiKepiz0Un5x8ru/view?usp=sharing

4.4.6 Fluxograma

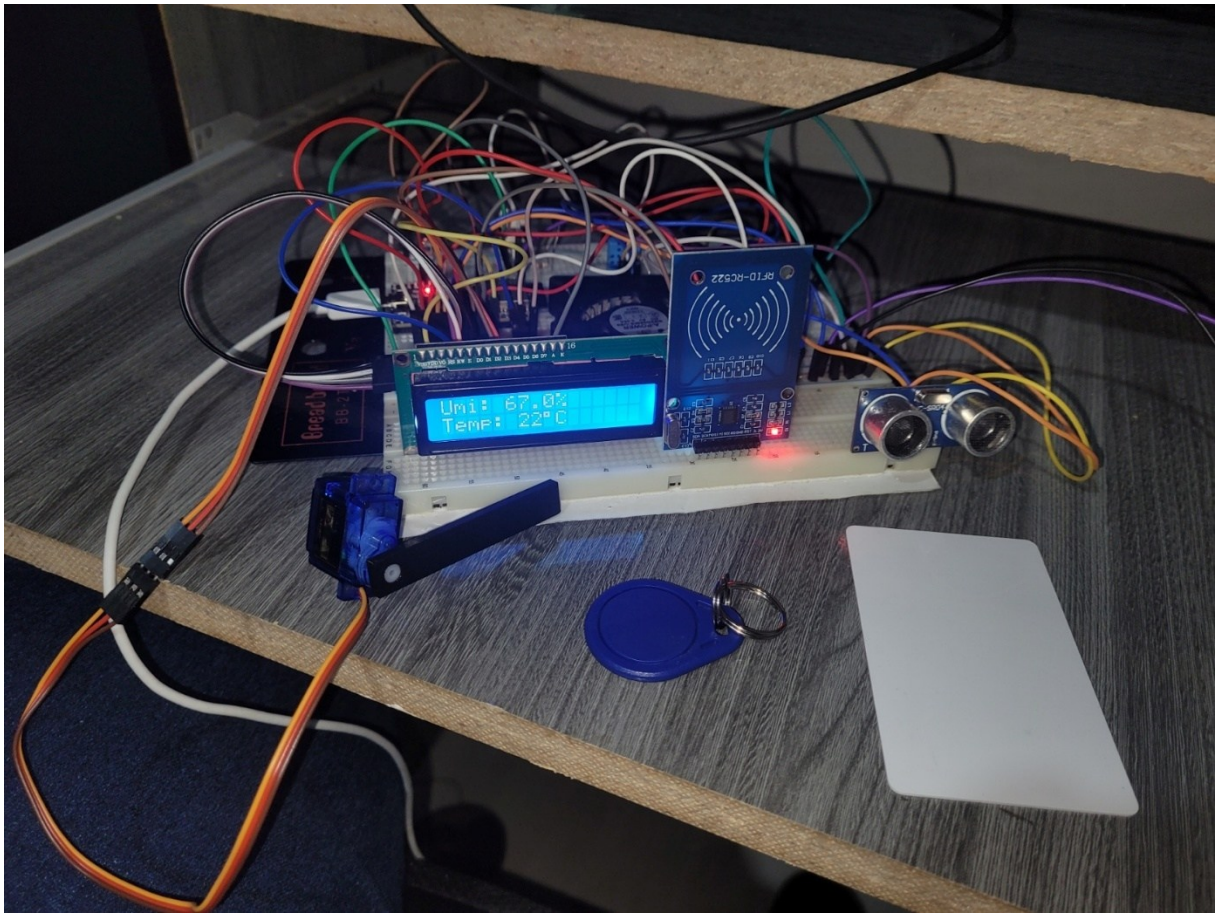
Para um melhor entendimento do código, foi elaborado um fluxograma do funcionamento completo do sistema, para sua visualização, realize a leitura do QR Code abaixo:

Link: https://abre.ai/m5MO

5 Implementação & Testes

5.1 Protótipo Inicial

A montagem final do projeto foi realizada em duas etapas. A primeira consistiu em uma montagem externa, utilizando exclusivamente uma protoboard, com o objetivo de testar a integração e funcionamento dos componentes. Essa etapa preliminar permitiu a validação da lógica e o ajuste de eventuais inconsistências no circuito. A imagem abaixo ilustra essa configuração de testes:



5.2 Maquete

Após a validação completa da lógica de funcionamento na protoboard, foi realizada a montagem final do projeto em uma maquete simplificada, utilizando uma caixa de papelão como estrutura física representativa.

Nessa etapa:

- O display LCD foi fixado de modo a ficar visível externamente, permitindo a leitura das informações de temperatura, umidade e status do sistema.

