



**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
TOCANTINS
CAMPUS PALMAS
BACHARELADO EM ENGENHARIA ELÉTRICA**

**ÁLVARO ALVES SOUSA
LUCAS KAIO PEREIRA DANTAS**

**SISTEMA DE AUTOMAÇÃO PARA AS SALAS DE AULA DO IFTO - CAMPUS
PALMAS UTILIZANDO DETECÇÃO DE PESSOAS BASEADO EM INTELIGÊNCIA
ARTIFICIAL**

PALMAS/TO

2024

ÁLVARO ALVES SOUSA
LUCAS KAIO PEREIRA DANTAS

**SISTEMA DE AUTOMAÇÃO PARA AS SALAS DE AULA DO IFTO - CAMPUS
PALMAS UTILIZANDO DETECÇÃO DE PESSOAS BASEADO EM INTELIGÊNCIA
ARTIFICIAL**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do Título de
Bacharel do Curso de Graduação em Engenharia
Elétrica do Instituto Federal do Tocantins, Campus
Palmas.

Orientador: Doutor, Valci Ferreira Victor.
Coorientador: Doutor, Antonio Marcelino da Silva
Filho.

PALMAS/TO
2024

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

S725s Sousa, Álvaro Alves
 Sistema de automação para as salas de aula do IFTO - campus
 Palmas utilizando detecção de pessoas baseado em inteligência
 artificial / Álvaro Alves Sousa, Lucas Kaio Pereira Dantas. –
 PALMAS, TO, 2024.
 111 f. : il. color.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia
Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia do
Tocantins, Campus Palmas, PALMAS, TO, 2024.

Orientador: Dr. Valci Ferreira Victor
Coorientador: Dr. Antonio Marcelino da Silva Filho

1. Automação. 2. Inteligência Artificial. 3. Economia de energia
elétrica. I. Dantas, Lucas Kaio Pereira. II. Ferreira Victor, Valci III. da
Silva Filho, Antonio Marcelino. IV. Título.

CDD 621

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins de estudo e pesquisa, desde que citada a fonte.

Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos pelo(a) autor(a).

ÁLVARO ALVES SOUSA
LUCAS KAIO PEREIRA DANTAS

**SISTEMA DE AUTOMAÇÃO PARA AS SALAS DE AULA DO IFTO - CAMPUS
PALMAS UTILIZANDO DETECÇÃO DE PESSOAS BASEADO EM INTELIGÊNCIA
ARTIFICIAL**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do Título de
Bacharel do Curso de Graduação em Engenharia
Elétrica do Instituto Federal do Tocantins, Campus
Palmas.

Orientador: Doutor, Valci Ferreira Victor.
Coorientador: Doutor, Antônio Marcelino da Silva
Filho.

Aprovado em: 27/08/2024

BANCA AVALIADORA

Documento assinado digitalmente



VALCI FERREIRA VICTOR

Data: 02/09/2024 14:43:20-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Valci Ferreira Victor
Orientador

Documento assinado digitalmente



ANTONIO MARCELINO DA SILVA FILHO

Data: 03/09/2024 09:27:39-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Antônio Marcelino da Silva Filho
Coorientador

Documento assinado digitalmente



MARCIO AUGUSTO TAMASHIRO

Data: 02/09/2024 17:22:33-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Márcio Augusto Tamashiro
Membro da Banca Avaliadora

Documento assinado digitalmente



WENDELL EDUARDO MOURA COSTA

Data: 03/09/2024 11:25:23-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Wendell Eduardo Moura Costa
Membro da Banca Avaliadora

RESUMO

Este trabalho de conclusão de curso tem por objetivo propor um sistema protótipo de automação aplicável às salas de aula do IFTO – Campus Palmas para reduzir o desperdício de energia elétrica. O trabalho traz uma nova abordagem ao utilizar inteligência artificial para detecção de pessoas nas salas de aula, através do *framework* YOLOv8, uma poderosa ferramenta para detecção e segmentação de objetos em imagens. O sistema proposto utiliza câmeras para captação de imagens, que são analisadas por um modelo de inteligência artificial do *framework* YOLOv8 para detectar pessoas no ambiente (sistema primário), a fim de enviar a informação para os circuitos de controle da iluminação e dos condicionadores de ar (sistema secundário), estes que realizam os comandos lógico através de módulos ESP32. Todos os subsistemas se comunicam através do protocolo HTTP, em um cenário de cliente-servidor. Uma pesquisa de campo mostrou que a aplicação do sistema auxilia na redução dos custos de energia elétrica, garantindo resultados satisfatórios quanto a proposta.

Palavras chave: Automação, Inteligência artificial, Eficiência energética.

ABSTRACT

This course completion work aims to propose a prototype automation system applicable to classrooms at IFTO – Campus Palmas to reduce electrical energy waste. The work brings a new approach to using artificial intelligence to detect people in classrooms, through the YOLOv8 framework, a powerful tool for detecting and segmenting objects in images. The proposed system uses cameras to capture images, which are analyzed by an artificial intelligence model from the YOLOv8 framework to detect people in the environment (primary system), in order to send the information to the lighting and air conditioner control circuits (secondary system), which carry out the logic commands via ESP32 modules. All the subsystems communicate via the HTTP protocol, in a client-server scenario. Field research has shown that the application of the system helps to reduce electricity costs, guaranteeing satisfactory results for the proposal.

Key words: Automation, Artificial intelligence, Energy efficiency.

SUMÁRIO

1 INTRODUÇÃO	8
1.1 Justificativa	10
1.2 Objetivos	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
2 REVISÃO DE LITERATURA	13
2.1 Eficiência Energética.....	13
2.2 Instalações Elétricas Prediais	13
2.3 Automação	14
2.4 Internet das Coisas (IoT)	15
2.4.1 Microcontroladores.....	16
2.4.2 Protocolo HTTP.....	20
2.4.3 API	22
2.5 Inteligência Artificial (IA).....	23
2.5.1 YOLOv8	24
3 METODOLOGIA	27
4 DESENVOLVIMENTO E FUNCIONAMENTO DO SISTEMA DE AUTOMAÇÃO .	28
4.1 Rede de Comunicação	28
4.2 Sistema de Detecção (Primário e Principal)	29
4.2.1 Programação.....	30
4.2.2 Princípio de Funcionamento e Funcionalidades Adicionais	31
4.2.3 Montagem	36
4.3 Sistema de Controle da Iluminação (Secundário).....	37
4.3.1 Programação.....	37
4.3.2 Princípio de Funcionamento e Funcionalidades Adicionais	38
4.3.3 Montagem	41

4.4 Sistema de Controle do Condicionador de Ar (Secundário).....	42
4.4.1 Programação.....	42
4.4.2 Montagem	45
4.5 Sistema de Acompanhamento (Terciário)	46
4.5.1 Programação e Princípio de Funcionamento	47
5 RESULTADOS.....	47
5.1 Teste Final do Sistema	48
5.2 Estimativas de Economia.....	57
5.3 Custos.....	59
5.4 Recomendações de Instalação.....	61
6 CONCLUSÃO	63
REFERÊNCIAS.....	65
APÊNDICE A - ALGORITMO DO SISTEMA DE DETECÇÃO	71
APÊNDICE B - ALGORITMO DO SISTEMA DE CONTROLE DA ILUMINAÇÃO ...	78
APÊNDICE C - ALGORITMO DO SISTEMA DE CONTROLE DO CONDICIONADOR DE AR	83
APÊNDICE D - ALGORITMOS DO SISTEMA DE ACOMPANHAMENTO	88
APÊNDICE E - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE DETECÇÃO	100
APÊNDICE F - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE CONTROLE DA ILUMINAÇÃO	104
APÊNDICE G - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE CONTROLE DO CONDICIONADOR DE AR.....	106
APÊNDICE H - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE ACOMPANHAMENTO.....	108
APÊNDICE I - BIBLIOTECAS DO PYTHON UTILIZADAS NO PROJETO E SUAS VERSÕES.....	109
APÊNDICE J - SUGESTÃO DE INSTALAÇÃO DO SISTEMA (CONDICIONADORES DE AR).....	110

APÊNDICE K - SUGESTÃO DE INSTALAÇÃO DO SISTEMA (ILUMINAÇÃO E CENTRAL DE PROCESSAMENTO	111
--	-----

1 INTRODUÇÃO

No IFTO - Campus Palmas, assim como em outras instituições de grande porte, é de suma importância a implementação de táticas para a redução do desperdício de energia elétrica. Isso se faz necessário devido à alta carga demandada da rede de distribuição por essas edificações, o que reflete diretamente em suas despesas orçamentárias e, conseqüentemente, na alocação de recursos para setores importantes. Quanto maior o desperdício, maiores serão essas despesas, dificultando, assim, uma melhor destinação desses recursos. Outro ponto importante é que reduzir o consumo de energia elétrica representa contribuir para o meio ambiente e o desenvolvimento sustentável.

No dia a dia, milhares de pessoas, dentre servidores, alunos e visitantes, utilizam as dependências do IFTO - Campus Palmas, e é um grande desafio realizar o controle dos sistemas de iluminação e climatização presentes nas salas de aula de modo a reduzir o desperdício de energia. Atualmente, esse controle é feito por servidores do instituto, contudo está sujeito à disponibilidade dos mesmos e ao erro humano, o que no longo prazo pode representar custos adicionais por desperdício de energia. Dessa forma, esse trabalho se propõe a apresentar uma solução de automação para esses dois tipos de sistemas, diminuindo a necessidade de intervenção humana e possibilitando uma maior assertividade no controle.

Para a realização deste trabalho, consideraram-se as condições das salas de aula do IFTO- *Campus* Palmas, com destaque aos blocos 14 a 17. A partir disso, observam-se algumas motivações para a necessidade da elaboração de um sistema automatizado nestes locais, tais como: grande fluxo de servidores e alunos, pois os ambientes sediam as salas de aula dos cursos técnicos, ensino superior e etc.; semelhança estrutural e de usos, onde os blocos apresentam uma construção padronizada e constituída em sua maioria por sala de aulas; e observação de situações isoladas de desperdício de energia elétrica nestes locais.

Nas salas de aula do IFTO - *Campus* Palmas, os sistemas de iluminação e climatização não possuem nenhum sistema automático de controle. Rotineiramente, esse controle é feito à medida que as pessoas utilizam os espaços de forma manual, tornando o acionamento dependente de ação humana. Nessa dinâmica, ainda há a intervenção de servidores da própria instituição que são responsáveis por garantir que as salas de aula não permaneçam com luzes acesas e com condicionadores de ar

ligados durante a ausência de pessoas no local. O sistema de climatização é controlado unicamente pelos servidores para evitar a perda ou extravio dos controles remotos dos equipamentos.

A preocupação quanto ao desperdício de energia elétrica, ocasionado pelo cenário citado acima, tem sido motivo de buscas para propostas de soluções quanto ao problema. Dois trabalhos de conclusão de curso foram realizados por Moreira e Neres, um em 2015 e o outro em 2019, ambos visando a elaboração de sistemas de automação para redução dos desperdícios de energia elétrica ocasionados pelo mal uso da iluminação e condicionadores de ar nas salas de aula do IFTO - Campus Palmas. O trabalho de 2015 comprova que há um desperdício de energia elétrica significativo, apresentando uma proposta de solução com controle automático robusto e industrial. O trabalho de 2019 apresenta uma proposta com sistema embarcado, mais viável economicamente, que foi instalado na sala 906. Apesar dos sistemas se mostrarem funcionais, não contemplam todas as salas de aula do campus, além de que se sujeitam a variáveis de falha, o que pode comprometer a efetividade do controle do desperdício.

Diante disto, pode-se levantar o seguinte questionamento; é possível reduzir o desperdício de energia elétrica nas salas de aula utilizando um sistema de automação? É essa pergunta que o sistema desenvolvido neste trabalho busca resolver para garantir a redução do desperdício e aumento da eficiência energética do IFTO - Campus Palmas.

A partir do cenário apresentado, este trabalho foi desenvolvido na seguinte estrutura: capítulo 1 – apresenta as justificativas para a realização deste trabalho, bem como os objetivos a serem alcançados; capítulo 2 – apresentará toda a base teórica, fundamentada nos principais conceitos que são necessários para a compreensão deste trabalho; capítulo 3: abordará a metodologia utilizada neste trabalho, quais métodos nortearam cada etapa; capítulo 4 – abordará sobre o sistema de automação proposto neste trabalho, passando pelos conceitos iniciais até seus princípios de funcionamento e processos de montagem; capítulo 5 – serão apresentados os resultados obtidos a partir da instalação do sistema de automação proposto; capítulo 6 – abordará as considerações finais do trabalho.

1.1 Justificativa

Nas instituições de ensino superior, uma manifestação do consumo de energia é a necessidade de utilização de energia elétrica para iluminação e climatização constantes, o que impacta diretamente nos custos operacionais. Os princípios e práticas de sustentabilidade devem ser vistos como valores. Ressalta-se que o uso racional e eficiente da energia elétrica deve estar presente em todas as atividades diárias (OLIVEIRA, BARROSO, et al, 2018).

Em uma instituição de ensino como o IFTO - Campus Palmas, não é diferente. Há a necessidade e preocupação com relação ao desperdício de energia elétrica presente na má utilização dos equipamentos de climatização e iluminação das salas de aula, onde há um grande fluxo de alunos durante todos os períodos matutino, vespertino e noturno.

O desperdício de energia elétrica nas salas de aulas do IFTO já foi constatado, comprovado e quantificado, através de pesquisa de campo (MOREIRA e NERES, 2015). O estudo foi realizado quando não haviam sido construídos os blocos 14 a 17. Os autores analisaram as condições de todas as salas de aula, desde as adequações de instalações elétricas dos equipamentos de ar-condicionado e iluminação, quanto o funcionamento de cada sistema, a fim de identificar os desperdícios acarretados pelo mau funcionamento destes. Através de um levantamento por meio de censo, foi constatado que há um desperdício significativo de energia elétrica, cujo principal fator está relacionado ao estado de funcionamento ocioso destes equipamentos em sala de aula.

Ao longo dos anos, houve iniciativas de implementação de sistemas para contribuir com a eficiência energética do campus, como um gerador a diesel e um sistema fotovoltaico, este instalado no bloco 9. Além disso, foi realizada a instalação de um sistema de automação para os condicionadores de ar e iluminação da sala de aula 906, como resultado dos estudos de um trabalho de conclusão de curso (MOREIRA e NERES, 2019). Porém, nem todos os locais da instituição são contemplados com este sistema, ou são abastecidos pelas gerações fotovoltaica e a diesel.

Neste fato, a falta de uma solução efetiva, visando a eficiência energética no campus como um todo, torna o levantamento de desperdícios realizado por Moreira e Neres (2015) algo válido, como uma projeção para o cenário atual. Haja vista que

houve ampliações estruturais ao longo dos anos para conter um maior fluxo de estudantes, é notório que o problema do desperdício de energia elétrica ainda se torne um gargalo para a instituição.

Um dos métodos implementados na redução de desperdícios de energia elétrica é a Automação. A automação de processos pode aumentar a eficiência em cerca de 60%. Outros benefícios potenciais da automação de processos incluem melhor qualidade do produto e esforço operacional reduzido. A substituição de redutores aumenta a eficiência média em 35%. Além disso, os custos de manutenção são reduzidos e a confiabilidade operacional e a vida útil dos equipamentos aumentam (SILVA, 2022 apud FIESC, 2022).

Com base neste raciocínio, entende-se que a elaboração de um sistema para automação dos condicionadores de ar e iluminação das salas de aula do IFTO - *Campus* Palmas faz-se necessária para mitigar o desperdício de energia elétrica. O controle atual é feito por servidores terceirizados, que periodicamente analisam as possíveis salas de aula ociosas que contenham equipamentos ligados. Um sistema efetivo de automação pode trazer um retorno econômico favorável para o campus, além de eliminar o fator humano que está sujeito a ineficiência.

1.2 Objetivos

1.2.1 Objetivo Geral

O presente trabalho tem como objetivo propor um protótipo de automação dos sistemas de climatização e de iluminação, aplicável às salas de aula do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - *Campus* Palmas, com finalidade de minimizar o desperdício de energia elétrica. O respectivo protótipo também visa uma melhoria no sistema já realizado em trabalhos anteriores (MOREIRA e NERES, 2019), trazendo uma nova abordagem do projeto, bem como a implementação de novas tecnologias, como uso de inteligência artificial (IA).

1.2.2 Objetivos Específicos

- 1) Desenvolver um protótipo de sistema de automação para controle de iluminação e climatização de salas de aula, cujo qual realize a detecção

de pessoas no ambiente através imagens capturadas por câmeras e processadas pelo modelo de inteligência artificial do YOLOv8;

- 2) Instalar temporariamente protótipo em sala de aula do IFTO - *Campus* Palmas, a título de testes e validação da solução proposta;
- 3) Realizar estimativas da economia de energia gerada, comparando cenários em que o sistema atua automaticamente com cenários de acionamento manual;
- 4) Realizar levantamento de requisitos e custos necessários para reprodução do sistema;

2 REVISÃO DE LITERATURA

2.1 Eficiência Energética

Um estudo que tem se apresentado em constante desenvolvimento, pautado em uma abordagem de interesse público desde a década de 80, é sobre o conceito de eficiência energética. Segundo a ABESCO (Associação Brasileira das Empresas de Serviços de Conservação de Energia), por definição “a eficiência energética consiste na relação entre a quantidade de energia empregada em uma atividade e aquela disponibilizada para sua realização” (ABESCO,2015). Estratégia que, se aplicada corretamente, garante a redução de custos e promove um consumo sustentável.

Este tema traz à tona a preocupação em relação ao consumo consciente de energia elétrica, seja no âmbito industrial, comercial ou residencial. Falar sobre eficiência energética é, também, pensar nas formas de contenção para o desperdício de energia elétrica. Ao mesmo tempo, este conceito também busca reduzir o gasto financeiro com energia. Para isso, é necessário investir em processos e produtos energeticamente eficientes (PORTAL DA INDÚSTRIA, [s.d.]).

No Brasil, estima-se que por ano é desperdiçado cerca de 43 Terawatt-hora (TWh) de energia elétrica. Isto corresponde ao consumo de cerca de 20 milhões de residências (JEREMIAS, 2022 apud FIESC, 2022). Estes dados demonstram a necessidade urgente de promover a eficiência energética em várias esferas da sociedade brasileira.

A implementação de medidas como a promoção de campanhas de conscientização da população para o consumo consciente, a implantação de sistemas de gestão energética e o encorajamento de empresas para ações de eficiência energética podem ser soluções frente ao problema de desperdício. Combater o desperdício representa não só reduzir os impactos ambientais, como também economizar recursos valiosos para o desenvolvimento sustentável do país.

2.2 Instalações Elétricas Prediais

Toda edificação, necessita utilizar de uma fonte de energia elétrica para suprir a demanda energética dos vários equipamentos instalados no local. Para isso, é

importante que haja a presença de uma instalação elétrica, que garanta a funcionalidade e eficiência de sistemas, tais como iluminação e climatização de um ambiente.

Por definição: “uma instalação elétrica qualquer se compreende pela implementação física dos componentes das ligações elétricas, a conexão entre a fonte geradora de energia elétrica e as cargas elétricas” (WATTSP, 2021). Em um cenário mais simples, aplicam-se as instalações elétricas prediais, que demandam um profundo conhecimento sobre a infraestrutura para garantir eficiência energética e reduzir riscos de acidentes. Estas são fundamentadas em normas técnicas, a fim de que sua execução seja correta e planejada em cada estabelecimento (WATTSP, 2021).

No Brasil, as instalações elétricas prediais são normatizadas pela NBR - 5410 de 2004 que define os aspectos em torno de instalações elétricas de baixa tensão. Estas enquadram instalações de até 1000 V em corrente alternada com frequência de até 400 Hz e até 1500 V em corrente contínua (ASSOCIAÇÃO BRASILEIRA DE NORMA TÉCNICAS, 2004, pág. 1).

Nesta norma são definidos os pré-requisitos técnicos para que estas instalações tenham condições mínimas de segurança de modo a evitar incêndios, choque elétrico e danos a equipamentos elétricos. Ela aborda aspectos como o projeto inicial, a execução, a operação e manutenção, perpassando por temas como dimensionamento de condutores, iluminação e aterramento.

2.3 Automação

Com base nos aspectos de segurança e funcionalidade das instalações elétricas, é pertinente trazer os conceitos de automação, pois é um campo cada vez mais presente nos projetos elétricos atuais. A Associação Brasileira de Automação Residencial e Predial (Aureside), em suas análises, consta que o número de construções que contam com algum tipo de sistema automatizado varia entre 1,2 e 2,2 milhões. Em 2016, havia cerca de 300 mil lares com algum tipo de automação no país (CELERE, 2021).

Segundo Prudente (2011, p. 18): “A automação predial e residencial (em inglês, *home & building automation*) é a tecnologia que estuda a automação de um prédio ou habitação”. Em outras palavras, são estudos que visam identificar as tecnologias que

permitem tornar automático os processos e operações em uma edificação (PRUDENTE, 2011, pág. 18). Neste caso, automatizar os processos pode acarretar uma série de vantagens, tais como: maior conforto, simplicidade no cabeamento elétrico, maior segurança do sistema, maior versatilidade da instalação e seus funcionamentos, e, principalmente, permite uma economia notável no custo da energia elétrica na gestão da instalação (PRUDENTE, 2011, pág. 20).

Quanto à estruturação e princípios de funcionamento dos sistemas de automação, caracterizam-se pelo uso de um computador central para processamento de dados e comando de sistemas. Esta central de comandos é responsável pelo gerenciamento e coordenação de vários subsistemas da edificação, sejam eles sistema de aquecimento, condicionadores de ar, iluminação, sistema de segurança e monitoramento, entre outros (TOTVS, 2022).

A partir dos conceitos apresentados, entende-se a importância da automação na otimização de sistemas, o que contribui para uma eficiência dos recursos energéticos. Automatizar processos é utilizar a tecnologia em favor da eficiência, do conforto e da praticidade. Uma vez que os custos de operação e manutenção de uma edificação podem ser comprometidos por subsistemas associados, como climatização e iluminação. Ter um sistema de automação facilita as rotinas e contribui para redução significativa dos desperdícios com energia elétrica.

2.4 Internet das Coisas (IoT)

Conforme o avanço tecnológico ao longo dos anos, vê-se que em vários âmbitos de trabalho, seja acadêmico ou profissional, a presença da internet e demais tipos de conectividade são de suma importância para a execução de atividades rotineiras. Diante deste cenário, surge-se uma certa necessidade cada vez maior de dispositivos trabalharem de forma “inteligente”, ainda mais quando um sistema atua de forma automática. A automação de processos pode usufruir de tecnologias que necessitam de comunicações entre a central de processamento e os demais dispositivos embarcados.

No contexto em que há uma rede de objetos “inteligentes” que se comunicam entre si e com outros sistemas ou dispositivos, podendo coletar e receber dados em nuvem através da internet, isto se remete à *IoT (Internet of Things)*, ou Internet das Coisas (IBM, [s.d.]). Através da computação de baixo custo, torna-se vantajoso o uso

da *IoT* para o controle de multissistemas, no qual se evita, em grande parte, a intervenção humana para realização de tarefas.

Um sistema típico *IoT* baseia-se na coleta e troca de dados em tempo real. Isto é possível por meio de sensores, que coletam informações do ambiente, levam-nas através de nuvem para que softwares processem a informação. Serviços e algoritmos integrados utilizam a tecnologia *machine learning* (aprendizado de máquina) para analisar os dados coletados, retornando ao dispositivo de *IoT* a melhor tomada de decisão, podendo ser uma resposta de um atuador ou uma interação com usuário por meio de uma interface gráfica (ORACLE, 2014).

Aplicações de *IoT* para melhoria de processos já é uma realidade situada em múltiplas áreas, como no **setor da saúde** em que são utilizados dispositivos *IoT* para coletar informações sobre sinais vitais de cada paciente; no **setor industrial**, quando processos de fabricação são otimizados devido ao monitoramento do desempenho das máquinas em operação; no **transporte**, para otimização de rotas e rastreamento de remessas, entre outras aplicações (IBM, [s.d.]). Deste modo, a Internet das Coisas possibilita uma série de benefícios, principalmente no que diz respeito a melhorar a eficiência e produtividade, bem como tornar a tomada de decisões mais “inteligente” para gestão de processos. Em proveito dessas vantagens, *campus* universitários e edifícios comerciais podem usufruir das aplicações de *IoT* para aumentar a eficiência operacional, pois os dispositivos de *IoT* acarretam redução do consumo de energia e de custos de manutenção, e utilização eficiente dos locais de trabalho (AWS, [s.d.]).

O sistema proposto neste trabalho utiliza os princípios de *IoT* para sua estrutura de funcionamento. A ideia principal se baseia na utilização de dispositivos *IoT* para detectar a presença de pessoas dentro da sala de aula, captar e levar a informação para uma central de processamentos, que envie sinais de comandos para o acionamento dos condicionadores de ar e da iluminação.

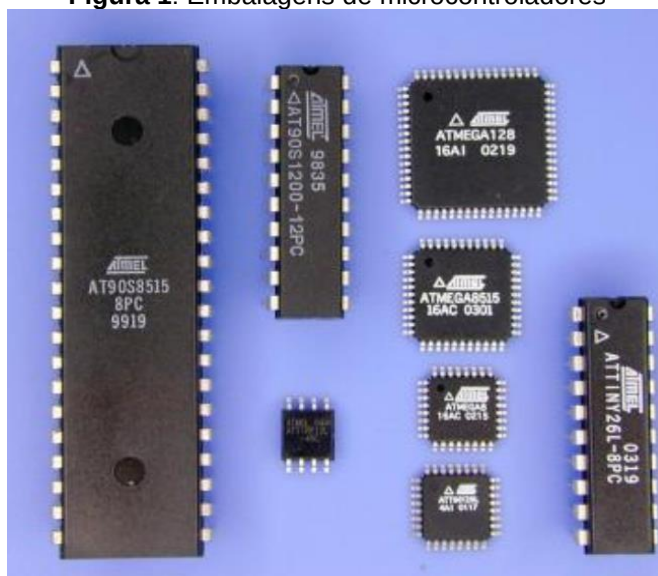
2.4.1 Microcontroladores

Um sistema de automação pode conter diversos dispositivos embarcados para realização de comandos. Haja vista a presença de ferramentas para coleta e transferência de dados, um equipamento que pode ser acionado como resposta de um sistema *IoT*, o faz devido uma informação previamente processada. Esta tarefa pode ser realizada por diversas tecnologias. No caso da praticidade e baixo custo,

destaca-se a utilização de microcontroladores. Estes são dispositivos com circuitos que integram diversas ferramentas, cujo seus periféricos são incorporados em um único componente. Os microcontroladores funcionam como um pequeno computador dentro de um chip, que podem realizar tarefas programáveis com alta capacidade (KASSOUF, 2023).

Em relação às características construtivas, um microcontrolador possui diversos componentes inseridos em um único circuito integrado, cujo qual é um pequeno chip capaz de armazenar diversos componentes eletrônicos, o que garante uma economia de espaço para instalação e aplicação. Além disso, um microcontrolador possui sua própria *CPU* (Unidade Central de Processamento) integrada, capaz de processar todos os dados fornecidos, além de avaliar problemas lógicos aplicados. Os dados coletados para processamento são armazenados e manuseados devido a presença de módulos de memória internos, como *RAM* (*Random Access Memory*) e *ROM* (*Read Only Memory*). Um microcontrolador usa um barramento interno para interagir com módulos em memória integrados (AWS, [s.d.]). A figura 1 apresenta alguns exemplos de microcontroladores que podem ser encontrados no mercado.

Figura 1: Embalagens de microcontroladores



Fonte: Kerschbaumer (2018)

A estrutura de um microcontrolador torna, ainda possível, a integração de componentes externos complementares, sejam conectores *USB*, bornes para acesso de terminais, módulos *Wi-fi* ou *Bluetooth*, entre outros (KANEHIRA, 2020). Estes recursos reforçam a versatilidade do uso de microcontroladores, uma vez que ele pode

ser amplamente usado para comandar diferentes circuitos e sistemas embarcados. Tais comandos são realizados dependente da lógica que está instalada no microcontrolador, podendo ser uma configuração de fábrica ou um algoritmo previamente programado e inserido na memória interna. Muitos deles possuem suporte a linguagens de programação comumente utilizadas no mercado (KANEHIRA, 2020).

A aplicação de microcontroladores se torna vantajosa, dada a sua versatilidade e baixo custo. Eles estão presentes em dispositivos eletrônicos cotidianos, como impressoras, calculadoras, relógios, televisores, entre outros. Sua aplicação também se estabelece na indústria, para sistemas de controle com escopo estreitamente definido (AWS, [s.d.]).

2.4.1.1 ESP32

Baseado na versatilidade dos microcontroladores, visto que são aplicáveis a diversas atividades de baixo custo, têm-se uma ampla gama de modelos no mercado. Neste contexto, o modelo ESP32 se torna uma excelente ferramenta para se atribuir a um sistema funcional de automação. Atrelado ao baixo custo e baixo consumo de energia, o ESP32 é um microcontrolador bastante robusto, versátil e confiável no mercado, essencialmente em termos de conectividade. Pode ser utilizado em projetos de *IoT*, robótica, automação residencial e outras situações que necessitam de conexões com internet (USINAINFO, [s.d.]).

Sua fabricação e desenvolvimento se deram através da empresa chinesa de tecnologia Espressif Systems, esta que disponibiliza uma série de módulos com o chip ESP32. O chip é responsável por todo o processamento de dados, e combina conectividades como *Wi-fi* e *Bluetooth* (ESPRESSIF, 2024, pág. 2). O módulo ESP32 pode ser dividido em diferentes modelos, no qual o mais utilizado consta com diversos periféricos e pinos de entrada e saída para controle de equipamentos, como displays, motores, sensores, módulos relés, emissor infravermelho, entre outros. Os diferentes tipos de modelos são produzidos por diferentes fabricantes, dentre os principais inclui-se: **ESP-WROOM-32** - módulo original da Espressif; **ESP32-PICO** - um módulo de encapsulamento menor, porém com recursos semelhantes ao ESP-WROOM-32; **ESP32-SOLO-1** - módulo mais compacto com poucos recursos comparado aos anteriores; **ESP32-CAM** - módulo com câmera embutida; entre outros modelos

(USINAINFO, [s.d.]). O módulo utilizado neste trabalho foi o ESP-WROOM-32, mostrado na Figura 2, que está contido em uma placa de desenvolvimento.

Figura 2: Módulo ESP-WROOM-32



Fonte: Mouser Eletronics (2022)

As especificações do módulo ESP-WROOM-32 podem ser encontradas diretamente nos documentos fornecidos pela Espressif. De acordo com o *datasheet* do produto, versão 3.4 de 2023, verifica-se todas as características, seja de conectividade, hardware, características físicas e elétricas, além de licenças e outros atributos relacionados ao módulo.

