

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JOSÉ BARRETO DOS SANTOS JUNIOR

**DESENVOLVIMENTO DE PROTÓTIPO PARA CONTROLE DE ACESSO
UTILIZANDO RECONHECIMENTO FACIAL**

CAMPO MOURÃO

2023

JOSÉ BARRETO DOS SANTOS JUNIOR

**DESENVOLVIMENTO DE PROTÓTIPO PARA CONTROLE DE ACESSO
UTILIZANDO RECONHECIMENTO FACIAL**

Development of a prototype for access control using facial recognition

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Eduardo Giometti Bertogna

CAMPO MOURÃO

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JOSÉ BARRETO DOS SANTOS JUNIOR

**DESENVOLVIMENTO DE PROTÓTIPO PARA CONTROLE DE ACESSO
UTILIZANDO RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Bacharel em Engenharia Eletrônica da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 24/novembro/2023

Marcio Rodrigues da Cunha
Doutor
Universidade Tecnológica Federal do Paraná

Osmar Tormena Junior
Doutor
Universidade Tecnológica Federal do Paraná

Eduardo Giometti Bertogna
Doutor
Universidade Tecnológica Federal do Paraná

**CAMPO MOURÃO
2023**

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Eduardo Bertonha, pelo apoio e tempo dedicado nesta trajetória e por sempre compartilhar conhecimento e experiência durante a elaboração do trabalho.

Aos meus colegas de sala, à minha esposa, e à meus pais, por sempre confiarem e acreditarem em mim, sem o apoio de vocês seria mais difícil superar este desafio.

RESUMO

A biometria, em sistemas de segurança, tem sido amplamente utilizada devido sua confiabilidade, sendo empregada para acessar contas bancárias, autenticação de estabelecimentos, pagamento em dispositivos móveis, controle de acesso, etc. Dentre as biometrias disponíveis, o reconhecimento facial se destaca pela sua praticidade e por ser uma das biometrias mais estudadas e utilizadas, podendo ser aplicada sem a necessidade de um contato físico. Diante desse contexto, o presente estudo teve como objetivo desenvolver um protótipo de reconhecimento facial para controle de acesso. Nesse processo, empregou-se uma placa ESP32-CAM para realizar o reconhecimento facial, em conjunto com um *display* LCD para simplificar a interação do usuário com o protótipo. Em termos gerais, o protótipo atendeu às expectativas, proporcionando a criação de um *hardware* compacto e de fácil utilização, permitindo a autenticação de usuários em tempo real.

Palavras-chave: biometria; processamento de imagem; esp32.

ABSTRACT

Biometrics, widely utilized in security systems, is known for its reliability and finds applications in accessing bank accounts, authenticating establishments, enabling mobile device payments, and managing access control, among other uses. Facial recognition, among the various biometric methods available, stands out due to its practicality and extensive study and usage. Moreover, it can be applied without the need for physical contact. In light of this context, the present study aims to develop a facial recognition prototype for access control. In this process, an ESP32-CAM board was employed to execute facial recognition, complemented by an LCD display to enhance user interaction with the prototype. In overall terms, the prototype has met expectations, delivering a compact and user-friendly hardware solution that facilitates real-time user authentication.

Keywords: biometry; image processing; esp32.

LISTA DE FIGURAS

Figura 1 – Método de Bertillon	17
Figura 2 – Modelo de impressão digital	18
Figura 3 – Reconhecimento baseado na localização dos olhos e nariz	19
Figura 4 – Escala de cinza	20
Figura 5 – Escala de cinza binarizada	20
Figura 6 – Fluxo de trabalho do MTNM	23
Figura 7 – Rostos em diferentes iluminações	24
Figura 8 – Pontos de referência a partir de características faciais	25
Figura 9 – Fluxograma do funcionamento do método <i>recognize face</i>	26
Figura 10 – ESP32-DevKitC.	26
Figura 11 – Fluxograma do <i>firmware</i>	27
Figura 12 – Diagrama de blocos do <i>hardware</i>	28
Figura 13 – ESP32-CAM	29
Figura 14 – GPIO disponíveis do ESP32-CAM	29
Figura 15 – <i>Display</i> LCD TFT 1.8"	30
Figura 16 – Módulo Relé	31
Figura 17 – Módulo Adaptador ESP32-CAM MB	32
Figura 18 – Criação e edição da tela do menu	34
Figura 19 – Menu KiCad	35
Figura 20 – Diagrama elétrico do protótipo	35
Figura 21 – Diagrama elétrico do ESP32-CAM	36
Figura 22 – Diagrama elétrico do <i>Display</i> TFT	36
Figura 23 – Diagrama elétrico dos botões	37
Figura 24 – Diagrama elétrico do módulo relé	37
Figura 25 – Projeto da placa de circuito impresso do protótipo	38
Figura 26 – Placa manufaturada	38
Figura 27 – Placa montada	39
Figura 28 – Telas de inicialização	39
Figura 29 – Fluxo de telas do login	39
Figura 30 – Fluxo de telas do cadastro	40

Figura 31 – Lista de arquivos salvos na memória <i>flash</i>	40
Figura 32 – Fluxo de telas para deletar usuário	41
Figura 33 – Interface física do protótipo	41

LISTA DE QUADROS

Quadro 1 – Portas de Entrada/Saída ESP32-CAM	30
--	----

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Função de detecção facial	32
Listagem 2 – Função de reconhecimento facial	33
Listagem 3 – Função de exibição do menu	34