- O leitor RFID foi posicionado na parte interna da caixa, permanecendo “oculto”, uma vez que sua distância de leitura é suficiente para atravessar a espessura do papelão sem comprometer o funcionamento.

- A ventoinha, o sensor de temperatura e umidade (DHT11) e o LED representando a iluminação foram instalados de forma visível na parte externa da maquete.
- Uma estrutura móvel de papelão simulando a porta da sala foi adicionada. Essa porta é travada mecanicamente por uma haste conectada ao servo motor, que impede sua abertura quando o sistema estiver bloqueado.

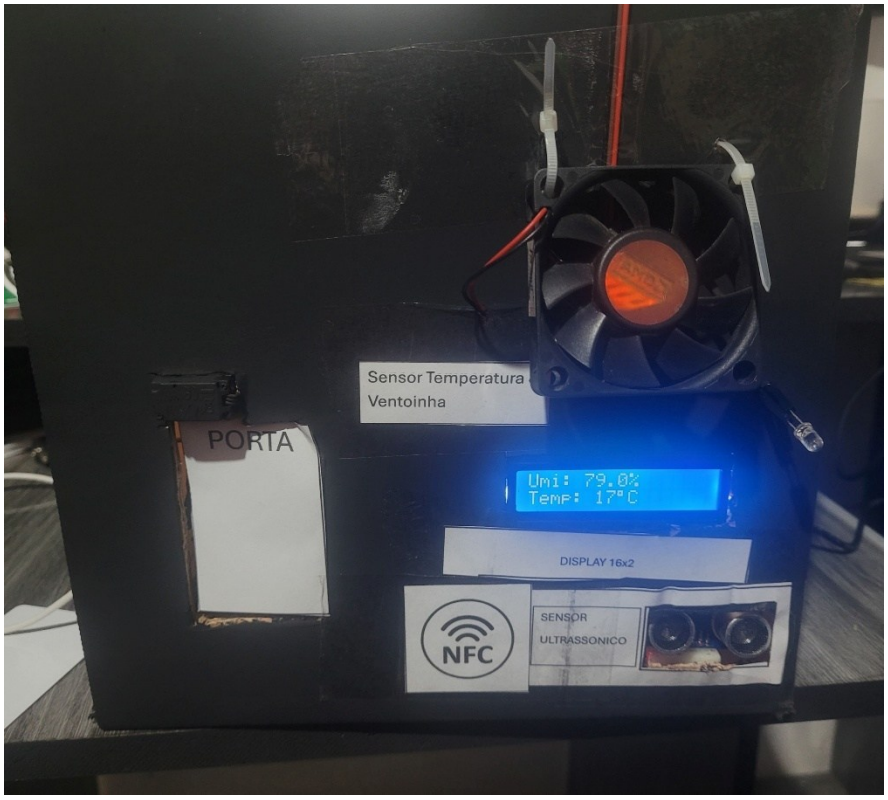
- O buzzer foi mantido no interior da caixa, já que sua função é apenas sonora, sem necessidade de visualização.

- Cada componente visível externamente foi rotulado com identificação em papel, facilitando a compreensão da montagem.

