

**Etec EURO ALBINO DE SOUZA**  
**ENSINO MÉDIO COM HABILITAÇÃO PROFISSIONAL DE TÉCNICO**  
**EM ELETRÔNICA**

**ANA LAURA BRITO**  
**ANA EVÉLYN LIMA**  
**MURILO MARIANO**  
**VITOR HUGO PEREIRA**

***ECHOFACE: CANCELA ELETRÔNICA EQUIPADA COM***  
**SISTEMA DE RECONHECIMENTO FACIAL**

**Mogi Guaçu – SP**  
**2025**

**Etec Euro Albino de Souza**  
**Ensino Médio com Habilitação Profissional de Técnico**  
**em Eletrônica**

**ANA LAURA BRITO**  
**ANA EVÉLYN LIMA**  
**MURILO MARIANO**  
**VITOR HUGO PEREIRA**

***ECHOFACE: CANCELA ELETRÔNICA EQUIPADA COM***  
**SISTEMA DE RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso apresentado à Etec Euro Albino de Souza, do Centro Estadual de Educação Tecnológica Paula Souza, submetida à banca examinadora do curso Ensino Médio com Habilitação Profissional, como requisito para obtenção do diploma de Técnico em Eletrônica.

**Orientador:** Prof. Ms. Diogo Pedriali.

**Mogi Guaçu – SP**  
**2025**

## **Ficha elaborada pelo Bibliotecário da Instituição.**

L732e

Brito, Ana Laura; Lima, Ana Évelyn; Mariano, Murilo; Pereira, Vitor Hugo.

ECHOFACE: Cancela Eletrônica Equipada com Sistema de Reconhecimento Facial  
– Mogi Guaçu – SP. 2025.

63 p.

Trabalho de Conclusão do Curso de Ensino Técnico Integrado ao Médio com  
Habilitação Profissional em Eletrônica da ETEC Euro Albino de Souza de Mogi  
Guaçu.

Orientador: Professor Mestre Diogo Pedriali.

1. Biometria. 2. Facial. 3. Acesso. I. Pedriali, Diogo (orientador) II. Título.

CDD 621.381 5

**ETEC EURO ALBINO DE SOUZA**  
**ENSINO MÉDIO COM HABILITAÇÃO PROFISSIONAL DE TÉCNICO**  
**EM ELETRÔNICA**

**ANA LAURA BRITO**  
**ANA EVÉLYN LIMA**  
**MURILO MARIANO**  
**VITOR HUGO PEREIRA**

***ECHOFACE: CANCELA ELETRÔNICA EQUIPADA COM***  
**SISTEMA DE RECONHECIMENTO FACIAL**

Monografia aprovada por banca examinadora em 18 de novembro de 2025.

**Banca Examinadora:**

Prof. Diogo Pedriali – (Orientador)

Prof. Luis Carlos Pompeu

Prof. José Carlos Monfredini

**Mogi Guaçu – SP**  
**2025**

## **DEDICATÓRIA**

Esse trabalho é dedicado aos pais dos desenvolvedores do projeto, professores e amigos próximos.

## **AGRADECIMENTOS**

Os agradecimentos são direcionados aos professores, amigos próximos e principalmente à Deus.

"Um ser humano deve transformar informação em inteligência ou conhecimento. Tendemos a esquecer que nenhum computador jamais fará uma nova pergunta." -

Grace Hopper.

## RESUMO

Este projeto consiste na elaboração de um protótipo de cancela eletrônica com sistema de reconhecimento biométrico facial, utilizando componentes como ESP32-CAM, Servo Motor e Conversor USB. Sua principal aplicação é a otimização do fluxo de alunos nas entradas e saídas das escolas, oferecendo uma solução de alto desempenho e acessibilidade, seu intuito não seria produção em larga escala e sim o uso exclusivo como experimento na instituição de ensino cujo foi desenvolvido, aprimorando seu atual sistema de controle de acesso, oferecendo maior eficácia, segurança e agilidade. A metodologia de pesquisa e desenvolvimento seguiu de uma revisão sistemática, adotando o protocolo PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*), que foi essencial para garantir uma análise precisa e estruturada dos dados. O uso desse protocolo contribuiu para aprimorar a qualidade da pesquisa e dos resultados. Após a implementação do protótipo, os resultados obtidos demonstraram um processamento de imagem eficiente, alta segurança, banco de dados eficaz, além de funcionamento ágil e uma tecnologia de baixo custo, tornando a solução tanto funcional quanto econômica. O projeto apesar de não ter atendido a todos os objetivos propostos, atua positivamente como contribuinte para a obtenção de avanços neste âmbito, e oferece uma alternativa acessível e inovadora para o controle de acesso em ambientes escolares.

**Palavras-chave:** Controle de acesso biométrico. Reconhecimento facial. Cancela automática. ESP-cam.

## **ABSTRACT**

This project consists of developing a prototype electronic gate with a biometric facial recognition system, using components such as ESP32-CAM, servo motor, and USB converter. Its main application is to optimize student flow at school entrances and exits, offering a high-performance and accessible solution. Its intention is not large-scale production, but rather exclusive use as an experiment at the educational institution where it was developed, improving its current access control system and offering greater efficiency, security, and speed. The research and development methodology followed a systematic review, adopting the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) protocol, which was essential to ensure a precise and structured analysis of the data. The use of this protocol contributed to improving the quality of the research and the results. After the prototype's implementation, the results obtained demonstrated efficient image processing, high security, an effective database, as well as fast operation and low-cost technology, making the solution both functional and economical. Although the project did not meet all of its proposed objectives, it positively contributes to achieving progress in this area and offers an accessible and innovative alternative for access control in school environments.

**Key words:** Biometric access control. Facial recognition. Automatic gate. ESP-cam.

## LISTA DE FIGURAS

Figura 1 - Cronograma do 1º Semestre de 2025.....	25
Figura 2 - Cronograma do 2º Semestre de 2025.....	26
Figura 3 - Diagrama eletrônico do projeto .....	26
Figura 4 - Layout do diagrama. ....	27
Figura 5 - Esquemático de ligação. ....	27
Figura 6- ESP32-CAM.....	29
Figura 7 - Protótipo montado em protoboard.....	30
Figura 8 - Maquete de representação do local de sua aplicação. ....	30
Figura 9 - Instruções de instalação da biblioteca do ESP32-CAM. ....	31
Figura 10 - Continuação de instruções.....	31
Figura 11 - Interface de desenvolvimento da ESP-CAM. ....	32
Figura 12 – Função de detecção facial ativada. ....	33
Figura 13 - Reconhecimento da face cadastrada.....	33
Figura 14 - " <i>intruder alert!</i> " Face não reconhecida. ....	34
Figura 15 - Implementação da variável ao código.....	34
Figura 16 - Implementação de variável externa. ....	35
Figura 17 - Implementação da função da variável.....	35
Figura 18 - Configuração final de função da variável. ....	36
Figura 19 – Definição de ligação do Servo Motor ao micro controlador. ....	36
Figura 20 - Diagrama eletrônico de segurança .....	42

## LISTA DE TABELAS

Tabela 1 - Investimento inicial.....	17
Tabela 2 - Custos do meio comercial.....	18
Tabela 3 - Dados da Pesquisa do Periódicos Capes.....	23
Tabela 4 - Dados de Pesquisa Google Acadêmico.....	23
Tabela 5 – Materiais necessários.....	25
Tabela 6 – Custo base.....	28
Tabela 7 - Esquemático de ligação do ESP com o FT232RL.....	29

## LISTA DE ABREVIATURAS E SIGLAS

<b>ESP</b>	<i>Expressif Sistem</i>
<b>ETEC</b>	Escola técnica estadual
<b>PRISMA</b>	<i>Preferred Reporting Items for Systematic reviews and Meta-Analyses</i>
<b>IA</b>	Inteligência Artificial
<b>IoT</b>	<i>Internet of Things</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IP</b>	<i>Internet Protocol</i>
<b>USB</b>	<i>Universal Serial Bus</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
1.1	OBJETIVO GERAL .....	15
1.1.1.	OBJETIVOS ESPECÍFICOS.....	15
1.2	JUSTIFICATIVA.....	16
1.2.1	ANÁLISE FINANCEIRA .....	17
1.2.2	ANÁLISE SOCIAL.....	18
1.2.3	ANÁLISE AMBIENTAL.....	19
1.2.4	ANÁLISE DE SEGURANÇA .....	19
1.3	RELEVÂNCIA .....	20
1.4	ACESSIBILIDADE.....	20
<b>2</b>	<b>DESENVOLVIMENTO</b> .....	<b>21</b>
2.1	REFERENCIAL TEÓRICO.....	21
2.1.1	METODOLOGIA CIENTÍFICA .....	22
2.2	TÓPICOS DE FUNDAMENTAÇÃO .....	24
2.3	RECURSOS NECESSÁRIOS.....	24
2.4	CRONOGRAMA.....	25
2.5	DESENHOS E DETALHAMENTOS.....	26
2.6	CUSTOS.....	28
2.7	CONSTRUÇÃO DO PROTÓTIPO .....	28
2.8	PROTOCOLO DE TESTES .....	37
2.8.1	EXECUÇÃO DOS TESTES .....	37
2.8.2	COLETA DE DADOS DOS TESTES .....	38
2.8.3	ANÁLISE DOS DADOS DOS TESTES.....	38
2.9	MELHORIAS.....	38
2.10	ASPECTOS POSITIVOS .....	39
2.11	RISCOS .....	39
2.12	DIFERENCIAIS.....	39
2.13	RECOMENDAÇÕES.....	40
<b>3</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>41</b>
3.1	TRABALHOS FUTUROS .....	41
	<b>REFERÊNCIAS</b> .....	<b>44</b>
	<b>APÊNDICES</b> .....	<b>45</b>

# 1 INTRODUÇÃO

No contexto atual, a crescente demanda por segurança em ambientes restritos impulsionou o desenvolvimento de tecnologias voltadas ao controle de acesso e circulação de pessoas por seus espaços, tornando-se essencial o desenvolvimento de soluções modernas, eficazes e confiáveis para o controle de entrada e saída de pessoas.

Nesse cenário, sistemas eletrônicos de controle de acesso têm ganhado destaque especialmente aqueles que utilizam métodos biométricos, como o reconhecimento facial e íris, por sua precisão e rapidez no processo de identificação. Sendo inclusivo para qualquer indivíduo, independentemente de suas limitações.

Este trabalho apresenta o desenvolvimento de um sistema eletrônico de controle de acesso, utilizando a cancela eletrônica integrada a um sistema de reconhecimento biométrico facial fornecido através do ESP32-CAM. A proposta do projeto tem como principal objetivo reforçar a segurança em ambientes que exigem acesso restrito, ao mesmo tempo em que agiliza o processo de identificação dos usuários e facilita a inclusão de pessoas de forma prática e confiável. A escolha da biometria facial como método de autenticação busca não apenas garantir maior precisão e confiabilidade, mas também oferecer uma experiência mais eficiente, segura e alinhada às exigências tecnológicas atuais de ambientes corporativos, educacionais e industriais.

A proposta simboliza a resposta imediata e inteligente do sistema ao reconhecimento facial, como se o próprio rosto provocasse uma reação automatizada. Além disso, a sonoridade moderna do nome “EchoFace” reforça o caráter tecnológico e inovador do projeto, alinhando-se ao objetivo de oferecer uma solução de controle de acesso segura e eficaz.

O atual trabalho apresenta-se dividido em três principais capítulos, estes sendo: Introdução, expõe previamente o projeto, a partir de uma breve explicação da elaboração do protótipo; Desenvolvimento, descreve detalhadamente o processo de construção e execução; Considerações Finais, que por fim encerra este trabalho expondo uma retomada dos principais resultados e reflexões, avalia se os objetivos foram alcançados e sugestões de melhorias ou estudos futuros. De forma clara e detalhada, todas as etapas envolvidas no desenvolvimento do projeto desde a sua

concepção inicial da ideia até a implementação do sistema final. São abordados os seus fundamentos teóricos, as tecnologias utilizadas, os testes realizados e os resultados alcançados. Mais do que mostrar como o sistema foi construído, o objetivo é evidenciar como o reconhecimento facial pode ser aplicado de maneira eficaz em sistemas de controle de acesso automatizados, oferecendo mais segurança, agilidade e modernidade aos ambientes que necessitam de controle rigoroso de entrada e saída de pessoas.

## **1.1 OBJETIVO GERAL**

O desenvolvimento do controle de acesso automatizado por meio da cancela eletrônica equipada a um sistema de reconhecimento biométrico facial, foi desenvolvido com o intuito de favorecer a segurança de ambientes restritos de uma maneira mais rápida e precisa, incluindo-a na instituição escolar dos desenvolvedores do projeto. O projeto busca com *software* e *hardware* integrados trazer modernidade e inclusão aos espaços onde o projeto for aplicado. Além de contribuir no combate a fraudes e gerar melhorias no tráfego e no movimento de entrada e saída em ambientes que demandem o controle rigoroso de circulação de pessoas e a disponibilização de tecnologias por um baixo-custo, um tópico fortemente considerado na elaboração deste projeto e assim executar com perfeição seu funcionamento.

### **1.1.1.OBJETIVOS ESPECÍFICOS**

O atual projeto foi desenvolvido com o objetivo de realizar uma operação satisfatória e funcional, almejando em primeiro lugar uma pesquisa detalhada para assim haver o conhecimento necessário para tal. Consequente a isso, o funcionamento adequado dos componentes eletrônicos é de extrema necessidade para uma programação que converse corretamente com o protótipo. Isso acontece através de uma programação base proporcionada pelo próprio IDE (*Integrated Development Environment*) que atua sobre o ESP32. Anseia-se que com isso o projeto opere de maneira funcional, promovendo uma execução apropriada ao que é designado pela programação.

Ao utilizar o reconhecimento biométrico facial, a ideia não é apenas modernizar os espaços, mas também garantir que qualquer pessoa tenha uma interação rápida e

descomplicada ao acessar ambientes restritos, mais do que segurança, existe o intuito de criar uma solução que traga mais autonomia para todos, permitindo que a tecnologia se torne uma aliada, facilitando o acesso e melhorando a experiência de quem circula por esses espaços.

## **1.2 JUSTIFICATIVA**

Atualmente, o controle de acesso da instituição de ensino dos desenvolvedores desse projeto é realizado de forma manual, onde a equipe de segurança precisa verificar, um a um, os cartões de identidade estudantil de todos os alunos nas entradas e saídas. Esse processo, além de implicar em tempo, está passível a falhas humanas, como a entrada de pessoas não autorizadas no campus ou mesmo a dificuldade em reconhecer rapidamente os alunos que estejam sem sua identificação seja por esquecimento ou perda, visto que a instituição de ensino enfrenta uma demanda de mais de 300 estudantes por dia. No contexto deste colégio que conta com a disponibilidade de muitos recursos tecnológicos, um sistema de câmeras de segurança, ambientes de computação e uma rede de internet estável.