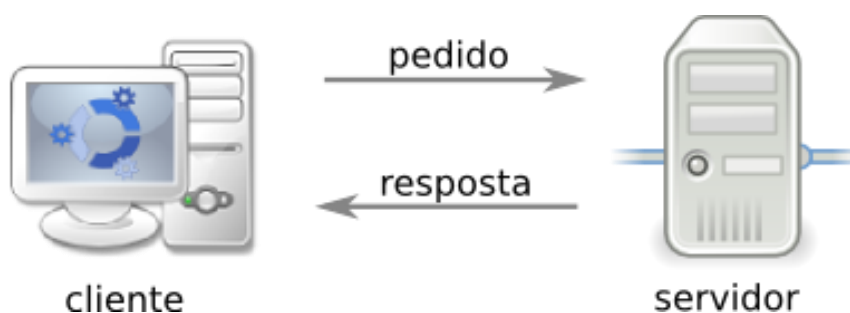
Para que seja possível integrar o módulo em diversos sistemas, é necessário a utilização de uma placa de prototipagem e desenvolvimento que contenha o módulo ESP32, trata-se da placa NodeMCU ESP32 que se baseia no microcontrolador ESP32 atrelado a diversos periféricos, como pinos de entrada/saída, para realização de projetos. Isto torna sua aplicação bastante versátil (USINAINFO, [s.d.]). Esta placa pode ser comumente encontrada no mercado, possuindo variações de diferentes fabricantes. A placa utilizada no sistema de automação proposto neste trabalho é a DOIT ESP32 Devkit V1. A Figura 3 apresenta um mapa da pinagem presente nesta variação.

páginas. A ação de uma página web que pode apontar para outras páginas através de links, nos leva ao conceito de hipertexto (BEOCK *et al.*, 2010, *apud* TANEMBAUM, 2003).

A partir deste cenário, surgiu diversos protocolos para regulamentar a transferência de dados na rede, como o caso do HTTP (*Hypertext Transfer Protocol* ou Protocolo de Transferência de Hipertexto), que padroniza a navegação de páginas por hipertexto. O HTTP se divide em dois programas, cliente e servidor, que são executados em máquinas diferentes, comunicando entre si através da estrutura de mensagens definida pelo protocolo (BEOCK *et al.*, 2010).

A Figura 4 ilustra um cenário no qual um cliente, podendo ser um dispositivo com conexão à rede ou navegador, solicita informações do servidor. De maneira prática, o cliente faz uma requisição de uma página web para o servidor, e este o responde com o protocolo HTTP, que permite a transmissão de informações através de uma conexão TCP (Protocolo de Controle de Transmissão) (MAZIERO, 2020).

Figura 4: Comunicação cliente servidor



Fonte: Maziero (2008)

O HTTP trabalha com critérios para realização da transferência de dados. A interação cliente-servidor pode ser feita por diferentes métodos de requisição disponíveis, por exemplo: **GET** - solicita algum recurso ou informação ao servidor; **PUT** - permite criar ou modificar dados do servidor; **POST** - permite enviar alguma informação ao servidor; **HEAD** - similar ao GET, porém parte da resposta é retornada; entre outros métodos (BEOCK *et al.*, 2010).

O sistema de automação proposto neste trabalho integra-se com o protocolo HTTP, uma vez que utiliza os seus recursos de comunicação para o tráfego de informações entre os seus subsistemas. As informações de como essa comunicação ocorre estão detalhadas na seção 4 do presente trabalho.

2.4.3 API

Aplicações diversa podem ser usadas para solicitar informações na rede, se comunicando com um servidor que possa conter os dados requisitados. De maneira análoga, tem-se o exemplo: um aplicativo de dados meteorológicos, presente na maioria dos smartphones, mostra ao usuário os dados climáticos de qualquer região que o indivíduo solicitar. Estas informações são fornecidas pelo sistema do instituto meteorológico correspondente, contidas em seu banco de dados. O aplicativo solicita a informação para o instituto, que a devolve no formato desejado. Esta interação é feita comumente através de uma API (*Application Programming Interface ou Interface de Programação de Aplicação*) (AWS, [s.d.]).

Segundo a IBM (s.d.), por definição “uma API (interface de programação de aplicativos) é um conjunto de normas que possibilitam a comunicação segura entre diferentes aplicativos e sistemas”. Ela funciona como uma ponte de acesso intermediário a informações específicas, normalmente contidas em grandes servidores. Como exemplo de aplicação, uma API pode auxiliar uma empresa a fornecer características e funcionalidades de seus aplicativos para desenvolvedores externos, consumidores finais, parceiros de negócio, entre outros.

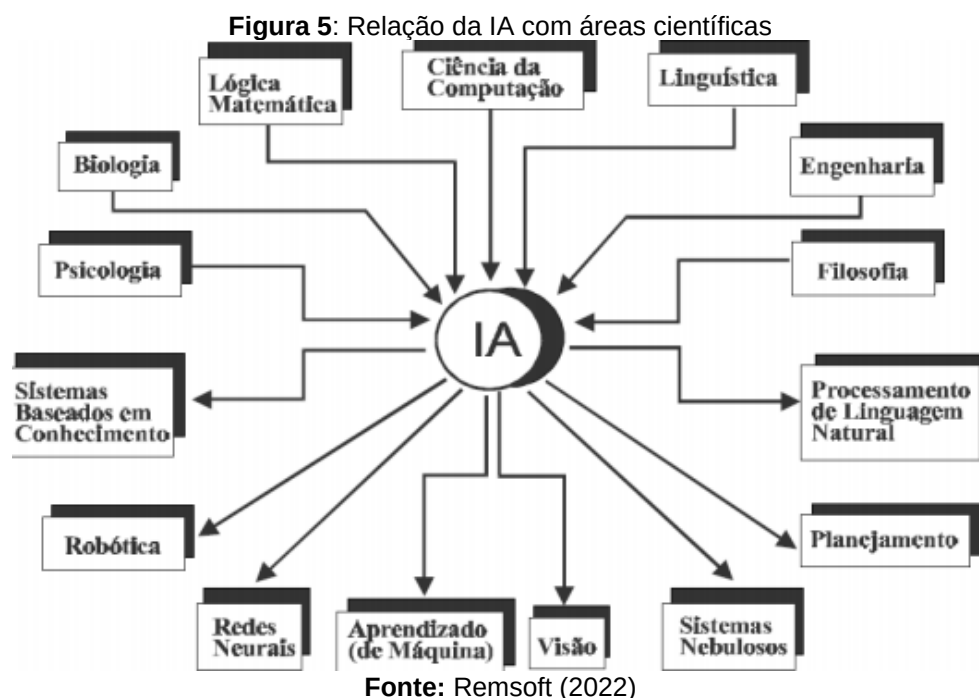
A respeito do princípio de funcionamento, uma API também possui uma arquitetura base de cliente-servidor. As funcionalidades mais comuns são APIs REST, no qual um cliente requisita informações ao servidor, cuja requisições se definem em funções como GET, PUT, DELETE, entre outras, trocando dados através de HTTP. Outra funcionalidade são as APIs WebSocket, como informa a AWS (s.d.):

A API de WebSocket é outro desenvolvimento de API da Web moderno que usa objetos JSON para transmitir dados. Uma API WebSocket oferece suporte à comunicação bidirecional entre aplicativos cliente e o servidor. O servidor pode enviar mensagens de retorno de chamada a clientes conectados, tornando-o mais eficiente que a API REST.

2.5 Inteligência Artificial (IA)

Automatizar processos pode exigir aplicação de várias tecnologias para garantir uma melhor efetividade da execução proposta, haja vista que a implementação de diversos equipamentos *IoT* podem aprimorar sistemas, agregando conectividade ao sistema de automação, o que pode torná-lo robusto e confiável. No proveito, existem importantes ferramentas que se baseiam em inteligência artificial, capazes de impulsionar o desenvolvimento de um sistema automático.

Inteligência Artificial (IA) é a área de estudo que visa o conjunto de tecnologias capazes de executar funções avançadas, de forma inteligente. O conceito se desenvolve em 4 linhas de pensamento: I – sistemas que pensam como seres humanos; II – sistemas que atuam como seres humanos; III – sistemas que pensam racionalmente; IV – sistemas que atuam racionalmente (GOMES, 2010). Por abordar um conceito amplo, a IA se relaciona com diversas áreas científicas, vide Figura 5.



Para se obter uma melhoria e desempenho aprimorado de uma IA, a grande maioria delas se permite ser “treinada”. Uma base de dados nova pode ser introduzida como fonte de aprendizado. Para uma melhor classificação, uma inteligência artificial pode ter um estágio de **máquina reativa**, se comportando de forma limitada conforme regras pré-programadas; ou pode ter um estágio de **memória limitada**, utilizando

memória para se aprimorar a partir de treinamentos com novos dados, por meio de redes neurais ou *machine learning* (GOOGLE, [s.d.]).

A utilização de inteligência artificial pode ser fundamental e benéfica em diversos aspectos, como automação de processos, redução do erro humano, solução para tarefas repetitivas, consistência e rapidez para tomada de decisões, pesquisa e desenvolvimento acelerados, entre outros (GOOGLE, [s.d.]). Uma importante aplicação de uma IA está no processamento de imagens, uma vez que se obtêm diversas classificações e identificações em uma análise de imagem. Esta é a aplicação base para o sistema de automação proposto neste trabalho, no qual têm-se um *framework* de detecção, baseado em inteligência artificial para analisar imagens das salas de aula, a fim de detectar a presença de pessoas no local.

2.5.1 YOLOv8

A ferramenta utilizada neste trabalho para processamento de imagens e detecção de pessoas dentro das salas de aula é a YOLO, versão 8. Desenvolvido pela empresa Ultralytics. YOLOv8 é um *framework* de IA que suporta múltiplas tarefas de visão computacional, cuja principal característica se baseia na detecção de objetos e segmentação de imagens em tempo real (ULTRALYTICS, 2024).

O YOLOv8, não é somente um modelo de detecção de objetos, é uma estrutura versátil concebida para abranger todo o ciclo de vida dos modelos de aprendizagem automática, partindo da coleta de dados até a validação das informações. O modelo possui diversos modos que oferecem flexibilidade e eficiência para várias tarefas e utilizações. Entender como cada modo funciona acarreta melhoria no desempenho da análise de imagens. Jocher (2024), fundador da Ultralytics, apresenta um panorama geral a respeito dos modos presentes no YOLOv8, são eles:

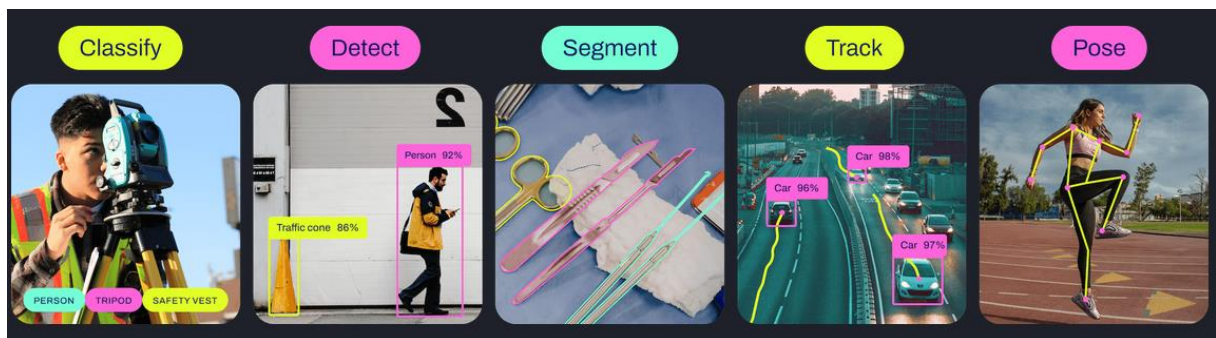
- **Train:** Modo de treinamento. A partir de um conjunto de dados, otimiza-se os parâmetros do modelo para obter melhor precisão nos resultados de análise de imagens;
- **Val:** Modo utilizado para validar o modelo após o treinamento. Ajusta e aprimora o desempenho, validando os dados de treinamento;
- **Predict:** Modo utilizado para processar a análise a partir de um arquivo específico para realizar as inferências, seja uma imagem ou vídeo informadas

pelo utilizador. Este é o modo que foi utilizado no sistema de automação proposto neste trabalho;

- **Export:** Modo que permite o modelo ser convertido para outro formato, a fim de ser utilizado em outros softwares ou dispositivos de hardware;
- **Track:** Modo utilizado para detecção de objetos em tempo real. O modelo é carregado em cima de um arquivo fornecido pelo utilizador, para retornar as informações em tempo real;
- **Benchmark:** Modo que permite os utilizadores optarem pelo melhor formato de exportação do modelo YOLOv8 para a utilização ideal, baseado nos requisitos de velocidade e precisão.

Como dito anteriormente, o YOLOv8 suporta múltiplas tarefas. Sua aplicação se torna ampla, uma vez que se tem diversas ferramentas de utilização, no qual inclui-se: detecção de objetos em um quadro de imagem ou vídeo com alta precisão; segmentação de regiões de uma imagem, com base no seu conteúdo; classificação de imagens com base em seu conteúdo; detecção de pontos-chaves para estimativa de poses; detecção de objetos orientados com alta precisão e velocidade. Vê-se a exemplificação de cada função na Figura 6:

Figura 6: Funções do YOLOv8



Fonte: Ultralytics (2024)

Os modelos do YOLOv8 permitem serem treinados e validados. Existe um conjunto de dados que abrange uma grande variedade de imagens diferentes, para categorizar uma gama de objetos sujeitos a detecção. O conjunto de dados de detecção, segmentação e legendagem de objetos em grande escala do YOLOv8 é chamado de COCO (*Common Objects in Context*) (Jocher, 2024). Dentro do conjunto de dados, têm-se os *coco-names*, que são categorias de objetos que o modelo é capaz de detectar, como “*person*”, “*laptop*”, “*chair*”, entre outros. Desta forma, o

modelo possibilita a sua utilização para o sistema de automação proposto, uma vez que pode detectar a categoria “pessoas” em sala de aula.

3 METODOLOGIA

O trabalho desenvolvido tem como metodologia uma abordagem prática, com o intuito de gerar conhecimento voltado para aplicações concretas e solucionar problemas específicos. Dessa forma, os estudos desenvolvidos condicionam-se para a solução do problema referente ao desperdício de energia elétrica.

Inicialmente, foi realizado uma abordagem de casos semelhantes ao enfrentado nesta pesquisa para melhor compreensão do cenário problemático. Além disso, fez-se uma pesquisa bibliográfica, estruturando um estudo que norteia os conhecimentos a serem aplicados para a compreensão do sistema de automação proposto, visando embasar e entender como a abordagem científica e acadêmica se aplica a respeito do assunto.

Posteriormente, o trabalho passou pela etapa de desenvolvimento de um sistema de automação para atuar nos condicionadores de ar e nas seções de iluminação. O sistema se baseia no uso de Inteligência Artificial atrelado a um sistema de câmeras para detectar pessoas dentro do ambiente, para que os microcontroladores atrelados a esta detecção sejam comunicados para realizar a tomada de decisões, a fim de ativar ou desativar aparelhos de ar-condicionado e iluminação da sala de aula. O devido sistema de automação foi aplicado, instalado temporariamente na sala 1S do bloco 16 do IFTO – Campus Palmas para ser testado e validado.

Por fim, foi realizado uma pesquisa de campo para obter informações rotineiras do campus, a fim de estimar o desperdício de energia elétrica ocasionado pelas salas de aula que contenham ar-condicionado e iluminação ligados em ausência de pessoas. As estimativas de desperdício serviram como base para certificar a economia gerada pelo sistema proposto, visando a eficiência energética, uma vez que evita cenários comprometedores e reduz a dependência humana.

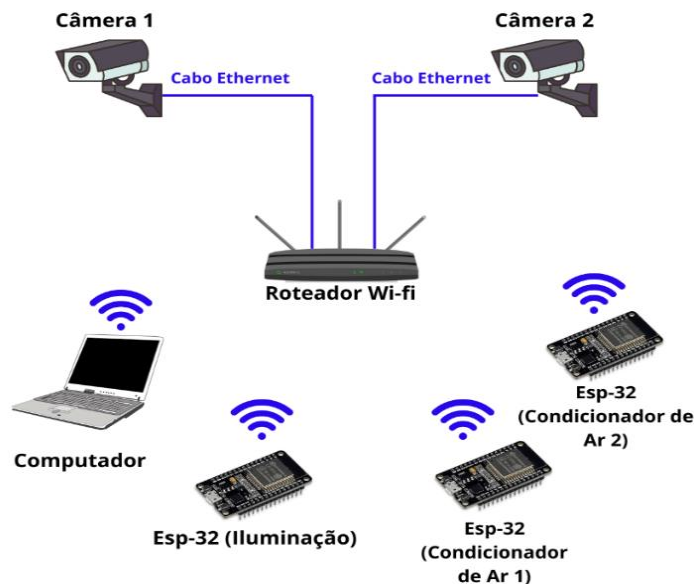
4 DESENVOLVIMENTO E FUNCIONAMENTO DO SISTEMA DE AUTOMAÇÃO

O sistema de automação proposto se baseia na integração dos seguintes subsistemas: **Sistema de Detecção (Principal e Primário)**, **Sistema de Controle da Iluminação (Secundário)**, **Sistema de Controle do Condicionador de Ar (Secundário)** e **Sistema de Acompanhamento (Terciário)**. Todos eles se comunicam entre si através do **protocolo HTTP**, amplamente utilizado nas mais variadas aplicações relacionadas à internet por todo o mundo (BEOCK *et al.*, 2010).

4.1 Rede de Comunicação

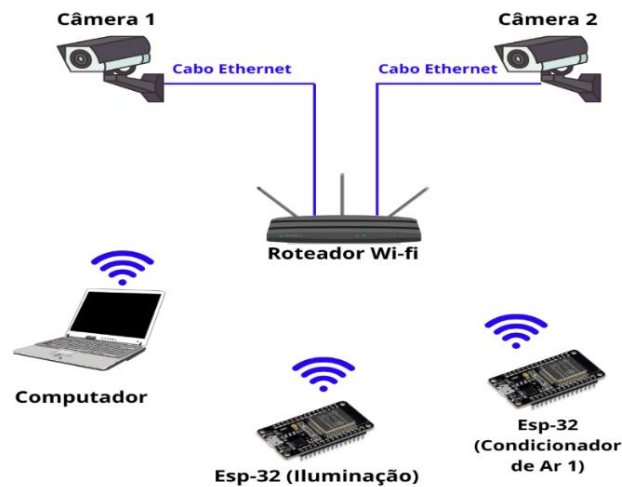
Para realizar a comunicação entre sistemas, foi criada uma rede local *offline* sediada em um roteador genérico da marca ZTE modelo ZXHN F660, mostrado na Figura 9. Este equipamento já era de posse dos autores e foi reaproveitado para este projeto. Os componentes do projeto se conectam a rede conforme Figuras 7 e 8 abaixo:

Figura 7: Esquema de conexão de dispositivos do sistema de automação para salas de aula com dois condicionadores de ar



Fonte: Acervo dos Autores (2024)

Figura 8: Esquema de conexão de dispositivos do sistema de automação para salas de aula com apenas um condicionador de ar



Fonte: Acervo dos Autores (2024)

Apesar da rede ser alocada em modo *offline*, esta escolha foi apenas para propósitos de teste e apresentação, sendo totalmente possível a alocação em modo *online*, desde que se observe os devidos cuidados com a cibersegurança.

Figura 9: Roteador utilizado ZTE ZHXX F660



Fonte: Acervo dos Autores (2024)

4.2 Sistema de Detecção (Primário e Principal)

O Sistema de Detecção recebe a classificação de **principal** e **primário** por realizar o processamento mais complexo do sistema de automação proposto e ser

responsável pela determinação dos estados dos demais subsistemas. Devido a sua robustez, foi necessário desenvolvê-lo em um equipamento que comportasse com uma boa margem de segurança todas as suas necessidades. Para isso, foi escolhido um notebook padrão como ambiente de desenvolvimento e execução, que possui as seguintes características principais:

- Modelo: Acer Aspire 3 A315-42G-R6FZ;
- Processador: AMD Ryzen 5 3500U;
- Memória RAM: 16 GB DDR4 2400Mhz;
- Armazenamento: HD de 1TB 5400 rpm + SSD NVme M.2 de 256GB;
- Placa de Vídeo Dedicada: AMD Radeon 540x;
- Sistema Operacional: Windows 10 Home versão 22H2

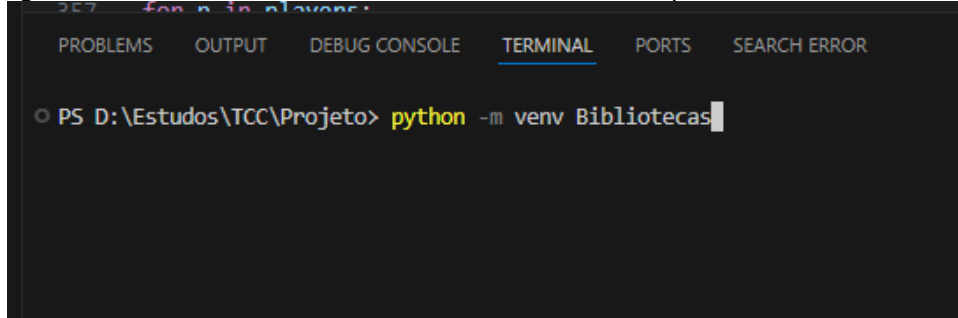
Vale ressaltar que as especificações apresentadas não são os requisitos mínimos necessários para o desenvolvimento do Sistema de Detecção. Na seção de resultados deste trabalho será abordada uma discussão mais aprofundada sobre o assunto.

4.2.1 Programação

O Sistema de Detecção foi desenvolvido utilizando o Ambiente de Desenvolvimento Integrado (do inglês IDE ou Integrated Development Environment) **Visual Studio Code ou VS Code** em sua versão 1.89.0. A linguagem de programação empregada foi o **Python** em sua versão 3.11.7, devido a sua compatibilidade com a biblioteca da inteligência artificial utilizada (YOLOv8). Outra opção seria o **C++**, porém há pouco conteúdo sobre disponível em comparação com o **Python** na época de publicação deste trabalho.

Com os programas de desenvolvimento devidamente baixados e instalados, prosseguiu-se com a instalação das bibliotecas adicionais a serem utilizadas no projeto. Para isso, criou-se um ambiente virtual de modo a isolar estas bibliotecas em diretório específico separado do restante do sistema operacional para facilitar seus registros e manipulações. Foi criada uma pasta para armazenar todo o conteúdo do projeto e com essa pasta aberta no VS Code foi inserido o comando no terminal `"python -m venv Bibliotecas"` para criar o ambiente virtual, onde "Bibliotecas" é o caminho para o diretório escolhido para armazenar as informações do ambiente (ANDRADE, 2023). A Figura 10 mostra o comando executado:

Figura 10: Comando executado no terminal do VS Code para criar ambiente virtual



Fonte: Acervo dos Autores (2024)

As principais bibliotecas adicionais utilizadas no projeto foram a “Ultralytics” instalada via terminal pelo comando “*pip install ultralytics*” e a “VLC” instalada via terminal pelo comando “*pip install python-vlc*”. Estas, por sua vez, instalam automaticamente bibliotecas complementares necessárias para o seu correto funcionamento. No **Apêndice I** estão listadas todas as bibliotecas utilizadas e suas respectivas versões, enfatizando que as bibliotecas não citadas nesta lista são padrões do python e encontram-se discriminadas no código-fonte do Sistema de Detecção, **Apêndice A**. A “Ultralytics” é responsável por gerenciar todos os comandos e funções relacionadas ao YOLOv8 e a “VLC” nos permite gerenciar a capturas de imagem das câmeras.

Com estes pré-requisitos cumpridos, o sistema foi desenvolvido e testado continuamente até se chegar no resultado discriminado no **Apêndice A**.

4.2.2 Princípio de Funcionamento e Funcionalidades Adicionais

O princípio de funcionamento do Sistema de Detecção, se baseia na captura de fotos do ambiente e análise destas pelo modelo de inteligência artificial do **YOLOv8**. Para realizar as capturas são posicionadas 2 câmeras apontadas para o ambiente de modo que seja possível visualizar todas as regiões. As imagens em tempo real destas câmeras são integradas com o algoritmo em **Python (Apêndice A)**, programado para realizar a captura de uma determinada quantidade de fotos com um intervalo de tempo entre si. Cada foto capturada é armazenada e analisada pelo modelo de inteligência artificial do YOLOv8 em seu modo **Predict**, onde ele irá determinar quais objetos estão presentes na cena com base nos nomes **COCO**. Com

os objetos determinados e listados pelo algoritmo, será feita a avaliação para verificar se algum destes objetos equivale ao objeto “Person” ou “Pessoa”. Se for detectado o objeto “Pessoa”, significa que a inteligência artificial detectou uma pessoa no ambiente e partir disso é feito o controle dos sistemas de iluminação e climatização.

Cada conjunto de fotos capturadas com o intervalo de tempo entre si, é chamado no algoritmo de “**Ciclo**” e é possível determinar um tempo entre cada ciclo para que esta análise não seja realizada a todo momento, economizando, assim, poder de processamento. Isso foi pensado com base na lógica de usos de uma sala de aula, haja vista que é um ambiente que não possui grande fluxo de pessoas: os alunos entram, assistem as aulas por um longo período e saem, não sendo necessário uma análise constante do estado do ambiente.

Os parâmetros como quantidade de fotos, tempo entre fotos e tempo entre ciclos são determinados pelo operador do sistema ao início do algoritmo, onde é recomendado que não se escolha tempos ou quantidades exacerbadas, considerando-se que isso pode afetar os tempos de resposta do sistema e por fim, a experiência dos usuários. O sistema apresentou ótimos resultados para os seguintes parâmetros:

- 3 a 6 fotos por ciclo;
- 1 a 30 segundos entre cada foto;
- 3 a 10 minutos entre cada ciclo;

Vale ressaltar também que os resultados da análise pelo **YOLOv8**, são salvos também no mesmo diretório em que as fotos são salvas, sendo possível determinar, no começo do algoritmo, quantos ciclos de análise serão armazenados para evitar acúmulo de dados.

Para mais detalhes sobre o funcionamento do Sistema de Detecção foi criado um fluxograma descrevendo as etapas de processamento. O fluxograma está discriminado no **Apêndice E**. Ao analisar este e os demais fluxogramas, deve-se considerar as seguintes informações:

- **Bloco de Processamento:** utilizado para descrever um ponto de processamento no sistema, vide Figura 11:

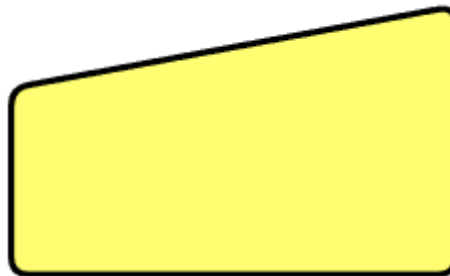
Figura 11: Bloco de Processamento



Fonte: Acervo dos Autores (2024)

- **Bloco de Entrada Manual:** utilizado para descrever um ponto em que o usuário deve inserir dados de forma manual no sistema, vide Figura 12:

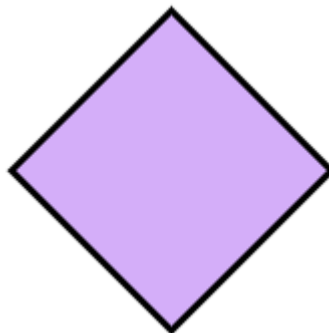
Figura 12: Bloco de Entrada Manual



Fonte: Acervo dos Autores (2024)

- **Bloco de Decisão:** utilizado para descrever um ponto em que há uma tomada de decisão no sistema, vide Figura 13:

Figura 13: Bloco de Decisão



Fonte: Acervo dos Autores (2024)

- **Bloco de Delay ou Atraso:** utilizado para descrever um ponto em que o sistema deve aguardar um determinado período, vide Figura 14:

Figura 14: Bloco de Delay



Fonte: Acervo dos Autores (2024)

- **Bloco de Início:** indica o início do sistema, vide Figura 15:

Figura 15: Bloco de Início



Fonte: Acervo dos Autores (2024)

- **Bloco de Fim:** indica o fim do sistema, vide Figura 16:

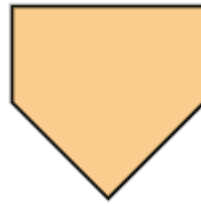
Figura 16: Bloco de Fim



Fonte: Acervo dos Autores (2024)

- **Bloco Conector Fora de Página:** indica que fluxo continua em outra página ou seção. São identificados com números e números iguais se conectam, vide Figura 17:

Figura 17: Bloco Conector Fora de Página



Fonte: Acervo dos Autores (2024)

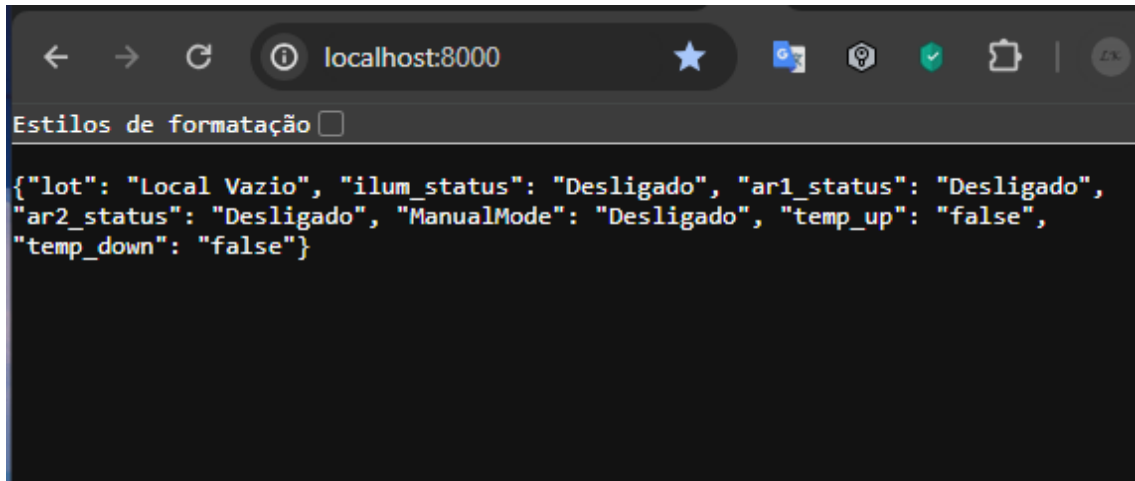
Além das funcionalidades informadas, o Sistema de Detecção ainda possui outra funcionalidade extra que se trata dos **Protocolos de Hora**. De forma simples, os Protocolos de Horas são dias ou horários em que o Sistema de Automação deve entrar em modo de suspensão, ou seja, o sistema não deve atuar mantendo o estado dos dispositivos controlados em “Desligado” até que se saia do protocolo. A função visa mitigar qualquer possibilidade de desperdício de energia elétrica em momentos em que não há necessidade de atuação dos sistemas controlados, como é o caso de intervalos e períodos em que não há aula no local. Os protocolos definidos estão listados abaixo:

- Segundas a sextas-feiras:
 - Protocolo estará ativo das 12:00 às 13:00 e das 23:00 às 07:00 (horário local).
- Sábados e Domingos:
 - Protocolo estará ativo durante todo o dia.