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	13
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	14
1.2	Justificativa	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Biometria	16
2.2	História da biometria	16
2.3	Tipos de biometria	17
2.4	Reconhecimento facial	18
2.4.1	Processamento de imagem	19
2.4.2	Redes neurais	21
2.4.3	Redes neurais convolucionais	21
2.4.4	Classificador em cascata	21
2.4.5	Redes convolucionais em cascata de multitarefas	22
2.4.6	Redes móveis de multitarefas	22
2.5	Biometria facial para controle de acesso	24
2.6	Microcontrolador	26
3	METODOLOGIA	27
3.1	Microcontrolador ESP32-CAM	28
3.1.1	Pinos de Entrada/Saída (E/S)	28
3.2	Interface gráfica	29
3.3	Módulo de acionamento	30
3.4	Desenvolvimento do <i>software</i>	31
3.4.1	PlatformIO	31
3.4.2	Biblioteca ESP-DL	32
3.4.3	Fluxo de telas	33
3.5	Desenvolvimento do protótipo	34
4	RESULTADOS E DISCUSSÃO	38
4.1	Protótipo <i>hardware</i>	38

4.2	Interface gráfica e física	39
4.3	Melhorias	42
5	CONCLUSÃO	43
	REFERÊNCIAS	44
	APÊNDICE A CÓDIGO PRINCIPAL	48
	APÊNDICE B ESQUEMA ELÉTRICO ESP32-CAM	52

1 INTRODUÇÃO

Desde o nascimento, os seres humanos desenvolvem habilidades de reconhecimento e identificação de objetos. Logo na primeira semana de vida, os bebês estabelecem rapidamente reconhecimentos individuais, discriminando e demonstrando preferência pela face, voz e odor de sua própria mãe (Vieira; Oliva, 2017, p. 20).

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, 2009). Como, por exemplo, a face, retina, íris, impressões digitais, geometria da mão, etc. A biometria se tornou uma nova área de estudo a partir do antropologista francês Alphonse Bertillon, em 1890, quando utilizou conceitos de biometria para a identificação de criminosos (Galimberti, 2018, p. 15).

Dentre as tecnologias atuais de segurança, a biometria tem sido amplamente utilizada, seja para acessar contas bancárias, aplicativos e até controlar o acesso a locais públicos e privados. Atualmente, o reconhecimento facial é uma das biometrias mais estudadas, pois além da praticidade, é considerada uma das formas mais seguras de identificação (Zhao; Chellappa; Phillips, 2003, p. 55).

Embora a identificação facial seja uma tarefa simples para os seres humanos, representa um desafio considerável para os computadores. Isso se deve em parte às restrições impostas pelo sistema biométrico facial, que abrangem o controle da iluminação e dos ângulos das imagens utilizadas. Além disso, várias variáveis estéticas, como barba, cabelo, uso de óculos e bonés, problemas na lente da câmera e até mesmo a possibilidade de inserção de dados incorretos, podem ocasionar falhas no processo de reconhecimento (Cavalcanti, 2005, p. 31).

Assim, aprimorar a precisão de um sistema biométrico requer atenção durante o desenvolvimento, com foco na minimização de falsos positivos e falsos negativos no reconhecimento facial. Para superar esse desafio, é essencial encontrar uma abordagem que seja mais adequada ao sistema de autenticação por imagem. Entre as várias abordagens disponíveis, é importante avaliar a taxa de identificações incorretas e a taxa de casos não detectados (Viola; Jones, 2004, p. 8).

Diante disso, a proposta deste trabalho foi desenvolver um dispositivo que emprega o reconhecimento facial para controle de acesso, acompanhado por uma interface gráfica e física intuitiva e fácil de usar.

1.1 Objetivos

Nesta seção serão apresentados os objetivos deste trabalho e as etapas necessárias para o desenvolvimento do protótipo. Na qual, além da implementação do *hardware*, também foram necessárias algumas etapas para a elaboração do *software*, tendo como finalidade, obter

um protótipo de controle de acesso por reconhecimento facial.

1.1.1 Objetivo geral

Desenvolver um protótipo para controle de acesso por meio de reconhecimento facial. Para isso, foram utilizados algoritmos de processamento de imagens e aprendizagem profunda no microcontrolador ESP32-CAM.

1.1.2 Objetivos específicos

Para atingir o objetivo geral, algumas etapas essenciais foram realizadas:

- Desenvolver o *hardware* para aquisição de imagens, considerando a luminosidade local e a qualidade da câmera;
- Implementar um código otimizado e organizado;
- Criar uma interface gráfica e física que permita aos usuários interagir e utilizar de maneira intuitiva e simples;
- Desenvolver o módulo de controle de acesso, que permitirá ao administrador gerenciar e cadastrar novos usuários, concedendo ou limitando o acesso conforme as necessidades.

1.2 Justificativa

Os sistemas de reconhecimento facial foram uma grande solução durante a retomada das atividades presenciais após a pandemia do coronavírus, ajudando empresas a promoverem uma maior segurança física, como também segurança sanitária, evitando contaminações e agilizando os processos. Ao contrário dos sistemas manuais, onde normalmente geram atrasos e demandam contato físico (TERRA, 2020).

No cotidiano, uma das formas mais comuns de garantir a segurança é por meio de chaves físicas, utilizadas tanto em fechaduras mecânicas quanto em fechaduras eletrônicas, onde essas chaves podem assumir a forma de dispositivos metálicos ou tags de proximidade. No entanto, apesar de sua praticidade, essas chaves apresentam vulnerabilidades relacionadas à possibilidade de fraudes, seja através de cópias não autorizadas ou do uso de técnicas como o "Bump Key", uma ameaça frequente em diversos tipos de chaves. Um exemplo atual é o caso do *Flipper Zero*, um dispositivo projetado para clonar frequências de NFC, Bluetooth e RFID. De acordo com a Anatel (Agência Nacional de Telecomunicações), esse aparelho possui um "potencial lesivo" aos usuários, permitindo a reprodução não autorizada de controles remotos de portões eletrônicos, sistemas de segurança residencial, chaves de carro, entre outros (CA-

NALTECH, 2023). Essa capacidade de clonagem destaca a necessidade de abordagens mais seguras e avançadas no desenvolvimento de tecnologias de acesso e segurança.