Após a nova implementação de controle de acesso houve um aumento na preocupação de alunos, visto que erros acontecem. Diante desse cenário, a elaboração de uma cancela com reconhecimento facial surge como uma solução inovadora e eficiente para otimizar não só o tempo mais também o fluxo de pessoas e fortalecer a segurança no ambiente escolar, atuando na redução de tempo gasto pela equipe de segurança na certificação manual, devido a maior precisão do sistema automatizado na identificação, também impede que o uso inadequado dos cartões de identidade estudantil ou falsificações ocorram. Isso proporciona um ambiente mais seguro para alunos, professores e demais funcionários, além de permitir que o setor de segurança possa direcionar seus trabalhos para outras atividades mais estratégicas dentro da escola.

Portanto, o desenvolvimento deste projeto justifica-se pela necessidade de aprimorar o sistema de controle de acesso da instituição, oferecendo mais agilidade, segurança e confiabilidade.

### 1.2.1 ANÁLISE FINANCEIRA

Atualmente, o mercado apresenta grande diferença quando se trata de uma cancela acoplada a um sistema de reconhecimento facial de alta qualidade, muitas vezes tornando-se menos acessíveis. Um dos pontos fundamentais para o desenvolvimento do protótipo é justamente a questão financeira, onde o baixo custo se torna um ponto crucial. Pode-se observar o valor investido a seguir:

Tabela 1 - Investimento inicial.

<b>INVESTIMENTO INICIAL</b>			
<b>COMPONENTE</b>	<b>QUANTIDADE</b>	<b>VALOR</b>	<b>VALOR FINAL (TOTA)</b>
Kit Esp32-Cam C/ Conversor USB 4 /FTDI	1	R\$89,99	R\$89,99
Servo motor SG90	1	R\$25,00	R\$25,00
Bateria recarregável 18650 3,7V	2	R\$15,00	R\$30,00
Carregador Bateria 18650	1	R\$25,00	R\$25,00
Suporte duplo Bateria 18650	1	R\$12,00	R\$12,00
Regulador de Tensão LM7805	1	R\$4,50	R\$4,50

Fonte: Autoria própria

Posteriormente à descoberta do valor final investido foi executado uma pesquisa detalhada, afim de melhor comparar o protótipo as opções existentes no mercado comercial. Na sequência é observado o resultado apurado:

Tabela 2 - Custos do meio comercial.

<b>DO MEIO CUSTOS COMERCIAL</b>				
<b>CANCELA</b>	<b>ALUGUEL / COMPRA</b>	<b>FRETE</b>	<b>INSTALAÇÃO</b>	<b>VALOR FINAL (TOTAL)</b>
Cancela de Controle de Acesso <i>Next</i> – (Loja do ponto)	Aluguel (48 meses) R\$799,00	R\$278,26	Varia de R\$150,00 a R\$500,00	R\$1.227,26
Cancela Facial Intelbras – ( <i>Ponto System</i> )	Compra R\$4.390,00	R\$304,53	Varia de R\$150,00 a R\$500,00	R\$4.844,53
Cancela Facial IDBlock – (BY Ponto)	Compra R\$8.287,50	R\$873,04	Varia de R\$150,00 a R\$500,00	R\$9.310,54

Fonte: Autoria própria

O protótipo é financeiramente vantajoso, principalmente para projetos acadêmicos devido ao baixo-custo, embora atenda aos objetivos propostos, ele não possui as funcionalidades e acabamentos de um produto comercializado em lojas.

### 1.2.2 ANÁLISE SOCIAL

O projeto busca trazer benefícios à sociedade como um todo, reconhecendo as adversidades e individualidades de cada pessoa. Com isso, almeja-se atender de maneira mútua e inclusiva a todos, sem excluir aqueles que, por qualquer motivo, enfrentam limitações físicas, cognitivas ou sociais. Quesitos que podem ser encarados por alguns como limitações ou obstáculos, como dificuldades de locomoção, baixa visão e analfabetismo são considerados no desenvolvimento do sistema.

A escolha pelo reconhecimento biométrico facial se dá justamente por ser uma tecnologia não invasiva, higiênica (por não exigir contato físico), e altamente precisa, o que possibilita seu uso por pessoas de diferentes condições, garantindo um acesso mais justo, seguro e autônomo.

Nesse sentido, o projeto vai além da proposta técnica, ele se posiciona como uma ferramenta de inclusão social e tecnológica, contribuindo para a construção de espaços mais acessíveis, modernos e comprometidos com a dignidade e a segurança de todos.

### **1.2.3 ANÁLISE AMBIENTAL**

O projeto visa modernizar o sistema atual oferecendo um ambiente mais seguro e confiável, contudo ainda está sujeito a resistência pela comunidade escolar (responsáveis, alunos e equipe escolar) devido à falta de conhecimento desta tecnologia.

Apesar de ser uma tecnologia pouco conhecida e ainda pouco utilizada na região, ela seria fundamental para maior segurança tanto dos alunos quanto do meio ambiente, visto que substituem o controle de acesso tradicional utilizado, evitando o descarte de materiais como, papéis, cartões e quaisquer dispositivos ou objetos descartáveis, reduzindo significativamente a geração de resíduos, reforçando também a ideia de um baixo impacto ambiental. O projeto proporciona um bom desempenho, sendo simples, acessível, e se encaixando em diversas áreas, quando se trata de controle de acesso.

Apesar do protótipo ser composto por materiais eletrônicos, é possível realizar o descarte adequado de seus componentes, minimizando os impactos ambientais significativamente. Para isso, o descarte deve ser realizado em pontos de coleta específicos, como centros de reciclagem ou estabelecimentos que recebam equipamentos eletrônicos para destinação correta. Como seu meio de aplicação será atualmente na escola ETEC Euro Albino de Souza o descarte não se torna um obstáculo, considerando que a instituição já oferece pontos de coleta.

### **1.2.4 ANÁLISE DE SEGURANÇA**

Apesar de ser um projeto que proporciona uma melhor qualidade de segurança aos alunos em comparação ao atual sistema de controle de acesso, durante a sua execução foram observadas algumas ameaças à segurança, tais como, possível vazamento de dados, erros no momento de cadastro ou falhas ao realizar o reconhecimento.

Com isso é de extrema necessidade a adoção de algumas medidas, para melhor execução do projeto, como por exemplo aprimoramento nas camadas proteção de dados, estes que serão coletados e armazenados com controle de permissões, em caso de erro no cadastro, será possível o recadastramento do aluno através de autenticação por intermédio exclusivo pela secretaria escolar, no entanto

falhas ao realizar o reconhecimento podem acontecer por diversos fatores, ângulo inadequado, iluminação insuficiente, proximidade incorreta em relação ao ESP32-CAM ou diversidade de expressões. Com isso o projeto busca aumentar o nível de segurança da instituição, oferecendo uma solução eficiente.

### **1.3 RELEVÂNCIA**

A relevância desse projeto, está relacionada com a necessidade do aprimoramento do sistema de segurança da escola. De acordo com as vivências do grupo, ao longo dos últimos anos, observa-se que grande parte das instituições de ensino em Mogi Guaçu ainda realizam o controle de acesso de forma manual, sendo ele, na maioria das vezes, executado por vigilantes, tendo grandes possibilidades de falha na identificação, além de gerar atrasos e riscos na segurança.

Em razão disso, conclui-se que esse projeto tem a finalidade de solucionar tais problemas com a implementação de uma cancela com reconhecimento facial, disponibilizando uma alternativa mais rápida, tecnológica, segura e eficaz. Além de ter o potencial de contribuir para a inovação tecnológica em ambientes escolares e tendo a capacidade de ser visto como referência para outras instituições para que adotem o mesmo método.

### **1.4 ACESSIBILIDADE**

O projeto propõe o desenvolvimento de um sistema eletrônico de controle de acesso que valoriza a acessibilidade e a inclusão. A escolha pelo reconhecimento biométrico facial foi motivada por suas características técnicas e únicas como alta precisão, rapidez e ausência ou pouca necessidade de contato físico o que tornam essa tecnologia especialmente adequada para atender pessoas com limitações físicas, sensoriais ou cognitivas. Diferente de sistemas tradicionais que exigem toques, digitação ou leitura manual de cartões, a solução apresentada permite um acesso mais autônomo, confortável e seguro. Ao considerar adversidades como dificuldades de locomoção, baixa estatura, visão reduzida e analfabetismo funcional, o projeto busca eliminar barreiras e garantir igualdade de participação para todos os usuários. Dessa forma, mais do que oferecer um recurso tecnológico avançado, o sistema contribui para ambientes mais humanos, inclusivos e socialmente comprometidos.

## 2 DESENVOLVIMENTO

O desenvolvimento do projeto *EchoFace* consiste em uma cancela automatizada acoplada ao reconhecimento biométrico facial, tendo como objetivo de modo geral alcançar a praticidade. O sistema terá uma função básica de realizar o reconhecimento de alunos e funcionários, concedendo ou negando a passagem. O termo “*echo*” remete a algo que ressoa ou responde rapidamente, simbolizando a resposta do sistema de reconhecimento facial, enquanto o termo “*face*” está intimamente relacionado ao reconhecimento facial e à biometria, sugerindo um sistema que identifica as pessoas com base em suas características faciais. Seu nome foi escolhido justamente por remeter à ideia de um “eco digital” gerado a partir da leitura da face do usuário.

O atual capítulo é destinado a demonstrar em diversos tópicos toda a produção do protótipo expondo sua estrutura e construção eletrônica e teórica, abordando seus principais conceitos como métodos de pesquisas, análises e resultados obtidos. O desenvolvimento é a chave fundamental para a execução satisfatória do projeto como um todo, contendo os fundamentos que sustentam o tema, etapas de efetivação e montagem, que conduzem à compreensão.

A princípio, é apresentado um referencial teórico com base em diversos autores criteriosamente selecionados após uma pesquisa minuciosa, na qual foi utilizada para a definição de diversos artigos que agregam ao protótipo, posteriormente é colocado em pauta o método de pesquisa utilizado, determinando palavras chaves principais para a revisão bibliográfica nas bases científicas Periódico Capes e Google Acadêmico, seguindo como base o protocolo PRISMA.

Por fim, é abordado a construção do protótipo, onde é demonstrado passo a passo do desenvolvimento, contando com testes e suas análises para considerar possíveis melhorias e possíveis riscos.

### 2.1 REFERENCIAL TEÓRICO

O termo biometria, do grego *bios* (vida) e *metron* (medida), pode ser definida como ramo da ciência que estuda a identificação de aspectos físicos, biológicos e até comportamentais dos seres vivos. Na qual, são utilizados para distinguir indivíduos, a

partir de suas características únicas (FERREIRA, apud SANTOS JUNIOR, 2023, p.13).

A biometria foi transformada em uma nova área de estudo pelo antropologista francês Alphonse Bertillon, em 1890, quando a utilizou para a identificação de criminosos (MORAES, apud GALIMBERTI, 2018, p.15). Atualmente este estudo é voltado a diversas áreas, como, controle de acesso. Com o avanço das técnicas de visão computacional e o aumento da capacidade de processamento, sistemas de controle de acesso baseados em biometria facial têm se tornado cada vez mais viáveis e precisos (LEMOS, 2024).

Segundo Silva; Rizzetti (2024, p.1) “Sistemas de controle de acesso são fundamentais para proteger áreas restritas. [...] O reconhecimento facial, explora características faciais únicas, que são difíceis de replicar ou falsificar.”

Apesar da elevada precisão e do caráter avançado das técnicas empregadas, o reconhecimento facial apresenta vulnerabilidades específicas.

Mudanças na iluminação e ângulos de face diferentes podem afetar significativamente a eficácia dos algoritmos de reconhecimento facial. A habilidade de um sistema para reconhecer com precisão um rosto em condições variadas de iluminação ou em diferentes poses continua a ser um problema crítico. (Rocha; Rocha, 2024, p.16)

### **2.1.1 METODOLOGIA CIENTÍFICA**

O método de pesquisa e desenvolvimento utilizado para a realização do projeto como uma revisão sistemática tem como base o protocolo PRISMA (*Preferred Reporting Items for Systematic reviews and Meta-Analyses*), o mesmo contribui na coletânea e seleção de informações, o que conseqüentemente favorece os resultados e a metodologia de pesquisa usada para estudos acadêmicos. As palavras chaves adotadas para a realização da pesquisa e para servirem como bases principais da fundamentação deste projeto foram: Controle de acesso biométrico; reconhecimento facial; cancela automática; ESPCAM. Também traduzidas e pesquisadas em inglês para maior quantidade de resultados, como: *Biometric access control; facial recognition; automatic cancellation; ESPCAM*.

A princípio o procedimento adotado para a então realização dos testes foi inicialmente constituído em pesquisas de diversas bases científicas, fundamentais

para a ampliação do conhecimento de dados gerais neste campo, sendo por sua parcela predominante com data a partir de 2020. Desse modo ao decorrer do projeto são abordados amplamente os elementos adquiridos através do estudo realizado. Seguindo a revisão a seguir, onde demonstra-se na primeira tabela a análise no Periódicos Capes e na segunda no Google Acadêmico:

Tabela 3 - Dados da Pesquisa do Periódicos Capes.

<b>PERIÓDICO CAPES</b>		
<b>PALAVRAS CHAVE</b>	<b>IDIOMA</b>	<b>Nº DE ARTIGOS ÚTEIS</b>
Reconhecimento Facial	Inglês/Português	13
Controle de acesso biométrico	Inglês/Português	0
Cancela Automática	Inglês/Português	0
ESPCAM	Inglês/Português	0

Fonte: Autoria própria

Tabela 4 - Dados de Pesquisa Google Acadêmico.

<b>GOOGLE ACADÊMICO</b>		
<b>PALAVRAS CHAVE</b>	<b>IDIOMA</b>	<b>Nº DE ARTIGOS ÚTEIS</b>
Reconhecimento Facial	Inglês/Português	1
Controle de Acesso Biométrico	Inglês/Português	4
Cancela Automática	Inglês/Português	1
ESP32-CAM	Inglês/Português	2

Fonte: Autoria própria

O Procedimento adotado pelos responsáveis para o desenvolvimento experimental do sistema, consiste em uso de tecnologias de informação específicas seguido de testes com voluntários para a confirmação de funcionamento do projeto. Para a coleta de dados específicos, registrando as tentativas de acesso e analisar a taxa de reconhecimento e o tempo de resposta do mesmo.

## 2.2 TÓPICOS DE FUNDAMENTAÇÃO

A popularização da biometria fundamenta-se na sua capacidade de ampliar os níveis de segurança, seja em transações eleitorais e financeiras, no controle de acesso a dispositivos e sistemas, ou ainda como meio de verificação de identidade e de não repúdio. Ademais, as tecnologias biométricas, quando associadas ao uso de câmeras, sistemas de monitoramento e técnicas de inteligência artificial, têm impulsionado o avanço do reconhecimento facial, que, por sua vez, pode ser empregado na autenticação de identidade em diferentes contextos, como no controle de fronteiras, de frequência escolar, entre outros. Em termos teóricos, os resultados decorrentes do uso dessa ferramenta mostram-se promissores; contudo, seus desafios na aplicação prática são significativos, uma vez que os dados biométricos são sensíveis e abrangem grandes parcelas da população, incluindo grupos em situação de vulnerabilidade (CEBRIAN et al., 2024).