Para trafegar as informações dos resultados das detecções bem como das demais variáveis de controle para os outros sistemas, o Sistema de Detecção cria um servidor HTTP que atua como API para os demais sistemas e disponibiliza essas informações no endereço “localhost:8000” formatadas em *Notação de Objetos JavaScript* ou *JavaScript Object Notation* (JSON), onde cada variável é estruturada em um par Key (Chave ou Identificador) e Value (Valor), como é mostrado na Figura 18. Essa estrutura serve para facilitar o tráfego e legibilidade das informações via HTTP entre os sistemas programados em diferentes linguagens (SOUZA, 2020). Os outros sistemas consultam e capturam o estado destas variáveis para utilizar em seus respectivos funcionamentos. Vale ressaltar que o termo “*localhost*” se refere ao

endereço do IP da máquina que está executando o servidor (no caso, o notebook) e em outras seções deste trabalho ele será referido como “*ip_do_servidor*”.

Figura 18: JSON servido no endereço localhost:8000



Fonte: Acervo dos Autores (2024)

4.2.3 Montagem

Na captura de imagens do ambiente, foram empregadas 2 câmeras IP (Internet Protocol) conectadas via cabo ethernet ao roteador. A primeira câmera é da marca Ezviz modelo C3N a qual possui como principais características: resolução de captura de 1080 pixels, visão noturna, compatibilidade com conexão ethernet e Wi-fi 2.4 Ghz, e resistência a água e poeira (EZVIZ, 2024). A segunda câmera é da marca Intelbras modelo VIP 1130 B G3 com resolução de 720 pixels, visão noturna, compatível apenas com conexão ethernet e resistente a água e poeira (INTELBRAS, 2024). Um fator importante para o bom funcionamento deste projeto é que ambas as câmeras, mostradas na Figura 19, são compatíveis com o protocolo Real Time Streaming Protocol (RTSP) ou com o próprio Protocolo HTTP, pois isto permite a captura das imagens via link direto, possibilitando assim, a integração delas diretamente com os algoritmos do projeto, como é visto no **Apêndice A**.

Figura 19: Câmera EZVIZ C3N (à esquerda) e câmera Intelbras VIP 1130 B G3 (à direita)



Fonte: Acervo dos Autores (2024)

4.3 Sistema de Controle da Iluminação (Secundário)

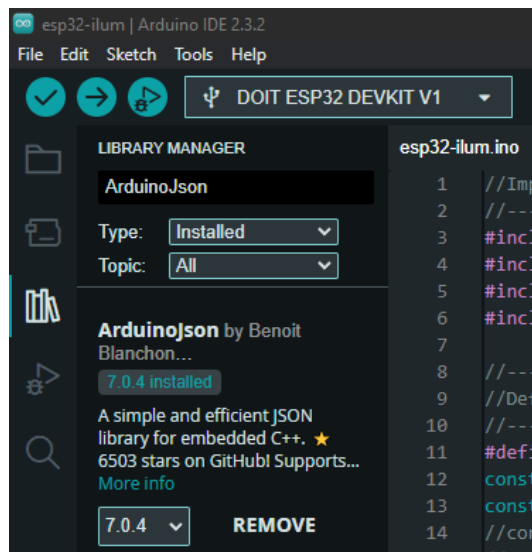
O Sistema de Controle da Iluminação é considerado secundário pois ele reage aos comandos enviados pelo Sistema de Detecção (Primário). Apesar disto, ele ainda possui algumas funcionalidades extras que implementam e completam o funcionamento do sistema como um todo. Nas seções a seguir serão apresentadas estas funcionalidades bem como suas construções.

Para realizar tanto o controle da iluminação quanto do condicionador de ar, foram implementados os módulos **ESP32** devido a sua alta compatibilidade com os propósitos do projeto: fácil programação, tamanho reduzido, compatibilidade com conexão Wi-fi, múltiplas portas de entrada e saída e a excelente rentabilidade. A variante do módulo utilizada é a *DOIT ESP32 Devkit V1*.

4.3.1 Programação

Para programar os módulos *ESP32* foi utilizado o *Arduino IDE* na sua versão 2.3.2, em conjunto com sua linguagem **C/C++** nativa, destinada a programação de módulos em geral. Conforme tutorial de Random Nerd Tutorials (2023), realizou-se a instalação do gerenciador do módulo e devidas configurações iniciais. No gerenciador de bibliotecas da própria IDE, instalou-se a biblioteca adicional “*ArduinoJson.h*” para manipular o JSON consultado no endereço “*localhost:8000*”, vide Figura 20. As demais bibliotecas utilizadas não citadas são padrões do gerenciador do módulo.

Figura 20: Biblioteca “*ArduinoJson.h*” instalada via Arduino IDE



Fonte: Acervo dos Autores (2024)

Por conseguinte, o algoritmo descrito no **Apêndice B** foi carregado no ESP32 que faz parte do módulo de controle. Detalhes da montagem estão descritos em seção a seguir.

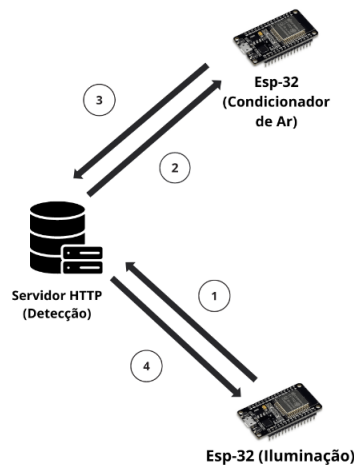
4.3.2 Princípio de Funcionamento e Funcionalidades Adicionais

O **ESP32** presente neste sistema é a peça central de todo o seu funcionamento. De modo simples, esse módulo se conecta via Wi-fi da rede do projeto e consulta a cada um segundo em “*ip_do_servidor:8000*”, os resultados das variáveis disponibilizadas pelo servidor HTTP do Sistema de Detecção. Nesse ponto, a variável que será o alvo principal da consulta é a variável “lot”, pois é ela que vai determinar se pessoas foram detectadas no ambiente ou se o local está vazio. Essa consulta é realizada através de uma requisição **GET**, onde o estado de “lot” é armazenado em variável correspondente já dentro do sistema. Dependendo do valor que “lot” assumir o sistema enviará comandos para o módulo relé que ativa ou desativa a iluminação. Por exemplo:

- Para “lot” igual a “Pessoas Detectadas” – **liga a iluminação.**
- Para “lot” igual a “Local Vazio” – **desliga a iluminação.**
- Para “lot” igual a “Suspensa” – Sistema de Detecção em Protocolo de Hora – **desliga a iluminação.**

Além destas funcionalidades padrões, foram implementadas funcionalidades extras visando complementar o Sistema de Automação com um todo. Uma delas é a possibilidade de os usuários conseguirem aumentar ou diminuir a temperatura do condicionador de ar usando o botão de funcionalidades do sistema. Para diminuir a temperatura em 1 grau Celsius basta pressioná-lo por um tempo que vai de 1 a 3 segundos e para aumentar a temperatura em 1 grau Celsius basta pressioná-lo por um tempo que vai de 3 a 5 segundos. A Figura 21 mostra como essa funcionalidade atua, destacando-se que é a mesma estrutura para aumentar ou diminuir a temperatura, o que muda são apenas as variáveis envolvidas (“*temp_up*” para aumentar e “*temp_down*” para diminuir):

Figura 21: Esquema de funcionamento da função para controle de temperatura



Fonte: Acervo dos Autores (2024)

1. Em caso de aumento de temperatura, no ESP-32 do Sistema de Controle da Iluminação, a variável “*temp_up*” que possui o valor “*false*” recebe valor “*true*” e é enviada para o servidor HTTP usando requisição **PUT**.
2. O servidor disponibiliza o novo valor dessa variável em “*localhost:8000*”, onde esse valor é capturado e armazenado usando **GET** em variável correspondente no ESP32 do Sistema de Controle do Condicionador de Ar.
3. Como o novo valor da variável “*temp_up*” é “*true*”, o ESP32 do Sistema de Controle do Condicionador de Ar é programado para, nesse caso, enviar um comando infravermelho para o condicionador de ar aumentar a temperatura. Após finalizar o comando, ele retorna “*temp_up*” para “*false*” e à envia de volta para o servidor HTTP usando **PUT**.

4. O servidor disponibiliza o novo valor dessa variável em “localhost:8000”, onde esse valor é capturado e armazenado usando **GET** na variável “temp_up” do ESP32 do Sistema de Controle da Iluminação, finalizando assim o comando.

Durante a construção do protótipo foi identificado que dependendo do momento em que o ciclo de detecção de pessoas está, pode levar um certo tempo até que a iluminação ative. Isso pode gerar um certo desconforto para quem está entrando na sala de aula após ela estar vazia, principalmente em períodos noturnos, onde a visibilidade é reduzida. Pensando nisso, foi implementando um sensor ultrassônico junto a central de controle da iluminação apontado para a porta da sala de aula para medir a distância. Se essa distância for menor do que 1,20 metros (distância medida entre sensor e região da porta, vide Figura 22), significa que algo passou em frente do sensor e, dessa forma, a iluminação é ligada por um período tempo suficiente para que o Sistema de Detecção detecte pessoas na sala, solucionando assim o problema. Recomenda-se que o tempo seja configurado de modo que seja igual ou maior do que o tempo de um ciclo de análise, para dar tempo de qualquer ciclo ativo ser finalizado e essa funcionalidade atuar corretamente. Para demais informações sobre este sistema, no **Apêndice F** consta o fluxograma de funcionalidades completo para análise.

Figura 22: Medição realizada entre o sensor ultrassônico e a região da porta da sala de aula

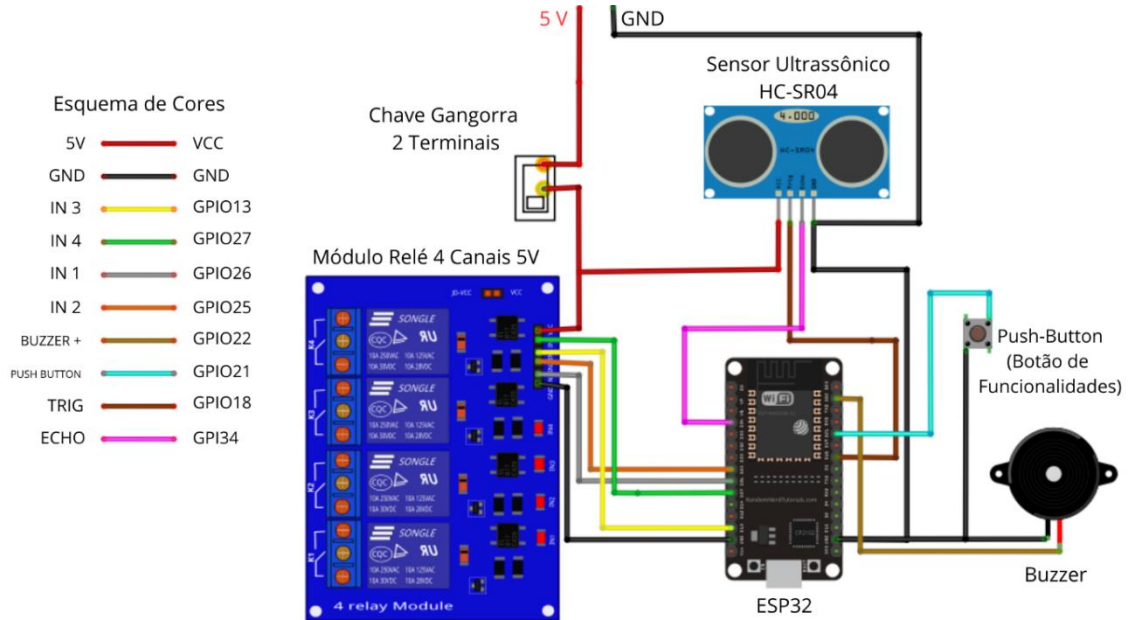


Fonte: Acervo dos Autores (2024)

4.3.3 Montagem

O esquema de controle de iluminação é mostrado nas Figuras 23 e 24:

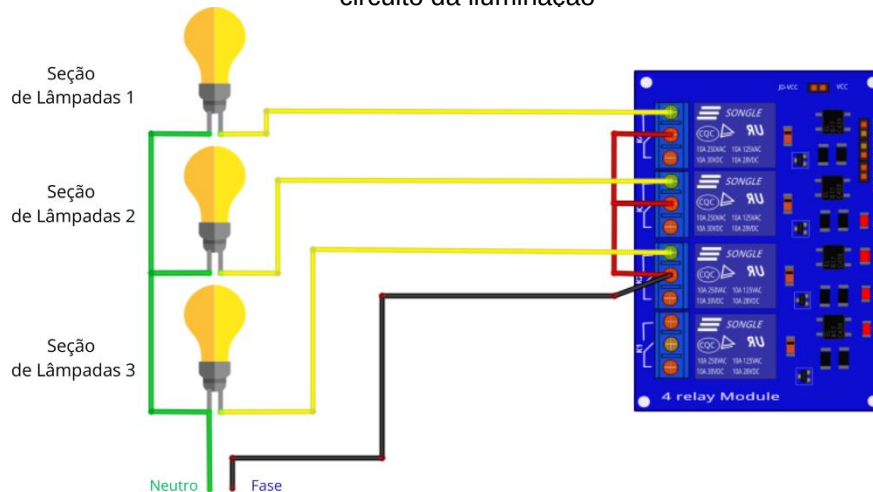
Figura 23: Esquemático do Sistema de Controle da Iluminação



Fonte: Acervo dos Autores (2024)

No controle da iluminação foi empregado o módulo relé de 4 terminais pois boa parte das salas presentes no bloco 14 a 17 possuem 3 seções de lâmpadas, onde é empregado um relé para cada seção. Para conexão do relé com o circuito presente na sala, seguiu-se o esquemático da Figura 24:

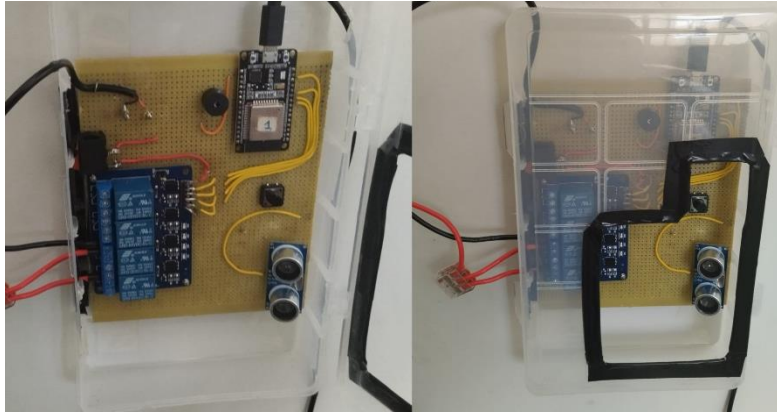
Figura 24: Esquemático de conexão do módulo relé 4 canais com circuito da iluminação



Fonte: Acervo dos Autores (2024)

O esquemático é também aplicável para salas com 1 ou 2 seções de lâmpadas, basta desconsiderar seções inexistentes. O resultado obtido é mostrado na Figura 25:

Figura 25: Módulo do Sistema de Controle da Iluminação



Fonte: Acervo dos Autores (2024)

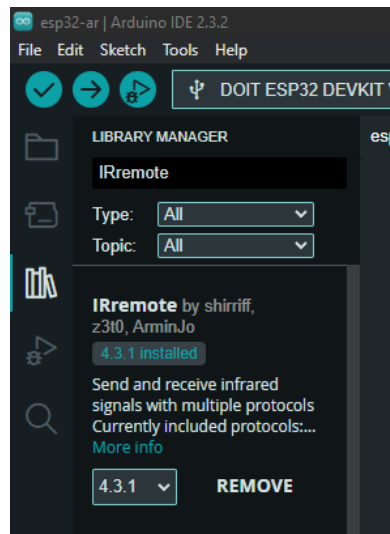
4.4 Sistema de Controle do Condicionador de Ar (Secundário)

O Sistema de Controle do Condicionador de Ar funciona de modo muito semelhante ao da Iluminação, realizando os mesmos controles de “liga” e “desliga” baseado nos estados da variável “/ot”, só que agora para o condicionador de ar. Este sistema acaba sendo mais simples por não possuir funcionalidades extras, além de responder ao controle de temperatura explicado anteriormente. No **Apêndice G** consta o fluxograma de funcionamento desse sistema com mais informações.

4.4.1 Programação

Devido ao fato de também utilizar o módulo **ESP32**, este sistema foi programado com o mesmo *Arduino IDE* citado anteriormente, sendo adicionado apenas a biblioteca “*IRremote.hpp*” para realizar a captura e emissão de códigos infravermelhos do condicionador de ar. Essa biblioteca foi baixada no próprio gerenciador de bibliotecas da IDE, conforme Figura 26:

Figura 26: Biblioteca “*IRremote.hpp*” instalada via Arduino IDE



Fonte: Acervo dos Autores (2024)

Para controlar o modelo de condicionador de ar utilizado na sala de aula onde seriam realizados os testes, foi necessário realizar uma visita ao local para capturar os códigos infravermelho do equipamento. A sala escolhida foi a sala do bloco 16 número 1S e a visita foi realizada no dia 15 de maio de 2024 no período noturno. O equipamento de ar-condicionado utilizado nas salas é da marca *Carrier* modelo 38CCV036515MC, conforme Figura 28. Os códigos infravermelhos foram coletados com o auxílio o algoritmo exemplo “*ReceiveandSend.ino*” disponibilizado pela biblioteca “*IRremote.hpp*” conforme Arduino-IRremote (2012?), e do controle remoto do equipamento. A Figura 27 registra o processo:

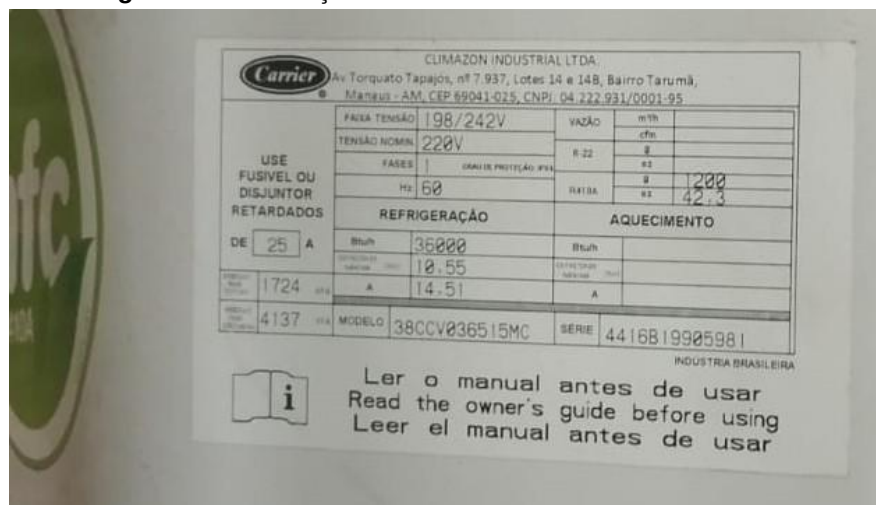
Figura 27: Processo de captura de códigos infravermelho



Fonte: Acervo dos Autores (2024)

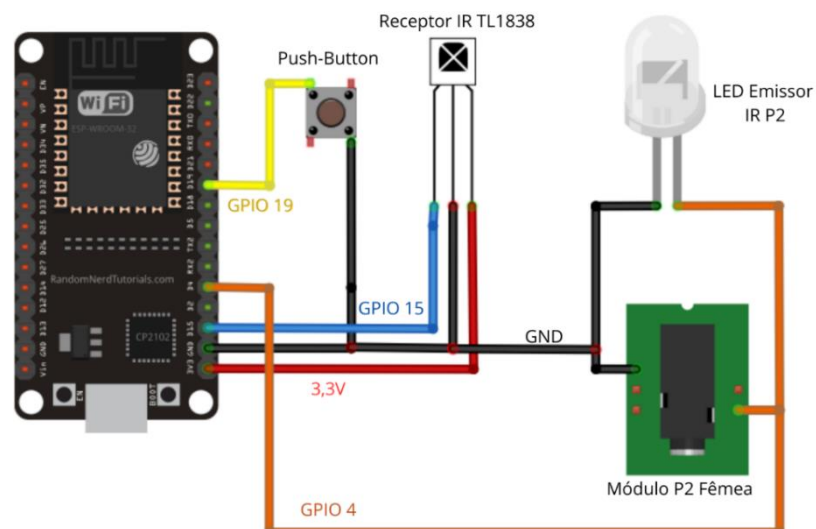
O algoritmo foi carregado para um dos ESP32 disponíveis, onde montou-se o circuito da Figura 29. O receptor infravermelho captura o código raw infravermelho do controle remoto e com um clique no botão, dispara esse mesmo código pelo emissor para o equipamento, com isso foi possível testar as distâncias de efeito da emissão. O módulo P2 fêmea se faz necessário porque foi utilizado um cabo extensor onde numa ponta há um conector P2 macho e na outra um led emissor infravermelho (IR), como é mostrado nas figuras 30 e 32.

Figura 28: Informações levantadas sobre o condicionador de ar



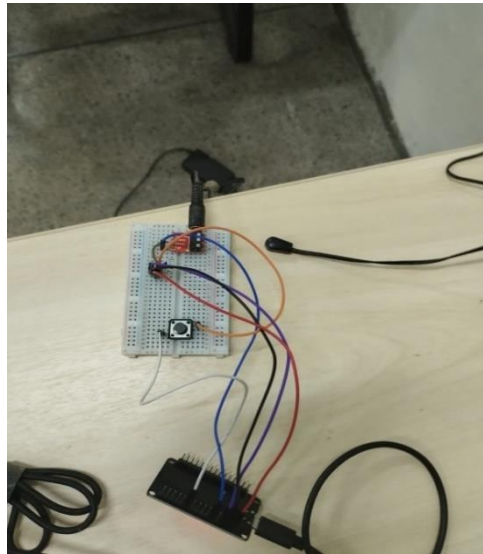
Fonte: Acervo dos Autores (2024)

Figura 29: Esquemático do circuito de captura de códigos infravermelho



Fonte: Acervo dos Autores (2024)

Figura 30: Circuito de captura de códigos infravermelho montado



Fonte: Acervo dos Autores (2024)

Os códigos capturados foram “Desligar”, “Ligar”, “Aumentar Temperatura” e “Diminuir Temperatura”, e eles estão registrados no algoritmo deste sistema no **Apêndice C**.

4.4.2 Montagem

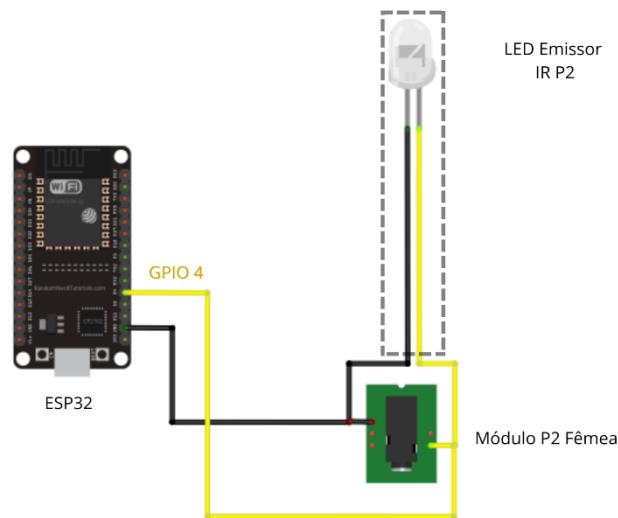
A montagem deste sistema é a mais simples dentro os demais, onde é necessário apenas fazer a conexão entre o módulo P2 fêmea e o segundo ESP32, conforme Figura 32. Na figura 31 é mostrado o extensor P2 utilizado para ser o emissor IR.

Figura 31: Cabo extensor emissor IR utilizado



Fonte: Acervo dos Autores (2024)

Figura 31: Esquemático do circuito do Sistema de Controle do Condicionador de Ar



Fonte: Acervo dos Autores (2024)

Ao final da montagem, obteve-se o resultado da Figura 33:

Figura 33: Módulo do Sistema de Controle do Condicionador de Ar



Fonte: Acervo dos Autores (2024)

4.5 Sistema de Acompanhamento (Terciário)

O Sistema de Acompanhamento é um recurso adicional ao sistema de automação proposto que visa apenas disponibilizar as informações trafegadas para o usuário final. Ele é considerado um sistema terciário porque não influencia no funcionamento do sistema com um todo, sendo apenas um recurso visual para os estados dos sistemas secundários e primário, bem como algumas outras informações adicionais.

4.5.1 Programação e Princípio de Funcionamento

A programação deste sistema foi realizada utilizando o *framework* **Flask**, pela facilidade de implementação e compatibilidade com o **Python**. O algoritmo foi escrito via **VS Code**, em arquivo separado do Sistema de Detecção cujo qual é executado paralelamente a ele. Seu funcionamento é como o de um site comum: usando o *framework*, o algoritmo inicia um novo servidor HTTP em endereço “localhost:5000” (perceba que a porta é diferente da utilizada na API, ou seja, é um servidor diferente com apenas o mesmo *host*). Este servidor serve 3 páginas diferentes:

- No endereço “localhost:5000” ou “localhost:5000/home”: serve a página do painel da automação;
- No endereço “localhost:5000/autores”: serve a página contendo as informações sobre os autores;
- No endereço “localhost:5000/manual”: serve a página contendo as informações sobre o sistema e como utilizá-lo;

Cada página servida possui um arquivo HTML atrelado para posicionar os componentes da página. Cada arquivo desse se integra com outro arquivo CSS para adicionar cores e personalizações a página. Na página do painel da automação ainda há o adicional de um script utilizando a linguagem de programação **JavaScript**, que em conjunto com o arquivo HTML e CSS tem a função de consultar os estados das variáveis disponibilizadas pela API em “localhost:8000” e atualizá-las em tempo real no painel, aplicando personalizações visuais dependendo dos estados destas variáveis. O fluxograma do **Apêndice H** detalha este funcionamento e os algoritmos na íntegra tanto do servidor utilizando *Flask* como dos arquivos HTML, CSS e JavaScript estão descritos no **Apêndice D**.

5 RESULTADOS

Nesta seção são apresentados os resultados obtidos através da aplicação das metodologias descritas anteriormente, bem como as discussões pertinentes a respeito do andamento e estado do projeto.