Quanto ao reconhecimento facial, um aspecto a ser considerado é a automação de processos. Embora a redução de custos seja um diferencial destacado por muitos gestores ao abordar a automação, é igualmente crucial ponderar sobre o potencial dessa tecnologia em proporcionar benefícios mais significativos para toda a cadeia produtiva. Isso implica não apenas em otimizar a eficiência operacional, mas também em maximizar as recompensas globais para os diversos setores envolvidos (TIINSIDE, 2023).

Por último, destaca-se a praticidade desse sistema, uma vez que elimina a necessidade de memorizarem senhas ou carregarem chaves consigo. Esse fator tem um impacto positivo de forma geral, mas vem sendo especialmente benéfico para grupos de pessoas com Déficit de Atenção. Ao considerar que indivíduos com Déficit de Atenção muitas vezes enfrentam dificuldades em lembrar-se de compromissos, memorizar senhas ou extraviam chaves (sejam da casa ou do carro) com facilidade (ABDA, 2023), desta forma, o sistema de reconhecimento facial surge como uma solução que pode oferecer suporte a essas pessoas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns conceitos importantes para o entendimento do trabalho, sendo abordados assuntos como biometria, processamento de imagens, aprendizado profundo e reconhecimento facial.

2.1 Biometria

Com o avanço da tecnologia, hoje é possível realizar transações e pagamentos de qualquer lugar, ou até mesmo, sem sair de casa, apenas com o uso de um dispositivo conectado a internet. Entretanto, também tornaram-se indispensáveis o uso de mecanismos de segurança, principalmente os que são capazes de identificar e comprovar quem realmente está utilizando esses serviços.

Por mais que existam outros processos de identificação, como por exemplo, cartões magnéticos, senhas, tags, etc., atualmente o processo considerado mais seguro é o baseado em biometria. A biometria pode ser definida como o processo de identificação dos seres vivos e tem por intuito distinguir os indivíduos por meio de suas características únicas. É uma técnica que foi utilizada até mesmo pelos egípcios para o processo de identificação, baseando-se em características da aparência dos indivíduos, como cor dos olhos e cicatrizes (Galimberti, 2018, p. 15).

Os sistemas biométricos podem ser divididos em dois grupos distintos: 1) os métodos invasivos, que requerem a colaboração ativa do sujeito para a identificação; e 2) os métodos não invasivos, que podem ser aplicados sem que o indivíduo identificado tenha conhecimento disso. Entre os métodos, incluem-se técnicas amplamente conhecidas, como a biometria da impressão digital, do rosto, da íris, da assinatura, entre outras (Teixeira, 2011, p. 20)

2.2 História da biometria

Contrariando o senso comum, é importante notar que o conceito de biometria não é algo recente, o que realmente se tornou inédito foi a sua aplicação em sistemas computacionais. Na verdade, práticas de identificação com base em características físicas datam de muito tempo atrás. Um exemplo notório é o uso que os faraós do Egito faziam de características como cicatrizes, cor dos olhos, arcada dentária e outros detalhes para distinguir indivíduos (Boechat, 2008, p. 19-25).

Na Europa, por volta de 1890, o antropologista e delegado policial parisiense Alphonse Bertillon viu na biometria uma maneira de identificar criminosos e a transformou em uma nova área de estudo (Galimberti, 2018, p. 15). Bertillon, com um profundo interesse em pesquisas criminalísticas, começou a medir diversas partes do corpo humano, incluindo o diâmetro trans-

versal da cabeça, comprimento dos pés, dedos, antebraços, estatura, entre outros e, registrou essas medidas em cartões. Embora os resultados obtidos por Bertillon não tenham sido conclusivos, a ideia de relacionar medidas do corpo humano para a identificação continuou a ser usada por autoridades policiais em todo o mundo, sendo conhecida como "Bertillonage" (Boechat, 2008, p. 18).

Na Figura 1, podem ser observadas várias técnicas de medição desenvolvidas por meio do método de Bertillon, abrangendo aferições dos braços, pernas, cabeça e até mesmo das orelhas.



Fonte: Adaptado de (Galimberti, 2018, p. 16).

Na Suíça, com a introdução da fotografia, essa técnica começou a ser usada de maneira exclusiva nos procedimentos de identificação criminal. Na Índia, um indivíduo chamado William James Hersche, insatisfeito com a falta de cumprimento de contratos por parte dos comerciantes locais, passou a exigir que as assinaturas fossem acompanhadas pela impressão digital nos documentos (Boechat, 2008, p. 18).

Por fim, nas últimas décadas, novos sistemas biométricos começaram a surgir à medida que novas aplicações biométricas eram desenvolvidas e se tornavam uma realidade comercial.

2.3 Tipos de biometria

De acordo com Moraes (2010), os principais tipos de biometria são:

- Orelhas: usa a anatomia da orelha para identificar indivíduos.
- Termograma da face e das mãos: o padrão de calor emitido pelo corpo humano é uma característica de cada pessoa e pode ser captado por infravermelho. Sistemas baseados

em imagens termográficas não requerem contato ou cooperação individual. No entanto, a captura de imagem continua sendo um desafio em ambientes não controlados, pois é afetada por fontes de calor que possivelmente podem estar próximas ao indivíduo.

- Impressão digital: como pode ser observado na Figura 2, recurso mais comumente usado em credenciais automatizadas em grande escala. Sua popularidade se deve em parte a dispositivos de coleta de baixo custo e desempenho de processo razoável. Embora a impressão digital não se modifique naturalmente ao longo dos anos, ela é sensível aos fatores ambientais aos quais os indivíduos estão submetidos, o que pode levar à sua alteração e deterioração. Trabalhadores manuais, por exemplo, podem ver suas impressões digitais constantemente alteradas devido a cortes profundos ou outros cortes em seus dedos.