Segundo LEMOS et al. (2024, p.3): “A tecnologia de reconhecimento facial oferece uma camada adicional de segurança em comparação com métodos tradicionais de autenticação. As técnicas de visão computacional permitem o desenvolvimento de sistemas adaptáveis e robustos, capazes de se ajustar às necessidades específicas de cada ambiente”. Tais informações tiveram o papel de servir ao planejamento de funcionamento e utilização do projeto, auxiliando na definição do local para sua aplicação.

SADEQ, S.; ZIDAN, K. A.; STEPHAN, J. J. (2021, p.5106-5107) afirmam que: “A palavra “biométrico” refere-se à autenticação e identificação de uma identidade única, baseada nas propriedades ou características distintivas das pessoas [1]. O sistema biométrico inclui características comportamentais e fisiológicas. As características comportamentais são um conjunto de dados biométricos que lidam com propriedades não fisiológicas ou não biológicas, organizadas por meio de um sistema biométrico. Elas compreendem quatro grupos principais: assinatura, reconhecimento de digitação (teclado), modo de andar e voz.”

## 2.3 RECURSOS NECESSÁRIOS

Para a completa execução do atual protótipo é de extrema necessidade alguns componentes eletrônicos, os quais são indicados a seguir:

Tabela 5 – Materiais necessários.

COMPONENTE	QUANTIDADE
Kit ESP32-CAM C/ Conversor USB 4 /FTDI	1
Servo Motor SG90	1
Bateria 3,7V	2
Suporte Bateria 18650	1
Regulador de tensão LM7805	1

Fonte: Autoria própria

O ESP32-CAM é crucial para a captura de face e o reconhecimento, sendo o único microcontrolador presente no protótipo e o Conversor USB é responsável pelo canal de transferência de dados para o ESP32-CAM.

## 2.4 CRONOGRAMA

Este campo aborda o planejamento de desenvolvimento deste protótipo, baseado em um cronograma gerenciado pelo professor orientador e moldado pelo grupo, sendo dividido por meses e datas importantes em sua execução, como compra de componentes, prazos, revisões e entregas.

É apresentado a seguir o cronograma referente ao primeiro semestre de 2025.

Figura 1 - Cronograma do 1º Semestre de 2025

1º SEMESTRE DE 2025			
MARÇO	ABRIL	MAIO	JUNHO
12/03 - INÍCIO DA ESCRITA DA MONOGRAFIA.	02/04 - REALIZAÇÃO DE BIBLIOMETRIA.	09/05 - CONCLUSÃO DA COMPRA DE TODOS OS COMPONENTES	09/06 - PRAZO PARA CONCLUSÃO DE DADOS DA FICHA CATALOGRÁFICA.
19/03 - ABORDAGEM SOBRE METODOLOGIA CIENTÍFICA.	23/04 - REVISÃO SOBRE CITAÇÕES BIBLIOGRÁFICAS.		11/06 - PRÉ-APRESENTAÇÃO DO DESENVOLVIMENTO DO PROJETO.
26/03 - REALIZAÇÃO DE REVISÃO BIBLIOGRÁFICA.			18/06 - ENTREGA DA (V) MONOGRAFIA EM PROCESSO DE DESENVOLVIMENTO, E OS (VI) DADOS PARA A CRIAÇÃO DA FICHA CATALOGRÁFICA.

Fonte: Autoria própria

Adiante é evidenciado a segunda parte do cronograma a respeito do segundo semestre de 2025, onde a data em destaque se refere a apresentação final do TCC para a banca avaliativa.

Figura 2 - Cronograma do 2ºSemestre de 2025

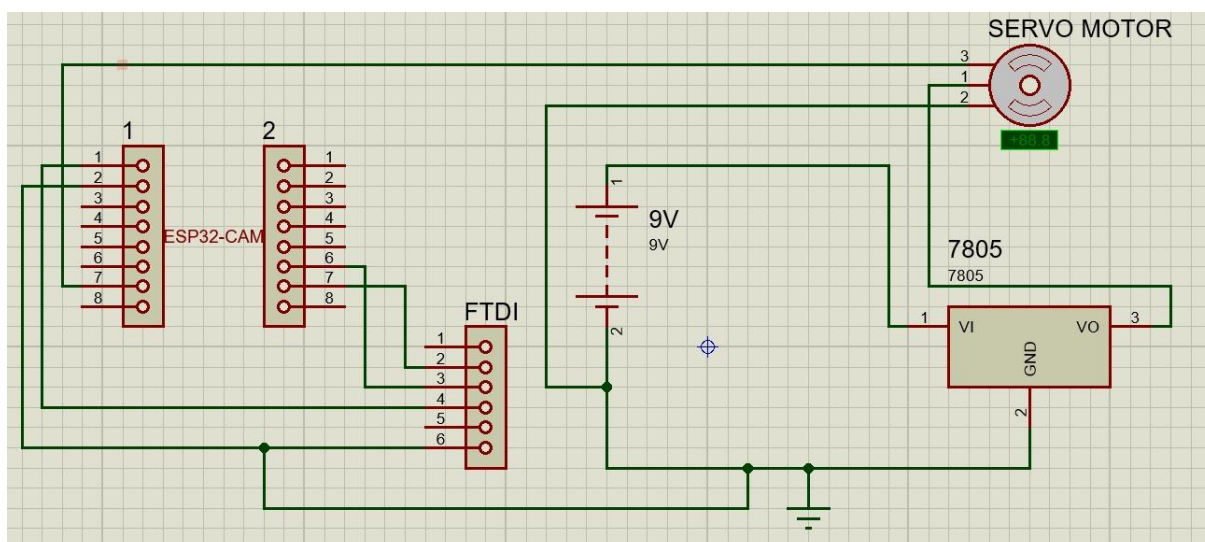
2º SEMESTRE DE 2025			
AGOSTO	SETEMBRO	NOVEMBRO	DEZEMBRO
06/08 - DESENVOLVIMENTO DO PROTÓTIPO, DA MONOGRAFIA E DO PÔSTER.	08/09 - CONCLUSÃO DE MONOGRAFIA	05/11 - ENTREGA DE 2 CÓPIAS IMPRESSAS DA MONOGRAFIA PARA OS COMPONENTES DA BANCA.	10/12 - ENTREGA DE MÍDIA REMOVÍVEL CONTENDO TODOS OS ARQUIVOS
	17/09 - ENTREGA DA MONOGRAFIA REVISADA E FINALIZADA.	12/11 - TREINO DE PRÉ-APRESENTAÇÃO DO PROJETO PARA A BANCA EXAMINADORA.	
	24/09 - ENTREGA DO PÔSTER REVISADO E FINALIZADO (VIA E-MAIL)	18/11 - APRESENTAÇÃO FINAL DO DTCC PARA A BANCA DE AVALIAÇÃO.	

Fonte: Autoria própria

## 2.5 DESENHOS E DETALHAMENTOS

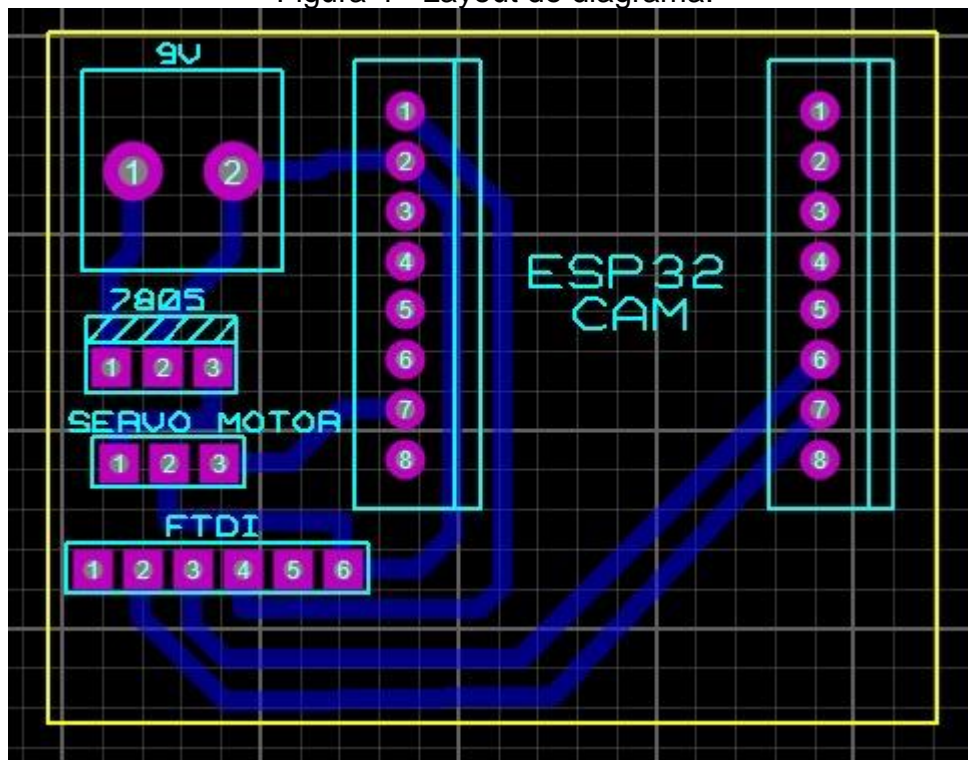
A Figura 3 ilustra o diagrama eletrônico do protótipo.

Figura 3 - Diagrama eletrônico do projeto



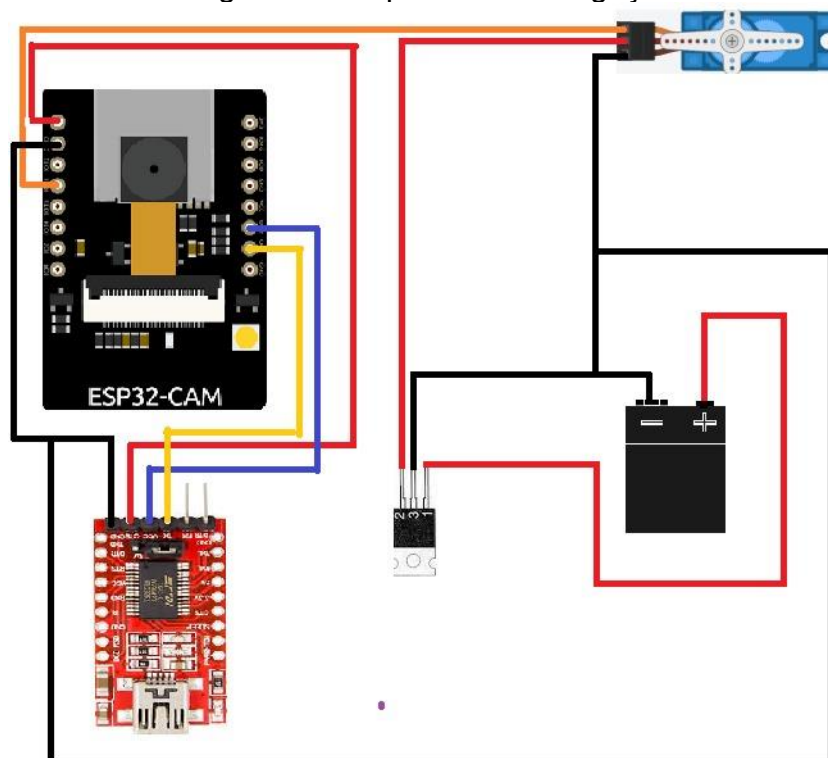
Fonte: Autoria própria

Figura 4 - Layout do diagrama.



Fonte: Autoria própria

Figura 5 - Esquemático de ligação.



Fonte: Autoria própria

## 2.6 CUSTOS

Para assegurar um bom desempenho e o funcionamento adequado do protótipo, torna-se essencial a utilização de determinados componentes que garantem sua precisão e eficiência. A seguir, apresenta-se uma tabela com a relação desses componentes, acompanhados do custo base correspondente a cada um.

Tabela 6 – Custo base.

COMPONENTE	QUANTIDADE	FINALIDADE	VALOR	VALOR FINAL (TOTAL)
Kit Esp32-Cam C/ Conversor USB 4 /FTDI	1	Elementos principais para o reconhecimento.	R\$89,99	R\$89,99
Servo Motor	1	Responsável pela movimentação da cancela.	R\$25,00	R\$25,00
Bateria 18650 3,7V	2	Encarregada pela alimentação do projeto.	R\$15,00	R\$30,00
Suporte duplo de Bateria 18650	1	Tem a função de designar a energia para o projeto.	R\$25,00	R\$25,00
Regulador de Tensão LM7805	1	Regula a tensão de 9V para 5V.	R\$4,50	R\$4,50

Fonte: Autoria própria.

## 2.7 CONSTRUÇÃO DO PROTÓTIPO

O protótipo conta com um microcontrolador principal, sendo ele o ESP32-CAM, mais conhecido como ESP-CAM.

O mesmo foi escolhido por conta de seu melhor custo benefício, sendo um microcontrolador produzido pela empresa *Espressif Systems®* e contando com sua variedade de recursos. Este componente possui um chip ESP32-S Xtensa® *dual core* LX6 de 32 bits um *clock* ajustável com uma frequência máxima de 240MHz, 4 mb de memória, *flash* e disponibiliza 16 pinos de entrada e saída (E/S), além de uma câmera que pode ser integrada a placa, o que o torna mais conhecido por sua função (JÚNIOR, 2023).

O ESP32 atua como o maior controlador do projeto, onde tem a finalidade principal de processamento de imagens, controle de motores, coleta de dados e

armazenamento. Seus 16 pinos agem de maneira crucial para o funcionamento do micro controlador.

A placa desse microcontrolador não conta com conector USB, foi preciso realizar sua conexão através de *jumpers* fêmea a um conversor USB, sendo especificamente nomeado “FT232RL”. Onde, para o funcionamento necessário e completo do microcontrolador, só seria possível seu desenvolvimento se ligado ao conversor.

Figura 6- ESP32-CAM.



Fonte: Autoria própria

Tabela 7 - Esquemático de ligação do ESP com o FT232RL.

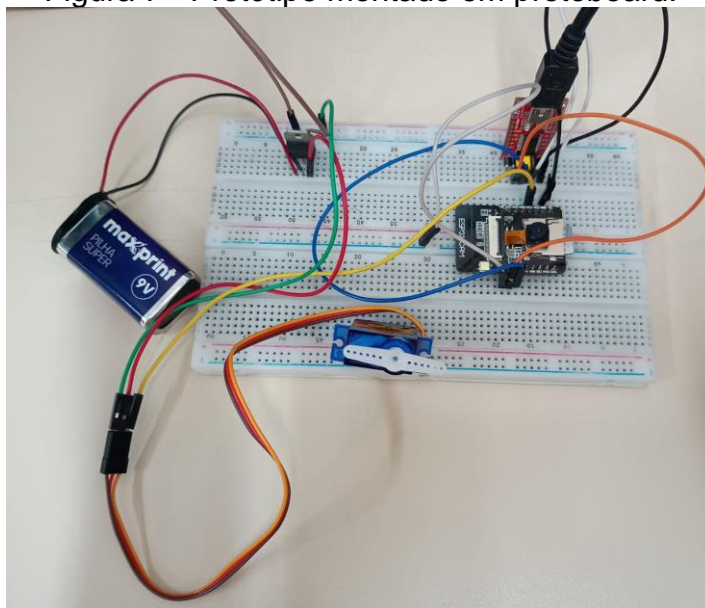
ESP32-CAM	FT232RL
5V	VCC
GND	GND
RX	UOT
TX	UOR

Fonte: Autoria própria

Para o funcionamento do sistema de cancela, utiliza-se um Servo Motor, onde será empregado apenas para a demonstração do resultado do estudo realizado. O motor conta com uma alimentação de 5 volts e é ligado o seu pino GPIO ao pino 12 do ESP32-CAM, onde ele é o responsável pelo acionamento e movimentação da cancela tendo uma codificação básica.