5.1 Teste Final do Sistema

O teste final do sistema de automação proposto foi realizado na sala escolhida, (Sala 1S do Bloco 16), no dia 30 de maio de 2024. Durante a montagem, enfrentamos duas principais dificuldades:

1. Dos 4 relés disponíveis no módulo relé, 3 deles (relés 1, 2 e 4), não estavam atuando como deveriam. Apesar de receberem o sinal correto do ESP32, nas saídas os estados não eram comutados. Para contornar o problema, ligamos ambas as seções de iluminação no único relé disponível. A Figura 34 ilustra a adaptação feita:

Figura 34: Adaptação realizada frente ao problema com os relés



Fonte: Acervo dos Autores (2024)

2. O aparelho de ar condicionado não estava respondendo a todos os comandos, mesmo se posicionássemos o emissor infravermelho a uma distância de 1 metro do condicionador de ar, o que é considerado uma distância curta. Para solucionar este problema, posicionamos o emissor imediatamente ao lado do receptor infravermelho do aparelho. A Figura 35 mostra como foi feito esse posicionamento:

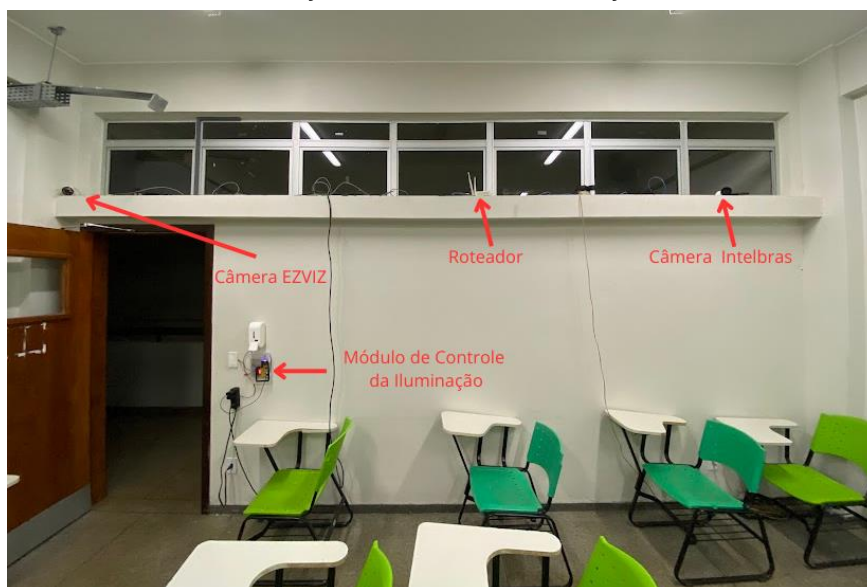
Figura 35: Posicionamento do emissor infravermelho no condicionador de ar



Fonte: Acervo dos Autores (2024)

Apesar das dificuldades enfrentadas, ao final dos testes o sistema funcionou conforme o previsto, executando com êxito todas as funcionalidades implementadas. Aproveitou-se a própria estrutura da sala para posicionar as câmeras, o roteador e seus respectivos cabeamentos de rede e alimentação, evitando assim, furações desnecessárias nas paredes da sala. A Figura 36 mostra esse posicionamento:

Figura 36: Posicionamento dos equipamentos dos Sistemas de Detecção e Controle da Iluminação



Fonte: Acervo dos Autores (2024)

A Figura 37 mostra mais detalhes sobre a montagem do sistema na sala:

Figura 37: Posicionamento dos Sistemas de Controle da Iluminação e do Condicionador de Ar



Fonte: Acervo dos Autores (2024)

Durantes os testes, as inferências realizadas pelo modelo de IA do YOLOv8 ao objeto “Pessoa” (“Person”) foram bastante precisas cumprindo com êxito os requisitos necessários para o projeto. Esta precisão foi tamanha que dispensa a necessidade de treinamento do modelo de inteligência artificial utilizado perante a aplicação no projeto. As Figuras 38 a 41 mostram alguns dos resultados destas inferências:

Figura 38: Resultados das inferências realizadas pela IA do YOLOv8 em foto capturada na câmera Intelbras (esquerda – foto capturada; direita - foto analisada)



Fonte: Acervo dos Autores (2024)

Figura 39: Resultados das inferências realizadas pela IA do YOLOv8 em foto capturada na câmera EZVIZ (em cima – foto capturada; embaixo - foto analisada)



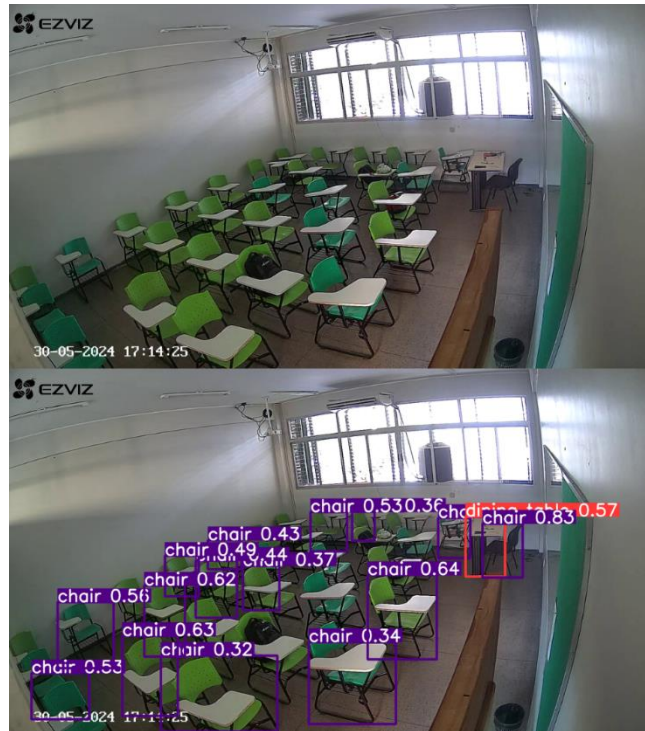
Fonte: Acervo dos Autores (2024)

Figura 40: Resultados das inferências realizadas pela IA do YOLOv8 em foto capturada na câmera Intelbras (esquerda – foto capturada; direita - foto analisada)



Fonte: Acervo dos Autores (2024)

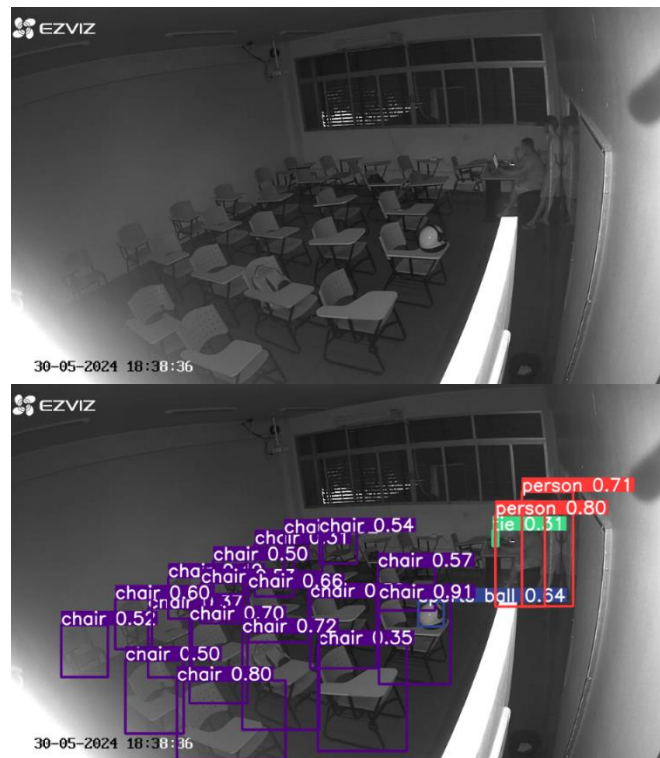
Figura 41: Resultados das inferências realizadas pela IA do YOLOv8 em foto capturada na câmera EZVIZ (em cima – foto capturada; embaixo - foto analisada)



Fonte: Acervo dos Autores (2024)

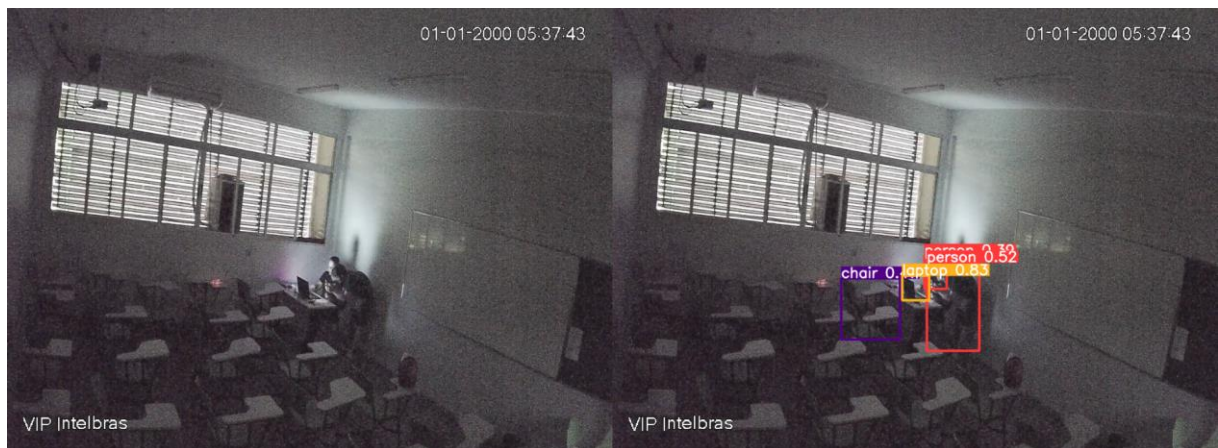
Mesmo com o modo “Visão Noturna” das câmeras ativado, as inferências realizadas foram satisfatórias para detecção de pessoas, conforme é mostrado nas Figuras 42 e 43:

Figura 42: Resultados das inferências realizadas pela IA do YOLOv8 em foto capturada na câmera EZVIZ com Visão Noturna Ativa (em cima – foto capturada; embaixo - foto analisada)



Fonte: Acervo dos Autores (2024)

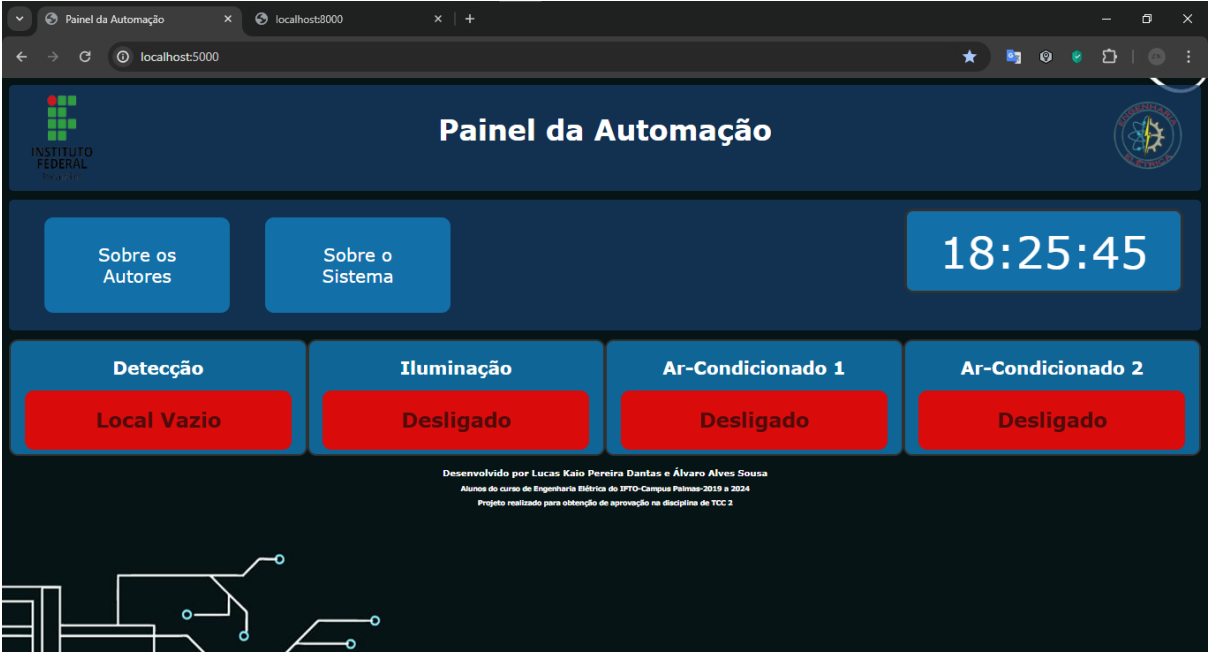
Figura 43: Resultados das inferências realizadas pela IA do YOLOv8 em foto capturada na câmera Intelbras com Visão Noturna Ativa (esquerda – foto capturada; direita - foto analisada)



Fonte: Acervo dos Autores (2024)

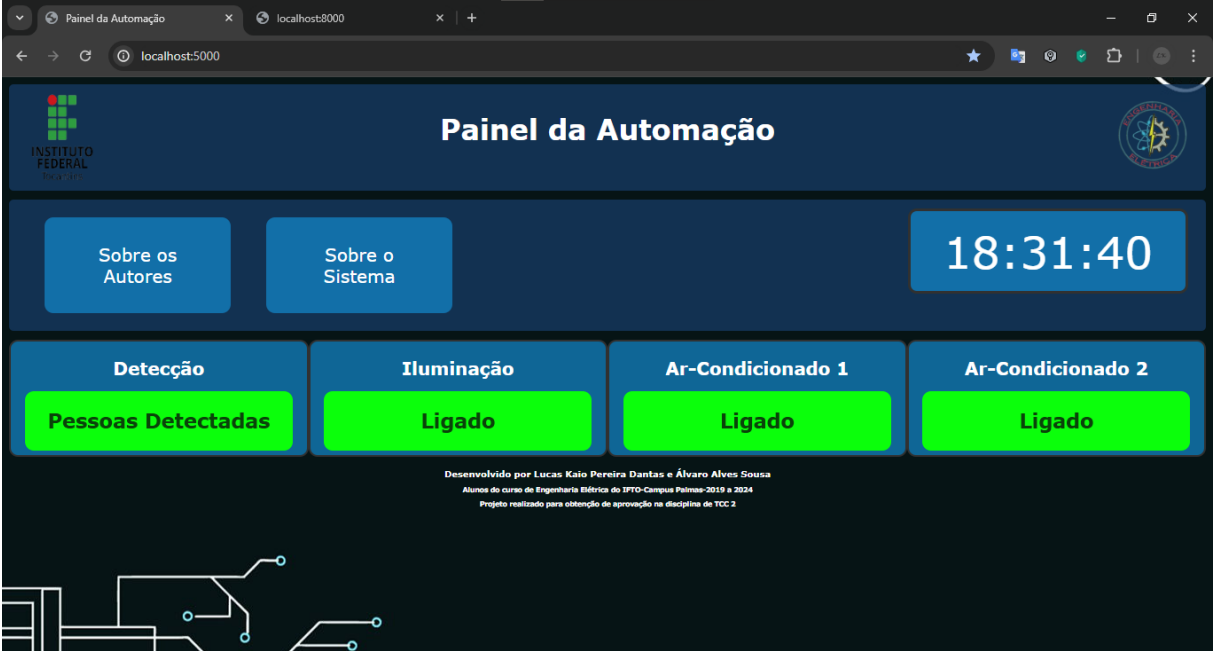
A interface gráfica do Sistema de Acompanhamento também funcionou conforme o esperado para cada estado do sistema. As Figuras 44 a 49 mostram o comportamento do painel da automação bem como as páginas com informações sobre os autores e sobre o sistema:

Figura 44: Página “Painel da Automação” do Sistema de Acompanhamento quando não são detectadas pessoas no ambiente



Fonte: Acervo dos Autores (2024)

Figura 45: Página “Painel da Automação” do Sistema de Acompanhamento quando são detectadas pessoas no ambiente



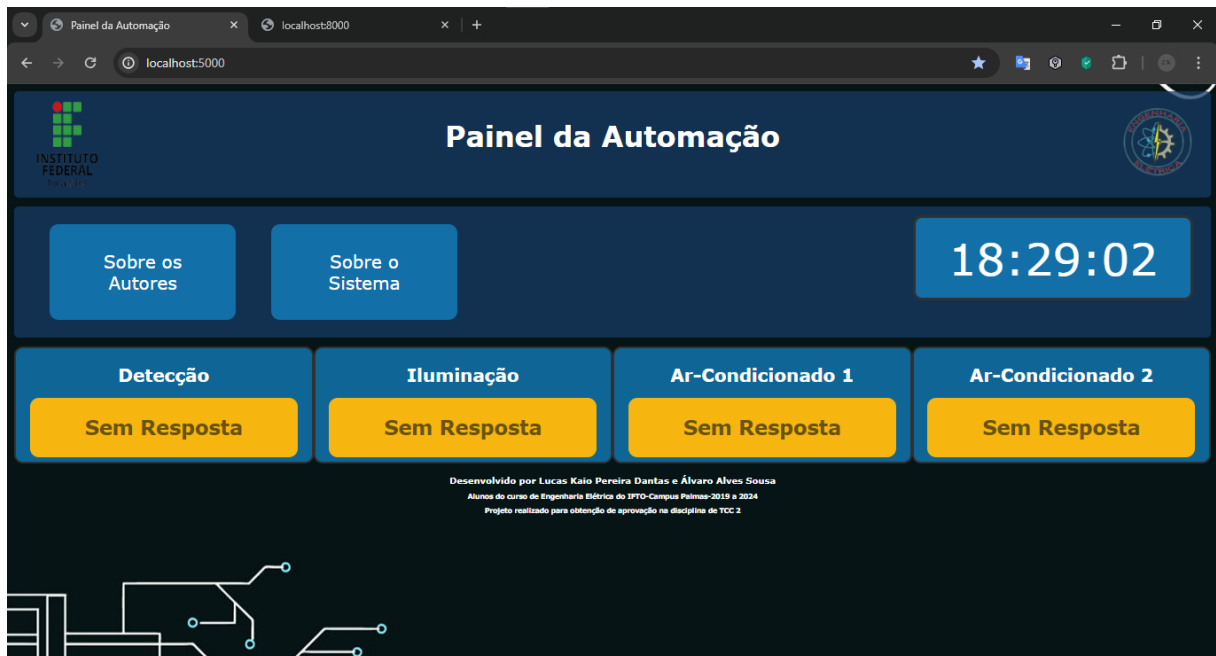
Fonte: Acervo dos Autores (2024)

Figura 46: Página “Painel da Automação” do Sistema de Acompanhamento quando o sistema está em “Protocolo de Hora”



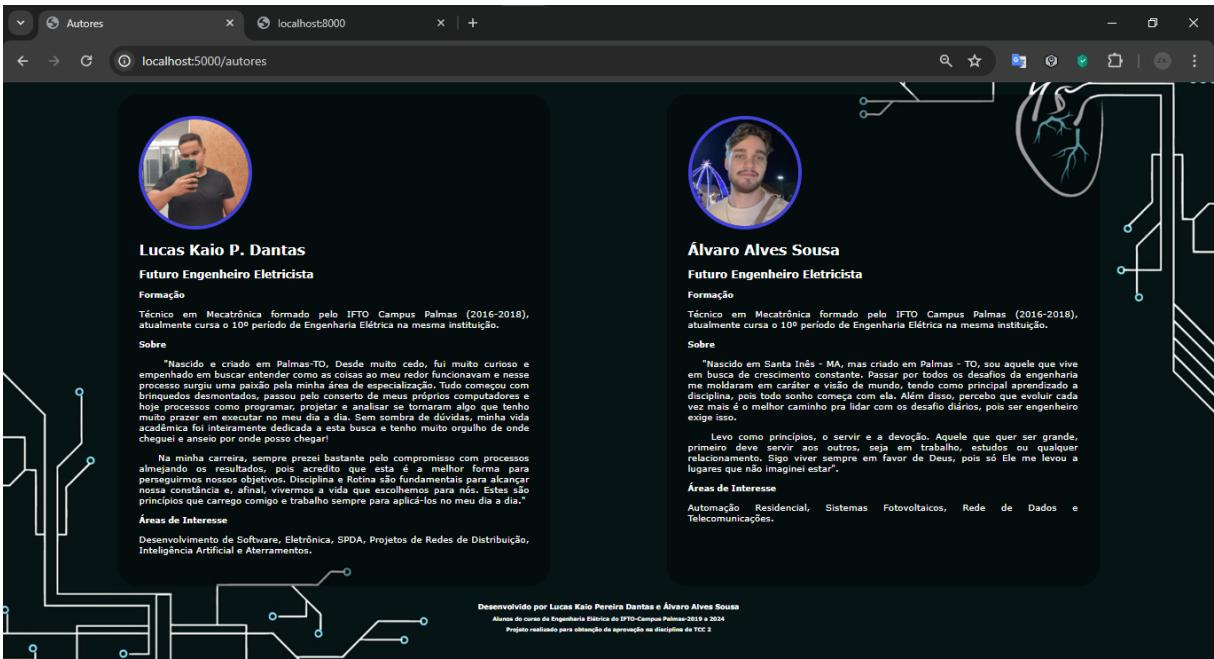
Fonte: Acervo dos Autores (2024)

Figura 47: Página “Painel da Automação” do Sistema de Acompanhamento quando o servidor da API não responde (fora do ar)



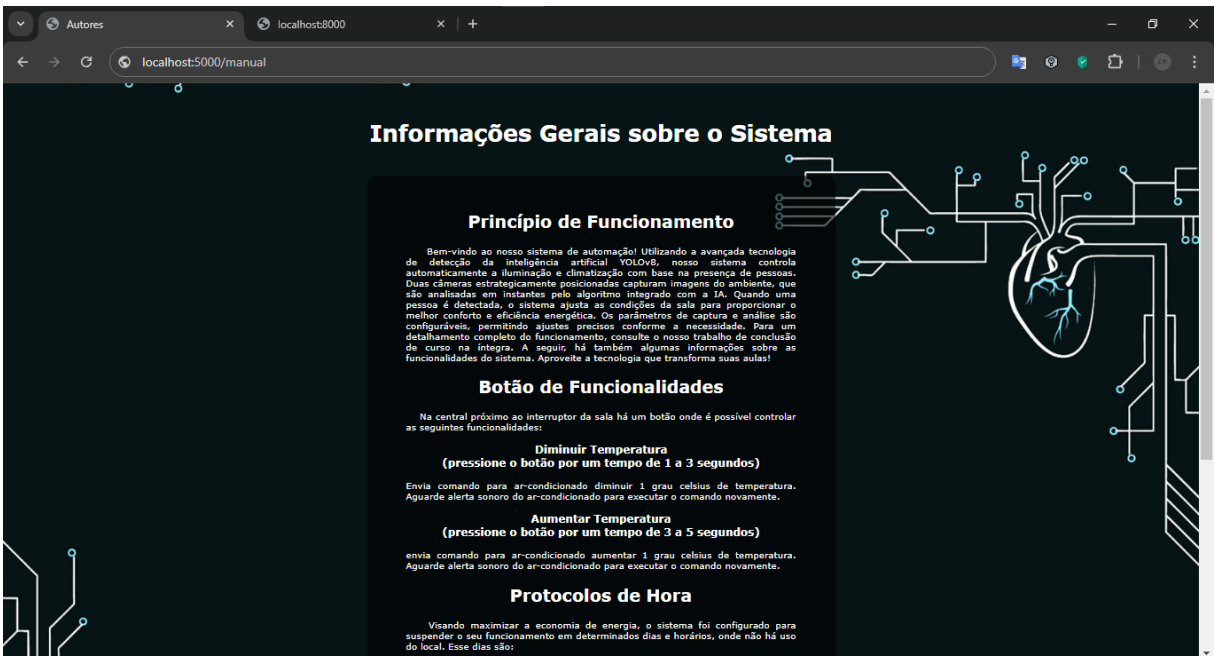
Fonte: Acervo dos Autores (2024)

Figura 48: Página “Autores” do Sistema de Acompanhamento



Fonte: Acervo dos Autores (2024)

Figura 49: Página “Manual” do Sistema de Acompanhamento



Fonte: Acervo dos Autores (2024)

5.2 Estimativas de Economia

Para se obter uma base da economia de energia gerada pelo sistema de automação, primeiro foi necessário entender como se dá a rotina de desperdícios de energia elétrica ocasionados pelo mau uso dos condicionadores de ar e da iluminação nas salas de aula. Para isso, fez-se uma pesquisa de campo para coletar as devidas informações. A pesquisa baseou-se em uma pequena entrevista com os servidores terceirizados, responsáveis por ligar e desligar os aparelhos de ar-condicionado e as luminárias de cada sala de aula, tanto do período matutino, vespertino e noturno.

A pesquisa teve por finalidade entender como funcionam as rotinas dos servidores para ligar e desligar os aparelhos, e ter uma base da frequência de situações de desperdício, ou seja, quando os condicionadores de ar e iluminação estão ligados sem que haja presença de pessoas no local. A entrevista constou com os principais questionamentos:

1. Qual o horário se inicia a rota para **ligar** os aparelhos das salas de aula?
2. Qual o horário se inicia a rota para **desligar** os aparelhos das salas de aula?
3. Ao final da rota, quantas salas de aula, em média, são encontradas com ar-condicionado e iluminação ligados sem que haja pessoas no local?

Ao todo, foram entrevistados 4 funcionários: 1 pessoa responsável pela rotina da manhã e tarde nos blocos 14 a 17; 1 pessoa responsável pela rotina da noite nos blocos 14 a 17; 1 pessoa responsável pela rotina da manhã e tarde nos blocos 3 a 13; 1 pessoa responsável pela rotina da noite nos blocos 3 a 13. Desta forma, obteve-se os seguintes resultados:

- Horário para ligar: 7:00 no período matutino, 13:00 no período vespertino e 18:00 no período noturno;
- Horário para desligar: 12:00 no período matutino, 17:50 no período vespertino e 23:00 no período noturno
- Quantidade média de salas de aula com ar-condicionado e luz ligados ao final das rotas de desligamento: 9 salas no período matutino, 10 salas no período vespertino e 10 salas no período noturno.

Em média, o início e fim das aulas se dá pelos horários 7:30 as 11:40, 13:30 as 17:40 e 18:40 as 22:30. Considerando o cenário de desperdício baseado na ausência de pessoas, percebe-se que rotineiramente já se tem um desperdício de energia elétrica devido os aparelhos estarem ligados antes das aulas iniciarem e após elas

finalizarem. Dessa forma, desconsiderando os horários úteis (aula), em um dia, os condicionadores de ar e iluminação ficam ligados durante 160 minutos (2h40min) em cenário de desperdício.

Para estimar o desperdício de energia elétrica gerado, considera-se que as salas possuem, em média, 9 luminárias do tipo fluorescente tubular e 2 condicionadores de ar de 30000 BTU, cujo consumo médio individual seja de 20W e 3,6kW respectivamente, conforme NDU 001 da Energisa (2023). Para o caso de uma sala de aula que fica em cenário de desperdício, considera-se os cálculos para consumo médio mensal de energia elétrica (tendo em vista o mês com 22 dias letivos):

1. Iluminação

$$9 \text{ luminárias} \times 20W \times 2,67h \text{ (160 minutos)} \times 22 \text{ dias} = 10,57 \text{ kWh/mês}$$

2. Condicionadores de ar

$$2 \text{ condicionadores de ar} \times 3,6kW \times 2,67h \times 22 \text{ dias} = 422,93 \text{ kWh/mês}$$

3. Total

$$10,57 + 422,93 = \underline{433,5 \text{ kWh/mês}}$$

Com base na pesquisa realizada, para o caso da quantidade de salas de aula em cenário de desperdício, no total 29 salas durante o dia, considera-se os seguintes cálculos para desperdício de energia elétrica:

$$433,5 \text{ kWh/mês (consumo de uma sala de aula)} \times 29 \text{ salas} = \underline{12571,5 \text{ kWh/mês}}$$

A partir dos cálculos realizados, percebe-se que há uma estimativa de desperdício de 12571,5 kWh/mês devido o cenário em que os condicionadores de ar e iluminação permanecem ligados sem necessidade. Isto eleva consideravelmente os gastos do IFTO – Campus Palmas com relação à fatura de energia elétrica, principalmente para os cenários em horário de ponta, o que compromete um maior valor da tarifa de energia.

Para contribuir com a eficiência energética da instituição, faz-se necessária a implementação de medidas na redução de desperdício de energia. A Relação de desperdício citada acima pode ser evitada ao aplicar-se o sistema de automação proposto neste trabalho. De forma eficiente, pode-se evitar cenários de desperdício, uma vez que o sistema garante que a iluminação e a climatização permaneçam ligadas somente na presença de pessoas dentro da sala de aula, trazendo uma economia considerável para o campus.

5.3 Custos

Como mostrado anteriormente, a sala de aula em que foram realizados os testes do sistema possui apenas duas seções de iluminação e um condicionador de ar. Devido a estas características, os custos a serem considerados para implantação do sistema com essas condições estão descritos na Tabela 1. Nas colunas de “Preço Unitário Pago” e “Preço Total Pago” alguns itens estão com o valor igual a 0, pois eles são produtos que não foram comprados. A forma de obtenção destes produtos está descrita na Tabela 3.

Tabela 1: Custos dos materiais para de implementação do Sistema de Automação para salas de aula contendo 1 condicionador de ar e 2 seções de iluminação

Descrição	QTD	Preço Unitário Pago (R\$)	Preço Total Pago (R\$)	Preço Unitário de Mercado (R\$)	Preço Total de Mercado (R\$)
Módulo DOIT ESP32 Devkit V1	2	48,99	97,98	48,99	97,98
Módulo Relé 4 Canais 5V	1	24,90	24,90	24,90	24,90
Módulo Sensor Ultrassônico HC-SR04	1	14,99	14,99	14,99	14,99
Módulo P2 Fêmea 3,5 mm	1	6,38	6,38	6,38	6,38
Chave Tátil (Push Button) 12x12x8,5 mm	1	0,75	0,75	0,75	0,75
Buzzer Ativo 3V	1	3,50	3,50	3,50	3,50
Chave Gangorra 2 Terminais 2 Posições	1	1,50	1,50	1,50	1,50
Conector Wago 221 3 vias	1	3,50	3,50	3,50	3,50
Conector Wago 221 2 vias	3	2,50	7,50	2,50	7,50
Caixa Organizadora 23x14 cm	1	17,99	17,99	17,99	17,99
Jumper	NA	0,00	0,00	0,00	0,00
Placa Circuito Ilhada 15x15 cm	1	0,00	0,00	25,00	25,00
Placa Circuito Ilhada 5x7 cm	1	0,00	0,00	5,00	5,00
Câmera EZVIZ C3N	1	0,00	0,00	429,90	429,90
Câmera Intelbras VIP 1130 B G3	1	0,00	0,00	250,00	250,00
Roteador Wi-fi ZTE ZXHN F660	1	0,00	0,00	125,00	125,00
Cabo de Rede Ethernet RJ45 8 m	1	0,00	0,00	40,00	40,00
Fonte 5V com Cabo (Carregador de Celular)	3	0,00	0,00	30,00	90,00
Cabo Extensor P2 Emissor IR 3 m	1	0,00	0,00	25,00	25,00
Caixa de Passagem com Espelho Cego 123x85x65 mm	1	10,00	10,00	10,00	10,00
Extensão Elétrica de 5m	2	0,00	0,00	15,00	30,00
Extensão Elétrica de 1m	1	0,00	0,00	10,00	10,00
Cabo Flexível 1,5mm (30cm)	1	1,00	1,00	1,00	1,00
Receptor IR TL1838 VS1838 38Khz	1	1,80	1,80	1,80	1,80
Totais (R\$)		137,80	191,79	1.092,70	1.221,69

Fonte: Dados dos Autores (2024)

A Tabela 2 mostra como ficariam os custos de implantação para uma sala de aula que possui 3 seções de iluminação e 2 condicionadores de ar.

Tabela 2: Custos dos materiais para implementação do Sistema de Automação para salas de aula contendo 2 condicionadores de ar e 3 seções de iluminação

Descrição	QTD	Preço Unitário Médio de Mercado	Preço Total Médio de Mercado
Módulo DOIT ESP32 Devkit V1	3	R\$ 48,99	R\$ 146,97
Módulo Relé 4 Canais 5V	1	R\$ 24,90	R\$ 24,90
Módulo Sensor Ultrassônico HC-SR04	1	R\$ 14,99	R\$ 14,99
Módulo P2 Fêmea 3,5 mm	2	R\$ 6,38	R\$ 12,76
Chave Tátil (Push Button) 12x12x8,5 mm	1	R\$ 0,75	R\$ 0,75
Buzzer Ativo 3V	1	R\$ 3,50	R\$ 3,50
Chave Gangorra 2 Terminais 2 Posições	1	R\$ 1,50	R\$ 1,50
Conector Wago 221 3 vias	2	R\$ 3,50	R\$ 7,00
Conector Wago 221 2 vias	4	R\$ 2,50	R\$ 10,00
Caixa Organizadora 23x14 cm	1	R\$ 17,99	R\$ 17,99
Jumper	NA	R\$ 0,00	R\$ 0,00
Placa Circuito Ilhada 15x15 cm	1	R\$ 25,00	R\$ 25,00
Placa Circuito Ilhada 5x7 cm	2	R\$ 5,00	R\$ 10,00
Câmera EZVIZ C3N	1	R\$ 429,90	R\$ 429,90
Câmera Intelbras VIP 1130 B G3	1	R\$ 250,00	R\$ 250,00
Roteador Wi-fi ZTE ZXHN F660	1	R\$ 125,00	R\$ 125,00
Cabo de Rede Ethernet RJ45 8 m	1	R\$ 40,00	R\$ 40,00
Fonte 5V com Cabo (Carregador de Celular)	4	R\$ 30,00	R\$ 120,00
Cabo Extensor P2 Emissor IR 3 m	2	R\$ 25,00	R\$ 50,00
Caixa de Passagem com Espelho Cego 123x85x65 mm	2	R\$ 10,00	R\$ 20,00
Extensão Elétrica de 5m	2	R\$ 15,00	R\$ 30,00
Extensão Elétrica de 1m	1	R\$ 10,00	R\$ 10,00
Cabo Flexível 1,5mm (30cm)	1	R\$ 1,00	R\$ 1,00
Receptor IR TL1838 VS1838 38Khz	1	R\$ 1,80	R\$ 1,80
Totais (R\$)		R\$ 1.092,70	R\$ 1.353,06

Fonte: Dados dos Autores (2024)

A Tabela 3 mostra como se obteve cada um dos equipamentos instalados na sala de aula escolhida e na coluna “Substituível?” estão indicados quais produtos acreditamos que podem ser facilmente substituídos por produtos ou soluções mais baratas e até gratuitas, baixando assim, o custo total de implantação do sistema.