Figura 2 – Modelo de impressão digital



Fonte: Adaptado de (Boechat, 2008, p. 10).

- Íris: formada durante o desenvolvimento fetal, estabiliza-se durante os dois primeiros anos de vida. Sua textura é extremamente complexa e fornece informações a serem utilizadas no reconhecimento facial. Tem um baixo grau de impostura, pois é difícil até cirurgicamente alterar a textura da íris. Seu ponto fraco está em sua capacidade de recuperação, requer equipamentos caros e complexos, bem como cooperação individual.
- Voz: união de biometria comportamental e fisiológica. Ele não muda em curtos períodos de tempo, mas é afetado por fatores como um simples frio, estado emocional e ruído de fundo. Possui baixa exclusividade e não é recomendado para identificação em larga escala. O ponto forte é a capacidade de coleta e aceitabilidade, além do baixo custo dos coletores. Geralmente indicado para verificação de identidade em conversas.

2.4 Reconhecimento facial

Desde a infância, o ser humano adquire e desenvolve sua capacidade de reconhecer traços faciais, que é uma particularidade da visão e inevitável para relações sociais (Rouhani, 2019, p. 16).

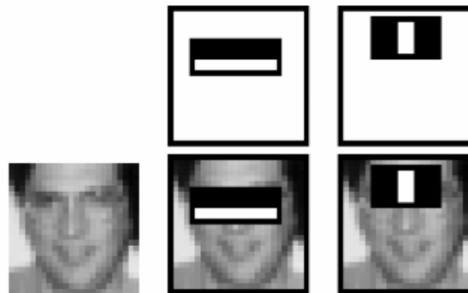
Existem estudos sobre automatização do reconhecimento facial desde os anos 60. Os projetos iniciais nessa área dependiam do administrador encontrar manualmente as características faciais nas imagens, só então o sistema calculava as distâncias entre elas e comparava suas dimensões normalizadas com as referenciadas.

Hoje o processo de reconhecimento facial pode ser descrito a partir de uma imagem ou vídeo estático, identificando um ou múltiplos indivíduos a partir de um banco de dados de rostos previamente cadastrados. Assim, existem três abordagens conhecidas para reconhecimento:

- Imagem a imagem: a amostra e a base de dados composta por imagens estáticas;
- Vídeo para vídeo: a amostra e o banco de dados que consiste em vídeos;
- Imagem para vídeo: o exemplo é um vídeo. O vídeo é comparado a um banco de dados de imagens estáticos.

Após a imagem ter sido lida e transformada, a mesma é duplicada e redimensionada proporcionalmente para uma altura fixa. A imagem original é mantida para ser utilizada posteriormente, em seguida, na cópia da imagem é realizado alguns processamentos para extrair suas características. Contudo, as abordagens mais populares usadas no problema de reconhecimento facial são baseadas na localização e análise de atributos faciais como olhos, nariz e boca (Figura 3).

Figura 3 – Reconhecimento baseado na localização dos olhos e nariz



Fonte: Adaptado de Viola e Jones (2004).

Ao comparar as informações extraídas com aquelas já conhecidas, é possível categorizar o objeto e determinar com grande precisão sua natureza ou identidade (Gonzalez; Woods, 2010, p. 409).

2.4.1 Processamento de imagem

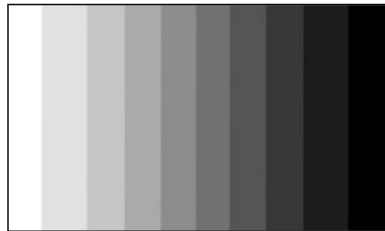
As técnicas de processamento de imagens começaram a surgir no final da década de 1960, para serem utilizadas no realce e restauração de imagens capturadas do espaço, como por exemplo, as imagens da missão Apollo. Logo em seguida, essa tecnologia começou a ser

empregada para processar imagens em diagnósticos médicos e, com o aumento do poder de processamento dos computadores, essas técnicas agora são empregadas nas mais diversas áreas de conhecimento (Gonzalez; Woods, 2010, p. 2).

Em processamento de imagens, um conceito bastante utilizado é a binarização de imagens, que consiste em duas classes distintas, o fundo e o objeto, esse processo serve para separar ambas as classes.

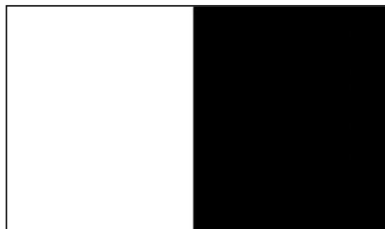
Sendo assim, a forma mais simples de processamento consiste na bipartição do histograma, dando valores iguais a 0 (branco) aos *pixels* que estiverem abaixo do valor de *threshold* (T) e iguais a 255 (preto) aos *pixels* que estiverem acima desse valor. Na Figura 4 é possível observar uma escala de tons de cinza e na Figura 5 verifica-se essa mesma escala pós-processamento, exemplificando o processo de binarização.

Figura 4 – Escala de cinza



Fonte: Autoria própria (2023).

Figura 5 – Escala de cinza binarizada



Fonte: Autoria própria (2023).

Especialmente durante o processo de reconhecimento de objetos, a segmentação é uma ferramenta indispensável para fins de análise e interpretação. A segmentação de uma imagem é um procedimento importante no que tange ao reconhecimento e extração objetos, uma vez que ela subdivide uma imagem em regiões que posteriormente serão ou não tidas como de interesse, o que pode variar muito de acordo com a aplicação (Gonzalez; Woods, 2010, p. 6).

Os principais algoritmos de segmentação de imagem têm suas bases em duas abordagens: descontinuidade e similaridade. A abordagem da descontinuidade envolve a divisão de uma imagem com base em mudanças abruptas de intensidade, como as que ocorrem nas bordas. Por outro lado, a técnica de similaridade segmenta a imagem com base na identificação de regiões que compartilham semelhanças de acordo com critérios predefinidos (Gonzalez; Woods, 2010, p. 6).