Abaixo segue o registro da montagem do protótipo em uma protoboard, onde o micro controlador está sendo alimentado por um notebook e o Servo Motor é alimentado por uma fonte externa, sendo ela uma bateria de 9 volts.

Figura 7 - Protótipo montado em protoboard.



Fonte: Autoria própria

A montagem registrada acima é apenas para a demonstração de funcionamento do sistema.

Figura 8 - Maquete de representação do local de sua aplicação.

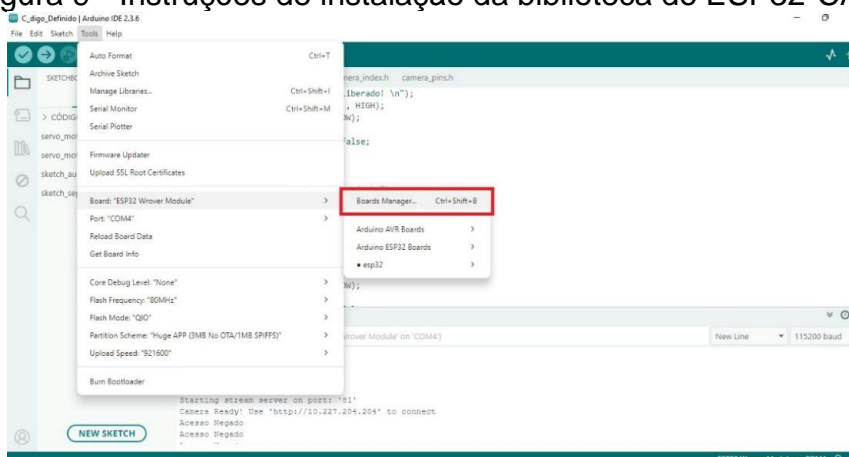


Fonte: Autoria própria

Para a ligação do Servo Motor, é necessário um regulador de tensão 7805 que irá transformar 9 volts em 5 volts, assim fazendo com que este trabalhe com êxito.

Este protótipo visa ter um trabalho que dure 12 horas por dia, portanto é necessário que aja um dimensionamento de uma bateria com alimentação suficiente para funcionar. De acordo com tais necessidades, o cálculo de dimensionamento de bateria consiste no cálculo de consumo diário que seria a corrente total multiplicada pela quantidade de tempo (horas) que o protótipo opera. Que nesse caso, seria a corrente total de 2 amperes/hora consumidos. Conclui-se então que a bateria necessária para a alimentação seria a de 5 volts e 10.000 miliamperes/hora. O ESP32 conta com a mesma interface de programação do Arduino, o que torna o seu desenvolvimento menos complicado. Para a ligação e compilação de um software no ESP32, primeiramente deve-se adicionar e instalar a biblioteca do ESP32 ao Arduino IDE do desenvolvedor do projeto.

Figura 9 - Instruções de instalação da biblioteca do ESP32-CAM.



Fonte: Autoria própria

Figura 10 - Continuação de instruções.



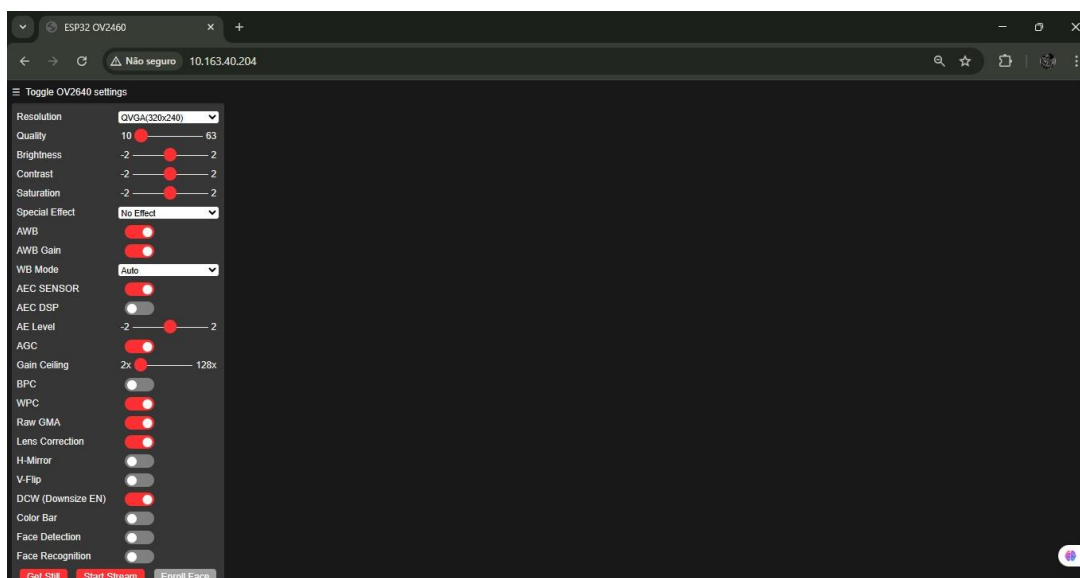
Fonte: Autoria própria

Em seguida, utilizando uma programação base e exercendo a opção de compilação da fonte para o micro controlador, consegue-se acessar, no monitor serial, o IP para visualização da interface de configuração do ESPCAM.

Após haver o acesso à interface de configuração do micro controlador base, pode-se notar que existem duas funções, onde é possível haver a detecção facial e o reconhecimento facial, essas funções serão essenciais para a funcionalidade perfeita do projeto de acordo com os objetivos propostos anteriormente. Ao utilizar essas funções é possível identificar um contorno em volta à face detectada, o que se resulta no reconhecimento facial, onde, em seu primeiro acionamento, ativa o programa base que imprime em volta da face detectada uma caixa vermelha sinalizando um alerta a intrusos. Para que se faça o cadastro de uma face de um usuário é necessário utilizar a função “*enroll face*”, que seria registrar a face. Com isso, após o cadastro do usuário, é impressa a interface, em volta do rosto do usuário cadastrado, um contorno verde, que indica “Olá pessoa 0”, sendo um número que aumentará de acordo com a quantidade de usuários cadastrados.

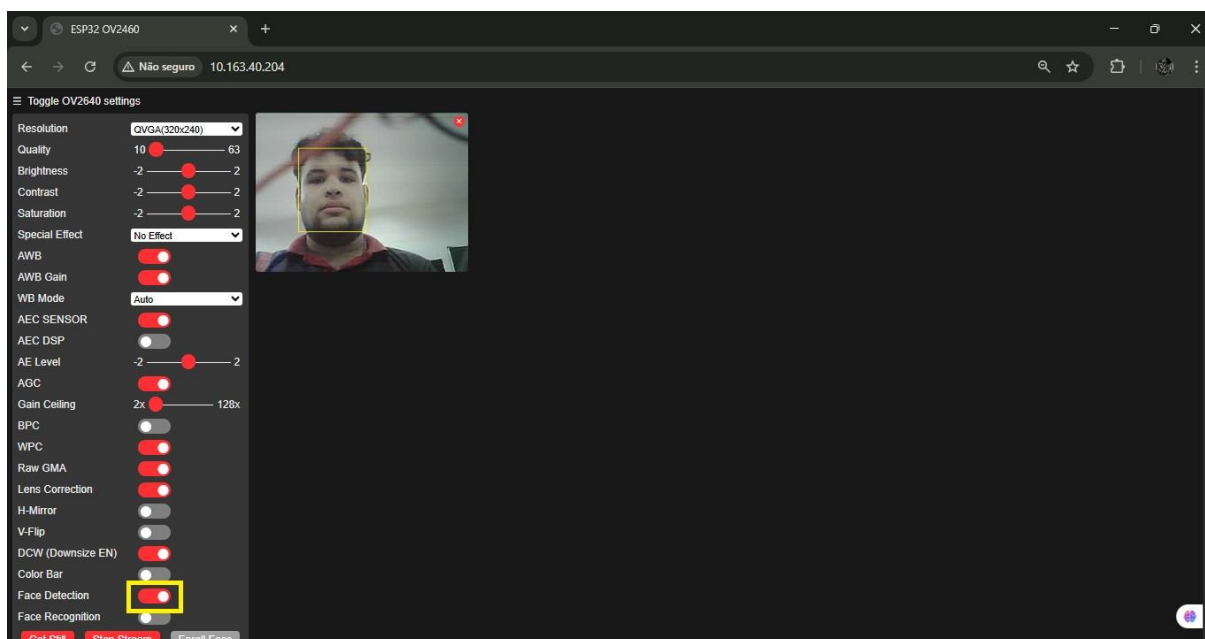
A seguir, uma sequência de imagens expondo a interface do *software* proveniente do próprio ESP32-CAM, com as etapas e orientações referentes ao registro de faces.

Figura 11 - Interface de desenvolvimento da ESP-CAM.



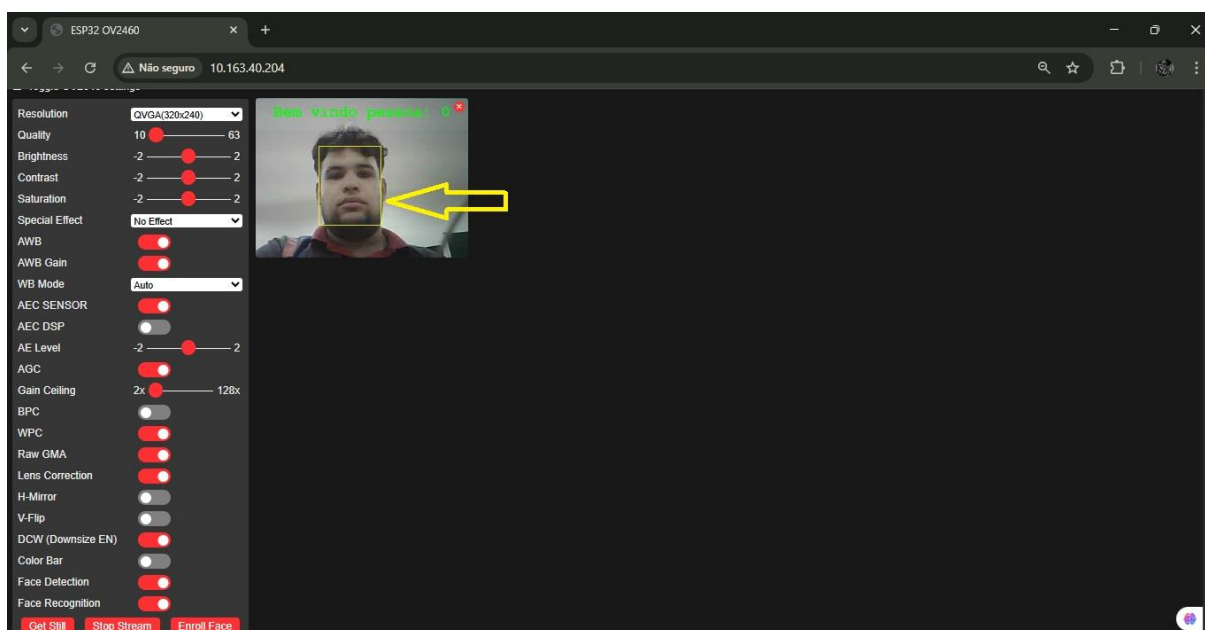
Fonte: Autoria própria

Figura 12 – Função de detecção facial ativada.



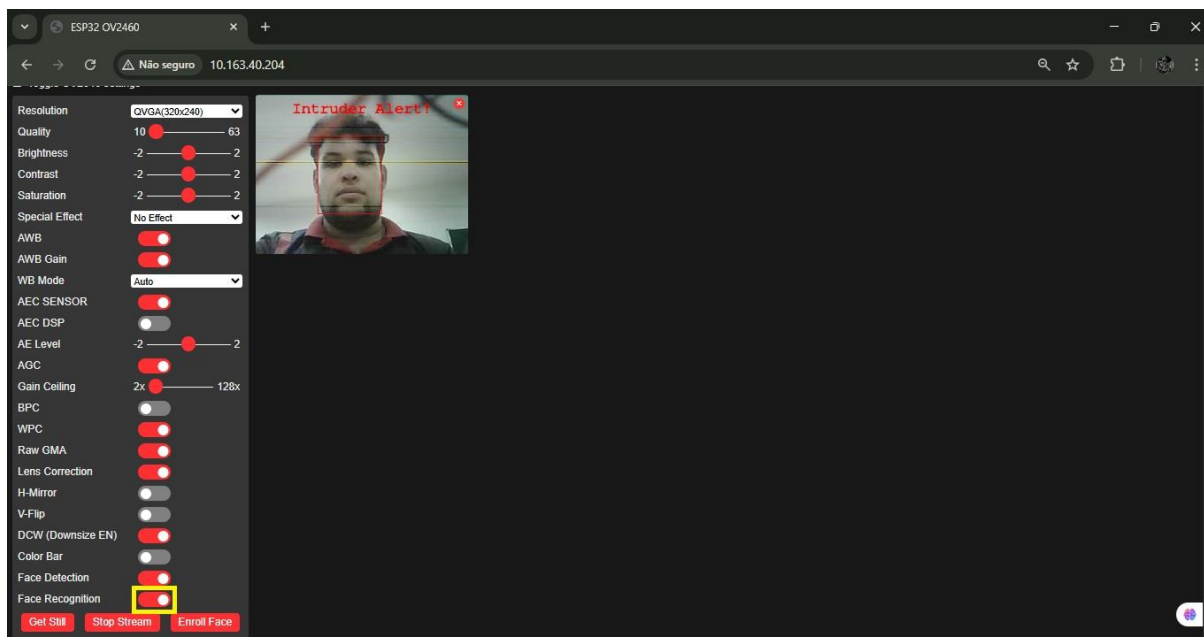
Fonte: Autoria própria

Figura 13 - Reconhecimento da face cadastrada.



Fonte: Autoria própria

Figura 14 - "intruder alert!" Face não reconhecida.



Fonte: Autoria própria

Este seria o tutorial base de aplicação do reconhecimento facial obtido pela ESP32-CAM, por todavia a construção do protótipo não se resume a apenas essa aplicação. Para que o protótipo funcione de maneira que seja feito um controle de acesso através do reconhecimento facial é necessário exercer alterações ao código base.