Tabela 3: Forma de obtenção de cada produto e quais deles recomenda-se a substituição por soluções mais econômicas

(continua)

Descrição	Forma de Obtenção	Substituível?
Módulo DOIT ESP32 Devkit V1	Compra	Não
Módulo Relé 4 Canais 5V	Compra	Não
Módulo Sensor Ultrassônico HC-SR04	Compra	Não
Módulo P2 Fêmea 3,5 mm	Compra	Sim

Tabela 3 – Forma de obtenção de cada produto e quais deles recomenda-se a substituição por soluções mais econômicas

Descrição	Forma de Obtenção	(conclusão)
		Substituível?
Chave Tátil (Push Button) 12x12x8,5 mm	Compra	Sim
Buzzer Ativo 3V	Compra	Não
Chave Gangorra 2 Terminais 2 Posições	Compra	Sim
Conector Wago 221 3 vias	Compra	Sim
Conector Wago 221 2 vias	Compra	Sim
Caixa Organizadora 23x14 cm	Compra	Sim
Jumper	Reciclagem	Não
Placa Circuito Ilhada 15x15 cm	Já Possuíamos	Sim
Placa Circuito Ilhada 5x7 cm	Já Possuíamos	Sim
Câmera EZVIZ C3N	Doação	Sim
Câmera Intelbras VIP 1130 B G3	Empréstimo	Sim
Roteador Wi-fi ZTE ZXHN F660	Já Possuíamos	Sim
Cabo de Rede Ethernet RJ45 8 m	Doação	Sim
Fonte 5V com Cabo (Carregador de Celular)	Já Possuíamos	Sim
Cabo Extensor P2 Emissor IR 3 m	Doação	Sim
Caixa de Passagem com Espelho Cego 123x85x65 mm	Compra	Sim
Extensão Elétrica de 5m	Já Possuíamos	Sim
Extensão Elétrica de 1m	Já Possuíamos	Sim
Cabo Flexível 1,5mm (30cm)	Compra	Não
Receptor IR TL1838 VS1838 38Khz	Compra	Não

Fonte: Dados dos Autores (2024)

Como podemos perceber, o notebook utilizado para embarcar o Sistema de Detecção não foi considerado juntos aos custos. Isso ocorreu porque observou-se que este sistema não consome totalmente a capacidade de desempenho da máquina utilizada, o que abre margem para substituí-la por um equipamento com especificações mais em conta. Além desse fator, o notebook utilizado também é de uso pessoal de um dos autores, o que leva a considerar que o mesmo também contém outras aplicações que funcionam em paralelo ao sistema executado, aumentando assim, o consumo de recursos. Uma máquina exclusiva para os propósitos do sistema não enfrentaria estas dificuldades. Outro fator importante, é que o algoritmo empregado pode ser otimizado para que seja possível embarcá-lo em equipamentos com especificações mais brandas.

5.4 Recomendações de Instalação

Esta seção apresenta sugestões de instalação do sistema de automação proposto, caso seja aplicado permanentemente em uma sala de aula. As sugestões

foram elaboradas pelos autores e constatadas nos **Apêndices J e K**, baseadas nos modelos de sala de aula presentes nos blocos 14 a 17 do IFTO – Campus Palmas.

O **Apêndice J** apresenta as devidas adaptações para instalação dos módulos ESP32 para o controle dos condicionadores de ar (no caso de uma sala com 2 aparelhos), assim também como a alocação de uma das câmeras para captação das imagens em melhor ângulo. O Apêndice K apresenta as adaptações a serem feitas para alocar o circuito de controle da iluminação, a central de controle e processamento e a outra câmera. Toda a sugestão visa o aproveitamento das canaletas já presentes nas salas de aula, que contém rotas de cabos de energia e dados para o projetor.

Apesar de algumas salas de aula possuírem divergências em seus tamanhos, o que resulta na diminuição do número de condicionadores de ar, todas apresentam semelhanças estruturais, o que facilita as devidas adaptações para a instalação do sistema. As sugestões apresentadas visam o melhor proveito com o mínimo de modificações quanto a infraestrutura do local, o que facilita sua aplicação e ampliação para o devido funcionamento.

6 CONCLUSÃO

O sistema proposto neste trabalho traz uma abordagem inovadora sobre automação de ambientes. Sua integração com inteligência artificial faz com que os resultados dos controles dos sistemas sejam cada vez mais inteligentes. O projeto como um todo, por ser desenvolvido com tecnologias atuais, possui uma certa adaptabilidade a outros tipos de demandas e propósitos, o que abre espaço para novas aplicações e interpretações.

Após uma análise detalhada dos resultados obtidos, pode-se concluir que o sistema de automação proposto cumpre com os objetivos apresentados, realizando o devido controle dos sistemas de iluminação e climatização da sala de aula onde foi temporariamente implementado no IFTO – Campus Palmas. As estimativas de economia de energia apresentadas mostram uma realidade de desperdício evidente que torna este projeto ótima opção para reduzir custos com energia elétrica, no contexto atual em que a instituição se insere.

Durante os testes, todos os acionamentos dos equipamentos bem como todas as funcionalidades extras criadas atuaram conforme o esperado, validando o funcionamento do protótipo. Os resultados da detecção utilizando o YOLOv8 se mostraram assertivos e satisfatórios mesmo em condições de dificuldade de iluminação e posicionamento das câmeras. Dessa forma, essa ferramenta se mostrou efetiva para os propósitos do projeto. Vale ressaltar que o Sistema de Acompanhamento também se destacou por apresentar em tempo real via interface gráfica a comutação dos sistemas de controle, demonstrando, assim, uma boa responsividade na conexão do protótipo como um todo.

Haja vista o estado alcançado do projeto, sugere-se a trabalhos futuros melhorias referentes a sua implantação e expansão. No que se refere a implantação, é recomendável uma reestruturação das placas eletrônicas desenvolvidas de modo a entregar um resultado ainda mais refinado. São bem-vindas, melhorias utilizando recursos de impressão 3D e placas de circuitos impresso nos módulos de controle da iluminação e climatização. Outro ponto que se pode melhorar também é o circuito de emissão de comandos infravermelho, pois no trabalho apresentando não foi possível implementar, com apenas um módulo ESP32, um circuito capaz de controlar mais de um equipamento a longas distâncias.

Sobre melhorias de expansão, ao considerar um cenário onde este sistema seja implantando em algumas ou todas as salas dos Blocos 14 a 17 do Campus Palmas, uma boa sugestão seria adaptar os algoritmos de comunicação da API para que se integrassem a uma rede única e central. Dessa forma, todas as informações sobre os estados dos sistemas de climatização e iluminação de cada sala de aula estariam disponíveis em um único lugar para serem compartilhadas, metrificadas ou até controladas. A infraestrutura atual do local pode ser aproveitada para estes fins, pois sabe-se que a instituição já possui servidores, roteadores e cabeamento de rede espalhados por suas edificações.

REFERÊNCIAS

ANDRADE, Emmanuel. **Guia Prático para Criação de uma venv no VS Code:** Instalação, Ativação, Desativação e Freeze dos Requirements. 2023. Disponível em: <<https://www.dio.me/articles/guia-pratico-para-criacao-de-uma-venv-no-vs-code-instalacao-ativacao-desativacao-e-freeze-dos-requirements>>. Acesso em 12 mar. 2024.

ALPHACODERS. **Technology Circuit HD Wallpaper**. Disponível em: <<https://wall.alphacoders.com/big.php?i=220582>>. Acesso em 07 mai. 2024.
ARDUINO. **Downloads**. 2024. Disponível em: <<https://www.arduino.cc/en/software>>. Acesso em 15 fev. 2024.

ARDUINO-IRREMOTE. **Arduino-IRremote**. [2012?]. Disponível em: <<https://github.com/Arduino-IRremote/Arduino-IRremote/discussions?page=2>>. Acesso em: 30 abr. 2024.

ARDUINO-IRREMOTE. **Arduino-IRremote/examples/ReceiveAndSend/**. [2012?]. Disponível em: <<https://github.com/Arduino-IRremote/Arduino-IRremote/tree/master/examples/ReceiveAndSend>>. Acesso em: 30 abr. 2024.

ASALI. **ESP32 GET JSON from API - ESP32 GET request from API receive JSON**. Youtube, 19 abr. 2019. 4min47s. Disponível em: <<https://www.youtube.com/watch?v=cgoCmYLeaSI&t=192s>>. Acesso em 16 abr. 2024.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SERVIÇOS DE CONSERVAÇÃO DE ENERGIA. **Eficiência Energética:** A saída para produzir e consumir melhor gastando menos. 2022. Disponível em: <http://www.abesco.com.br/novidade/6754/>. Acesso em: 04 jun. 2023.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SERVIÇOS DE CONSERVAÇÃO DE ENERGIA. **O que é Eficiência Energética? (EE)**. 2015. Disponível em: <http://www.abesco.com.br/pt/o-que-e-eficiencia-energetica-ee/>. Acesso em: 04 jun. 2023.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Instalações Elétricas de Baixa Tensão:** NBR 5410. Rio de Janeiro, 2004, 209 p.

AUBERT, Olivier. **Python-vlc**. [2012?]. Disponível em: <<https://github.com/oaubert/python-vlc?tab=GPL-2.0-2-ov-file>>. Acesso em: 29 abr. 2024.

AUGMENTED AI. **YOLOv8 for Beginners: A Complete Tutorial on Windows 11**. Youtube, 30 jan. 2023. 8min59s. Disponível em: <<https://www.youtube.com/watch?v=C4i8Bia83TM&t=115s>>. Acesso em 28 mar. 2024.

AWS. Qual é a diferença entre microprocessadores e microcontroladores?

Disponível em: <<https://aws.amazon.com/pt/compare/the-difference-between-microprocessors-microcontrollers/>>. Acesso em: 04 maio. 2024.

AWS. O que é a Internet das Coisas (IoT)? Disponível em:

<[https://aws.amazon.com/pt/what-is/iot/#:~:text=oAWS20IoT-,OqueéaInternetdasCoisas\(IoT\),comoentreosprópriosdispositivos.](https://aws.amazon.com/pt/what-is/iot/#:~:text=oAWS20IoT-,OqueéaInternetdasCoisas(IoT),comoentreosprópriosdispositivos.)> Acesso em 26 abr. 2024.

AWS. O que é uma API (interface de programação de aplicações)? Disponível em: <<https://aws.amazon.com/pt/what-is/api/>>. Acesso em: 23 maio. 2024.

BEOCK, Leandro Rodrigo Preis, et al. **Protocolo HTTP**. 2011. Disponível em: <<https://www.vivaolinux.com.br/imagens/artigos/comunidade/ProtocoloHTTP.pdf>>. Acesso em: 17 maio. 2024.

BLACHON, Benoît. **Deserialization tutorial**. 2024. Disponível em:

<<https://arduinojson.org/v6/doc/deserialization/>>. Acesso em: 14 abr. 2024.

BLACHON, Benoît. **Documentation**. 2024. Disponível em:

<<https://arduinojson.org/v6/doc/>>. Acesso em: 14 abr. 2024.

BLACHON, Benoît. **JsonArray**. 2024. Disponível em:

<<https://arduinojson.org/v6/api/jsonarray/>>. Acesso em: 14 abr. 2024.

CELERE. **Automação residencial e predial e seus impactos na construção civil**.

2021. Disponível em: <https://celere-ce.com.br/tecnologia/automacao-residencial-e-predial/>. Acesso em: 07 jun. 2023.

CHAURASIA, Ayush *et al.* **Previsão de modelos com Ultralytics YOLO**. 2023.

Disponível em: <<https://docs.ultralytics.com/pt/modes/predict/>>. Acesso em 5 abr. 2024.

CHAURASIA, Ayush; JOCHER, Glenn; MUNAWAR, Muhammad Rizwan. **Home**.

2024. Disponível em: <<https://docs.ultralytics.com/>>. Acesso em 5 abr. 2024.

CUNHA, Matheus. **Criando um Relógio Digital com JavaScript e CSS**. 2023.

Disponível em: <<https://www.dio.me/articles/criando-um-relogio-digital-com-javascript-e-css>>. Acesso em 15 mai. 2024.

CURVELLO, André. **ESP32 - Um grande aliado para o Maker IoT**. 2018. Disponível

em: <<https://www.makerhero.com/blog/esp32-um-grande-aliado-para-o-maker-iot/>>. Acesso em: 11 maio. 2024.

DEV IDEIAS. **Sistema de Alerta Contra Invasão | Visão Computacional | YOLOv8 e Opencv**. Youtube, 20 ago. 2023. 33min53s. Disponível em:

<<https://www.youtube.com/watch?v=EJE2aOfwqxY>>. Acesso em 26 mar. 2024.

DRONEBOT WORKSHOP. **IR Remotes & Microcontrollers: Arduino & ESP32**.

Youtube. 23 out. 2023. 1h31min49s. Disponível em:

<<https://www.youtube.com/watch?v=OB4jAc2033o&t=3992s>>. Acesso em 15 mai. 2024.

ENERGISA. **NDU001 - Fornecimento de energia elétrica em tensão secundária a edificações individuais ou agrupadas em até três unidades consumidoras.**

v.6.3. João Pessoa – PB. Novembro 2023. Disponível em:

<<https://www.energisa.com.br/Paginas/informacoes/taxas-prazos-e-normas/normas-tecnicas.aspx>>. Acesso em: 02 jun. 2024.

ESP32. Disponível em: <<https://www.usinainfo.com.br/esp32-611>>. Acesso em: 8 maio. 2024.

ESPRESSIF. **Arduino-esp32.** [2016?]. Disponível em:

<<https://github.com/espressif/arduino-esp32/commits/master/?since=2016-10-06&until=2016-10-06>>. Acesso em: 15 fev. 2024.

ESPRESSIF. **ESP32WROOM32 - Datasheet.** v.3.4 Shanghai: Espressif Systems, 2023.

EZVIZ. **C3N:** Câmera externa inteligente com Wi-fi. 2024. Disponível em:

<<https://www.ezviz.com/br/product/C3N/21697>>. Acesso em 10 abr. 2024.

F4TORNADO. **How do I prevent a python server from writing to the terminal window?** 2018. Disponível em:

<<https://stackoverflow.com/questions/53422825/how-do-i-prevent-a-python-server-from-writing-to-the-terminal-window>>. Acesso em 10 abr. 2024.

FLASK. **Quickstart.** [ca. 2020]. Disponível em:

<<https://flask.palletsprojects.com/en/3.0.x/quickstart/>>. Acesso em 15 mai. 2024.

FIESC. **Desperdício elétrico no Brasil equivale ao consumo de 20 milhões de residências.** 2022. Disponível em: <https://fiesc.com.br/pt-br/imprensa/desperdicio-eletrico-no-brasil-equivale-ao-consumo-de-20-milhoes-de-residencias>. Acesso em: 07 jun. 2023.

GOOGLE. **O que é inteligência artificial (IA)?** Disponível em:

<<https://cloud.google.com/learn/what-is-artificial-intelligence?hl=pt-br>>. Acesso em: 25 maio. 2024.

GUIMARÃES, Flávio. **Um Botão, Múltiplas Funções no Arduino**

#ComoProgramar. Youtube. 10 mai. 2022. 17min33s. Disponível em:

<<https://www.youtube.com/watch?v=kFwaSB1pHAc&t=840s>>. Acesso em 27 abr. 2024.

IBM. **O que é internet das coisas?** Disponível em: <<https://www.ibm.com/br-pt/topics/internet-of-things>>. Acesso em 24 abr. 2024.

IBM. **O que é uma API (Interface de Programação de Aplicativos)?** Disponível em: <<https://www.ibm.com/br-pt/topics/api>>. Acesso em: 23 maio. 2024.

INTELBRAS. **Câmera Bullet compatível com a tecnologia PoE**. 2024. Disponível em: <<https://www.intelbras.com/pt-br/camera-bullet-compativel-com-a-tecnologia-poe-vip-1130-b-g3>>. Acesso em 25 abr. 2024.

JOCHER, Glen, MUNAWAR, Hizwan, EXEL, Ayush. **YOLOv8**. Disponível em: <<https://docs.ultralytics.com/>>. Acesso em: 29 maio. 2024.

KANEHIRA, C. **Microcontroladores: o que são e aplicações**. Disponível em: <<https://eescjr.com.br/blog/microcontroladores-o-que-sao-e-aplicacoes/>>. Acesso em: 04 maio. 2024.

KASSOUF, S. **Microcontrolador: o que é, como funciona e quais as funções**. Disponível em: <<https://blog.kalatec.com.br/microcontrolador/>>. Acesso em: 04 maio. 2024.

KERSCHBAUMER, R. **Microcontroladores**. 2018. Disponível em: <<https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/wp-content/uploads/sites/43/2018/02/Apostila-Microcontroladores.pdf>>.

MAYER, Christian. **How to Check if the Current Time is in a Range in Python?**. 2021. Disponível em: <<https://blog.finxter.com/how-to-check-if-the-current-time-is-in-a-range-in-python/#:~:text=Therefore%2C%20you%20can%20use%20the,and%20current%20are%20datetime%20objects.>>. Acesso em 25 abr. 2024.

MAZIERO, Carlos. **Introdução aos Serviços de Rede**. Disponível em: <<https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:introducao>>. Acesso em: 17 maio. 2024.

MAZIERO, Carlos. **O Serviço HTTP**. Disponível em: <[https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:servico_http&s\[\]=protocolo&s\[\]=http](https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:servico_http&s[]=protocolo&s[]=http)>. Acesso em: 18 maio. 2024.

MCCHAN. **Enable access control on simple HTTP server**. 2014. Disponível em: <<https://stackoverflow.com/questions/21956683/enable-access-control-on-simple-http-server>>. Acesso em 5 mai. 2024.

MOREIRA, Levi Dias; NERES, Arabutan Marques. **Proposta de automação dos condicionadores de ar e da iluminação das salas de aula do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - Campus Palmas**. 2015. 89 f. TCC (Graduação) - Curso de Engenharia Elétrica, Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - Campus Palmas, Palmas, 2015.

MOREIRA, Levi Dias; NERES, Arabutan Marques. **Sistema microcontrolado voltado à automação dos condicionadores de ar e da iluminação das salas de aula do IFTO - Campus Palmas**. 2019. 54 f. TCC (Graduação) - Curso de Engenharia Elétrica, Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - Campus Palmas, Palmas, 2019.

OLIVEIRA, Brehme Rezende de et al. Potencial de economia de energia elétrica na iluminação e refrigeração da Universo/JF. 2018. 8 f. **Artigo - Curso de Engenharia de Produção**, Universidade Federal de Goiás – Regional Catalão, Catalão, 2018.

ORACLE. **O que é IoT?** Disponível em: <<https://www.oracle.com/br/internet-of-things/what-is-iot/>>. Acesso em 24 abr. 2024.

PINTEREST. **Logotipo da Engenharia Elétrica**. Disponível em: <<https://i.pinimg.com/originals/e8/d2/06/e8d206546457718f4bb1c09337fa518f.jpg>>. Acesso em 07 mai. 2024.

PORTAL DA INDUSTRIA. **Eficiência Energética**. Disponível em: <https://www.portaldaindustria.com.br/industria-de-a-z/eficiencia-energetica/#:~:text=Oqueeefficienciaenergetica,processoseprodutosenergeticamente eficientes>. Acesso em: 07 jun. 2023.

PRUDENTE, Francesco. **Automação Predial e Residencial: Uma Introdução**. [Reimpr.]. Rio de Janeiro: Ltc, 2017. 228 p.

PYTHON. **Download the latest version for Windows**. 2024. Disponível em: <<https://www.python.org/downloads/>>. Acesso em 01 mar. 2024.

RANDOM NERD TUTORIALS. **ESP32 Client-Server Wi-Fi Communication Between Two Boards**. 2024. Disponível em: <<https://randomnerdtutorials.com/esp32-client-server-wi-fi/>>. Acesso em 13 abr. 2024.

RANDOM NERD TUTORIALS. **ESP32 HTTP GET and HTTP POST with Arduino IDE (JSON, URL Encoded, Text)**. 2024. Disponível em: <<https://randomnerdtutorials.com/esp32-http-get-post-arduino/>>. Acesso em 13 abr. 2024.

RANDOM NERD TUTORIALS. **ESP32 with HC-SR04 Ultrasonic Sensor with Arduino IDE**. 2024. Disponível em: <<https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino/>>. Acesso em 25 mai. 2024.

RANDOM NERD TUTORIALS. **Getting Started with the ESP32 Development Board**. 2023. Disponível em: <<https://randomnerdtutorials.com/getting-started-with-esp32/>>. Acesso em 15 fev. 2024.

SATY, Adriana. **HTML + CSS NA PRÁTICA: Faça seu primeiro site**. Youtube. 24 mar. 2022. 16min18s. Disponível em: <https://www.youtube.com/watch?v=n_Etdr7Dbjs&t=487s>. Acesso em 20 abr. 2024.

SOUZA, Ivan de. **Afinal, o que é JSON e para que ele serve? Descubra agora!**. 2020. Disponível em: <<https://rockcontent.com/br/blog/json/>>. Acesso em 25 mai. 2024.

SOUZA, Kymberli B. de. **Conheça os ícones da notação fluxograma**. [ca. 2020]. Disponível em: <<https://zeev.it/blog/icones-da-notacao-fluxograma/>>. Acesso em 26 mai. 2024.

TECH WITH TIM. **Flask Tutorial #9 - Static Files (Custom CSS, Images & Javascript)**. Youtube, 18 nov. 2019. 10min46s. Disponível em: <<https://www.youtube.com/watch?v=tXpFERibRaU>>. Acesso em 15 mai. 2024.

USER1937459. **Multithreaded web server in python**. 2012. Disponível em: <<https://stackoverflow.com/questions/14088294/multithreaded-web-server-in-python>>. Acesso em 20 mai. 2024.

VISUAL STUDIO CODE. **Code editing**. 2024. Disponível em: <<https://code.visualstudio.com/>>. Acesso em 01 mar. 2024.

WATT-SP. **O que são instalações elétricas prediais? Entenda tudo!** Disponível em: <<https://www.wattsp.com.br/blog/o-que-sao-instalacoes-eletricas-prediais-entenda-tudo/>>. Acesso em: 07 mar. 2024.

WELSH, Jason. **1304 Arduino Chapter13 Sending Raw IR codes to consumer products**. Youtube. 5 jan. 2013. 8min42s. Disponível em: <<https://www.youtube.com/watch?v=eR8sQq3pl20>>. Acesso em 15 mai. 2024.

WIKIPÉDIA. **Instituto Federal do Tocantins**. 2022. Disponível em: <https://pt.wikipedia.org/wiki/Instituto_Federal_do_Tocantins#/media/Ficheiro:Instituto_Federal_do_Tocantins_-_Marca_Vertical_2015.svg>. Acesso em 07 mai. 2024.

APÊNDICE A - ALGORITMO DO SISTEMA DE DETECÇÃO

```
#Importação de Bibliotecas
#-----
from ultralytics import YOLO #biblioteca para trabalhar com YOLOv8
import time #biblioteca para trabalhar com intervalos de tempo
import datetime #biblioteca para trabalhar com data e hora
import threading #biblioteca para trabalhar múltiplos processamentos simultâneos
from http.server import BaseHTTPRequestHandler, HTTPServer #biblioteca para trabalhar com
servidor HTTP
import json #biblioteca para trabalhar com JSON
import vlc #biblioteca para trabalhar com VLC
import os #biblioteca para trabalhar com manipulação de pastas do sistema
import shutil #biblioteca para deletar árvore de pastas
from socketserver import ThreadingMixIn #biblioteca para gerenciar servidor HTTP
multithreaded

#-----
#Definição de funções que serão utilizadas pelo algoritmo
#-----
# Classe que gerencia as requisições HTTP
class RequestHandler(BaseHTTPRequestHandler):
    # Função que silencia as mensagens de log das conexões do servidor
    def log_message(self, format, *args):
        pass
    # Função utilizada para responder à requisição GET
    def do_GET(self):
        # Define o tipo de conteúdo da resposta
        self.send_response(200)
        self.send_header('Content-Type', 'application/json')
        self.send_header('Access-Control-Allow-Origin', '*') #Supressor de políticas CORS
        self.end_headers()

        #variáveis a serem enviadas via API configuradas como JSON
        data = {"lot": lot,
                "ilum_status":ilum_status,
                "ar1_status":ar1_status,
                "ar2_status":ar2_status,
                "temp_up":temp_up,
                "temp_down":temp_down
                }
        #formata string "data" para json
        json_data = json.dumps(data)
        # Envia a resposta no servidor HTTP
        self.wfile.write(json_data.encode('utf-8'))

    # Método para responder à requisição PUT
    def do_PUT(self):
        global temp_up, temp_down
        # Define o tipo de conteúdo da resposta
        self.send_response(200)
        self.end_headers()

        # Lê o conteúdo da requisição PUT
        content_length = int(self.headers['Content-Length'])
        put_data = self.rfile.read(content_length)

        # Converte o conteúdo da requisição PUT para JSON
        put_json = json.loads(put_data.decode('utf-8'))

        # Verificação de qual informação está contida no JSON da requisição PUT
        if "temp_up" in put_json: #se put_json tiver a informação "temp_up"
            temp_up = put_json['temp_up']
        elif "temp_down" in put_json: #se put_json tiver a informação "temp_down"
            temp_down = put_json['temp_down']
```

```

# Classe para responder requisições em threads separadas
class ThreadingSimpleServer(ThreadingMixIn, HTTPServer):
    pass

# Função para iniciar o servidor HTTP
def start_server():
    global server
    with ThreadingSimpleServer(('', 8000), RequestHandler) as server: #configuração para
servir no localhost porta 8000
        print("Servidor HTTP rodando na porta 8000...")
        print ('-'*70) #mostra divisória entre análises
        server.serve_forever()

# Função que realiza o processamento de imagens na detecção
def image_processing(id):

    cls = [] #lista para armazenar as classes detectadas por câmera
    capture_attempts = 5 #define o número de tentativas de captura de imagem
    global quant_analise
    global time_btw_photo

    name_predict = f"Foto{quant_analise}" #variável que armazena o nome da pasta que será
armazenada as fotos salvas
    cicle = f"Resultados\{moment_exec}\Ciclo{cicle_counter}\Camera{id+1}" #variável que
armazena o nome do diretório onde as fotos serão salvas
    dir = f".\{cicle}\{name_predict}\." #variável e concatena name_predict e cicle
    os.makedirs(os.path.dirname(dir)) #cria a pasta para salvar as fotos

    print("Capturando foto...")
    players[id].video_take_snapshot(0,dir,0,0) #captura a foto e salva na pasta
especificada

    if empty_folder(dir): #verifica se a pasta está vazia
        while empty_folder(dir) and capture_attempts != 0:
            print("Foto não capturada, tentando novamente...") #caso a foto não tenha sido
capturada, tenta novamente
            players[id].video_take_snapshot(0,dir,0,0) #captura a foto e salva na pasta
especificada
            capture_attempts -= 1 #decrementa o número de tentativas
            if empty_folder(dir):
                regressive_count(5)

        if empty_folder(dir):
            cls = []
            quant_analise = 0
            return cls

    if not empty_folder(dir):
        print("Foto capturada, analisando...")
        results = model.predict(dir, save = True, conf = 0.3, project = cicle, name =
name_predict)
        #na linha acima o yolo vai detectar objetos na imagem com confianças acima de 30% e
salvar os resultados nas pastas especificadas

        if quant_analise != 1:
            regressive_count(time_btw_photo) #define tempo em segundos entre uma foto e
outra

        for result in results:
            boxes = result.boxes
            for box in boxes:
                cls.append(int(box.cls)) #obtenção das classes detectadas em uma lista
        return cls

```

```

# Função para verificar se uma pasta do sistema está vazia
def empty_folder (directory):
    content_folder = os.listdir(directory)
    return len(content_folder) == 0

#Função para validar se variáveis digitadas são números inteiros
def variable_validation(variable):
    while True:
        try:
            variable = int(input(""))
            if variable <=0:
                raise ValueError
            break
        except ValueError:
            print("entrada inválida! tente novamente...")
    return variable

#Função para realizar contagens regressivas
def regressive_count(seconds):
    while seconds > 0:
        print(f"{seconds} segundos", "\33[k", end='\r')
        time.sleep(1)
        seconds -= 1

# Função para verificar se um determinado horário está ou não entre um intervalo de
horários
def time_in_range(start, end, current):
    return start<=current<=end

#-----
#Definições gerais de sistema
#-----
print ("***** (SISTEMA
INICIALIZADO)*****")

#Variáveis de Sistema
cicle_counter = 1 # variável que conta quantos ciclos de análise foram realizados
system_status = True #status de funcionamento: True - sistema operando; False - sistema
desligado
today = datetime.datetime.now() #captura o momento da execução
moment_exec= today.strftime('%d-%m-%Y_%H-%M-%S') #formata o momento da execução para texto
no formato especificado
aux = True #variável auxiliar que serve apenas para, em regime de suspensão, o sistema
mostre as informações de suspensão apenas 1 vez.
temp_up = "false" #variável que manipulará o aumento de temperatura dos condicionadores de
ar
temp_down = "false" #variável que manipulará a diminuição de temperatura dos
condicionadores de ar

#Variáveis de Detecção
model = YOLO("yolov8x.pt") #importa o modelo a ser utilizado na detecção
quant_analise_global = 0 #quantidade de fotos a serem tiradas
time_btw_photo = 0 #tempo entre fotos
time_btw_analise = 0 #tempo entre análises
range_delete = 0 #quanta ciclos serão contados do atual para trás
reconnection_attempts = 5 #número de tentativas de reconexão quando todas as câmeras
falharem
url = ["rtsp://admin:camera123@192.168.1.6:554/cam/realmonitor?channel=1&subtype=0",
      "rtsp://admin:XNDKFO@192.168.1.3:554/H264/Streaming/Channels/101"
      ] #url das câmeras
instances = [] #variável que armazena as instâncias do vlc
medias = [] #variável que armazena as mídias para o vlc
players = [] #variável que armazena os player de cada mídia
state_cameras = [] #variável que armazena o estado das câmera em caso de falha de ambas
line = []
i = []