2.4.2 Redes neurais

As Redes Neurais Artificiais (ANN) representam modelos computacionais compostos por conjuntos de neurônios artificiais, desenvolvidos para identificar padrões em dados de entrada mediante um processo de treinamento prévio. Essas redes podem variar em complexidade, desde estruturas simples com um único neurônio até configurações mais sofisticadas com múltiplas camadas totalmente conectadas. O emprego de redes neurais apresenta um bom desempenho na solução de diversos problemas, destacando-se notadamente na área de classificação (Noe, 2021, p. 19).

A tarefa de classificação envolve o cálculo das probabilidades associadas a um determinado dado de entrada pertencente aos conjuntos de saídas possíveis para um dado problema. No contexto do aprendizado supervisionado, uma abordagem comum para gerar um modelo de classificação é a coleta experimental de características. Nesse processo, são apresentadas entradas (na camada de entrada) com seus valores de saída correspondentes (na camada de saída). A rede, então, busca representar as características comuns identificadas nos parâmetros treinados, localizados na camada oculta (Noe, 2021, p. 19).

2.4.3 Redes neurais convolucionais

Quando se aborda o reconhecimento de padrões em imagens, é comum que cada amostra contenha uma quantidade significativa de *pixels* irrelevantes para a classificação desejada. Além disso, devido às possíveis variações na iluminação, angulação e até distorções, uma abordagem de amostragem simples demandaria uma quantidade exorbitante de amostras, tornando-se impraticável para aplicações de aprendizado computacional. Por esses motivos, as Redes Neurais Convolucionais (CNN) oferecem metodologias e arquiteturas específicas para efetivamente reduzir o número de parâmetros que precisam ser ajustados pela rede (Noe, 2021, p. 22).

A CNN foi inicialmente apresentada pela arquitetura *LeNet*, porém apenas recebeu mais destaque posteriormente pela arquitetura *AlexNet*. Esta rede pode ser definida como uma rede neural que utiliza a operação matemática convolução no lugar da multiplicação geral da matriz em pelo menos uma de suas camadas (Lecun *et al.*, 1998, p. 6).

2.4.4 Classificador em cascata

O classificador em cascata possui uma certa semelhança com a técnica de *Haar-like*, que é essencialmente quadros retangulares escaláveis, usados para comparar e analisar como os *pixels* se relacionam entre si, essa técnica é conhecida como "*Haar-like features*". O classificador em cascata é chamado assim, pois combina vários classificadores fracos, ou seja,

sozinhos não podem classificar uma imagem, porém uma soma ponderada deles é capaz de formar um classificador mais forte, que é chamado de método de reforço. A principal vantagem deste método é sua baixa complexidade computacional e sua paralelização, sendo uma das técnicas mais utilizadas na indústria ou no meio científico para detecção de faces ou objetos (Viola; Jones, 2004, p. 2).

2.4.5 Redes convolucionais em cascata de multitarefas

Diferentemente das abordagens manuais, as Redes Neurais Convolucionais (CNNs) desenvolvem descritores automaticamente a partir das imagens de treino, sem fazer qualquer pré-suposição sobre padrões ou características específicas da imagem a ser analisada (Zafeiriou; Zhang; Zhang, 2015, p. 14). Embora seja verdade que o número de camadas está diretamente relacionado à qualidade das previsões da rede, é importante ressaltar que esse aumento na complexidade também resulta em um significativo incremento no custo computacional (Mukherjee *et al.*, 2017, p. 3).

O algoritmo *Multi-task Cascaded Convolutional Neural Networks* (MTCNN) é um modelo fundamentado em cascata de Redes Neurais Convolucionais (CNN), adotando uma estrutura sequencial composta por três redes neurais convolucionais. Essas redes são capazes de prever a localização de um rosto e seus pontos-chave em uma imagem (Zhang *et al.*, 2016, p. 1). Embora o modelo MTCNN alcance elevadas taxas de precisão, é importante notar que sua implementação pode exigir considerável poder computacional.

2.4.6 Redes móveis de multitarefas

O modelo utilizado neste projeto foi o MTMN (*Multi-Task Mobile Nets*), ou simplesmente Redes Móveis de Multitarefa. Esse modelo foi especificamente projetado para microcontroladores, apresentando uma operação eficiente que se baseia na arquitetura móvel *MobileNetV2* e tem como base o algoritmo MTCNN (ESPRESSIF SYSTEMS, 2019b).

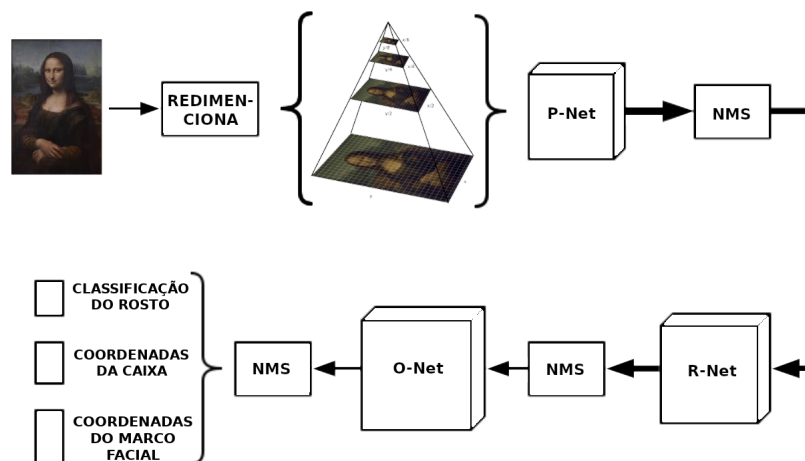
O *MobileNetV2* faz parte de uma nova geração de redes neurais convolucionais profundas voltadas para a visão computacional. Essa arquitetura permite a implementação em tempo real de aplicativos de classificação e detecção, destacando-se por sua capacidade de processamento leve (Luna *et al.*, 2022, p. 4).