Para haver a simulação da aplicação do reconhecimento facial em um controle de acesso é utilizado o Servo Motor SG90, onde irá se mover até 90°, bloqueando passagem. Assim indicando o acesso não permitido. No caso contrário irá se mover até 180°, liberando passagem. Assim concedendo o acesso permitido.

A alteração no código é o elemento principal para o melhor funcionamento, então, foi necessário a implementação de uma variável booleana chamada "Autorizacao\_Acesso" ao início do código, assim como é mostrado na figura, colocando-a como "falsa", sendo assim iniciando-a do zero.

Figura 15 - Implementação da variável ao código.

```

1 #include "esp_camera.h"
2 #include <WiFi.h>
3 boolean Autorizacao_Acesso = false;
4

```

Fonte: Autoria própria

Após isso é necessário, em outra aba do código, adicionar a mesma variável, porem de forma externa, pois ela já foi declarada anteriormente ao código principal, vide figura abaixo.

Figura 16 - Implementação de variável externa.

```

64
65 extern boolean Autorizacao_Acesso;
66
67 static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t sample_size){
68     memset(filter, 0, sizeof(ra_filter_t));
69

```

Fonte: Autoria própria

Com o processo anterior feito, é necessário, no local de configuração do reconhecimento facial, onde é construído o que será feito se o rosto for reconhecido ou não. Sendo assim, se a face for reconhecida, a função “Autorizacao\_Acesso” será acionada como “falsa”, tornando o acesso negado, caso contrário, a função será acionada como “verdadeira” e o acesso será concedido, assim como mostrado na figura abaixo.

Figura 17 - Implementação da função da variável.

```

188     } else {
189         matched_id = recognize_face(sid_list, aligned_face);
190         if (matched_id >= 0) {
191             Serial.printf("Match Face ID: %u\n", matched_id);
192             rgb_printf(image_matrix, FACE_COLOR_GREEN, "Bem vindo pessoa: %u", matched_id);
193             Autorizacao_Acesso = true;
194         }
195         else
196         {
197             Serial.println("No Match Found");
198             rgb_print(image_matrix, FACE_COLOR_RED, "Intruder Alert!");
199             matched_id = -1;
200             Autorizacao_Acesso = false;
201         }

```

Fonte: Autoria própria

A partir das alterações executadas, é preciso configurar o que a função comandará ao protótipo. Acessando, na aba principal, a configuração de ações em repetição, dentro do Arduino IDE, é necessário indicar que quando a função “Autorizacao\_Acesso” for considerada “verdadeira” o Servo Motor será acionado movimentando assim a cancela em 180°, assim relatando que uma face foi

reconhecida e o acesso foi liberado. Caso contrário, a função será reconhecida como “falsa”, controlando o Servo Motor até posição 90°, indicando que o acesso foi negado. O resultado do processo é retratado na figura abaixo.

Figura 18 - Configuração final de função da variável.

```

86 void loop()
87 {
88     if(Autorizacao_Acesso == true)
89     {
90         Serial.print("Acesso Liberado! \n");
91         meuServo.write(90);
92         delay(1000);
93         Autorizacao_Acesso = false;
94     }
95     else
96     {
97         Serial.print("Acesso Negado \n");
98         meuServo.write(180);
99         delay(1000);
100    }
101    // put your main code here, to run repeatedly:
102    delay(1000);
103 }

```

Fonte: Autoria própria

Por fim, o último processo será definir o Servo Motor utilizado como saída de informação e referenciá-los a qual pino do micro controlador serão ligados, assim de acordo com as figuras abaixo.

Figura 19 – Definição de ligação do Servo Motor ao micro controlador.

```

12
13 #define SERVO 2

```

Fonte: Autoria própria

O protótipo conta com uma programação extensa, e ela será explicada detalhadamente de acordo com suas funções ao decorrer do capítulo. A seguir, o código obtido ao final das alterações:

## 2.8 PROTOCOLO DE TESTES

Para o projeto desenvolvido exercer uma boa performance, é necessário que haja testes. Por isso foi desenvolvido uma sequência de etapas, nas quais o protótipo seria necessário passar para realmente provar o seu funcionamento.

Sendo os testes: 1°Conectar ESP32-CAM à rede Wi-Fi; 2°Iniciar módulo de reconhecimento facial; 3°Capturar rosto autorizado; 4°Capturar rosto não autorizado; 5°Verificar acionamento do Servo Motor com rosto autorizado; 6°Verificar acionamento do Servo Motor com rosto não autorizado; 7°Testar estabilidade com múltiplas tentativas; 8° Verificar tempo de resposta do reconhecimento à ação do servo; 9° Reconhecer mais de um rosto por vez; 10° Reconhecer rostos de pessoas com óculos;

Análise de resposta esperadas de acordo com cada numeração dos testes: 1°Conexão estabelecida, IP atribuído; 2°Módulo inicializa sem erros; 3°Sistema identifica rosto cadastrado; 4°Sistema reconhece, acesso permitido; 5° Servo Motor gira 180° liberando a passagem; 6° Sistema não reconhece, acesso negado; 7° Servo não aciona, cancela permanece fechada; 8° Sistema mantém precisão e resposta; 9° Tempo de resposta  $\leq 2$  segundos.

### 2.8.1 EXECUÇÃO DOS TESTES

Foi executada a alimentação do projeto e seu funcionamento, onde se conecta com a rede Wi-Fi 2,4GHz.

Módulo de reconhecimento facial iniciado com a passagem da programação ao ESP sendo alimentado pela fonte de compilação.

Foi iniciada a captura de rosto cadastrado pelo micro controlador com a conexão Wi-Fi já estabelecida, com distancia de menos de um metro do roteador.

Face não autorizada contém sua captura e reconhecimento.

O Servo Motor na situação de reconhecimento de um rosto cadastrado foi analisado com alimentação de 5 volts.

O Servo Motor na situação de reconhecimento de um rosto não cadastrado foi analisado com alimentação de 5 volts.

Foram testadas múltiplas tentativas de entrada quando não reconhecido.

O tempo de resposta foi observado em relação ao reconhecimento e ao acionamento do Servo Motor.

## **2.8.2 COLETA DE DADOS DOS TESTES**

Após ser iniciada a conexão Wi-Fi ao ESP, foi obtida ótima conexão e disponibilidade Wi-Fi, porém enfrenta problemas para ser aberta a interface de configuração ESPCAM, onde é necessário atualizar diversas vezes o navegador na qual está sendo acessado.

O módulo de reconhecimento facial atua com total funcionalidade, porém sofre com processo lento devido à complexidade do reconhecimento.

O reconhecimento facial é executado com sucesso, porém enfrenta problemas durante o processamento de imagens. Contudo o rosto cadastrado é totalmente reconhecido. Ao caso do rosto não cadastrado é enfrentado o mesmo problema de processamento visual da identificação facial.

## **2.8.3 ANÁLISE DOS DADOS DOS TESTES**

A conexão Wi-Fi é extremamente essencial, que de acordo com os resultados dos testes foi totalmente concedida sendo algo considerável para o avanço do projeto, que torna o próximo passo possivelmente concedido onde o modulo de reconhecimento facial exerce totalmente sua função.

A questão de processamento de imagens é vista como um problema por prejudicar a praticidade do sistema de controle de acesso, porem o básico necessário é obtido.

O Servo Motor age de forma funcional e possui habilidades para o controle de acesso, o que é extremamente importante para o bom funcionamento.

## **2.9 MELHORIAS**

Após testes e avaliações de desempenho alguns ajustes futuros seriam possíveis se adicionados ao protótipo para melhor performance, sendo eles: maior capacidade de armazenamento de imagens e dados, otimização do sistema de reconhecimento tornando-o mais ágil e preciso; integração com controles de ponto ou presença para uso como registro de entrada em escolas e/ou empresas, capacitor de Backup ou Fonte Ininterrupta (UPS) para garantir funcionamento mesmo durante quedas rápidas de energia.

Tem-se também grande aspiração por atualizações e avanços que incremente ou substituam o sistema de reconhecimento facial para o reconhecimento biométrico de íris, mas esta ideia se aplica como um projeto futuro.

## **2.10 ASPECTOS POSITIVOS**

Aspectos que agem de maneira positiva são: predominância de recursos que colaboram para o aprimoramento na segurança dos espaços onde for aplicado reconhecimento facial é difícil de burlar e garante que apenas pessoas autorizadas tenham acesso, aumentando a proteção de ambientes restritos; automação e praticidade, pois elimina o uso de crachás, senhas ou cartões, tornando o acesso mais rápido, eficiente e com menos riscos de perda ou esquecimento de identificadores e por consequência gerando redução de custos a longo prazo, apesar do custo inicial de instalação, reduz gastos com insumos físicos e manutenção de equipamentos como leitores de cartão ou botões. Acesso sem contato físico, o sistema funciona à distância, sem necessidade de toque, o que promove higiene, agilidade e é ideal para locais com grande fluxo de pessoas.

## **2.11 RISCOS**

O protótipo pode trazer alguns riscos à privacidade dos dados biométricos, caso não haja segurança adequada. Também há riscos técnicos, como falhas no reconhecimento facial ou no funcionamento da cancela. Além disso, o sistema pode excluir usuários com deficiência visual, caso não existam métodos alternativos de autenticação e a baixa aceitação do público, principalmente por desconfiança quanto ao uso da biometria pode trazer riscos a integridade do projeto.

## **2.12 DIFERENCIAIS**

Tem-se como principais diferenciais: baixo-custo, uso de materiais reaproveitados para a confecção do protótipo, facilidade de manutenção, com estrutura simples que permite troca de componentes com agilidade e reconhecimento sem contato físico, promovendo maior higiene. Tais aspectos permitem melhor

desempenho em sua aplicação prática e conseqüentemente provável melhor recepção do produto no mercado.

### **2.13 RECOMENDAÇÕES**

Recomenda-se que para que seja possível a elaboração deste projeto e obtenção de resultados convenientes, que se dê relevância aos requerimentos exigidos. É necessário adquirir um conhecimento prévio e básico a respeito dos componentes usados em especial o ESP32-CAM, este estabelecido como peça central do sistema, conhecimentos primários em programação de softwares, mais especificamente em Linguagem C++, também se demonstra essencial a obtenção do Driver do próprio ESP32-CAM para a aquisição do código base do reconhecimento facial.

Este discernimento é fundamental para superar possíveis desafios encontrados ao decorrer da montagem do projeto. A partir disso torna-se mais fácil otimizar o código, calibrar as peças e consertar os erros. Esse preparo é o que garante a precisão do projeto, melhora seu desempenho e, no fim das contas, nos permite entregar um produto final muito mais confiável e eficiente.

### **3 CONSIDERAÇÕES FINAIS**

O atual trabalho teve como principal objetivo fornecer segurança e praticidade a ambientes restritos com hardware e software integrados agindo de maneira eficiente por um baixo custo, motivado pela necessidade de sanar possíveis erros e fraudes, além de agilizar e gerar melhorias ao tráfego e na atividade de entrada e saída em ambientes que demandem rigor no controle de circulação de pessoas.

Ao longo desta pesquisa visou-se não apenas explorar a busca por oferecer modernidade e segurança, mas também analisar e tomar como parâmetro temas que envolvam inclusão, acessibilidades e sustentabilidade a fim de trazer melhorias ao cotidiano das pessoas. Apesar das melhores intenções postas em pauta no desenvolvimento deste trabalho.

O protótipo apresentou diversas limitações. O sistema de reconhecimento facial demonstrou sensibilidade a variações de iluminação e ângulos de captura, podendo ter seu desempenho afetado em ambientes externos com luz solar direta ou durante a noite sem iluminação adequada. Além disso, a capacidade de processamento do hardware utilizado limitou o número de faces que o sistema pode identificar em tempo real, indicando sua incapacidade para aplicações em larga escala sem uma rigorosa otimização e implementações que no geral não são promissoras devido ao custo e demais desavenças. Apesar das limitações enfrentadas, o servo apresenta um bom desempenho.

Avanços mais complexos e aprimorados como a integração de IA e funcionamento por energia solar também são vistos com anseio pelo grupo e almeja-se sua integração. Em suma, este projeto validou o conceito de uma cancela eletrônica autônoma baseada em reconhecimento facial, demonstrando que a fusão de diferentes tecnologias pode resultar em soluções inovadoras e eficazes para desafios de segurança contemporâneos. A jornada de desenvolvimento e os resultados obtidos confirmam o potencial dessa tecnologia e abrem as portas para a criação de sistemas de controle de acesso ainda mais inteligentes e integrados no futuro.

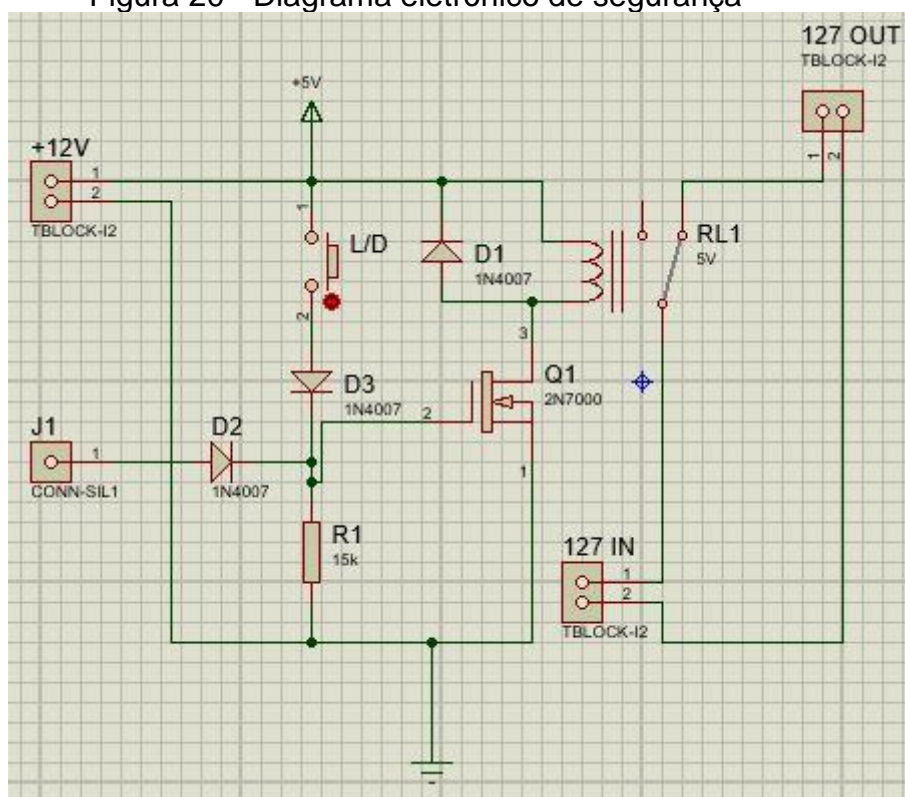
#### **3.1 TRABALHOS FUTUROS**

Com base nos resultados obtidos e nos diagnósticos limitantes do protótipo, o potencial para o futuro do projeto é positivo. Propõe-se visando ao futuro que pesquisas se concentrem na otimização do sistema, aprimorando os algoritmos de

reconhecimento facial para que funcionem em condições adversas de iluminação ou com múltiplas pessoas com possível implementação de Micro SD. Outro caminho promissor é a integração com novas tecnologias, como a Internet das Coisas (IoT), Inteligência Artificial (IA) ou o processamento em nuvem, para melhorar o desempenho, reduzir custos e permitir o monitoramento em tempo real do fluxo de pessoas. Além disso, a expansão da aplicação do sistema para outros ambientes, como estacionamentos ou sistemas de ponto eletrônico, é um tópico relevante para comprovar sua viabilidade em diferentes cenários.