```

```

#Variáveis de controle
lot = "Local Vazio" #status inicial da variável que armazena a lotação da sala
ilum_status, ar1_status, ar2_status = "Desligado","Desligado","Desligado" #status inicial
dos sistemas à serem controlados

#Protocolos de hora
protocol_1_start = datetime.time(12,0,0) #início do primeiro protocolo de hora: 12h
protocol_1_end = datetime.time(13,0,0) #fim do primeiro protocolo de hora: 13h
protocol_2_start = datetime.time(23,0,0) #início do segundo protocolo de hora: 23h
protocol_2_end = datetime.time(23,59,59) #fim do segundo protocolo de hora: 23h59min
protocol_3_start = datetime.time(0,0,0) #início do terceiro protocolo de hora: 00h
protocol_3_end = datetime.time(7,0,0) #fim do terceiro protocolo de hora: 07h

# Inicia o servidor HTTP em uma thread separada
server_thread = threading.Thread(target=start_server)
server_thread.start()
regressive_count(3)

#Captura as configurações do usuário
print("Quantidades de fotos a serem tiradas:")
quant_analise_global = variable_validation(quant_analise_global)
print ("Tempo mínimo em segundos entre fotos:")
time_bt看_photo = variable_validation(time_bt看_photo)
print("Tempo mínimo em segundos entre ciclos:")
time_bt看_analise = variable_validation(time_bt看_analise)
print("Quantos ciclos deseja que o sistema mantenha os resultados salvos, contados do ciclo
atual para trás:")
range_delete = variable_validation(range_delete)
print ('-'*70) #mostra divisória

# Inicialização das câmeras no vlc
print("Ligando Câmeras!")
for idx, fname in enumerate(url):
    print(f"Carregando {idx}:{fname}")
    instances.append(vlc.Instance()) #cria as instâncias do vlc
    medias.append(instances[idx].media_new(fname)) #cria as mídias
    players.append(vlc.MediaPlayer()) #cria os players
    players[idx].set_media(medias[idx]) #seta as mídias nos players
    players[idx].play() #inicializa todas as câmeras ao mesmo tempo

regressive_count(5)

#-----
#Loop
#-----
while system_status:
    now = datetime.datetime.now().time() #captura o momento atual
    hour = now.strftime('%H:%M:%S') #captura hora, minutos e segundos do momento atual
    weekday = datetime.datetime.now().weekday() #captura o dia da semana do momento atual
    em formato de número inteiro sequencial - 0 par Segunda-Feira e 6 para domingo

    # a função "time_in_range" não considera mudança de dia. Essa seção do código é apenas
    evitar a execução de um ciclo
    # de detecção entre 23:59:59 e 00:00:00
    if hour == "23:59:59":
        time.sleep(2)
        now = datetime.datetime.now().time()

    if not (time_in_range(protocol_1_start,protocol_1_end, now) or
time_in_range(protocol_2_start,protocol_2_end, now) or
time_in_range(protocol_3_start,protocol_3_end, now) or weekday == 5 or weekday== 6):
        status_cam1 = str(players[0].get_state()) #armazena em "status_cam1" o estado da
câmera 1
        status_cam2 = str(players[1].get_state()) #armazena em "status_cam2" o estado da
câmera 2
        aux = True

```

```

print(f"Ciclo de Análise {cicle_counter}") # mostra o ciclo atual de operação

classes = [] #define e limpa a variável classes que armazena as classe detectadas
classes_concat = [] #define e limpa a variável que armazena as listas de classes concatenadas em uma única lista

if status_cam1 == "State.Playing" and status_cam2 == "State.Playing": #câmera 1 ok e câmera 2 ok
    print("Status Câmera 1: OK")
    print("Status Câmera 2: OK")
    quant_analise = quant_analise_global #quantidade de fotos a serem tiradas
    while quant_analise>0:
        results1 = [] #cria e limpa variável que vai armazenar os resultados do processamento da câmera 1
        results2 = [] #cria e limpa variável que vai armazenar os resultados do processamento da câmera 2
        results1 = image_processing(0)
        results2 = image_processing(1)
        if results1 is not None: #Para evitar objetos com valores "None"
            classes.append(results1)
        if results2 is not None: #Para evitar objetos com valores "None"
            classes.append(results2)
        quant_analise-=1

    elif status_cam1 == "State.Playing" and status_cam2 != "State.Playing": #câmera 1 ok e câmera 2 inativa
        print("Status Câmera 1: OK")
        print("Status Câmera 2: Falha! Verifique o dispositivo")
        quant_analise = quant_analise_global #quantidade de fotos a serem tiradas
        while quant_analise>0:
            results1 = [] #cria e limpa variável que vai armazenar os resultados do processamento da câmera 1
            results1 = image_processing(0)
            if results1 is not None: #Para evitar objetos com valores "None"
                classes.append(results1)
            quant_analise-=1
        players[1].set_media(medias[1]) #seta a mídia a câmera 2
        players[1].play() #tenta iniciar a câmera 2
        time.sleep(1)

    elif status_cam1 != "State.Playing" and status_cam2 == "State.Playing": #câmera 1 inativa e câmera 2 ok
        print("Status Câmera 1: Falha! Verifique o dispositivo")
        print("Status Câmera 2: OK")
        quant_analise = quant_analise_global #quantidade de fotos a serem tiradas
        while quant_analise>0:
            results2 = [] #cria e limpa variável que vai armazenar os resultados do processamento da câmera 2
            results2 = image_processing(1)
            if results2 is not None: #Para evitar objetos com valores "None"
                classes.append(results2)
            quant_analise-=1
        players[0].set_media(medias[0]) #seta a mídia a câmera 1
        players[0].play() #tenta iniciar a câmera 1
        time.sleep(1)

    else: #ambas as câmeras falharam
        print("Todos os dispositivos de captura falharam! Tentando reconectar com câmeras...")
        system_status = False #para evitar que o while system_status se repita e caia novamente neste loop
        reconnection_attempts = 5 #número de tentativas de reconexão quando todas as câmeras falharem
        counter_attempts = 1 #contador de tentativas de reconexão
        while reconnection_attempts > 0 and state_cameras != "State.Playing":

```

```

        print ('-'*70) #mostra divisória
        print(f"Tentativa de reconexão {counter_attempts}, aguarde...\n")

        if reconnection_attempts != 0 or reconnection_attempts!=5: #evita que o
sistema realize contagem no primeiro e último ciclo
            regressive_count(30)

        if reconnection_attempts != 0 or state_cameras != "State.Playing" :
#tentativa de reconexão
            for idx, fname in enumerate(url):
                print(f"Carregando {idx}:{fname}")
                state_cameras = []
                instances.append(vlc.Instance())
                medias.append(instances[idx].media_new(fname))
                players.append(vlc.MediaPlayer())
                players[idx].set_media(medias[idx])
                players[idx].play()
                time.sleep(3)
                state_cameras = str(players[idx].get_state())
                print (state_cameras)
                if state_cameras == "State.Playing": #se a câmera estiver
funcionando, o loop é interrompido
                    break
            else:
                break
            reconnection_attempts -=1
            counter_attempts +=1

        if state_cameras == "State.Playing":
            print("Câmera encontrada, conectando!")
            system_status = True

        if reconnection_attempts == 0:
            print("todas as tentativas de reconexão falharam. Desligando sistemas de
detecção e controle...")
            break

    #concatenação de listas obtidas na variável classes em lista única para análise:
    for line in classes:
        for i in line:
            classes_concat.append(i)

    #verificação de pessoas detectadas e definição de variáveis que farão o controle
dos sistemas
    count_people = classes_concat.count(0)
    if count_people>0:
        print("Tem pessoas no local! Sistemas: Ligados")
        lot = "Pessoas Detectadas"
        ilum_status, ar1_status, ar2_status = "Ligado","Ligado","Ligado"

    else:
        print("Nenhuma pessoa detectada. Sistemas: Desligados")
        lot = "Local Vazio"
        ilum_status, ar1_status, ar2_status = "Desligado","Desligado","Desligado"

    #contagem regressiva entre análises
    print(f"tempo ate a próxima verificação:")
    regressive_count(time_btw_analise)

    #deleção de resultados antigos conforme definido por usuário
    if cycle_counter - range_delete > 0 and range_delete!=0:
        deleted_path = f"Resultados\{moment_exec}\Ciclo{cycle_counter-range_delete}"
        shutil.rmtree(deleted_path)

    print("\nIniciando nova análise de ambiente...")

```

```

        print ('-'*70) #mostra divisória entre análises
        cicle_counter+=1 #aumenta 1 no contador de ciclo
    else:
        if aux == True:
            print ('-'*70) #mostra divisória
            print("Execução do Sistema Suspenso!")
            print(f"Horários de Suspensão:\n{protocol_1_start} a
{protocol_1_end}\n{protocol_2_start} a {protocol_2_end}\n{protocol_3_start} a
{protocol_3_end}\n Sábados e Domingos")
            lot = "Suspensa"
            ilum_status, ar1_status, ar2_status = "Desligado","Desligado","Desligado" #envia
status de "Desligado para Sistemas"
            aux = False

#finalização de todas as câmeras abertas
for p in players:
    p.stop()
server.shutdown() #encerra o servidor
print ("Servidor encerrado!")
server_thread.join() #encerra a thread do servidor
print("Sistema Desligado!...") #mensagem de interrupção

```

APÊNDICE B - ALGORITMO DO SISTEMA DE CONTROLE DA ILUMINAÇÃO

```
//Importação de bibliotecas
//-----
#include <Arduino.h> //biblioteca padrão para trabalhar com placas microcontroladoras
compatíveis com a plataforma Arduino
#include <WiFi.h> //biblioteca para trabalhar com conexões Wi-fi
#include <HTTPClient.h> //biblioteca para trabalhar com conexões HTTP
#include <ArduinoJson.h> //biblioteca para manipular JSON

//-----
//Definições variáveis do sistema
//-----

#define SOUND_SPEED 0.034 //constante de velocidade do som em cm/uS
const char* ssid = "acesspoint"; //SSID da rede Wi-fi
const char* password = "Tcc#1501"; //senha da rede Wi-fi
const char* serverName_pc = "http://192.168.1.2:8000"; //endereço do servidor HTTP(API)
const char* temp_up = "false"; //variável que controla o aumento de temperatura
const char* temp_down = "false"; //variável que controla a diminuição de temperatura
int Button = 21; //porta GPIO que vai receber os comandos do botão de funcionalidades do
sistema
int trigPin = 18; //porta GPIO que se conecta com o trigger do ultrassônico
int echoPin = 34; //porta GPIO que se conecta com o echo do ultrassônico
int LED_BUILTIN = 2; //definição da porta do led on-board
int RELAY_1 = 26; //porta GPIO que vai controlar o relé 1 extra
int RELAY_2 = 25; //porta GPIO que vai controlar o relé da seção 1 de lâmpadas
int RELAY_3 = 13; //porta GPIO que vai controlar o relé da seção 2 de lâmpadas
int RELAY_4 = 27; //porta GPIO que vai controlar o relé da seção 3 de lâmpadas
int BUZZER = 22; //porta GPIO que vai controlar o Buzzer
int httpResponseCode; //variável que armazena o código de resposta do servidor HTTP(API)
String request_server; //variável que armazena a informação contida na requisição GET feita
ao servidor HTTP(API)
String lot; //variável que armazena a presença no local obtida através do servidor
HTTP(API)
String temp_up_string; //variável para converter resposta da variável "temp_up" vinda do do
servidor HTTP(API) em "string"
String temp_down_string; //variável para converter resposta da variável "temp_down" vinda
do do servidor HTTP(API) em "string"
unsigned long Initialtime1 = 0; //variável que armazena o momento que o botão de
funcionalidades é pressionado
unsigned long Buttontime1 = 0; //variável que armazena o tempo que o botão de
funcionalidades foi mantido pressionado
unsigned long previousMillis = 0; //variável de controle para calcular tempo de requisições
bool ButtonState1; //variável auxiliar para armazenar o estado do botão de funcionalidades
do sistema
bool ButtonPreviousState1; //variável auxiliar para armazenar o estado anterior do botão de
funcionalidades do sistema
bool DetectedbySensor = false; //variável que indica se a detecção foi feita pelo
ultrassônico
const long interval = 1000; //variável tempo em milissegundos entre cada requisição GET
long duration; //variável que armazena o duration do pulso do sensor ultrassônico
float distanceCm; //variável que armazena a distância lida pelo sensor ultrassônico

//-----
//Definição de funções que serão utilizadas pelo algoritmo
//-----

//Função para gerenciar requisições "GET" via HTTP ao servidor HTTP(API)
String httpGETRequest(const char* serverName) {
    //Definição de parâmetros iniciais para requisição
    WiFiClient client;
    HTTPClient http;
```

```

//Acessando o domínio do servidor HTTP(API)
http.begin(client, serverName);

int httpResponseCode = http.GET(); //faz requisição "GET" e armazena em
"httpResponseCode" o código de resposta
String payload = "--"; //define payload com o valor de "--"

if (httpResponseCode>0) { //caso o código de resposta da requisição seja maior do que 0
    Serial.print("HTTP GET Response code: ");
    Serial.println(httpResponseCode); //mostra na porta serial o código de resposta da
requisição
    payload = http.getString(); //armazena em payload o conteúdo da resposta
}
else {
    Serial.print("Error code: "); //caso seja igual a 0 ou menor (servidor HTTP não
responde), retorna o erro
    Serial.println(httpResponseCode); //mostra na porta serial o conteúdo a resposta
    StaticJsonDocument<100> doc; //cria o JSON document em doc com o tamanho de 100 bytes
    doc["lot"] = "Sem Resposta"; //na key "lot" armazena o valor "Sem Resposta"
    serializeJson(doc, payload); //serializa doc no payload para o mesmo não ficar vazio
}
http.end(); // libera os recursos da requisição
return payload; //retorna o a resposta obtida na requisição "GET"
}

//Função para gerenciar requisições "PUT" via HTTP ao servidor HTTP(API)
void putRequest(const char* variableName, const char* variableValue) {
    //Definição de parâmetros iniciais para requisição
    WiFiClient client;
    HTTPClient http;

    // Acessando o domínio do servidor HTTP(API)
    http.begin(client, serverName_pc);

    http.addHeader("Content-Type", "application/json"); //envia um cabeçalho HTTP
informando o conteúdo da requisição (um JSON)
    String httpRequestData = "{" + String(variableName) + "":" + String(variableValue)
+ "\"}"; //formata o JSON a ser enviado
    int httpResponseCode = http.PUT(httpRequestData); //realiza o "PUT"

    if (httpResponseCode>0) { //caso o código de resposta da requisição seja maior do que 0
(há resposta servidor HTTP(API))
        Serial.print("HTTP PUT Response code: ");
        Serial.println(httpResponseCode); //mostra na porta serial o código de resposta da
requisição
    }
    else {
        Serial.print("Error code: "); //caso seja igual a 0 ou menor (servidor HTTP(API) não
responde), retorna o erro
        Serial.println(httpResponseCode); //mostra na porta serial o conteúdo a resposta
    }
    http.end(); // Libera os recursos da requisição
}

// Função para gerar um beep curto
void shortBeep (int repeat){
    for (int k = 0; k < repeat; k++){
        digitalWrite(BUZZER, HIGH); // Liga o buzzer
        delay (100);
        digitalWrite(BUZZER, LOW); // Desliga o buzzer
        delay (100);
    }
}

// Função para gerar um beep longo

```

```

void longBeep (int repeat){
  for (int k = 0; k < repeat; k++){
    digitalWrite(BUZZER, HIGH); // Liga o buzzer
    delay (500);
    digitalWrite(BUZZER, LOW); // Desliga o buzzer
    delay (500);
  }
}

//-----
//Definições gerais de sistema
//-----

void setup(){
  Serial.begin(115200); // inicia a porta serial na velocidade 115200
  pinMode(Button, INPUT_PULLUP); //configura porta do botão de funcionalidades do sistema
  para entrada
  pinMode(LED_BUILTIN, OUTPUT); //configura o led on-board para saída. Ele indicará
  presença no local
  pinMode(trigPin, OUTPUT); // configura o trigger do ultrassônico como saída
  pinMode(echoPin, INPUT); // Sets the echoPin as do ultrassônico como entrada
  pinMode(RELAY_1, OUTPUT); // configura a porta de sinal do relé 1 para saída.
  pinMode(RELAY_2, OUTPUT); // configura a porta de sinal do relé 2 para saída.
  pinMode(RELAY_3, OUTPUT); // configura a porta de sinal do relé 3 para saída.
  pinMode(RELAY_4, OUTPUT); // configura a porta de sinal do relé 4 para saída.
  pinMode(BUZZER, OUTPUT); // configura a porta de sinal do buzzer para saída.
  digitalWrite(LED_BUILTIN, LOW); //envia "low" para o led on-board para desativá-lo ao
  iniciar o programa
  digitalWrite(RELAY_1, HIGH); //envia "high" para o relé 1 para desativá-lo ao iniciar o
  programa (o relé escolhido ativa com LOW)
  digitalWrite(RELAY_2, HIGH); //envia "high" para o relé 2 para desativá-lo ao iniciar o
  programa (o relé escolhido ativa com LOW)
  digitalWrite(RELAY_3, HIGH); //envia "high" para o relé 3 para desativá-lo ao iniciar o
  programa (o relé escolhido ativa com LOW)
  digitalWrite(RELAY_4, HIGH); //envia "high" para o relé 4 para desativá-lo ao iniciar o
  programa (o relé escolhido ativa com LOW)
  digitalWrite(BUZZER, LOW); //envia "low" para o buzzer para desativá-lo ao iniciar o
  programa

  //Conexão ao Wi-Fi
  WiFi.begin(ssid, password); //tenta logar no wifi usando SSID e senhas fornecidos
  Serial.println("Conectando ao Wifi..");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  //Caso conecte, mostra na porta serial o ip do ESP-32
  Serial.println("Wifi conectado");
  Serial.println(WiFi.localIP());
  delay(3000);
  shortBeep(3); //3 beeps curtos
}

//-----
//Loop
//-----

void loop(){
  if (lot == "Local Vazio"){
    delay(10);
    // limpa trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    // mudar trigPin para HIGH por 10 microsegundos
    digitalWrite(trigPin, HIGH);
  }
}

```

```

    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH); //Lê o EchoPin, retorna o tempo da viagem da onda
    sonora em microsegundos

    distanceCm = duration * SOUND_SPEED/2; //calcula a distância em centímetros

    Serial.print("Distancia (cm): ");
    Serial.println(distanceCm);

    if ((distanceCm) < 120 && (distanceCm > 0)) {
        DetectedbySensor = true;
    }
}

ButtonState1 = !digitalRead(Button); //Lê o estado do botão

//Ao apertar o botão de funcionalidades
if (ButtonState1 && !ButtonPreviousState1) {
    if (Initialtime1 == 0) {
        Initialtime1 = millis(); //registro o momento do toque
    }
}

//Ao soltar o botão
if (Initialtime1 > 200) { //Filtro Debounce
    if (!ButtonState1 && ButtonPreviousState1) {
        Buttontime1 = millis() - Initialtime1; //calculo o tempo que o botão se manteve
        precionado
        Initialtime1 = 0; //reseto a variável Initialtime1
    }
}

//Caso o botão seja precionado por um período de tempo de 1 a 3 segundos (diminui
temperatura)
if ((Buttontime1>1000)&&(Buttontime1<=3000)){
    shortBeep(1); //1 beep curto
    Buttontime1 = 0;
    temp_down = "true";
    putRequest("temp_down",temp_down); // faz requisição PUT para trafegar informação ao
    ESP32 de controle condicionador de ar
}

//Caso o botão seja precionado por um período de tempo de 3 a 5 segundos (aumenta
temperatura)
if ((Buttontime1>3000)&&(Buttontime1<=5000)){
    shortBeep(2); //2 beeps curtos
    Buttontime1 = 0;
    temp_up = "true";
    putRequest("temp_up",temp_up); // faz requisição PUT para trafegar informação ao ESP32
    de controle condicionador de ar
}

ButtonPreviousState1 = ButtonState1;

unsigned long currentMillis = millis(); //tempo em milissegundos da execução atual do
programa
if (currentMillis - previousMillis >= interval){

    if (WiFi.status() == WL_CONNECTED){

        request_server = httpGETRequest(serverName_pc); //requisição "GET" ao servidor
        HTTP(API)
    }
}

```


APÊNDICE C - ALGORITMO DO SISTEMA DE CONTROLE DO CONDICIONADOR DE AR

```
#include <Arduino.h> //Biblioteca padrão para trabalhar com placas microcontroladoras
#include <WiFi.h> //Biblioteca para trabalhar com wifi
#include <HttpClient.h> //Biblioteca para trabalhar com requisições
#include <ArduinoJson.h> //Biblioteca para manipular json
#include <IRremote.hpp> //Biblioteca para manipular emissores e receptores IR
```

```
IRsend irsend; //Configuração inicial para envio de códigos Raw
```

```
//Código Raw (Infravermelho) para ligar o condicionador de ar
```

```
const uint16_t turn_on [199] = {4450, 4450,
    550,1650, 500, 550, 550,1650, 500,1650, 500,
    600, 450, 600, 500,1650, 550, 550, 500, 600,
    500,1650, 500, 550, 550, 550, 500,1650, 550,
    1650, 500, 550, 500,1650, 550, 550, 550,1600,
    550, 550, 500,1650, 500,1650, 550,1650, 550,
    1600, 550,1600, 550,1650, 550, 550, 500,1650,
    550, 550, 500, 550, 550, 550, 500, 550, 550,
    550, 500,1650, 550, 550, 500,1650, 550,1600,
    550, 550, 550, 500, 550, 550, 550, 500, 550,
    550, 550,1600, 550, 550, 550, 500, 550,1650,
    550,1600, 550,1650, 500,1650, 550,5250,4450,
    4450, 550,1600, 550, 550, 550,1600, 550,1650,
    500, 550, 550, 550, 500,1650, 550, 550, 500,
    550, 550,1650, 500, 550, 550, 550, 500,1650,
    550,1600, 550, 550, 550,1600, 550, 550, 550,
    1600, 550, 550, 550,1600, 550,1600, 550,1650,
    550,1600, 550,1650, 500,1650, 550, 500, 550,
    1650, 550, 500, 550, 550, 550, 550, 500, 550,
    550, 550, 500,1650, 550, 550, 500,1650, 550,
    1600, 550, 550, 550, 550, 500, 550, 550, 550,
    500, 600, 500,1650, 500, 600, 500, 550, 500,
    1650, 550,1650, 500,1650, 500,1700, 500};
```

```
//Código Raw (Infravermelho) para Desligar o condicionador de ar
```

```
const uint16_t turn_off [199] = {4450,4450
    , 550,1650 , 500, 550 , 550,1650 , 500,1650
    , 550, 550 , 500, 550 , 550,1650 , 500, 550
    , 550, 550 , 500,1650 , 500, 600 , 500, 550
    , 550,1650 , 500,1650 , 500, 600 , 500,1650
    , 500, 600 , 500,1650 , 500,1650 , 500,1700
    , 500,1650 , 500, 600 , 500,1650 , 500,1650
    , 550,1650 , 500, 550 , 500, 600 , 500, 550
    , 550, 550 , 500,1650 , 500, 600 , 500, 550
    , 550,1650 , 500,1650 , 500,1650 , 550, 550
    , 500, 600 , 500, 550 , 500, 600 , 500, 550
    , 500, 600 , 500, 550 , 550, 550 , 550,1600
    , 500,1650 , 550,1600 , 550,1650 , 550,1600
    , 550,5250 ,4500,4400 , 550,1600 , 550, 550
    , 550,1600 , 550,1650 , 550, 500 , 550, 550
    , 550,1600 , 550, 550 , 550, 500 , 550,1650
    , 550, 500 , 550, 550 , 550,1600 , 550,1650
    , 500, 550 , 550,1650 , 500, 550 , 550,1600
    , 550,1650 , 550,1600 , 550,1650 , 500, 550
    , 550,1600 , 550,1650 , 550,1600 , 550, 550
    , 500, 550 , 550, 550 , 550, 500 , 550,1650
    , 550, 500 , 550, 550 , 550,1600 , 550,1650
    , 500,1650 , 550, 550 , 500, 550 , 550, 550
    , 500, 550 , 550, 550 , 550, 500 , 550, 550
    , 550, 550 , 500,1650 , 550,1650 , 500,1650
    , 500,1650 , 550,1650 , 500};
```

```
//Código Raw (Infravermelho) para aumentar a temperatura do condicionador de ar
```

```

const uint16_t increase_temp [199] = {4450,4450
, 550,1650 , 500, 550 , 500,1700 , 500,1650
, 500, 550 , 550, 550 , 500,1650 , 550, 550
, 500, 550 , 550,1650 , 500, 550 , 550, 550
, 500,1650 , 550,1650 , 500, 550 , 550,1650
, 500, 550 , 500, 600 , 500,1650 , 500,1650
, 550,1650 , 500,1650 , 500,1700 , 500,1650
, 500,1650 , 550,1650 , 500, 550 , 500, 600
, 500, 550 , 550, 550 , 500, 550 , 550, 550
, 500,1650 , 550, 550 , 500,1650 , 550,1650
, 500, 550 , 500, 600 , 500, 550 , 550, 550
, 500, 550 , 550,1650 , 500, 550 , 550, 550
, 500,1650 , 550,1650 , 500,1650 , 500,1650
, 550,5250 ,4450,4450 , 550,1650 , 500, 550
, 550,1650 , 500,1650 , 500, 600 , 500, 550
, 500,1650 , 550, 550 , 500, 600 , 500,1600
, 550, 600 , 500, 550 , 500,1650 , 550,1600
, 550, 550 , 550,1600 , 550, 550 , 550, 500
, 550,1650 , 500,1650 , 550,1600 , 550,1650
, 550,1600 , 550,1600 , 550,1650 , 550,1600
, 550, 550 , 550, 500 , 550, 550 , 550, 500
, 550, 550 , 550, 550 , 500,1650 , 550, 500
, 550,1650 , 550,1600 , 550, 550 , 550, 500
, 550, 550 , 550, 500 , 550, 550 , 550,1600
, 550, 550 , 550, 550 , 500,1650 , 550,1600
, 550,1650 , 500,1650 , 550};

```

```

//Código Raw (Infravermelho) para diminuir a temperatura do condicionador de ar
const uint16_t decrease_temp [199] = {4450,4450

```

```

, 550,1650 , 500, 550 , 550,1650 , 500,1650
, 500, 600 , 500, 550 , 500,1700 , 500, 550
, 500, 600 , 500,1650 , 500, 600 , 500, 550
, 500,1650 , 550,1650 , 500, 550 , 550,1650
, 500, 550 , 550, 550 , 500,1650 , 550,1650
, 500,1650 , 500,1650 , 550,1650 , 500,1650
, 500,1650 , 550,1650 , 500, 550 , 550, 550
, 500, 550 , 550, 550 , 500, 600 , 500, 550
, 500,1650 , 550, 550 , 500, 600 , 500, 550
, 500, 600 , 500, 550 , 550, 550 , 500, 550
, 550, 550 , 500,1650 , 550,1650 , 500,1650
, 500,1650 , 550,1600 , 550,1650 , 500,1650
, 550,5250 ,4450,4450 , 550,1600 , 550, 550
, 550,1600 , 550,1650 , 500, 550 , 550, 550
, 550,1600 , 550, 550 , 500, 550 , 550,1650
, 500, 550 , 550, 550 , 500,1650 , 550,1600
, 550, 550 , 550,1600 , 550, 550 , 550, 500
, 550,1650 , 550,1600 , 550,1650 , 500,1650
, 550,1600 , 550,1650 , 500,1650 , 550,1600
, 550, 550 , 550, 500 , 550, 550 , 550, 550
, 500, 550 , 550, 550 , 500,1700 , 500, 550
, 500, 600 , 500, 550 , 550, 550 , 500, 550
, 550, 550 , 500, 550 , 550, 550 , 500,1650
, 550,1650 , 500,1650 , 500,1700 , 500,1650
, 500,1650 , 500,1700 , 500};

```

```

// SSID e senha da rede Wi-Fi - Roteador do Projeto
const char* ssid = "acesspoint";
const char* password = "Tcc##1501";

```

```

int i = 0; //variável auxiliar para loop's "for"
int LED_BULTIN = 2; //definição da porta de led on-board
int IR_PIN = 4; //definição da porta que emitirá os comandos IR
String request_server; //variável que armazena a requisição a API
String lot; //variável que armazena a presença no local obtida através da API
String temp_up_string; //variável para converter resposta da variável temp_up vinda do
servidor

```

```

String temp_down_string; //variável para converter resposta da variável temp_down vinda do
servidor
String system_state = "Desligado"; //variável auxiliar que monitora o estado do
condicionador de ar(usada para evitar que comando se repitam a cada requisição GET)
const char* temp_up = "false"; //variável que controla o aumento de temperatura
const char* temp_down = "false"; //variável que controla a diminuição de temperatura
const char* serverName_pc = "http://192.168.1.2:8000"; //IP da API/Servidor HTTP(API)
unsigned long previousMillis = 0; //variável de controle para calcular tempo de requisições
const long interval = 1000; //tempo em milisegundos entre cada requisição GET

//Função para gerenciar requisições GET via HTTP ao Servidor HTTP(API)
String httpGETRequest(const char* serverName) {
    //Definição de parâmetros iniciais para requisição
    WiFiClient client;
    HTTPClient http;

    //Acessando o domínio do Servidor HTTP(API)
    http.begin(client, serverName);

    int httpResponseCode = http.GET(); //faz requisição GET e armazena em httpResponseCode o
código de resposta
    String payload = "--"; //define payload com o valor de "--"

    if (httpResponseCode>0) { //caso o código de resposta da requisição seja maior do que 0
        Serial.print("HTTP GET Response code: ");
        Serial.println(httpResponseCode); //mostra na porta serial o código de resposta da
requisição
        payload = http.getString(); //armazena em payload o conteúdo da resposta
    }
    else {
        Serial.print("Error code: "); //caso seja igual a 0 ou menor (Servidor HTTP(API) não
responde), retorna o erro
        Serial.println(httpResponseCode); //mostra na porta serial o conteúdo a resposta
        StaticJsonDocument<100> doc; //cria o JSON document em doc com o tamanho de 100 bytes
        doc["lot"] = "Sem Resposta"; //na key "lot" armazena o valor "Sem Resposta"
        serializeJson(doc, payload); //serializa doc no payload para o mesmo não ficar vazio
    }
    http.end(); //libera os recursos da requisição
    return payload; //retorna o a resposta obtida na requisição GET
}

//Função para gerenciar requisições PUT via HTTP ao Servidor HTTP(API)
void putRequest(const char* variableName, const char* variableValue) {
    //Definição de parâmetros iniciais para requisição
    WiFiClient client;
    HTTPClient http;

    // Acessando o domínio do Servidor HTTP(API)
    http.begin(client, serverName_pc);

    http.addHeader("Content-Type", "application/json"); //envia um cabeçalho HTTP informa o
conteúdo da requisição (um JSON)
    String httpRequestData = "{" + String(variableName) + "":" + String(variableValue)
+ "}" + "; //formata o JSON a ser enviado
    int httpResponseCode = http.PUT(httpRequestData); //

    if (httpResponseCode>0) { //caso o código de resposta da requisição seja maior do que 0
(há resposta Servidor HTTP(API))
        Serial.print("HTTP PUT Response code: "); //faz requisição PUT e armazena em
httpResponseCode o código de resposta
        Serial.println(httpResponseCode); //mostra na porta serial o código de resposta da
requisição
    }
    else {
        Serial.print("Error code: "); //caso seja igual a 0 ou menor (Servidor HTTP(API) não
responde), retorna o erro
    }
}