Uma característica distintiva do MTMN é sua estrutura multitarefa de aprendizagem em cascata. Essa abordagem permite que o modelo resolva desafios encontrados em ambientes diversos. O MTMN é composto por três estágios de redes convolucionais profundas que preveem a localização de faces e pontos de referência, proporcionando uma detecção precisa que varia de informações gerais a detalhes específicos (Luna *et al.*, 2022, p. 4).

Conforme o fluxo de trabalho do MTMN, demonstrado na Figura 6, inicialmente uma

imagem é redimensionada em diferentes escalas para formar uma pirâmide e que servirá de entrada para os três estágios da rede. No primeiro estágio o P-Net (*Proposal Network*) utiliza uma rede convolucional sem as camadas densas e tem o propósito de obter quadros com vetor da caixa delimitadora, que são as coordenadas do quadrado onde engloba a face detectada. Após isso é executado o método NMS (*Non-Maximum Suppression*) para juntar os quadros que possuem mais sobreposição e com isso a saída será utilizada para alimentar o próximo estágio. O R-Net (*Refine Network*) recebe todas os quadros escolhidas pela P-Net, que agora possui camadas densas após as convoluções, essa etapa é responsável por diminuir o número de falsos quadros e calibrar o vetor da caixa delimitadora e também é executado o NMS como no estágio anterior. A última etapa, o O-Net (*Output Network*) é uma etapa similar a R-Net, porém foca em descrever a face em mais detalhes e com isso atualizar o NMS. O resultado do processamento sempre terá 3 valores principais (classificação do rosto, coordenadas da caixa delimitadora e as coordenadas das referências faciais) (Zhang *et al.*, 2016, p. 2-3).

Figura 6 – Fluxo de trabalho do MTNM



Fonte: Adaptado de Espressif Systems (2019b).

A MTNM é treinada a partir de conjuntos de amostras que abrangem imagens positivas e uma quantidade substancial de imagens negativas:

- Amostras positivas: Estas consistem em imagens que destacam o objeto de interesse, apresentando diversas representações do objeto em diferentes perspectivas, condições de iluminação, tamanhos, entre outros.
- Amostras negativas: Englobam imagens que não incluem o objeto de interesse.
- O número de imagens necessárias para o treinamento é influenciado por diversos fatores, como a qualidade das imagens, as características específicas do objeto de interesse e a capacidade de processamento disponível.

O MTMN se beneficia de redes neurais pré-treinadas, que foram desenvolvidas com base em um vasto banco de dados contendo mais de um milhão de imagens do *ImageNet*. Essas redes aprenderam representações de recursos de mais de 1.000 categorias de objetos, fornecendo uma base sólida para a detecção de rostos (Luna *et al.*, 2022, p. 5).

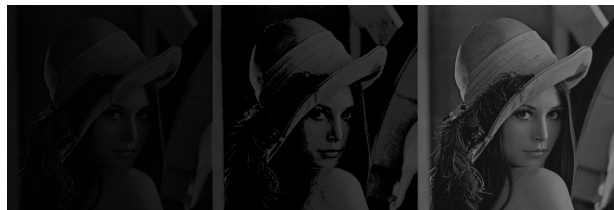
2.5 Biometria facial para controle de acesso

Os sistemas de identificação baseados em biometria são essencialmente sistemas de reconhecimento que, dadas informações biométricas, são capazes de distinguir padrões e classificá-los em diferentes classes ou categorias (Moraes, 2010, p. 22). Ainda de acordo com o autor, algumas das principais características anatômicas, fisiológicas e comportamentais utilizadas em sistemas biométricos incluem impressão digital, impressão da mão, aparência facial, temperatura da face, retina, voz, assinatura, entre outras.

A biometria facial representa a técnica biométrica mais adotada atualmente. Embora o reconhecimento de rostos seja uma tarefa simples para as pessoas, ela se revela notavelmente complexa para os computadores. Mesmo em condições desafiadoras, o cérebro humano é capaz de identificar com precisão uma pessoa com base em sua imagem facial, apesar das variações na iluminação, distorções ou deformações.

Embora os sistemas biométricos faciais apresentem um desempenho aceitável em ambientes comerciais, ainda estão sujeitos a restrições relacionadas ao ambiente, como variações na iluminação (Figura 7) e ângulos das imagens. Segundo Cavalcanti (2005, p. 31), alterações estéticas, como cabelo e barba, uso de acessórios, como óculos e bonés, são fatores que aumentam as chances de falha no processo de reconhecimento facial.

Figura 7 – Rostos em diferentes iluminações



Fonte: Adaptado de OpenCV (2023b).

Para utilizar a face em sistemas biométricos é preciso seguir três etapas fundamentais. São elas:

- Detecção facial: responsável por definir e localizar uma ou mais faces;
- Extração de características: esta fase é responsável por remover o excesso de informações que rodeiam as faces detectadas, assim como selecionar as melhores características para serem utilizadas na próxima etapa;

- Reconhecimento facial: esta fase compara as características selecionadas pela fase anterior com outras previamente cadastradas em um banco de dados, sendo responsável por encontrar um registro que se assemelhe ao que precisa ser identificado.

Essas etapas são importantes na avaliação da imagem, eliminando informações redundantes e irrelevantes. Por exemplo, se o algoritmo identifica uma ou mais faces na imagem, essas são extraídas da imagem original para análise individual. É importante notar que, quando a entrada do sistema é uma sequência de vídeo, a dimensão temporal também é considerada, tornando necessário que o algoritmo utilize mais recursos de processamento, para permitir a detecção em tempo real (Boechat, 2008, p. 44).