Seriam necessárias a substituição de alguns componentes para a execução em maior escala, como, o servo Motor e a alimentação. Onde poderiam ser utilizados um motor de torque maior e uma alimentação com placas solares ou diretamente da rede de energia. Para que essa nova fonte de alimentação possa ser implementada pode ser necessário a adição de uma eletrônica de segurança utilizando um relé, diodos, transistores, e outros materiais, assim como é mostrado na imagem abaixo.

Figura 20 - Diagrama eletrônico de segurança



Fonte: Autoria própria

Por fim, uma análise aprofundada da viabilidade econômica do protótipo, incluindo custos de produção e modelo de negócio, seria essencial para transformar a inovação em um produto comercializável.

## REFERÊNCIAS

CEBRIAN, Fabiana S. P. Faraco et al. **Radar Tecnológico nº 2: Biometria e Reconhecimento Facial – estudos preliminares**. Brasília, DF: Autoridade Nacional de Proteção de Dados – ANPD, jun. 2024. Disponível em: <https://www.gov.br/anpd/pt-br/centrais-de-conteudo/documentos-tecnicos-orientativos/radar-tecnologico-biometria-anpd.pdf>. Acesso em: 30 out. 2025.

GALIMBERTI, Luiz Henrique de Oliveira. **ESTUDO COMPARATIVO DE ALGORITMOS DE BIOMETRIA FACIAL DISPONIBILIZADOS PELA BIBLIOTECA OPENCV PARA CONTROLE DE ACESSO**. 2018. Disponível em: <https://www.univates.br/bduserver/api/core/bitstreams/a5321730-0ad6-4bb4-9a1f-78d6752c59dc/content>. Acesso em: 28 out. 2025.

LEMOS, Rodrigo Pinto. **Sistema de controle de acesso através de reconhecimento facial com monitoramento remoto**. 2024. Disponível em: <https://rbmc.org.br/rbmc/article/download/175/101/1163>. Acesso em: 18 jun. 2025.

ROCHA, Nataly M.; ROCHA, Tiago F. da. **Checkinclassroom: sistema de controle de presença com reconhecimento facial**. Monteiro-PB: Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Campus Monteiro, 2024. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas). Disponível em: <https://repositorio.ifpb.edu.br/bitstream/177683/4371/1/TCC%20TIAGO%20NATALY.pdf>. Acesso em: 18 jun. 2025.

SADEQ, Sadeer; ZIDAN, Khamis A.; STEPHAN, Jane J. **A Review on Iris Recognition System Based Different Classification Techniques. Solid State Technology**. v. 64, n. 2, p. 5106-5120, 2021. Disponível em: [https://www.researchgate.net/profile/Khamis-Zidan/publication/350793662\\_A\\_Review\\_on\\_Iris\\_Recognition\\_System\\_Based\\_Different\\_Classification\\_Techniques/links/6072d20b4585150fe99bd632/A-Review-on-Iris-Recognition-System-Based-Different-Classification-Techniques.pdf?\\_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19](https://www.researchgate.net/profile/Khamis-Zidan/publication/350793662_A_Review_on_Iris_Recognition_System_Based_Different_Classification_Techniques/links/6072d20b4585150fe99bd632/A-Review-on-Iris-Recognition-System-Based-Different-Classification-Techniques.pdf?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19). Acesso em: 28 out. 2025.

SILVA, Pedro Lucas Martins da; RIZZETTI, Tiago Antonio. **Aperfeiçoando Sistemas de Controle de Acesso utilizando Reconhecimento Facial**. 2024. Disponível em: <https://sol.sbc.org.br/index.php/errc/article/view/35459/35246>. Acesso em: 18 jun. 2025.

## APÊNDICE – Código do projeto

```
#include "esp_camera.h" // Inclui biblioteca para controle da
câmera ESP32-CAM, permitindo
captura e configuração de imagens.
#include <WiFi.h> // inclui biblioteca para gerenciamento de
conexões Wi-Fi, essencial para
transmissão de dados.
#include <ESP32Servo.h> // inclui biblioteca para controle de
servo motores, utilizando PWM para
posicionamento angular.
boolean Autorizacao_Acesso = false; // Declara variável booleana
para controle de autorização,
inicializada como false (acesso negado).
#define CAMERA_MODEL_AI_THINKER // Define o modelo da
câmera como AI_THINKER,
configurando pinos automaticamente via "camera_pins.h".
#include "camera_pins.h" // Inclui arquivo com mapeamento de
pinos GPIO para o modelo
AI_THINKER.
const char* ssid = "eumurilo"; // Define SSID da rede Wi-Fi para
conexão.
const char* password = "murilo94"; // Define senha da rede Wi-
Fi.
#define SERVO 2 // define pino GPIO 2 para o servo; limitação:
pode causar conflitos com
câmera ou LED integrado.
void startCameraServer(); // Declara função externa para iniciar
servidor de câmera.
Servo meuServo; // Declara objeto Servo para controle do
atuador.
void setup() { // Função setup(), executada uma vez para
inicializar hardware e software.
  Serial.begin(115200); // Inicializa comunicação serial a 115200
bps para debug.
  Serial.setDebugOutput(true); // Habilita saída de debug
detalhada.
  Serial.println(); // Imprime linha em branco para separar logs.
  meuServo.attach(SERVO); // Anexa servo ao pino GPIO 2;
limitação: executado antes da câmera,
```

pode causar interferência PWM.

```
camera_config_t config; // Declara estrutura de configuração
da câmera.
config.ledc_channel = LEDC_CHANNEL_0; // Define canal PWM
0 para câmera.
config.ledc_timer = LEDC_TIMER_0; // Define timer PWM para
câmera.
config.pin_d0 = Y2_GPIO_NUM; // Define pino para bit de dados
0.
config.pin_d1 = Y3_GPIO_NUM; // Define pino para bit de dados
1.
config.pin_d2 = Y4_GPIO_NUM; // Define pino para bit de dados
2.
config.pin_d3 = Y5_GPIO_NUM; // Define pino para bit de dados
3.
config.pin_d4 = Y6_GPIO_NUM; // Define pino para bit de dados
4.
config.pin_d5 = Y7_GPIO_NUM; // Define pino para bit de dados
5.
config.pin_d6 = Y8_GPIO_NUM; // Define pino para bit de dados
6.
config.pin_d7 = Y9_GPIO_NUM; // Define pino para bit de dados
7.
config.pin_xclk = XCLK_GPIO_NUM; // Define pino para clock
externo.
config.pin_pclk = PCLK_GPIO_NUM; // Define pino para clock
de pixel.
config.pin_vsync = VSYNC_GPIO_NUM; // Define pino para
sincronização vertical.
config.pin_href = HREF_GPIO_NUM; // Define pino para
referência horizontal.
config.pin_sscb_sda = SIOD_GPIO_NUM; // Define pino SDA para
I2C.
config.pin_sscb_scl = SIOC_GPIO_NUM; // Define pino SCL para
I2C.
config.pin_pwdn = PWDN_GPIO_NUM; // Define pino para power
down.
config.pin_reset = RESET_GPIO_NUM; // Define pino para reset.
config.xclk_freq_hz = 20000000; // Define frequência do clock
externo (20 MHz).
```

```

    config.pixel_format = PIXFORMAT_JPEG; // Define formato de
    pixel como JPEG.
    // Configuração condicional baseada em PSRAM.
    if(psramFound()){ // Verifica se PSRAM está disponível.
        config.frame_size = FRAMESIZE_UXGA; // Define resolução alta
        se PSRAM presente.
        config.jpeg_quality = 10; // Define qualidade JPEG baixa.
        config.fb_count = 2; // Define 2 buffers de frame.
    } else {
        config.frame_size = FRAMESIZE_SVGA; // Define resolução
        média se sem PSRAM.
        config.jpeg_quality = 12; // Define qualidade JPEG média.
        config.fb_count = 1; // Define 1 buffer de frame.
    }
    // Inicialização da câmera.
    esp_err_t err = esp_camera_init(&config); // Tenta inicializar
    câmera.
    if (err != ESP_OK) { // Verifica erro.
        Serial.printf("Camera init failed with error 0x%x", err); // Log de
        erro.
        return; // Interrompe se falhar.
    }
    sensor_t * s = esp_camera_sensor_get(); // Obtém ponteiro
    para sensor.
    s->set_framesize(s, FRAMESIZE_QVGA); // Define resolução
    para QVGA.
    WiFi.begin(ssid, password); // Inicia conexão Wi-Fi.
    while (WiFi.status() != WL_CONNECTED) { // Loop até conectar.
        delay(500); // Delay de 500ms.
        Serial.print("."); // Imprime ponto para progresso.
    }
    Serial.println(""); // Linha em branco.
    Serial.println("WiFi connected"); // Confirmação de conexão.
    startCameraServer(); // Inicia servidor de câmera.
    Serial.print("Camera Ready! Use 'http://"); // Mensagem de
    prontidão.
    Serial.print(WiFi.localIP()); // Imprime IP local.
    Serial.println(" to connect"); // Completa mensagem.
}
void loop() { // Função loop(), executada continuamente após
    setup().

```

```

    if(Autorizacao_Acesso == true) { // Verifica se autorização é
true.
    Serial.print("Acesso Liberado! \n"); // Log de liberação.
    meuServo.write(180); // Move servo para 180 graus (posição de
abertura).
    delay(1000); // Mantém por 1 segundo
    Autorizacao_Acesso = false; // Reseta flag.
    } else { // Caso contrário.
    Serial.print("Acesso Negado \n"); // Log de negação.
    meuServo.write(90); // Move servo para 90 graus (posição de
fechamento).
    delay(1000); // Mantém por 1 segundo.
    }
    // put your main code here, to run repeatedly:
    delay(1000); // Delay geral de 1 segundo para controlar loop.
}
#include "esp_http_server.h" // Inclui biblioteca para servidor
HTTP no ESP32.
#include "esp_timer.h" // Inclui biblioteca para medição de tempo
precisa.
#include "esp_camera.h" // Inclui biblioteca para controle da
câmera ESP32-CAM.
#include "img_converters.h" // Inclui biblioteca para conversão de
imagens.
#include "camera_index.h" // Inclui arquivo com interface HTML
para câmera.
#include "Arduino.h" // Inclui biblioteca Arduino para
compatibilidade.
#include "fb_gfx.h" // Inclui biblioteca para gráficos em
framebuffer.
#include "fd_forward.h" // Inclui biblioteca para detecção facial.
#include "fr_forward.h" // Inclui biblioteca para reconhecimento
facial.
#define ENROLL_CONFIRM_TIMES 5 // Define número de amostras
para enrollment.
#define FACE_ID_SAVE_NUMBER 7 // Define máximo de IDs de
faces.
#define FACE_COLOR_WHITE 0x00FFFFFF // Define cor branca.
#define FACE_COLOR_BLACK 0x00000000 // Define cor preta.
#define FACE_COLOR_RED 0x000000FF // Define cor vermelha.
#define FACE_COLOR_GREEN 0x0000FF00 // Define cor verde.
#define FACE_COLOR_BLUE 0x00FF0000 // Define cor azul.

```

```

#define     FACE_COLOR_YELLOW      (FACE_COLOR_RED      |
FACE_COLOR_GREEN) // Define cor
amarela.
#define     FACE_COLOR_CYAN       (FACE_COLOR_BLUE     |
FACE_COLOR_GREEN) // Define cor ciano.
#define     FACE_COLOR_PURPLE     (FACE_COLOR_BLUE     |
FACE_COLOR_RED) // Define cor roxa.
typedef struct { // Define estrutura para filtro de média móvel.
    size_t size; // Campo para tamanho do filtro.
    size_t index; // Campo para índice atual.
    size_t count; // Campo para contagem de valores.
    int sum; // Campo para soma acumulada.
    int * values; // Campo para array de valores.
} ra_filter_t;
typedef struct { // Define estrutura para chunking de JPEG.
    httpd_req_t *req; // Campo para ponteiro de requisição HTTP.
    size_t len; // Campo para comprimento dos dados.
} jpg_chunking_t;
#define PART_BOUNDARY "12345678900000000000000987654321"
// Define delimitador para
streaming.
static const char* _STREAM_CONTENT_TYPE = "multipart/x-
mixed-replace;boundary="
PART_BOUNDARY; // Define tipo de conteúdo para streaming.
static const char* _STREAM_BOUNDARY = "\r\n--"
PART_BOUNDARY "\r\n"; // Define boundary para
partes.
static const char* _STREAM_PART = "Content-Type:
image/jpeg\r\nContent-Length: %u\r\n\r\n"; //
Define cabeçalho para partes JPEG.
static ra_filter_t ra_filter; // Declara instância do filtro.
httpd_handle_t stream_httpd = NULL; // Declara handle para
servidor de streaming.
httpd_handle_t camera_httpd = NULL; // Declara handle para
servidor de câmera.
static mtmn_config_t mtmn_config = {0}; // Declara
configuração para detecção MTMN.
static int8_t detection_enabled = 0; // Declara flag para
detecção.
static int8_t recognition_enabled = 0; // Declara flag para
reconhecimento.
static int8_t is_enrolling = 0; // Declara flag para enrollment.