```

```

        Serial.println(httpResponseCode); //mostra na porta serial o conteúdo a resposta
    }
    http.end(); // Libera os recursos da requisição
}

void setup(){
    Serial.begin(115200); // Inicial a porta serial na velocidade 115200 apenas para
propósito de debugs
    pinMode(LED_BUILTIN, OUTPUT); //Configura o led onboard para saída. Ele indicará presença
no local
    pinMode(IR_PIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW); //envia "low" para o led on-board para desativar-lo ao
iniciar o programa
    IrSender.begin(IR_PIN); //inicializa o emissão de códigos IR na porta escolhida

    for (i = 0; i < 3; i++) { //Ao inicializar o sistema envia comando para Desligar
condicionador de Ar
        irsend.sendRaw(turn_off, 199, 38); //envia Raw para Desligar de 199 posições na
frequência de em 38kHz
        delay(300);
    }

    // Conexão ao Wi-Fi
    WiFi.begin(ssid, password);
    Serial.println("Conectando ao Wifi..");
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.print(".");
    }
    // Mostrando na porta serial, o ip do ESP-32
    Serial.println("Wifi conectado");
    Serial.println(WiFi.localIP());
}

void loop() {

    unsigned long currentMillis = millis(); //tempo em milisegundos de execução atual do
programa
    if (currentMillis - previousMillis >= interval){
        if (WiFi.status() == WL_CONNECTED){

            request_server = httpGETRequest(serverName_pc); //requisição GET à API

            StaticJsonDocument<200> doc; //requisição GET ao Servidor HTTP(API)
            deserializeJson(doc, request_server); //deserializa o doc em request_server
            lot = doc["lot"].as<String>(); //formata "lot" do JSON como string para que seja
manipulada
            temp_up_string = doc["temp_up"].as<String>(); //formata "temp_up" do JSON como
"string" para que seja manipulada
            temp_down_string = doc["temp_down"].as<String>(); //formata "temp_down" do JSON como
string para que seja manipulada
            temp_up = temp_up_string.c_str(); //armazena o conteúdo da "string" obtida na "const
char" da manipula o aumento de temperatura
            temp_down = temp_down_string.c_str(); //armazena o conteúdo da "string" obtida na
"const char" da manipula o aumento de temperatura

            Serial.print("JSON = "); //Mostra na porta serial o valor de "lot" obtido
            Serial.println(lot);

            if (temp_up_string == "true"){
                irsend.sendRaw(increase_temp, 199, 38); //envia Raw para aumentar temperatura de
199 posições na frequência de em 38kHz
                temp_up = "false"; //retornar "temp_up" para false (isto serve para reportar que o
comando foi finalizado)
                putRequest("temp_up",temp_up); // faz requisição PUT para informar o novo estado de
"temp_up"
            }
        }
    }
}

```

```

    }

    if (temp_down_string == "true"){
        irsend.sendRaw(decrease_temp, 199, 38); //envia Raw para diminuir temperatura de
199 posições na frequência de em 38kHz
        temp_down = "false"; //retornar "temp_down" para false (isto serve para reportar
que o comando foi finalizado)
        putRequest("temp_down",temp_down); // faz requisição PUT para informar o novo
estado de "temp_up"
    }

    if ((lot == "Pessoas Detectadas")&&(system_state == "Desligado")){ //Se pessoas forem
de detectadas
        digitalWrite(LED_BUILTIN, HIGH); //ativa led on-board
        for (i = 0; i < 3; i++) { //Laço "for" para repetir comando 3 vezes
            irsend.sendRaw(turn_on, 199, 38); //envia Raw para Ligar de 199 posições na
frequência de em 38kHz
            delay(300);
        }
        system_state = "Ligado";
    }
    if ((lot == "Sem Resposta")&&(system_state == "Ligado")){
        for (i = 0; i < 3; i++) { //Laço "for" para repetir comando 3 vezes
            irsend.sendRaw(turn_off, 199, 38); //envia Raw para Desligar de 199 posições na
frequência de em 38kHz
            delay(300);
        }
        system_state = "Desligado";
    }
    if ((lot == "Local Vazio")&&(system_state == "Ligado")){ //Caso pessoas não sejam
detectadas
        digitalWrite(LED_BUILTIN, LOW); //desativa led on-board
        for (i = 0; i < 3; i++) { //Laço "for" para repetir comando 3 vezes
            irsend.sendRaw(turn_off, 199, 38); //envia Raw para Desligar de 199 posições na
frequência de em 38kHz
            delay(300);
        }
        system_state = "Desligado";
    }
    if ((lot == "Suspensa")&&(system_state == "Ligado")){ //Caso o Sistema de Detecção
esteja suspenso
        digitalWrite(LED_BUILTIN, LOW); //desativa led on-board
        for (i = 0; i < 3; i++) { //Laço "for" para repetir comando 3 vezes
            irsend.sendRaw(turn_off, 199, 38); //envia Raw para Desligar de 199 posições na
frequência de em 38kHz
            delay(300);
        }
        system_state = "Desligado";
    }
    previousMillis = currentMillis; //manipulação de tempo para próxima requisição get
}
else{
    Serial.println("Wifi Desconectado!"); //Caso não haja resposta do Wifi
}
}
}
}

```

APÊNDICE D - ALGORITMOS DO SISTEMA DE ACOMPANHAMENTO

Servidor HTTP (Python)

```

from flask import Flask,render_template #importa biblioteca do Flask e função
render_template

app = Flask(__name__) #cria aplicação Flask

@app.route("/") #define a rota "/" para a função "home"
@app.route("/home") #define a rota "/home" para a função "home"
def home (): #cria a função "home" para servir a página inicial da interface gráfica
    response = render_template('home.html') #renderiza em response o html "home.html" para
    servi-lo na rota "/" ou "/home"
    return response #retorna response como o resultado da função quando o usuário acessar a
    rota

@app.route("/autores") #define a rota "/autores" para a função "autores"
def autores(): #cria a função "autores" para servir a página com informações sobre os
    autores
    response = render_template('autores.html') #renderiza em response o html "autores.html"
    para servi-lo na rota "/autores"
    return response #retorna response como o resultado da função quando o usuário acessar a
    rota

@app.route("/manual") #define a rota "/manual" para a função "manual"
def manual(): #cria a função "manual" para servir a página com informações sobre o sistema
    response = render_template('instrucoes_uso.html') #renderiza em response o html
    "instrucoes_uso.html" para servi-lo na rota "/manual"
    return response #retorna response como o resultado da função quando o usuário acessar a
    rota

# Inicia aplicação
if __name__ == "__main__":
    app.run(host = '0.0.0.0', port= 5000, debug=True) #servidor iniciaria no endereço
    "localhost", porta "5000"

```

HTML da Página do Painel da Automação (HTML)

```

<!DOCTYPE html>
<html lang="pt-br">
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content = "width=device-width, initial-scale=1.0" />
        <title>Painel da Automação</title>
        <link rel="stylesheet" href="{{ url_for ('static', filename='Style/home.css')}}"/>
        <script src="{{ url_for ('static', filename = 'JavaScript/home.js')}}" defer></script>
    </head>
    <body>
        <header>
            
            <h1>Painel da Automação</h1>
            
        </header>
        <div class="funcionalidades">
            <div class="info_projeto">
                <a href="http://192.168.1.2:5000/autores">Sobre os Autores</a>
                <a href="http://192.168.1.2:5000/manual">Sobre o Sistema</a>
            </div>
            <div class="relogio">
                <div class = "hora" id = "hora">12:00:00</div>
            </div>
        </div>
        <div class="status_sistemas">

```

```

<div class="deteccao">
  <h1>Detecção</h1>
  <div class = "deteccao_lot">
    <h2 id = "lot-local"></h2>
  </div>
</div>
<div class="ilum">
  <h1>Iluminação</h1>
  <div class = "ilum_status">
    <h2 id = "ilum_status"></h2>
  </div>
</div>
<div class="ar1">
  <h1>Ar-Condicionado 1</h1>
  <div class = "ar1_status">
    <h2 id = "ar1_status"></h2>
  </div>
</div>
<div class="ar2">
  <h1>Ar-Condicionado 2</h1>
  <div class = "ar2_status">
    <h2 id = "ar2_status"></h2>
  </div>
</div>
</body>
<footer>
  <br>
  <h1>Desenvolvido por Lucas Kaio Pereira Dantas e Álvaro Alves Sousa</h1>
  <h2>Alunos do curso de Engenharia Elétrica do IFTO-Campus Palmas-2019 a 2024</h2>
  <h2>Projeto realizado para obtenção de aprovação na disciplina de TCC 2</h2>
</footer>
</html>

```

CSS da Página do Paine! da Automação (CSS)

```

*{
  font-family: Verdana;
}
header{
  color: white;
  background-color: rgba(22, 59, 102, 0.74);
  padding: 10px;
  background-position: center;
  border-radius: 5px;
  display: flex;
  justify-content: space-between;
  list-style-type: none;
}
img{
  width: 100px;
  height: 100px;
}
.funionalidades{
  display: flex;
  background-color: rgba(22, 59, 102, 0.74);
  justify-content: space-between;
  margin-top: 10px;
  margin-bottom: 10px;
  border-radius: 5px;
  padding-left: 20px;
  padding-right: 20px;
}
.relogio{
  font-family: Arial, sans-serif;
  font-size: 50px;
  justify-content: center;
}

```

```

    align-items: center;
    display: flex;
    margin-top: 10px;
    background: rgba(18, 138, 207, 0.699);
    width: 310px;
    height: 90px;
    border-radius: 10px;
    border-style: solid;
    border-color: rgb(49, 49, 49);
}
.info_projeto{
    list-style-type: none;
    display: flex;
    justify-content: space-around;
    align-items: center;
}
a{
    margin: 20px;
    color: white;
    text-decoration: None;
    background: rgba(18, 138, 207, 0.699);
    border-radius: 10px;
    padding: 30px;
    width: 150px;
    font-size: 20px;
}
body {
    background-color: rgb(31, 31, 31);
    color: white;
    text-align: center;
    background-image: url("Images/background.jpg");
    background-position: center;
}
.status_sistemas{
    display: flex;
    justify-content: space-between;
}
}
.deteccao,.ilum,.ar1,.ar2{
    color : rgb(255, 255, 255);
    font-size: 10px;
    background: rgba(18, 138, 207, 0.699);
    width: 420px;
    margin: 0 auto;
    border-radius: 10px;
    padding: 5px;
    border: 2px;
    border-style: solid;
    border-color: rgb(49, 49, 49);
}
.deteccao_lot.green-bg,.ilum_status.green-bg,.ar1_status.green-bg,.ar2_status.green-bg {
    font-size: 15px;
    width: 300px;
    margin: 0 auto;
    border-radius: 10px;
    padding: 1px;
    background-color: rgb(11, 255, 11);
}
.green-text {
    color: rgb(10, 66, 10);
}
.deteccao_lot.red-bg,.ilum_status.red-bg,.ar1_status.red-bg,.ar2_status.red-bg {
    font-size: 15px;
    width: 300px;
    margin: 0 auto;
    border-radius: 10px;
}

```

```

padding: 1px;
background-color: rgb(218, 11, 11);
}
.red-text {
color: rgb(82, 11, 11);
}
.deteccao_lot.yellow-bg,.ilum_status.yellow-bg,.ar1_status.yellow-bg,.ar2_status.yellow-bg
{
font-size: 15px;
width:300px;
margin: 0 auto;
border-radius: 10px;
padding: 1px;
background-color: rgb(247, 181, 15);
}
.yellow-text {
color: rgb(105, 83, 11);
}
footer{
font-size: 5px;
text-align: bottom;
}

@media (max-width:450px){
*{
font-family: Verdana;
}
header{
color: white;
background-color: rgba(22, 59, 102, 0.74);
padding: 10px;
background-position: center;
border-radius: 5px;
display: flex;
justify-content: space-between;
list-style-type: none;
width: 97%;
font-size: 10px;
}
img{
width: 50px;
height: 50px;
}
.funcionalidades{
display: list-item;
background-color: rgba(22, 59, 102, 0.74);
list-style: disc;
align-items: center;
justify-content: center;
margin-top: 10px;
margin-bottom: 10px;
border-radius: 5px;
margin: 10px;
margin-left: 0;
margin-right: 0;
}
.relogio{
font-family: Arial, sans-serif;
font-size: 50px;
justify-content: center;
align-items: center;
display: flex;
margin-top: 10px;
background: rgba(18, 138, 207, 0.699);
width:325px;

```

```

    height: 90px;
    border-radius: 10px;
    border-style: solid;
    border-color: rgb(49, 49, 49);
    margin: 5px;
}

body {
    background-color: rgb(31, 31, 31);
    color: white;
    text-align: center;
    background-image: url("Images/background.jpg");
    background-position: center;
    background-size: 200%;
}

.status_sistemas{
    display: list-item;
    justify-content: center;
}

}

.deteccao,.ilum,.ar1,.ar2{
    color : rgb(255, 255, 255);
    font-size: 15px;
    background: rgba(18, 138, 207, 0.699);
    width:350px;
    margin: 0 auto;
    border-radius: 10px;
    padding: 5px;
    border: 2px;
    border-style: solid;
    border-color: rgb(49, 49, 49);
}

.deteccao_lot.green-bg,.ilum_status.green-bg,.ar1_status.green-bg,.ar2_status.green-bg {
    font-size: 15px;
    width:300px;
    margin: 0 auto;
    margin-bottom: 3px;
    border-radius: 10px;
    padding: 1px;
    background-color: rgb(11, 255, 11);
}

.green-text {
    color: rgb(10, 66, 10);
}

.deteccao_lot.red-bg,.ilum_status.red-bg,.ar1_status.red-bg,.ar2_status.red-bg {
    font-size: 15px;
    width:300px;
    margin: 0 auto;
    margin-bottom: 3px;
    border-radius: 10px;
    padding: 1px;
    background-color: rgb(218, 11, 11);
}

.red-text {
    color: rgb(82, 11, 11);
}

.deteccao_lot.yellow-bg,.ilum_status.yellow-bg,.ar1_status.yellow-bg,.ar2_status.yellow-bg{
    font-size: 15px;
    width:300px;
    margin: 0 auto;
    margin-bottom: 3px;
    border-radius: 10px;
    padding: 1px;
    background-color: rgb(247, 181, 15);
}
}

```

JavaScript da Página do Pannel da Automação (JavaScript)

```
function atualizarStatus(){
  fetch('http://192.168.1.2:8000/')
    .then(response => response.json())
    .then(data => {
      const lotElement = document.getElementById('lot-local');
      const ilumElement = document.getElementById('ilum_status');
      const ar1Element = document.getElementById('ar1_status');
      const ar2Element = document.getElementById('ar2_status');

      const deteccaolotElement = document.querySelector('.deteccao_lot');
      const ilumStatusElement = document.querySelector('.ilum_status');
      const ar1StatusElement = document.querySelector('.ar1_status');
      const ar2StatusElement = document.querySelector('.ar2_status');

      deteccaolotElement.classList.remove('green-bg', 'red-bg', 'yellow-bg',
      'green-text', 'red-text', 'yellow-text');
      ilumStatusElement.classList.remove('green-bg', 'red-bg', 'yellow-bg',
      'green-text', 'red-text', 'yellow-text');
      ar1StatusElement.classList.remove('green-bg', 'red-bg', 'yellow-bg',
      'green-text', 'red-text', 'yellow-text');
      ar2StatusElement.classList.remove('green-bg', 'red-bg', 'yellow-bg',
      'green-text', 'red-text', 'yellow-text');

      if (data.lot === "Pessoas Detectadas") {
        lotElement.textContent = data.lot;
        deteccaolotElement.classList.add('green-bg');
        deteccaolotElement.classList.add('green-text');

        ilumElement.textContent = data.ilum_status;
        ilumStatusElement.classList.add('green-bg');
        ilumStatusElement.classList.add('green-text');

        ar1Element.textContent = data.ar1_status;
        ar1StatusElement.classList.add('green-bg');
        ar1StatusElement.classList.add('green-text');

        ar2Element.textContent = data.ar2_status;
        ar2StatusElement.classList.add('green-bg');
        ar2StatusElement.classList.add('green-text');
      } else if (data.lot === "Local Vazio") {
        lotElement.textContent = data.lot;
        deteccaolotElement.classList.add('red-bg');
        deteccaolotElement.classList.add('red-text');

        ilumElement.textContent = data.ilum_status;
        ilumStatusElement.classList.add('red-bg');
        ilumStatusElement.classList.add('red-text');

        ar1Element.textContent = data.ar1_status;
        ar1StatusElement.classList.add('red-bg');
        ar1StatusElement.classList.add('red-text');
      }
    })
}
```

```

        ar2Element.textContent = data.ar2_status;
        ar2StatusElement.classList.add('red-bg');
        ar2StatusElement.classList.add('red-text');
    }else if (data.lot === "Suspensa") {
        lotElement.textContent = data.lot;
        deteccaoLotElement.classList.add('yellow-bg');
        deteccaoLotElement.classList.add('yellow-text');

        ilumElement.textContent = data.ilum_status;
        ilumStatusElement.classList.add('red-bg');
        ilumStatusElement.classList.add('red-text');

        ar1Element.textContent = data.ar1_status;
        ar1StatusElement.classList.add('red-bg');
        ar1StatusElement.classList.add('red-text');

        ar2Element.textContent = data.ar2_status;
        ar2StatusElement.classList.add('red-bg');
        ar2StatusElement.classList.add('red-text');
    }
    setTimeout(atualizarStatus, 1000);
})
.catch(error => {
    console.error('Error:', error);
    const lotElement = document.getElementById('lot-local');
    const ilumElement = document.getElementById('ilum_status');
    const ar1Element = document.getElementById('ar1_status');
    const ar2Element = document.getElementById('ar2_status');

    const deteccaoLotElement = document.querySelector('.deteccao_lot');
    const ilumStatusElement = document.querySelector('.ilum_status');
    const ar1StatusElement = document.querySelector('.ar1_status');
    const ar2StatusElement = document.querySelector('.ar2_status');

    lotElement.textContent = "Sem Resposta";
    deteccaoLotElement.classList.add('yellow-bg');
    deteccaoLotElement.classList.add('yellow-text');

    ilumElement.textContent = "Sem Resposta";
    ilumStatusElement.classList.add('yellow-bg');
    ilumStatusElement.classList.add('yellow-text');

    ar1Element.textContent = "Sem Resposta";
    ar1StatusElement.classList.add('yellow-bg');
    ar1StatusElement.classList.add('yellow-text');

    ar2Element.textContent = "Sem Resposta";
    ar2StatusElement.classList.add('yellow-bg');
    ar2StatusElement.classList.add('yellow-text');
    setTimeout(atualizarStatus, 1000);
});
}
function updateTime() {
    const timeElement = document.getElementById('hora');
    const now = new Date();
    const hours = now.getHours().toString().padStart(2, '0');

```


ajusta as condições da sala para proporcionar o melhor conforto e eficiência energética. Os parâmetros de captura e análise são configuráveis, permitindo ajustes precisos conforme a necessidade.

Para um detalhamento completo do funcionamento, consulte o nosso trabalho de conclusão de curso na íntegra. A seguir, há também algumas informações sobre as funcionalidades do sistema. Aproveite a tecnologia que transforma suas aulas!

Botão de Funcionalidades

Na central próximo ao interruptor da sala há um botão onde é possível controlar as seguintes funcionalidades:

Diminuir Temperatura

(pressione o botão por um tempo de 1 a 3 segundos)

Envia comando para ar-condicionado diminuir 1 grau celsius de temperatura.

Aguarde alerta sonoro do ar-condicionado para executar o comando novamente.

Aumentar Temperatura

(pressione o botão por um tempo de 3 a 5 segundos)

Envia comando para ar-condicionado aumentar 1 grau celsius de temperatura.

Aguarde alerta sonoro do ar-condicionado para executar o comando novamente.

Protocolos de Hora

Visando maximizar a economia de energia, o sistema foi configurado para suspender o seu funcionamento em determinados dias e horários, onde não há uso do local. Esses dias são:

- Segunda a Sexta: das 12h às 13h.
- Segunda a Sexta: das 23h às 07h.
- Sábados e Domingos: dia todo.

No início de cada protocolo de hora os sistemas de iluminação e climatização receberão comando para desligarem.

Alertas Sonoros

O sistema da iluminação localizado próximo ao interruptor pode emitir beep's sonoros indicando algumas informações sobre o funcionamento do sistema:

- 1 beep curto - função para diminuir temperatura executada!
- 2 beep's curtos - função para aumentar temperatura executada!
- 3 beep's curtos - Sistema Iniciado!

Desenvolvido por Lucas Kaio Pereira Dantas e Álvaro Alves Sousa

Alunos do curso de Engenharia Elétrica do IFTO-Campus Palmas-2019 a 2024

Projeto realizado para obtenção de aprovação na disciplina de TCC 2

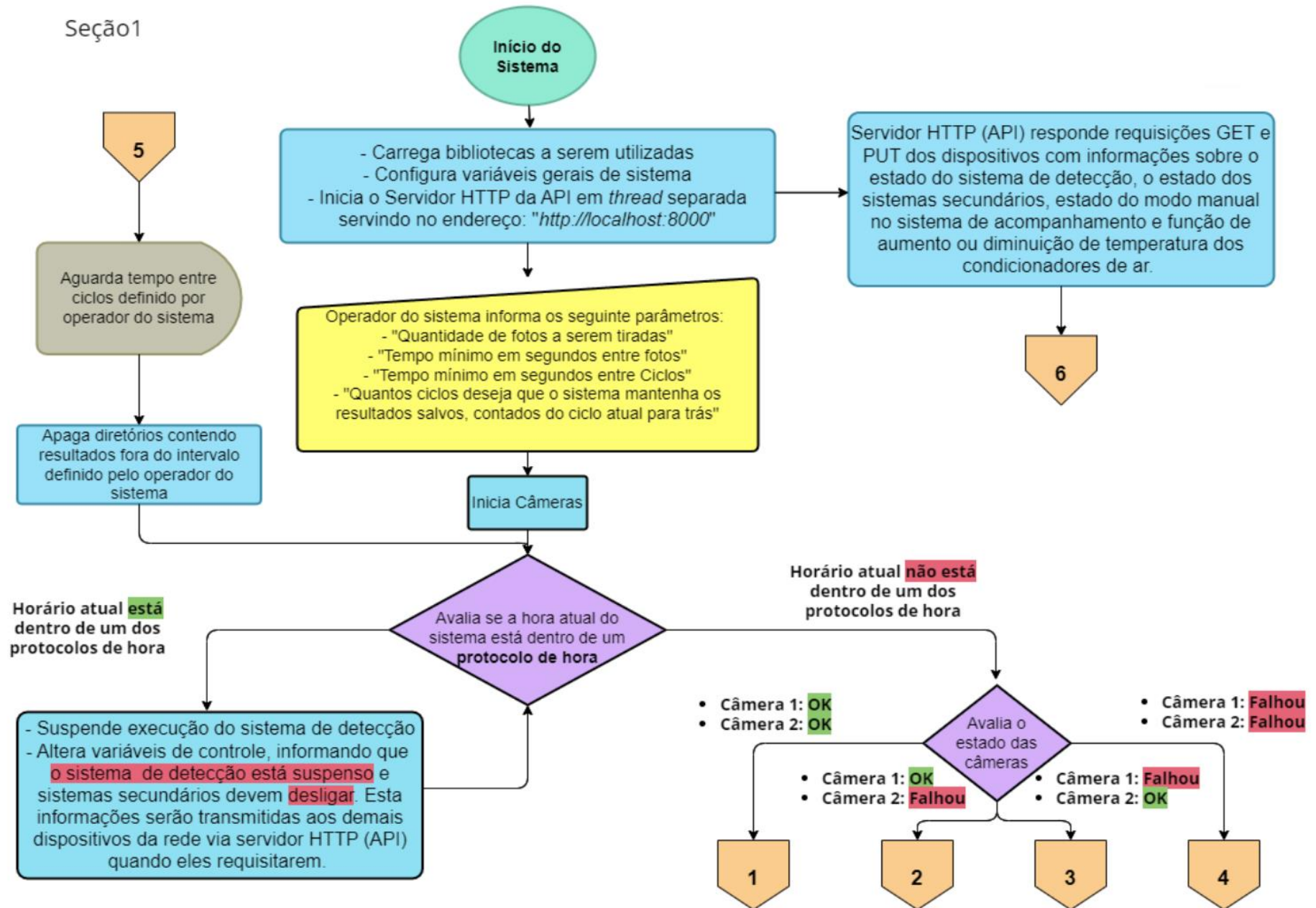
CSS da Página Sobre o Sistema (CSS)

```

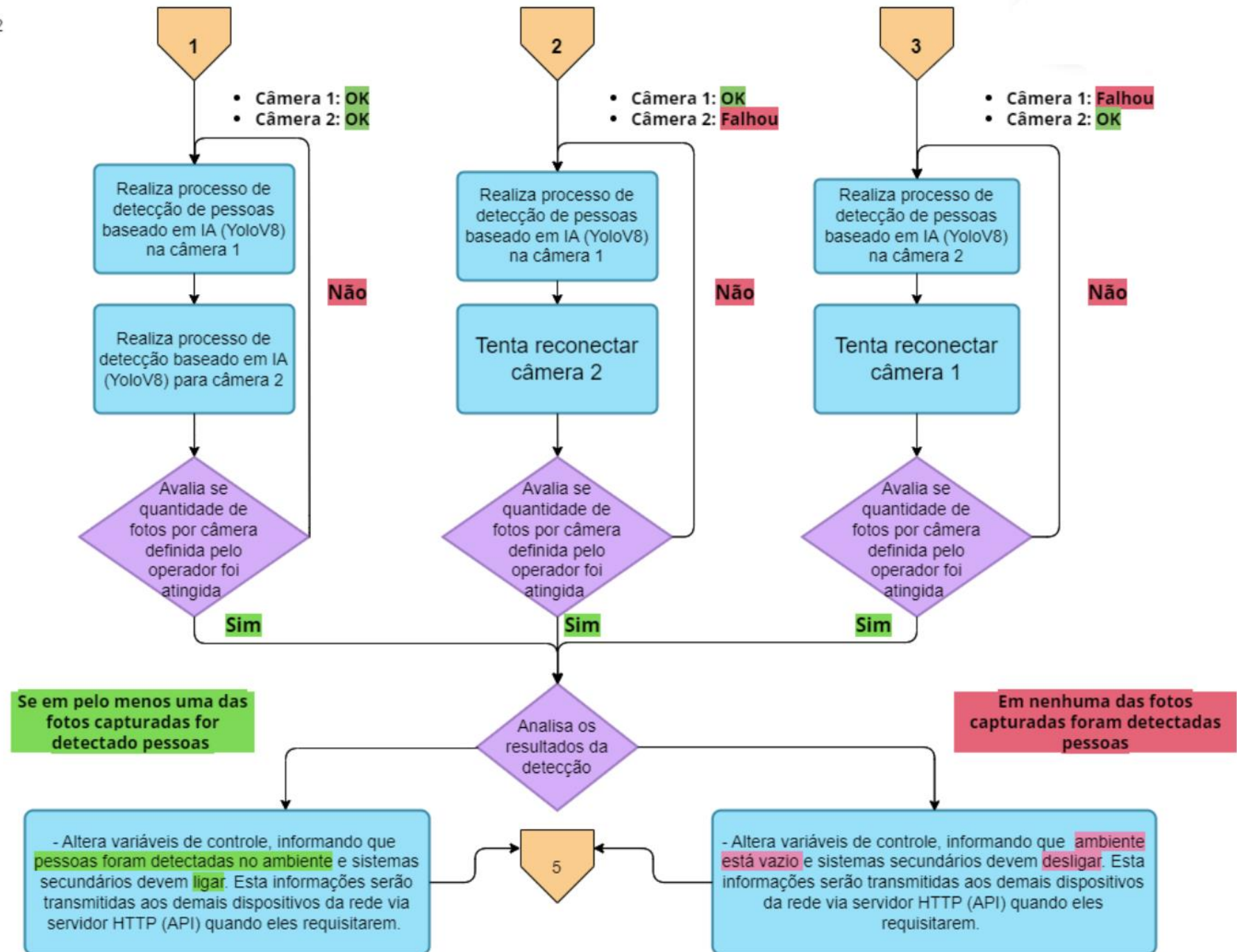
*{
    font-family: Verdana;
}
header{
    padding: 20px;
    justify-content: center;
}
body {
    background-color: rgb(31, 31, 31);
    color: white;
    text-align: center;
    background-image: url("Images/background.jpg");
    background-position: center;
}
p{
    margin: 0 auto;
    font-size: 13px;
    text-align: justify;
    width: 550px;
}
footer{
    font-size: 5px;
    text-align: bottom;
}
h3{
    font-size: 15px;
}
.text{
    background-color: rgba(0, 0, 0, 0.589);
    width: 600px;
    margin: auto;
    padding: 30px;
    border-radius: 15px;
}
@media (max-width: 450px){
    p{
        width: 300px;
    }
    .text{
        background-color: rgba(0, 0, 0, 0.589);
        width: 300px;
        border-radius: 15px;
    }
    header{
        justify-content: center;
    }
}