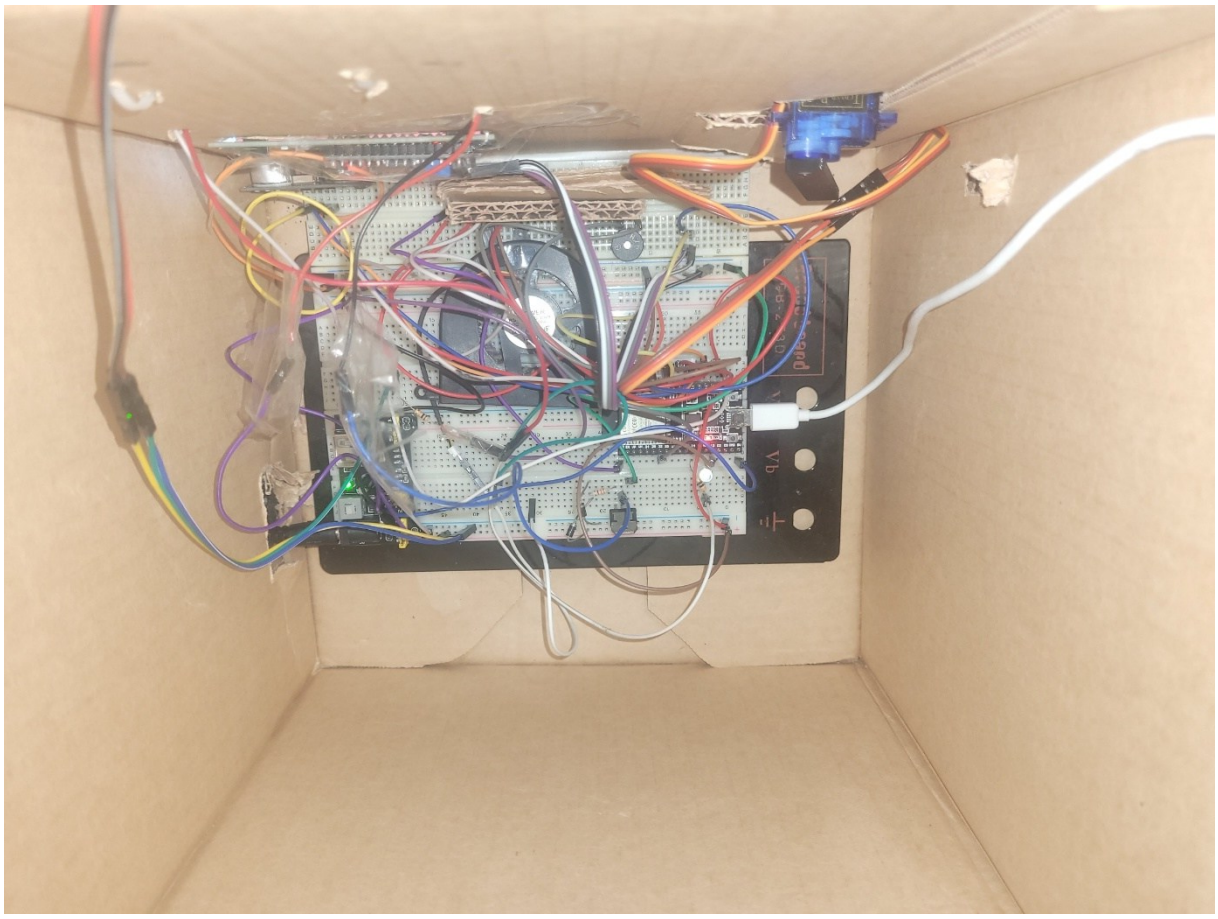
- Para fins estéticos e de apresentação, toda a caixa foi pintada na cor preta.

Essa maquete teve como objetivo representar visualmente o funcionamento do sistema de forma compacta, didática e funcional, servindo como suporte para apresentações e demonstrações.
Na próxima página, são apresentadas imagens da visão frontal e da estrutura interna da maquete.

5.2.1 Visão frontal:



5.2.2 Visão Superior:



Observação: devido ao espaço físico reduzido, a fiação interna apresenta-se visualmente desorganizada, o que é comum em protótipos compactos. Para uma implementação real em ambiente físico, o ideal seria a utilização de placas de circuito impresso (PCBs) independentes, distribuídas em pontos estratégicos da sala e interligadas por meio de tecnologias como Wi-Fi, RF ou Bluetooth Low Energy (BLE), garantindo organização, escalabilidade e eficiência no sistema final.

5.2.3 Vídeo de funcionamento:

Abaixo, encontra-se o QR Code que direciona para um vídeo publicado na plataforma YouTube, demonstrando o funcionamento prático do projeto em operação:

Link: https://youtu.be/5RqwbafUiyc

5.3 Comentários Finais:

A finalização do projeto foi altamente satisfatória, considerando que todas as funcionalidades inicialmente planejadas foram implementadas com sucesso, além de melhorias e incrementos adicionados ao longo do desenvolvimento. No entanto, como é comum em projetos dessa natureza, uma série de intempéries técnicas e operacionais foram enfrentadas e superadas. Um dos principais desafios ocorreu em relação à sensibilidade do protocolo I2C, afetado por interferência eletrostática. Esse problema comprometia a comunicação entre o ESP32 e o display LCD, resultando em perda de sincronia e exibição de caracteres aleatórios. A solução prática encontrada foi o reinício completo do microcontrolador sempre que a falha era detectada, o que, apesar de eficaz, expôs uma fragilidade importante do sistema em ambientes sujeitos a ruídos elétricos.

Também houve dificuldades na utilização simultânea da função tone () (nativa da plataforma Arduino) com o controle de servo motor. Foi observado que a emissão de sons interferia no funcionamento dos timers internos utilizados para o posicionamento do servo, exigindo o desligamento temporário do servo antes da execução da função sonora e a inserção de pequenos atrasos para garantir a estabilidade do sistema.

Além disso, o servidor web embarcado no próprio ESP32 apresentou limitações de desempenho, refletidas em lentidão e travamentos ocasionais, especialmente quando vários periféricos estavam ativos. A alimentação elétrica do conjunto também se mostrou um ponto crítico. A corrente fornecida via USB foi insuficiente para sustentar, de forma estável, todos os dispositivos conectados (servo motor, display, LED, buzzer e ventoinha), sendo necessário adotar uma fonte de alimentação externa para evitar reinicializações inesperadas.
Outra dificuldade significativa foi a limitação das ferramentas de simulação virtual.