```

```

static face_id_list id_list = {0}; // Declara lista de IDs de faces.
extern boolean Autorizacao_Acesso; // Declara variável externa
para autorização.
static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t
sample_size) { // Função para inicializar filtro.
    memset(filter, 0, sizeof(ra_filter_t)); // Zera a estrutura.
    filter->values = (int *)malloc(sample_size * sizeof(int)); //
Aloca memória para array.
    if (!filter->values) { // Verifica se alocação falhou.
        return NULL; // Retorna NULL se falhou.
    }
    memset(filter->values, 0, sample_size * sizeof(int)); // Inicializa
array com zeros.
    filter->size = sample_size; // Define tamanho.
    return filter; // Retorna ponteiro.
}
static int ra_filter_run(ra_filter_t * filter, int value) { // Função
para executar filtro.
    if (!filter->values) { // Verifica se filtro inicializado.
        return value; // Retorna valor original.
    }
    filter->sum -= filter->values[filter->index]; // Subtrai valor
antigo.
    filter->values[filter->index] = value; // Insere novo valor.
    filter->sum += filter->values[filter->index]; // Adiciona novo
valor.
    filter->index++; // Incrementa índice.
    filter->index = filter->index % filter->size; // Cicla índice.
    if (filter->count < filter->size) { // Incrementa contagem.
        filter->count++;
    }
    return filter->sum / filter->count; // Retorna média.
}
static void rgb_print(dl_matrix3du_t *image_matrix, uint32_t
color, const char * str) { // Função para
imprimir texto em imagem.
    fb_data_t fb; // Declara estrutura framebuffer.
    fb.width = image_matrix->w; // Define largura.
    fb.height = image_matrix->h; // Define altura.
    fb.data = image_matrix->item; // Define dados.
    fb.bytes_per_pixel = 3; // Define bytes por pixel.
    fb.format = FB_BGR888; // Define formato.
}

```

```

    fb_gfx_print(&fb, (fb.width - (strlen(str) * 14)) / 2, 10, color,
str); // Imprime texto.
}
static int rgb_printf(dl_matrix3du_t *image_matrix, uint32_t
color, const char *format, ...) { // Função
para imprimir texto formatado.
    char loc_buf[64]; // Declara buffer local.
    char * temp = loc_buf; // Ponteiro temporário.
    int len; // Comprimento.
    va_list arg, copy; // Argumentos variáveis.
    va_start(arg, format); // Inicia argumentos.
    va_copy(copy, arg); // Copia argumentos.
    len = vsnprintf(loc_buf, sizeof(loc_buf), format, arg); //
Formata string.
    va_end(copy); // Finaliza cópia.
    if (len >= sizeof(loc_buf)) { // Se buffer insuficiente.
        temp = (char*)malloc(len + 1); // Aloca memória.
        if (temp == NULL) { // Verifica falha.
            return 0; // Retorna 0.
        }
    }
    vsnprintf(temp, len + 1, format, arg); // Formata novamente.
    va_end(arg); // Finaliza argumentos.
    rgb_print(image_matrix, color, temp); // Imprime texto.
    if (len > 64) { // Se alocado dinamicamente.
        free(temp); // Libera memória.
    }
    return len; // Retorna comprimento.
}
static void draw_face_boxes(dl_matrix3du_t *image_matrix,
box_array_t *boxes, int face_id) { // Função
para desenhar caixas em faces.
    int x, y, w, h, i; // Declara variáveis para coordenadas.
    uint32_t color = FACE_COLOR_YELLOW; // Define cor padrão.
    if (face_id < 0) { // Se ID negativo.
        color = FACE_COLOR_RED; // Muda para vermelho.
    } else if (face_id > 0) { // Se ID positivo.
        color = FACE_COLOR_GREEN; // Muda para verde.
    }
    fb_data_t fb; // Declara framebuffer.
    fb.width = image_matrix->w; // Define largura.
    fb.height = image_matrix->h; // Define altura.

```

```

fb.data = image_matrix->item; // Define dados.
fb.bytes_per_pixel = 3; // Define bytes por pixel.
fb.format = FB_BGR888; // Define formato.
for (i = 0; i < boxes->len; i++) { // Loop para cada face.
x = (int)boxes->box[i].box_p[0]; // Coordenada X.
y = (int)boxes->box[i].box_p[1]; // Coordenada Y.
w = (int)boxes->box[i].box_p[2] - x + 1; // Largura.
h = (int)boxes->box[i].box_p[3] - y + 1; // Altura.
fb_gfx_drawFastHLine(&fb, x, y, w, color); // Desenha linha
superior.
fb_gfx_drawFastHLine(&fb, x, y + h - 1, w, color); // Desenha
linha inferior.
fb_gfx_drawFastVLine(&fb, x, y, h, color); // Desenha linha
esquerda.
fb_gfx_drawFastVLine(&fb, x + w - 1, y, h, color); // Desenha
linha direita.
#if 0 // Código comentado para landmarks.
int x0, y0, j;
for (j = 0; j < 10; j += 2) {
x0 = (int)boxes->landmark[i].landmark_p[j];
y0 = (int)boxes->landmark[i].landmark_p[j + 1];
fb_gfx_fillRect(&fb, x0, y0, 3, 3, color);
}
#endif
}
}
static int run_face_recognition(dl_matrix3du_t *image_matrix,
box_array_t *net_boxes) { // Função para
reconhecimento facial.
dl_matrix3du_t *aligned_face = NULL; // Declara buffer para
face alinhada.
int matched_id = 0; // Declara ID reconhecido.
aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH,
FACE_HEIGHT, 3); // Aloca buffer.
if (!aligned_face) { // Verifica falha.
Serial.println("Could not allocate face recognition buffer"); //
Log erro.
return matched_id; // Retorna.
}
if (align_face(net_boxes, image_matrix, aligned_face) ==
ESP_OK) { // Alinha face.
if (is_enrolling == 1) { // Se em enrollment.

```

```

    int8_t left_sample_face = enroll_face(&id_list, aligned_face);
    // Registra face.
    if (left_sample_face == (ENROLL_CONFIRM_TIMES - 1)) {
        Serial.printf("Enrolling Face ID: %d\n", id_list.tail);
    }
    Serial.printf("Enrolling Face ID: %d sample %d\n", id_list.tail,
        ENROLL_CONFIRM_TIMES -
        left_sample_face);
    rgb_printf(image_matrix, FACE_COLOR_CYAN, "ID[%u]
    Sample[%u]", id_list.tail,
        ENROLL_CONFIRM_TIMES - left_sample_face);
    if (left_sample_face == 0) {
        is_enrolling = 0;
        Serial.printf("Enrolled Face ID: %d\n", id_list.tail);
    }
    } else { // Modo reconhecimento.
        matched_id = recognize_face(&id_list, aligned_face); //
        Reconhece face.
        if (matched_id >= 0) { // Se reconhecida.
            Serial.printf("Match Face ID: %u\n", matched_id);
            rgb_printf(image_matrix, FACE_COLOR_GREEN, "Bem vindo
            pessoa: %u", matched_id);
            Autorizacao_Acesso = true; // Seta autorização.
        } else { // Se não reconhecida.
            Serial.println("No Match Found");
            rgb_print(image_matrix, FACE_COLOR_RED, "Intruder Alert!");
            matched_id = -1;
            Autorizacao_Acesso = false; // Nega autorização.
        }
    }
    } else { // Se alinhamento falhou.
        Serial.println("Face Not Aligned");
    }
    dl_matrix3du_free(aligned_face); // Libera buffer.
    return matched_id; // Retorna ID.
}

static size_t jpg_encode_stream(void * arg, size_t index, const
void* data, size_t len) { // Callback para
codificação JPEG.
    jpg_chunking_t *j = (jpg_chunking_t *)arg; // Ponteiro para
estrutura.
    if (!index) { // Se primeiro chunk.

```

```

    j->len = 0; // Reseta comprimento.
    }
    if (httpd_resp_send_chunk(j->req, (const char *)data, len) !=
ESP_OK) { // Envia chunk.
    return 0; // Retorna 0 se falhou.
    }
    j->len += len; // Atualiza comprimento.
    return len; // Retorna comprimento enviado.
    }
static esp_err_t capture_handler(httpd_req_t *req) { // Handler
para captura.
    camera_fb_t * fb = NULL; // Declara buffer de frame.
    esp_err_t res = ESP_OK; // Declara resultado.
    int64_t fr_start = esp_timer_get_time(); // Tempo inicial.
    fb = esp_camera_fb_get(); // Captura frame.
    if (!fb) { // Verifica falha.
        Serial.println("Camera capture failed");
        httpd_resp_send_500(req); // Resposta erro.
        return ESP_FAIL;
    }
    httpd_resp_set_type(req, "image/jpeg"); // Define tipo de
resposta.
    httpd_resp_set_hdr(req, "Content-Disposition", "inline;
filename=capture.jpg");
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    size_t out_len, out_width, out_height; // Declara dimensões.
    uint8_t * out_buf; // Declara buffer de saída
    bool s; // Declara flag.
    bool detected = false; // Declara flag de detecção.
    int face_id = 0; // Declara ID da face.
    if (!detection_enabled || fb->width > 400) { // Se detecção
desabilitada ou resolução alta.
        size_t fb_len = 0; // Declara comprimento.
        if (fb->format == PIXFORMAT_JPEG) { // Se já JPEG.
            fb_len = fb->len; // Define comprimento.
            res = httpd_resp_send(req, (const char *)fb->buf, fb->len); //
Envia resposta.
        } else { // Se não JPEG.
            jpg_chunking_t jchunk = {req, 0}; // Declara estrutura.
            res = frame2jpg_cb(fb, 80, jpg_encode_stream, &jchunk) ?
ESP_OK : ESP_FAIL; // Converte.
            httpd_resp_send_chunk(req, NULL, 0); // Envia chunk final.

```

```

fb_len = jchunk.len; // Define comprimento.
}
esp_camera_fb_return(fb); // Retorna frame.
int64_t fr_end = esp_timer_get_time(); // Tempo final.
Serial.printf("JPG:      %uB      %ums\n",      (uint32_t)(fb_len),
(uint32_t)((fr_end - fr_start) / 1000)); // Log.
return res; // Retorna resultado.
}
dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, fb-
>width, fb->height, 3); // Aloca matriz de
imagem.
if (!image_matrix) { // Verifica falha.
esp_camera_fb_return(fb);
Serial.println("dl_matrix3du_alloc failed");
httpd_resp_send_500(req);
return ESP_FAIL;
}
out_buf = image_matrix->item; // Define buffer de saída.
out_len = fb->width * fb->height * 3; // Define comprimento.
out_width = fb->width; // Define largura.
out_height = fb->height; // Define altura.
s = fmt2rgb888(fb->buf, fb->len, fb->format, out_buf); //
Converte para RGB.
esp_camera_fb_return(fb); // Retorna frame.
if (!s) { // Verifica falha.
dl_matrix3du_free(image_matrix);
Serial.println("to rgb888 failed");
httpd_resp_send_500(req);
return ESP_FAIL;
}
box_array_t *net_boxes = face_detect(image_matrix,
&mtmn_config); // Detecta faces.
if (net_boxes) { // Se faces detectadas.
detected = true; // Seta flag.
if (recognition_enabled) { // Se reconhecimento habilitado.
face_id = run_face_recognition(image_matrix, net_boxes); //
Reconhece.
}
draw_face_boxes(image_matrix, net_boxes, face_id); //
Desenha caixas.
free(net_boxes->score); // Libera memória.
free(net_boxes->box);

```

```

free(net_boxes->landmark);
free(net_boxes);
}
jpg_chunking_t jchunk = {req, 0}; // Declara estrutura.
s = fmt2jpg_cb(out_buf, out_len, out_width, out_height,
PIXFORMAT_RGB888, 90,
jpg_encode_stream, &jchunk); // Converte para JPEG.
dl_matrix3du_free(image_matrix); // Libera matriz.
if (!s) { // Verifica falha.
Serial.println("JPEG compression failed");
return ESP_FAIL;
}
int64_t fr_end = esp_timer_get_time(); // Tempo final.
Serial.printf("FACE: %uB %ums %s%d\n", (uint32_t)(jchunk.len),
(uint32_t)((fr_end - fr_start) / 1000),
detected ? "DETECTED " : "", face_id); // Log.
return res; // Retorna resultado.
}
static esp_err_t stream_handler(httpd_req_t *req) { // Handler
para streaming.
camera_fb_t * fb = NULL; // Declara buffer de frame.
esp_err_t res = ESP_OK; // Declara resultado.
size_t _jpg_buf_len = 0; // Declara comprimento do buffer
JPEG.
uint8_t * _jpg_buf = NULL; // Declara buffer JPEG.
char * part_buf[64]; // Declara buffer para partes.
dl_matrix3du_t *image_matrix = NULL; // Declara matriz de
imagem.
bool detected = false; // Declara flag de detecção.
int face_id = 0;
const uint8_t index_ov2640_html_gz[] = { // Declara um array
constante de bytes (uint8_t) contendo o
HTML comprimido em gzip. Esse array é usado pelo servidor HTTP
para servir a página web ao
navegador.
0x1F, 0x8B, 0x08, 0x08, 0x50, 0x5C, 0xAE, 0x5C, 0x00, 0x03,
0x69, 0x6E, 0x64, 0x65, 0x78, 0x5F, //
Bytes iniciais do arquivo gzip, incluindo cabeçalho de compressão
(0x1F 0x8B indica gzip).
0x6F, 0x76, 0x32, 0x36, 0x34, 0x30, 0x2E, 0x68, 0x74, 0x6D,
0x6C, 0x00, 0xE5, 0x5D, 0x7B, 0x73, //

```

Continuação dos dados comprimidos, representando o conteúdo HTML codificado.

```
0xD3, 0xC6, 0x16, 0xFF, 0x9F, 0x4F, 0x21, 0x04, 0x25, 0xF6,  
0x34, 0x76, 0x6C, 0xC7, 0x84, 0xE0, //
```