```

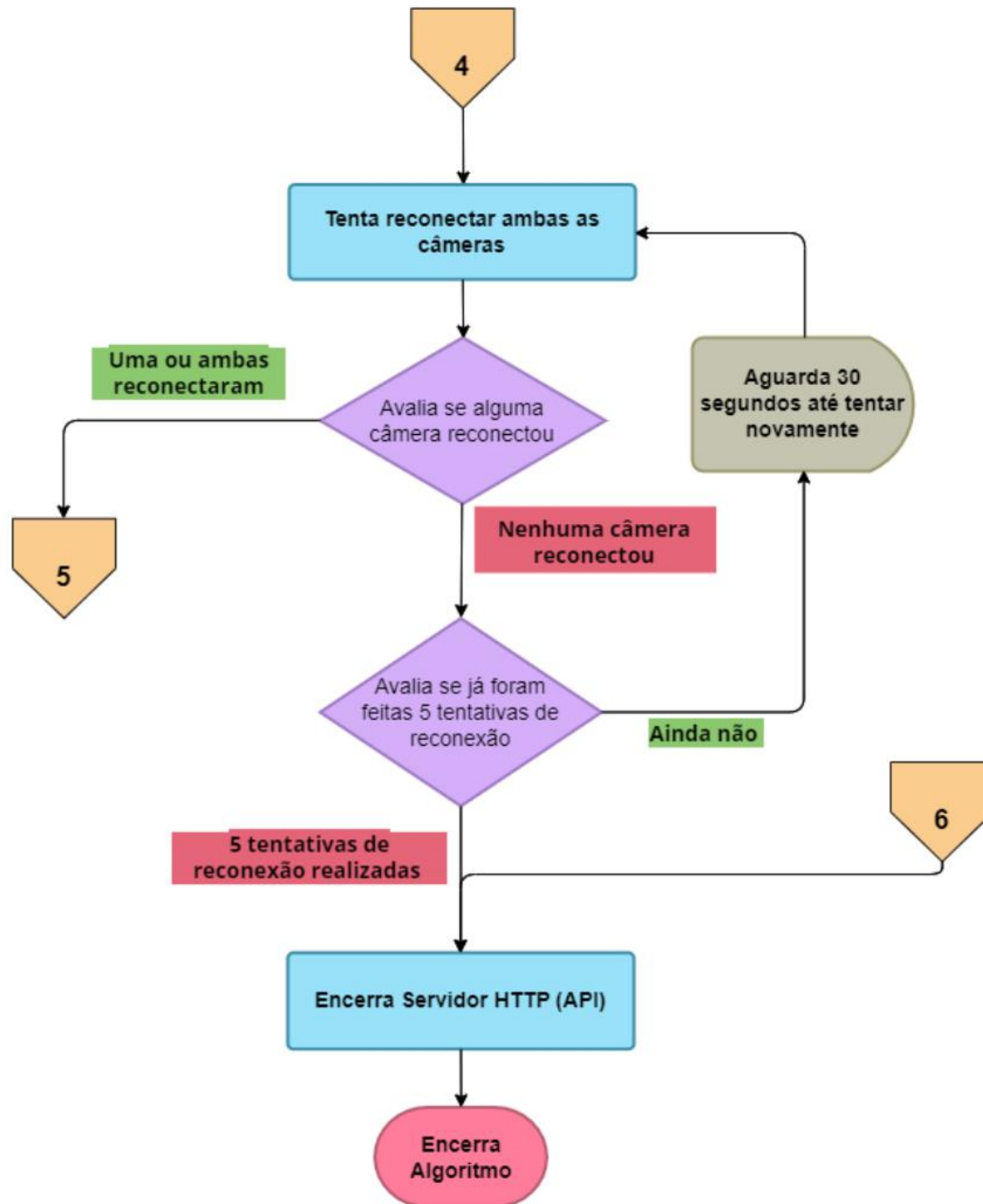
APÊNDICE E - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE DETECÇÃO



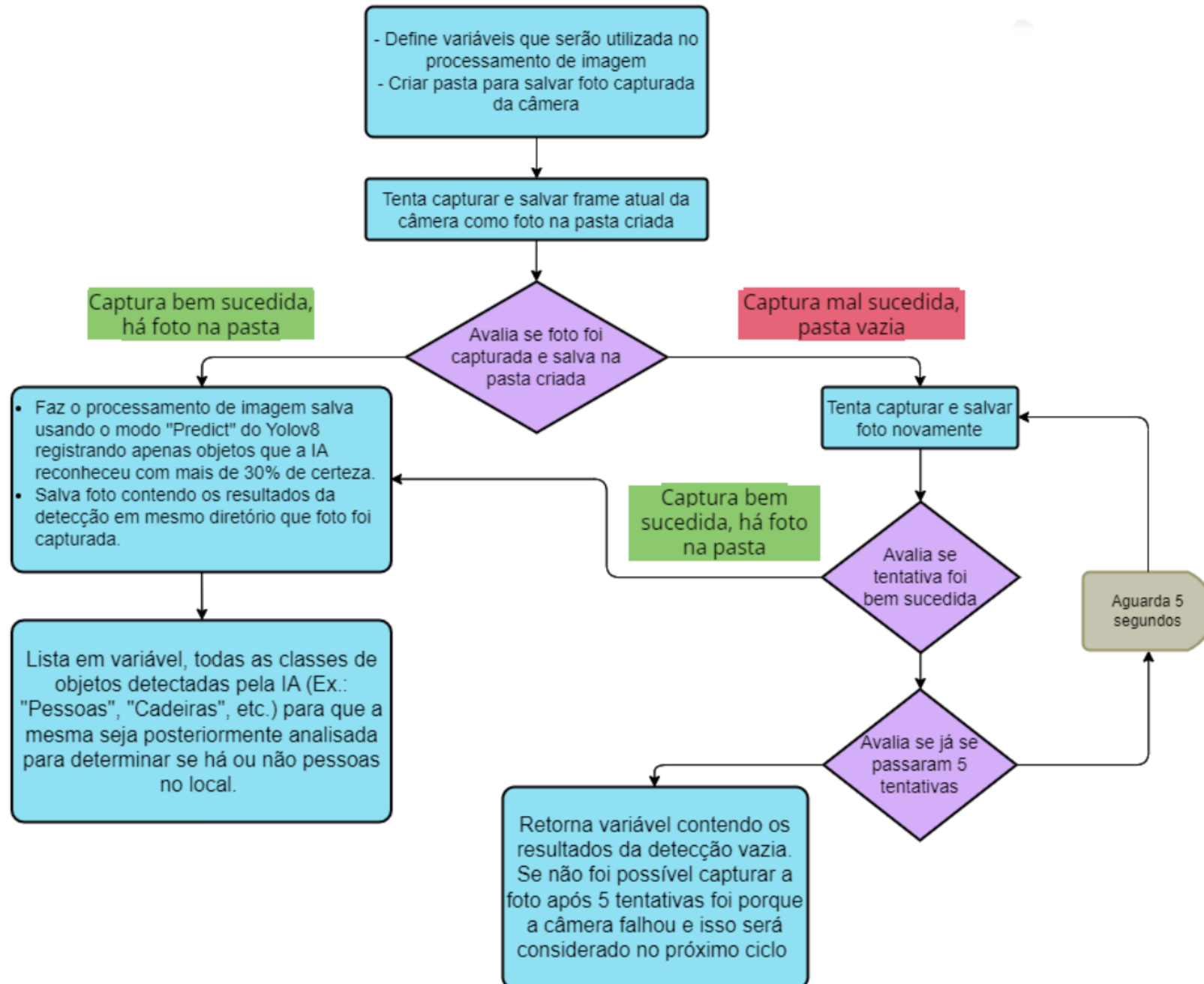
Seção 2



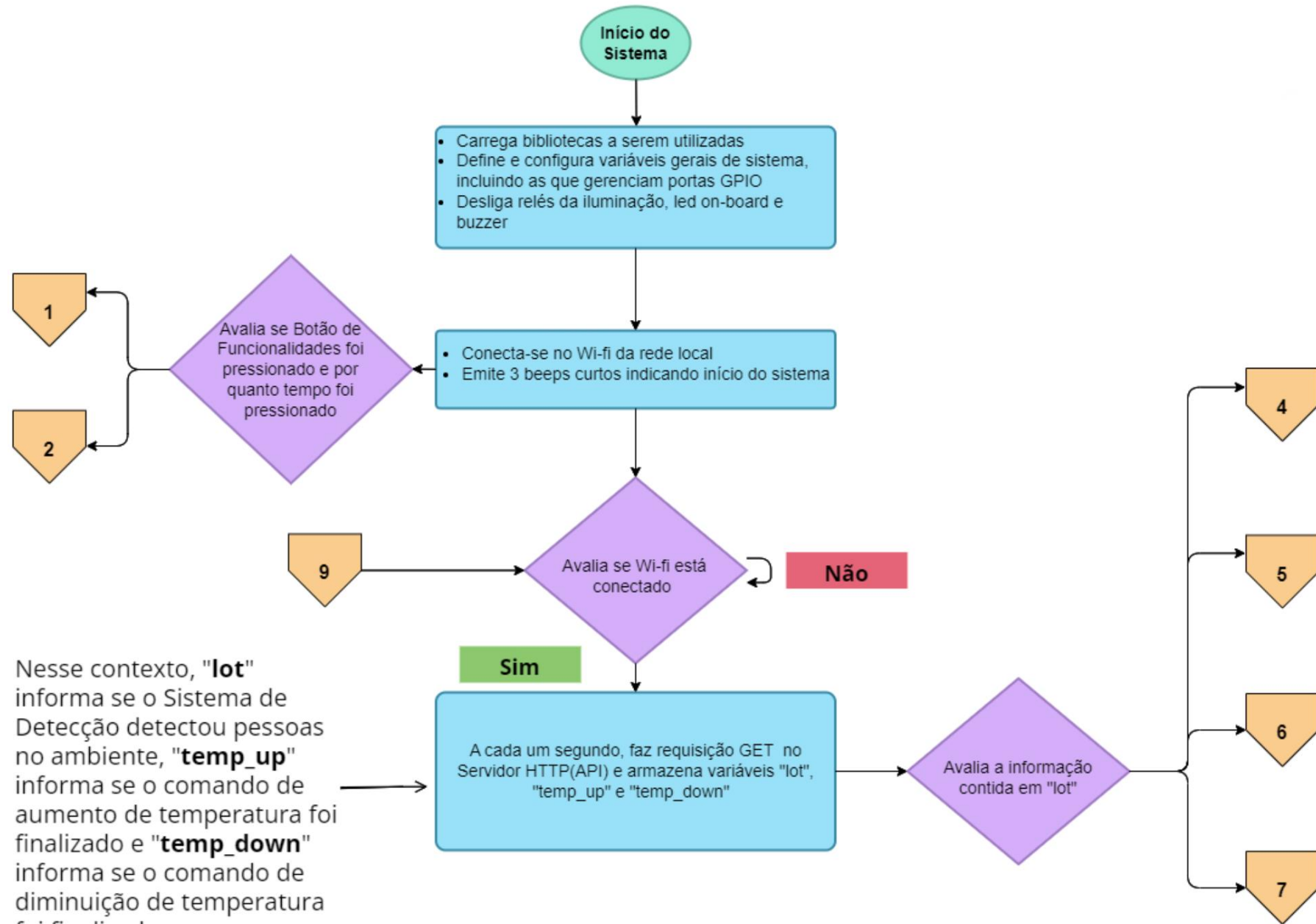
Seção 3

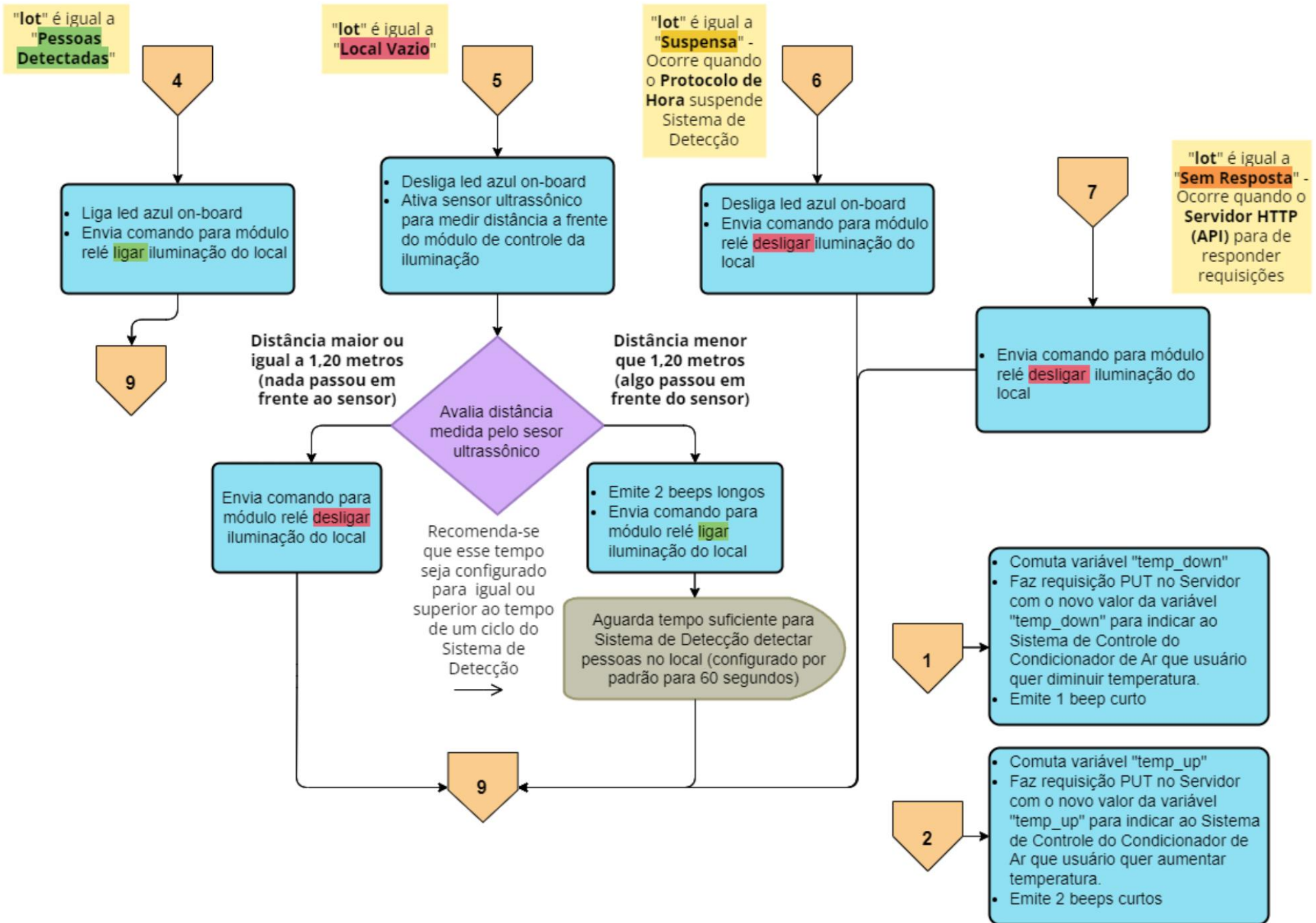


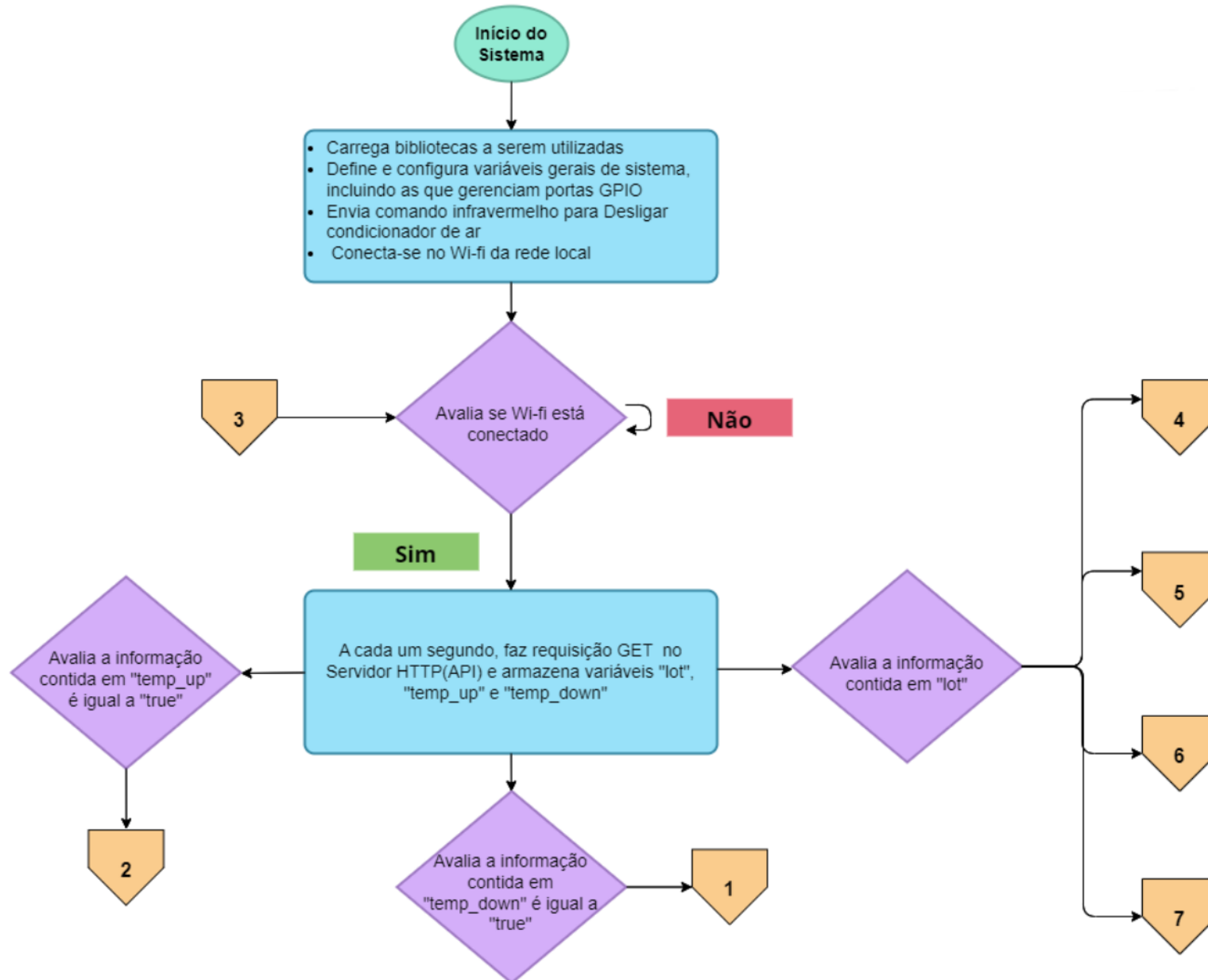
Processo de Detecção Baseado em IA (Yolov8)

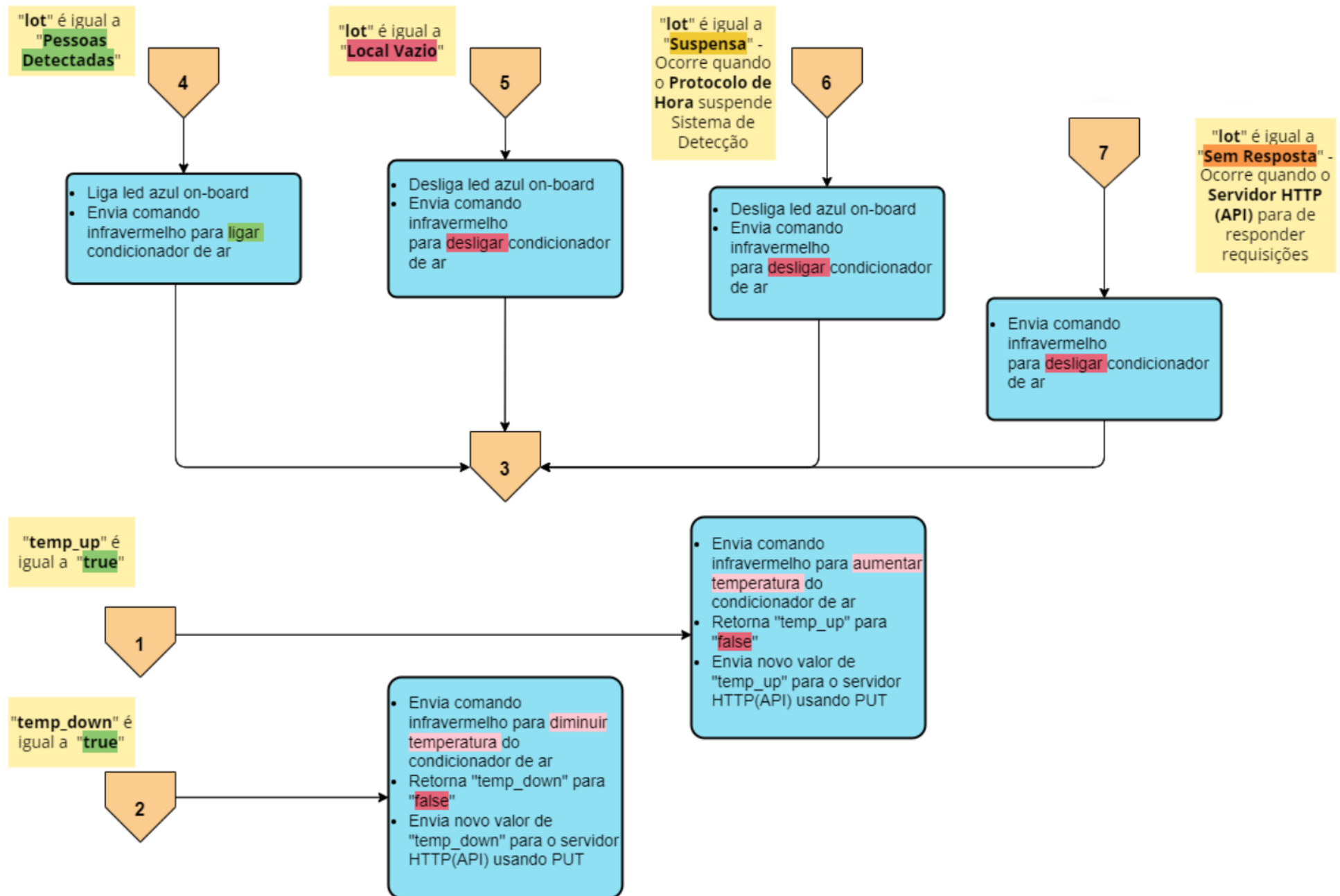


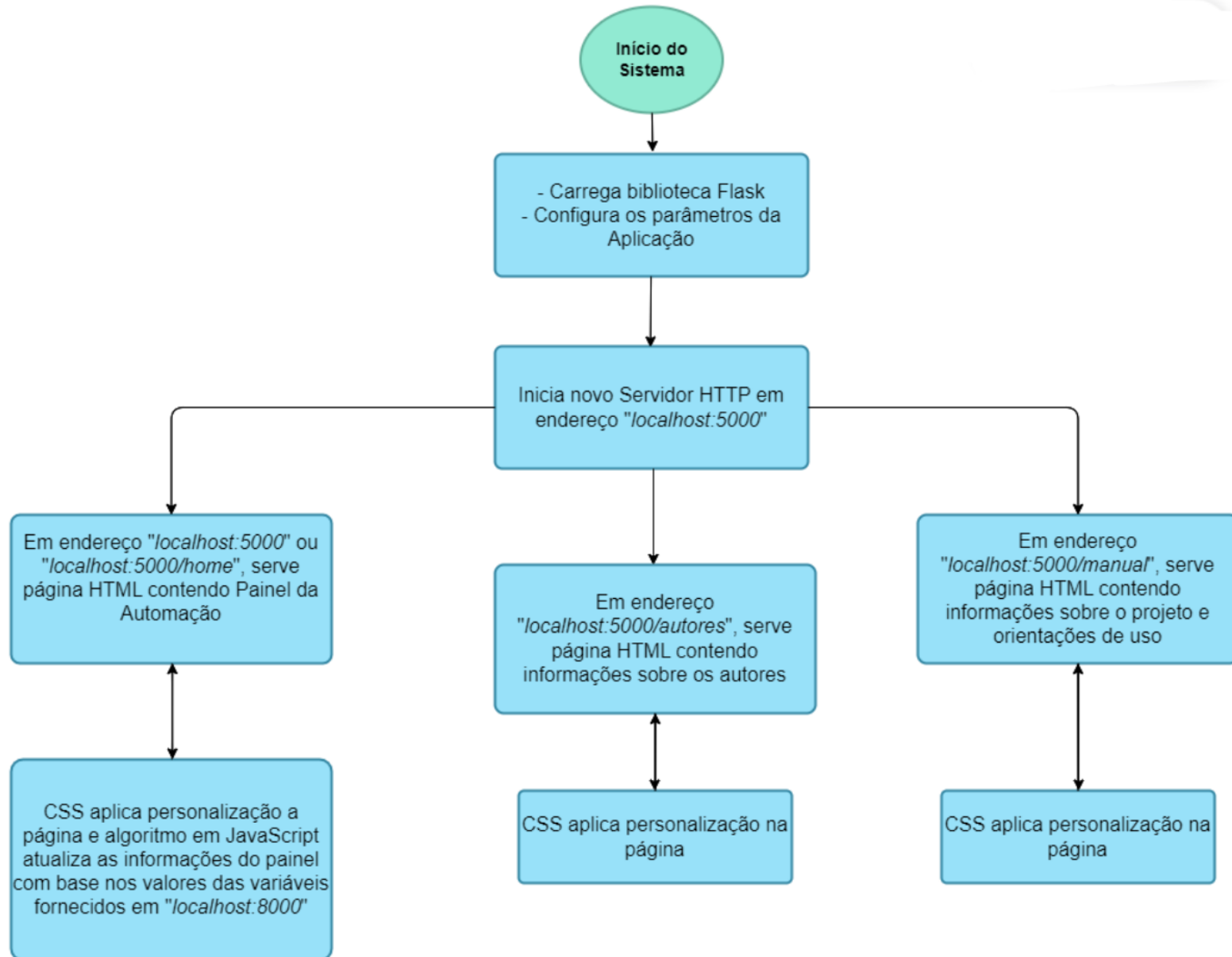
APÊNDICE F - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE CONTROLE DA ILUMINAÇÃO





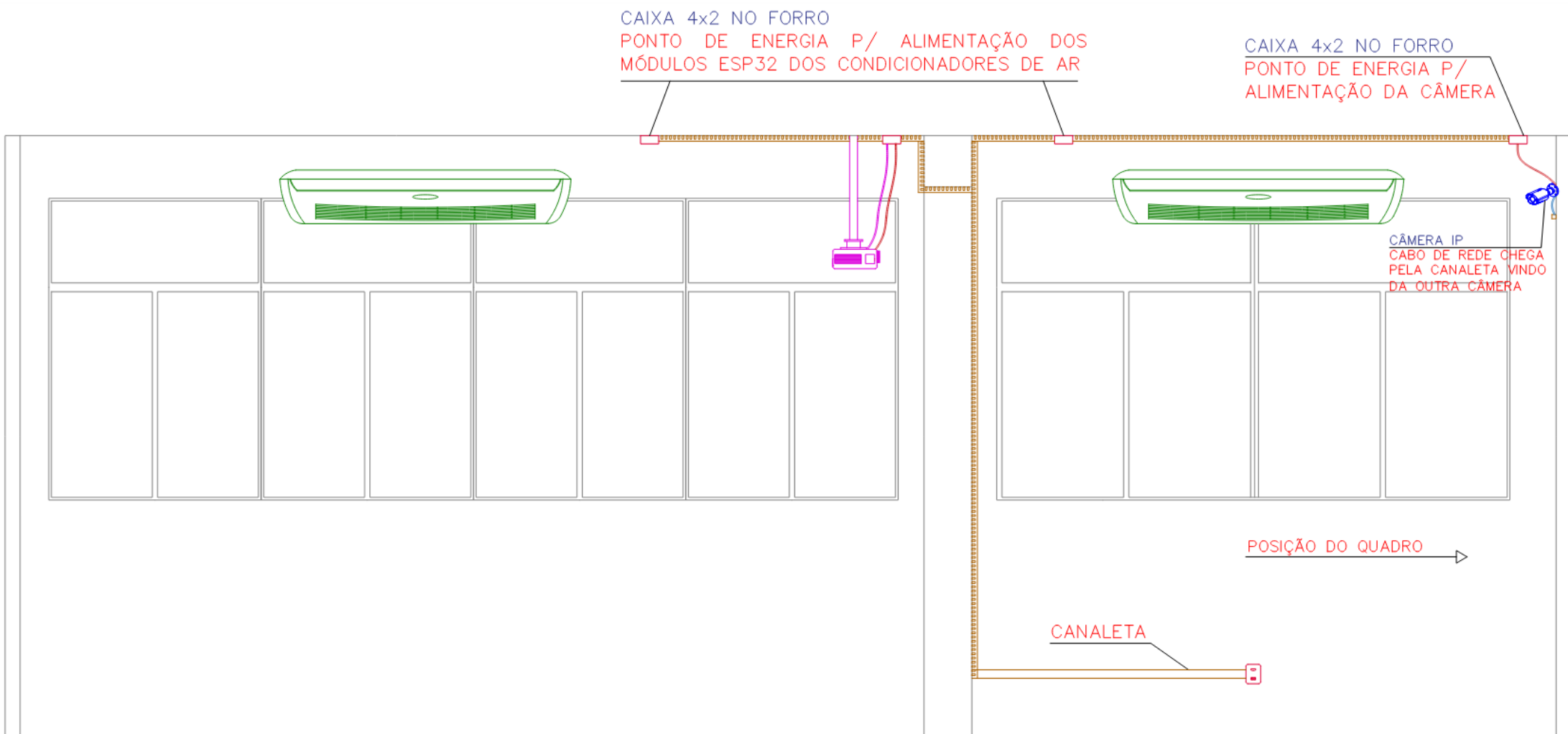
APÊNDICE G - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE CONTROLE DO CONDICIONADOR DE AR



APÊNDICE H - FLUXOGRAMA DE FUNCIONAMENTO DO SISTEMA DE ACOMPANHAMENTO

APÊNDICE I - BIBLIOTECAS DO PYTHON UTILIZADAS NO PROJETO E SUAS VERSÕES

```
blinker==1.8.1
certifi==2024.2.2
charset-normalizer==3.3.2
click==8.1.7
colorama==0.4.6
contourpy==1.2.1
cycler==0.12.1
filelock==3.14.0
Flask==3.0.3
fonttools==4.51.0
fsspec==2024.3.1
idna==3.7
intel-openmp==2021.4.0
itsdangerous==2.2.0
Jinja2==3.1.3
kiwisolver==1.4.5
MarkupSafe==2.1.5
matplotlib==3.8.4
mkl==2021.4.0
mpmath==1.3.0
networkx==3.3
numpy==1.26.4
opencv-python==4.9.0.80
packaging==24.0
pandas==2.2.2
pillow==10.3.0
psutil==5.9.8
py-cpuinfo==9.0.0
pyparsing==3.1.2
python-dateutil==2.9.0.post0
python-vlc==3.0.20123
pytz==2024.1
PyYAML==6.0.1
requests==2.31.0
scipy==1.13.0
seaborn==0.13.2
six==1.16.0
sympy==1.12
tbb==2021.12.0
thop==0.1.1.post2209072238
torch==2.3.0
torchvision==0.18.0
tqdm==4.66.2
typing_extensions==4.11.0
tzdata==2024.1
ultralalytics==8.2.6
urllib3==2.2.1
Werkzeug==3.0.2
```

APÊNDICE J - SUGESTÃO DE INSTALAÇÃO DO SISTEMA (CONDICIONADORES DE AR)

APÊNDICE K - SUGESTÃO DE INSTALAÇÃO DO SISTEMA (ILUMINAÇÃO E CENTRAL DE PROCESSAMENTO