Plataformas amplamente utilizadas, como o Wokwi e o TinkerCad, não ofereceram suporte completo a todos os componentes do projeto, impossibilitando a verificação funcional do circuito em ambiente simulado. Como alternativa, foi utilizada uma representação visual estática do circuito, por meio de sobreposição de imagens, com o objetivo de documentar a montagem física de forma esquemática.

Por fim, a tentativa de implementar autenticação por teclado matricial 4x3 teve de ser descartada devido a falha no módulo adquirido. Essa funcionalidade permanece como uma possibilidade de expansão futura do sistema.

Apesar dessas intempéries, o projeto se manteve funcional, estável e alinhado com os objetivos propostos. A montagem física, com maquete em papelão e integração dos componentes de forma didática, contribuiu significativamente para a clareza na demonstração prática. A colaboração entre os membros do grupo foi essencial para contornar os desafios técnicos e operacionais enfrentados, promovendo uma experiência de aprendizado rica e multifacetada.

6 Conclusão

O presente trabalho resultou na concepção, desenvolvimento e validação de um sistema embarcado voltado para automação de acesso e monitoramento ambiental, utilizando o microcontrolador ESP32 como núcleo da aplicação. A integração entre sensores, atuadores e interface web local demonstrou a viabilidade de criar soluções práticas e funcionais mesmo com recursos limitados.

Além da concretização dos objetivos técnicos, o projeto permitiu o enfrentamento de diversos obstáculos reais comuns em sistemas embarcados — desde interferências eletromagnéticas e limitações de hardware até restrições de simulação e falhas de componentes. Esses desafios proporcionaram oportunidades valiosas de aprendizado, exigindo análise crítica, busca por soluções alternativas e adaptação contínua durante o desenvolvimento.

A experiência consolidou conhecimentos importantes em programação C++, integração de hardware e software, organização de tarefas em equipe, e elaboração de documentação técnica. O resultado é um sistema funcional, que serve como prova de conceito aplicável a diferentes contextos e, ao mesmo tempo, uma base sólida para evoluções futuras.

Entre as possibilidades de expansão estão a integração com serviços em nuvem, o uso de autenticação via teclado matricial ou biometria, a criação de uma interface gráfica mais robusta, e a montagem do sistema em placas de circuito impresso (PCB), aproximando ainda mais o protótipo de um produto.

Mais do que a entrega de um sistema funcional, este projeto representa um marco importante na formação dos autores, consolidando habilidades técnicas e reforçando o entendimento prático sobre as aplicações reais de sistemas embarcados e Internet das Coisas (IoT) com ESP32.

7 Referências

16 x 2 Character LCD. [s.l: s.n.]. Disponível em: <https://www.vishay.com/docs/37484/lcd016n002bcfhet.pdf>. Acesso em: 8 jul. 2025.

CORREIA VIANA, C. **Como utilizar o display LCD 16×02 com módulo I2C no Arduino – Blog da Robótica**. Disponível em: <https://www.blogdarobotica.com/2022/05/02/como-utilizar-o-display-lcd-16x02-com-modulo-i2c-no-arduino/>. Acesso em: 8 jul. 2025.

DHT11 Humidity & Temperature Sensor. Disponível em: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf?srsltid=AfmBOopewvFB-UOqpZFlNz-2xKOY4dBOdEHlAcInWTsOfEdXze8obQYf>. Acesso em: 8 jul. 2025.

ELECFREAKS. **Ultrasonic Ranging Module HC -SR04**. [s.l: s.n.]. Disponível em: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>. Acesso em: 8 jul. 2025.

ESPRESSIF. **ESP32-WROOM-32 Datasheet**. [s.l: s.n.]. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>. Acesso em: 8 jul. 2025.

MFRC522 Standard performance MIFARE and NTAG frontend. [s.l: s.n.]. Disponível em: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>. Acesso em: 8 jul. 2025.
RFID RC522 Interfacing with ESP32 | ESP32. Disponível em: <https://www.electronicwings.com/esp32/rfid-rc522-interfacing-with-esp32>. Acesso em: 8 jul. 2025.

SANTOS, S. **Building an ESP32 Web Server: The Complete Guide for Beginners | Random Nerd Tutorials**. Disponível em: <https://randomnerdtutorials.com/esp32-web-server-beginners-guide/>. Acesso em: 8 jul. 2025.

SERVO MOTOR SG90 DATA SHEET. [s.l: s.n.]. Disponível em: <http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf>. Acesso em: 8 jul. 2025.

Technical Documentation - Design | onsemi. Disponível em: <https://www.onsemi.com/pdf/datasheet/tip120-d.pdf>. Acesso em: 8 jul. 2025.