Os algoritmos de reconhecimento facial têm a capacidade de identificar indivíduos com base em características específicas, como o tamanho dos olhos, nariz e boca. Essas características são usadas para localizar imagens correspondentes que se assemelham à imagem da face capturada. Além disso, os algoritmos desse tipo normalmente armazenam informações relevantes apenas da região da imagem que contém as características de interesse, conforme pode ser observado na Figura 8. De acordo com (Viola; Jones, 1993, p. 6), um dos primeiros sistemas de reconhecimento facial baseava-se em um modelo de técnicas aplicadas a um conjunto de características faciais, resultando em uma representação facial compacta e correspondente.

Figura 8 – Pontos de referência a partir de características faciais

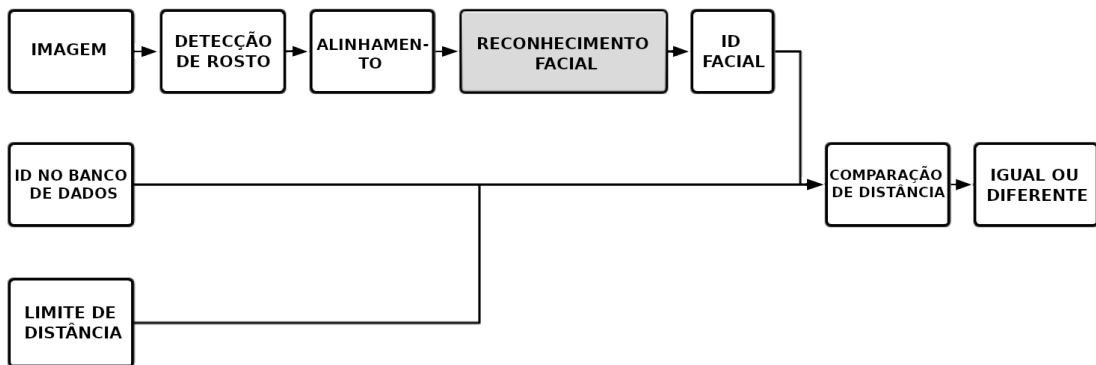


Fonte: Adaptado de OpenCV (2023a).

O modelo de reconhecimento utilizado neste trabalho foi o FRMN (*Face Recognition Mobile Nets* ou Redes Móveis de Reconhecimento Facial), que também se baseia na arquitetura *MobileNetV2* e emprega o algoritmo *ArcFace*. Para otimizar a complexidade computacional, as imagens foram treinadas em dimensões reduzidas (56x56). Na Figura 9, é possível visualizar todas as etapas do algoritmo no processo de reconhecimento facial.

Esse procedimento não apenas proporciona uma eficiente capacidade de reconhecimento facial, mas também viabiliza uma comparação precisa entre o rosto identificado e as informações previamente cadastradas. Tal abordagem contribui significativamente para soluções de alta qualidade em sistemas de identificação e autenticação. Contudo, a execução desse

Figura 9 – Fluxograma do funcionamento do método *recognize face*



Fonte: Adaptado de Espressif Systems (2019a).

procedimento demanda o uso de um dispositivo ou microcontrolador com a capacidade de processar esses dados em tempo real.

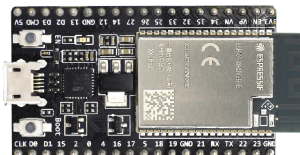
2.6 Microcontrolador

Microcontrolador é um computador em um único *chip* que incorpora: processador, memória, periféricos de entrada e saída, temporizadores e dispositivos de comunicação serial. Eles surgiram como uma evolução natural dos circuitos digitais devido à crescente complexidade. Chegou um ponto em que foi mais prático e econômico substituir a lógica das portas digitais por um conjunto de processador e *software* (Penido; Trindade, 2013, p. 15).

O primeiro microcontrolador, o "8048", foi lançado pela Intel em 1977 e evoluiu para a família "8051". Esses *chips* são programados em linguagem *Assembly* e possuem um conjunto de instruções poderoso (Penido; Trindade, 2013, p.16).

Atualmente, quando se trata de microcontrolador, uma boa opção é o ESP32 (Figura 10). Mesmo não sendo o modelo mais potente, nem o mais compacto, ainda assim, possui um ótimo custo-benefício ao considerar sua simplicidade e poder de processamento (ESPRESSIF SYSTEMS, 2022a).

Figura 10 – ESP32-DevKitC.