Dados binários comprimidos; não são legíveis como texto, mas descomprimidos formam o HTML da interface.

```
0xDA, 0xE2, 0x42, 0x08, 0xD0, 0x19, 0x5E, 0x25, 0x2D, 0x74,  
0xA6, 0xD3, 0x81, 0xB5, 0xB4, 0xB2,
```

```
// Parte do stream comprimido, otimizado para transmissão  
eficiente via WiFi.
```

```
0x55, 0x64, 0xC9, 0x95, 0x56, 0x76, 0x52, 0x26, 0x9F, 0xE3,  
0x7E, 0xA0, 0xFB, 0xC5, 0xEE, 0xD9,
```

```
// Continuação dos bytes; a compressão reduz o tamanho de  
~4KB para economizar memória no  
ESP32.
```

```
0x87, 0xA4, 0x95, 0xBC, 0x7A, 0xD8, 0x26, 0x36, 0x97, 0xEB,  
0xCC, 0x14, 0xD9, 0xDA, 0x73, 0xF6,
```

```
// Dados comprimidos; ao descomprimir, revela elementos HTML  
como botões para detecção facial e  
streaming.
```

```
0x9C, 0xF3, 0x3B, 0xAF, 0x5D, 0x3D, 0x3A, 0xBC, 0x6D, 0xF9,  
0x26, 0xB9, 0x9A, 0x63, 0x6D, 0x4A,
```

```
// Parte do arquivo gzip; inclui scripts JavaScript para interação  
em tempo real com a câmera.
```

```
0x66, 0xAE, 0x71, 0x6B, 0xC8, 0xFF, 0xD1, 0xE0, 0x33, 0x9C,  
0x62, 0x64, 0xF1, 0x43, 0xF6, 0x75,
```

```
// Bytes comprimidos; o HTML inclui controles para resolução,  
brilho e ativação de reconhecimento  
facial.
```

```
0x86, 0x09, 0xD2, 0xCC, 0x29, 0x0A, 0x42, 0x4C, 0x46, 0x7A,  
0x44, 0xEC, 0xD6, 0xA9, 0x9E, 0x3F,
```

```
// Continuação; a interface permite enrollment de faces e  
visualização de stream.
```

```
0xED, 0xA1, 0x19, 0x1E, 0xE9, 0x0B, 0x07, 0x2F, 0xE7, 0x7E,  
0x40, 0x74, 0xCD, 0xF4, 0x3D, 0x82,
```

```
// Dados binários; descomprimidos, formam uma página  
responsiva para dispositivos móveis.
```

```
0x3D, 0x18, 0xBE, 0x74, 0x2C, 0x32, 0x1D, 0x59, 0x78, 0xE1,  
0x98, 0xB8, 0xC5, 0xBE, 0x1C, 0x3A,
```

```
// Parte do gzip; inclui estilos CSS para layout da interface.
```

```

0x9E, 0x43, 0x1C, 0xE4, 0xB6, 0x42, 0x13, 0xB9, 0x78, 0xD4,
0x95, 0x79, 0x11, 0x87, 0xB8, 0xD8, //
Bytes comprimidos; a página HTML é servida diretamente do
ESP32 sem arquivos externos.
0x38, 0xBF, 0x78, 0x7B, 0xDC, 0xD3, 0xDE, 0xBC, 0xEF, 0xF5,
0x4F, 0x3A, 0xC3, 0x23, 0xFE, 0x5B,
// Continuação do array; a compressão permite armazenar a
interface inteira em ~4KB.
0x3A, 0x26, 0x24, 0x57, 0xF2, 0x77, 0xFA, 0x19, 0xFB, 0xD6,
0x95, 0xF6, 0x25, 0xF3, 0x13, 0xFD, //
Dados gzip; inclui elementos interativos como botões para "Start
Stream" e "Face Detection".
0xD8, 0x20, 0x44, 0xCB, 0x46, 0x33, 0xC7, 0xBD, 0x1A, 0x68,
0x8F, 0x03, 0x98, 0xF3, 0xF0, 0x05,
// Parte comprimida; o HTML é otimizado para baixa latência
em redes WiFi.
0x76, 0x17, 0x98, 0x38, 0x26, 0x3A, 0x0C, 0x91, 0x17, 0xB6,
0x42, 0x1C, 0x38, 0xF6, 0x4F, 0x2B, //
Continuação; inclui validações JavaScript para comandos enviados
ao ESP32.
0x84, 0x63, 0x64, 0x7E, 0x9E, 0x04, 0x7E, 0xE4, 0x59, 0x03,
0xED, 0x4E, 0xF7, 0x94, 0xFE, 0xAD,
// Bytes do gzip; a interface permite ajustes em tempo real,
como qualidade de imagem.
0x0E, 0x32, 0x7D, 0xD7, 0x0F, 0xE0, 0xFC, 0xF9, 0x33, 0xFA,
0xB7, 0x7A, 0x9E, 0xCD, 0x1E, 0x3A,
// Dados comprimidos; descomprimidos, revelam formulários para
controle de parâmetros.
0xFF, 0xE0, 0x81, 0xD6, 0x3D, 0x99, 0x5F, 0x66, 0xCE, 0x5F,
0xDF, 0xCA, 0x7C, 0x9D, 0xF6, 0x8A,
// Parte do array; a compressão reduz overhead de memória no
ESP32.
0xA4, 0x17, 0xF4, 0xA7, 0xE5, 0xF4, 0x21, 0x36, 0x89, 0xE3,
0x7B, 0xED, 0x19, 0x72, 0x3C, 0x05,
// Continuação; inclui handlers para eventos como clique em
botões.
0x27, 0xCB, 0x09, 0xE7, 0x2E, 0x02, 0x1B, 0xD8, 0x2E, 0x2E,
0xE5, 0x73, 0x67, 0x86, 0xBD, 0xE8,
// Bytes gzip; o HTML é auto-contido, sem dependências
externas.
0xB0, 0x82, 0x1B, 0x65, 0xD2, 0xB2, 0x9C, 0x80, 0x8F, 0x1A,
0x50, 0x3B, 0x44, 0x33, 0xAF, 0x92,

```

```

// Dados comprimidos; permite controle remoto sem software
adicional.
0x6D, 0x99, 0x5C, 0x9E, 0xEF, 0x61, 0x85, 0x01, 0xE9, 0x44,
0xCB, 0x00, 0xCD, 0xE9, 0x00, 0xFA,
// Parte do gzip; inclui feedback visual para status de conexão.
0xEF, 0xEA, 0x90, 0x99, 0xE3, 0x71, 0xA7, 0x1A, 0x68, 0xC7,
0xFD, 0xCE, 0xFC, 0xB2, 0x02, 0xCA,
// Continuação; a interface é responsiva para diferentes
resoluções de tela.
0xE3, 0x13, 0xFA, 0xB7, 0x3A, 0x68, 0x8E, 0x2C, 0xCB, 0xF1,
0x26, 0x03, 0xED, 0x54, 0xC9, 0xC2,
// Bytes comprimidos; otimizada para baixa potência do ESP32.
0x0F, 0x2C, 0x1C, 0xB4, 0x02, 0x64, 0x39, 0x51, 0x38, 0xD0,
0xFA, 0xAA, 0x31, 0x33, 0x14, 0x4C,
// Parte do array; inclui validações para evitar comandos
inválidos.
0x40, 0x16, 0xE2, 0x83, 0xB0, 0xAD, 0xAE, 0x52, 0x12, 0x31,
0x24, 0x70, 0x26, 0x53, 0x02, 0x90, //
Dados gzip; a página HTML é servida com cabeçalho gzip para
eficiência.
0xAE, 0x8C, 0xC9, 0x1B, 0x4D, 0x84, 0x50, 0x15, 0x9E, 0xA5,
0x76, 0x53, 0x5B, 0x0D, 0xB9, 0xCE,
// Continuação; permite monitoramento em tempo real do
sistema de reconhecimento.
0xC4, 0x6B, 0x39, 0x04, 0xCF, 0x40, 0x9D, 0x90, 0x04, 0x98,
0x98, 0xD3, 0x32, 0x51, 0x6C, 0x67, //
Bytes comprimidos; inclui elementos para exibir caixas de
detecção facial.
0x12, 0x05, 0x58, 0x21, 0x48, 0x62, 0xB7, 0x12, 0x85, 0xE1,
0xE4, 0xEA, 0xA9, 0xD6, 0x12, 0x8F, //
Parte do gzip; a interface é acessível via IP do ESP32.
0x3F, 0x3B, 0xA4, 0x25, 0x6C, 0x32, 0xC6, 0xB6, 0x1F, 0x60,
0xE5, 0xC8, 0x78, 0x84, 0xEB, 0x9B,
// Dados binários; descomprimidos, formam uma UI intuitiva para
usuários finais.
0x9F, 0x5B, 0x21, 0x41, 0x01, 0xA9, 0xC3, 0x10, 0xD9, 0x04,
0x07, 0xD5, 0xFC, 0x30, 0xF5, 0x8A,
// Continuação; inclui scripts para atualizar o stream
automaticamente.
0x6A, 0x6E, 0xC5, 0xD3, 0x8A, 0x01, 0x8E, 0xE7, 0x3A, 0x1E,
0xAE, 0x2F, 0x5E, 0xD1, 0xBC, 0x59,

```

```

// Parte comprimida; permite ajustes de câmera sem
reinicialização.
0x76, 0x7C, 0x54, 0x0D, 0x60, 0x9C, 0xD9, 0xA4, 0xCC, 0x4B,
0x98, 0xAE, 0xAB, 0x93, 0x89, 0xB8,
// Bytes gzip; a compressão é essencial para sistemas com
memória limitada.
0xE9, 0x76, 0x3A, 0x3F, 0xAC, 0x9E, 0x
#if defined(CAMERA_MODEL_WROVER_KIT) // Verifica se o modelo
WROVER_KIT está definido;
usado para câmeras com PSRAM integrada, otimizado para
protótipos.
#define PWDN_GPIO_NUM -1 // Pino para power down (desligar
câmera); -1 indica não usado, pois
o WROVER_KIT gerencia energia internamente.
#define RESET_GPIO_NUM -1 // Pino para reset da câmera; -1
indica não usado, confiando no reset
automático do módulo.
#define XCLK_GPIO_NUM 21 // Pino para clock externo (XCLK),
gerando sinal de temporização
para o sensor OV2640.
#define SIOD_GPIO_NUM 26 // Pino SDA para comunicação I2C
(dados), usado para configurar o
sensor da câmera.
#define SIOC_GPIO_NUM 27 // Pino SCL para comunicação I2C
(clock), sincronizando comandos
de configuração.
#define Y9_GPIO_NUM 35 // Pino para bit de dados Y9 do
barramento paralelo da câmera.
#define Y8_GPIO_NUM 34 // Pino para bit de dados Y8.
#define Y7_GPIO_NUM 39 // Pino para bit de dados Y7.
#define Y6_GPIO_NUM 36 // Pino para bit de dados Y6.
#define Y5_GPIO_NUM 19 // Pino para bit de dados Y5.
#define Y4_GPIO_NUM 18 // Pino para bit de dados Y4.
#define Y3_GPIO_NUM 5 // Pino para bit de dados Y3.
#define Y2_GPIO_NUM 4 // Pino para bit de dados Y2.
#define VSYNC_GPIO_NUM 25 // Pino para sincronização vertical
(VSYNC), indicando início de
quadro de imagem.
#define HREF_GPIO_NUM 23 // Pino para referência horizontal
(HREF), controlando linhas de pixels.
#define PCLK_GPIO_NUM 22 // Pino para clock de pixel (PCLK),
sincronizando captura de dados.

```

```

#elif defined(CAMERA_MODEL_ESP_EYE) // Verifica se o modelo
ESP_EYE está definido; usado
para módulos compactos com câmera integrada.
#define PWDN_GPIO_NUM -1 // Pino para power down; -1 indica
não usado, pois o ESP_EYE tem
controle de energia simplificado.
#define RESET_GPIO_NUM -1 // Pino para reset; -1 indica não
usado.
#define XCLK_GPIO_NUM 4 // Pino para clock externo, ajustado
para o layout do ESP_EYE.
#define SIOD_GPIO_NUM 18 // Pino SDA para I2C.
#define SIOC_GPIO_NUM 23 // Pino SCL para I2C.
#define Y9_GPIO_NUM 36 // Pino para bit de dados Y9, mapeado
conforme o hardware do
ESP_EYE.
#define Y8_GPIO_NUM 37 // Pino para bit de dados Y8.
#define Y7_GPIO_NUM 38 // Pino para bit de dados Y7.
#define Y6_GPIO_NUM 39 // Pino para bit de dados Y6.
#define Y5_GPIO_NUM 35 // Pino para bit de dados Y5.
#define Y4_GPIO_NUM 14 // Pino para bit de dados Y4.
#define Y3_GPIO_NUM 13 // Pino para bit de dados Y3.
#define Y2_GPIO_NUM 34 // Pino para bit de dados Y2.
#define VSYNC_GPIO_NUM 5 // Pino para VSYNC.
#define HREF_GPIO_NUM 27 // Pino para HREF.
#define PCLK_GPIO_NUM 25 // Pino para PCLK.
#elif defined(CAMERA_MODEL_M5STACK_PSRAM) // Verifica se o
modelo M5STACK_PSRAM está
definido; usado para stacks M5 com PSRAM extra.
#define PWDN_GPIO_NUM -1 // Pino para power down; -1 indica
não usado.
#define RESET_GPIO_NUM 15 // Pino para reset, usado para
reinicialização manual no M5STACK.
#define XCLK_GPIO_NUM 27 // Pino para clock externo.
#define SIOD_GPIO_NUM 25 // Pino SDA para I2C.
#define SIOC_GPIO_NUM 23 // Pino SCL para I2C.
#define Y9_GPIO_NUM 19 // Pino para bit de dados Y9, ajustado
para o layout do M5STACK.
#define Y8_GPIO_NUM 36 // Pino para bit de dados Y8.
#define Y7_GPIO_NUM 18 // Pino para bit de dados Y7.
#define Y6_GPIO_NUM 39 // Pino para bit de dados Y6.
#define Y5_GPIO_NUM 5 // Pino para bit de dados Y5.
#define Y4_GPIO_NUM 34 // Pino para bit de dados Y4.

```

```

#define Y3_GPIO_NUM 35 // Pino para bit de dados Y3.
#define Y2_GPIO_NUM 32 // Pino para bit de dados Y2.
#define VSYNC_GPIO_NUM 22 // Pino para VSYNC.
#define HREF_GPIO_NUM 26 // Pino para HREF.
#define PCLK_GPIO_NUM 21 // Pino para PCLK.
#elif defined(CAMERA_MODEL_M5STACK_WIDE) // Verifica se o
modelo M5STACK_WIDE está
definido; variante wide-angle do M5STACK.
#define PWDN_GPIO_NUM -1 // Pino para power down; -1 indica
não usado.
#define RESET_GPIO_NUM 15 // Pino para reset.
#define XCLK_GPIO_NUM 27 // Pino para clock externo.
#define SIOD_GPIO_NUM 22 // Pino SDA para I2C, ajustado para
wide-angle.
#define SIOC_GPIO_NUM 23 // Pino SCL para I2C.
#define Y9_GPIO_NUM 19 // Pino para bit de dados Y9.
#define Y8_GPIO_NUM 36 // Pino para bit de dados Y8.
#define Y7_GPIO_NUM 18 // Pino para bit de dados Y7.
#define Y6_GPIO_NUM 39 // Pino para bit de dados Y6.
#define Y5_GPIO_NUM 5 // Pino para bit de dados Y5.
#define Y4_GPIO_NUM 34 // Pino para bit de dados Y4.
#define Y3_GPIO_NUM 35 // Pino para bit de dados Y3.
#define Y2_GPIO_NUM 32 // Pino para bit de dados Y2.
#define VSYNC_GPIO_NUM 25 // Pino para VSYNC, diferente do
PSRAM para evitar conflitos.
#define HREF_GPIO_NUM 26 // Pino para HREF.
#define PCLK_GPIO_NUM 21 // Pino para PCLK.
#elif defined(CAMERA_MODEL_AI_THINKER) // Verifica se o
modelo AI_THINKER está definido; o
mais comum, usado em projetos DIY.
#define PWDN_GPIO_NUM 32 // Pino para power down, usado
para economizar energia no
AI_THINKER.
#define RESET_GPIO_NUM -1 // Pino para reset; -1 indica não
usado.
#define XCLK_GPIO_NUM 0 // Pino para clock externo, mapeado
para GPIO 0 no AI_THINKER.
#define SIOD_GPIO_NUM 26 // Pino SDA para I2C.
#define SIOC_GPIO_NUM 27 // Pino SCL para I2C.
#define Y9_GPIO_NUM 35 // Pino para bit de dados Y9.
#define Y8_GPIO_NUM 34 // Pino para bit de dados Y8.
#define Y7_GPIO_NUM 39 // Pino para bit de dados Y7.

```

```
#define Y6_GPIO_NUM 36 // Pino para bit de dados Y6.
#define Y5_GPIO_NUM 21 // Pino para bit de dados Y5.
#define Y4_GPIO_NUM 19 // Pino para bit de dados Y4.
#define Y3_GPIO_NUM 18 // Pino para bit de dados Y3.
#define Y2_GPIO_NUM 5 // Pino para bit de dados Y2.
#define VSYNC_GPIO_NUM 25 // Pino para VSYNC.
#define HREF_GPIO_NUM 23 // Pino para HREF.
#define PCLK_GPIO_NUM 22 // Pino para PCLK.
#else // Caso nenhum modelo seja definido, gera erro de
compilação.
#error "Camera model not selected" // Mensagem de erro para
forçar seleção de modelo, evitando
configurações incorretas.
#endif // Fim das diretivas condicionais.
```