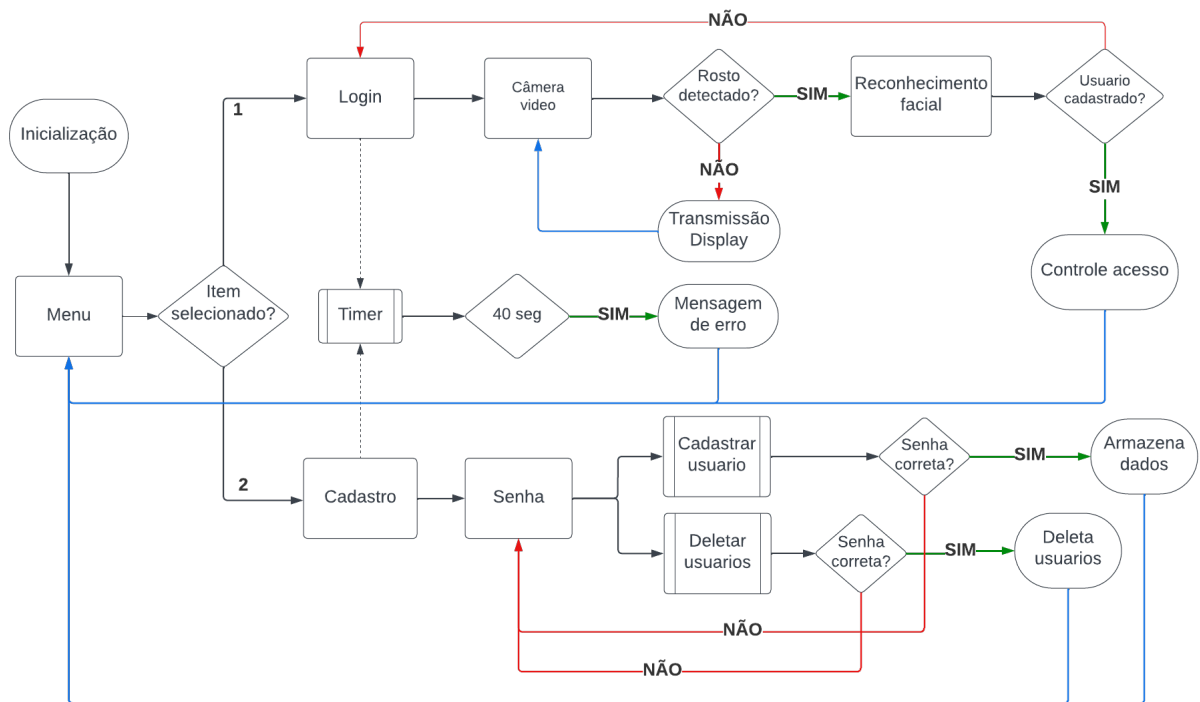
Fonte: Adaptado de Espressif Systems (2022c).

3 METODOLOGIA

O projeto foi estruturado em duas etapas principais: o desenvolvimento da infraestrutura de *hardware* e a criação do *software*. Além da criação do protótipo físico, foi elaborado um código com o objetivo de alcançar desempenho otimizado, atendendo a todos os requisitos do projeto. Isso incluiu desde a fase inicial de inicialização até a implementação de uma interface gráfica para os usuários, a entrada de dados, o reconhecimento facial, o cadastro e exclusão de usuários, por fim, o módulo de acionamento.

O fluxograma apresentado na Figura 11 oferece uma visão simplificada do funcionamento do *software* do projeto. O sistema de controle de acesso por biometria facial opera em dois ciclos principais: o primeiro é dedicado à autenticação do usuário, enquanto o segundo é responsável pelo cadastro de usuários. Além disso, há um ciclo obrigatório com um temporizador em execução em segundo plano. Esse ciclo é ativado automaticamente sempre que um dos ciclos principais é iniciado, com o intuito de encerrar quaisquer atividades pendentes e prevenir possíveis *loops* dentro do sistema.

Figura 11 – Fluxograma do *firmware*

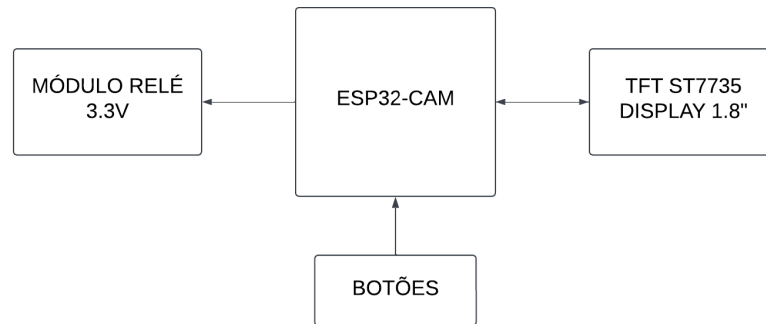


Fonte: Autoria própria (2023).

Para a execução desse programa, é necessário o uso de um *hardware* capaz de processá-lo. Portanto, a primeira etapa do projeto foi dedicada à construção do protótipo físico. E com o intuito de facilitar a compreensão desta parte do projeto, foi criado o diagrama apresentado na Figura 12. Conforme evidenciado, o ESP32-CAM é o módulo central, encar-

regado do processamento de dados e da coordenação das informações aos demais módulos. Para melhorar a interação com os usuários, é adicionado o módulo com botões e uma interface gráfica (*display*). Por fim, o módulo relé é responsável pelo controle de acesso, podendo acionar diferentes tipos de fechaduras elétricas.

Figura 12 – Diagrama de blocos do hardware



Fonte: Autoria própria (2023).

Como um dos objetivos do projeto é o desenvolvimento de um protótipo de baixo custo, a versão selecionada para essa finalidade é o ESP32-CAM, que se destaca por integrar um *chip* ESP32, uma câmera, uma entrada para cartão SD e LED de alto brilho.

3.1 Microcontrolador ESP32-CAM

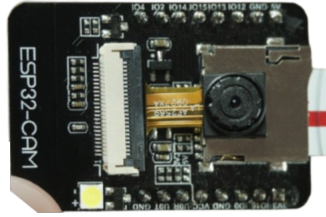
O ESP32-CAM (conforme ilustrado na Figura 13) é um microcontrolador desenvolvido pela empresa *Espressif Systems*® e que se destaca pelo seu custo benefício. Embora compacto, foi uma escolha ideal para este projeto devido à sua variedade de recursos. Seu *chip* é o ESP32-S, possui uma câmera integrada à placa, um processador *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/Saída (E/S), como pode ser visto no esquemático do Apêndice B.

Neste projeto, o ESP32-CAM desempenhou um papel central, sendo responsável pelo processamento de dados, análise das informações e controle dos demais componentes de *hardware*. Isso incluiu o gerenciamento de dispositivos adicionais e a coordenação das funções necessárias para a aplicação proposta.

3.1.1 Pinos de Entrada/Saída (E/S)

Os 16 pinos de Entrada/Saída (E/S) do ESP32-CAM (Figura 14) desempenham um papel crucial na versatilidade e funcionalidade deste microcontrolador. Esses pinos oferecem uma interface flexível para conectar o ESP32-CAM a uma ampla variedade de dispositivos e peri-

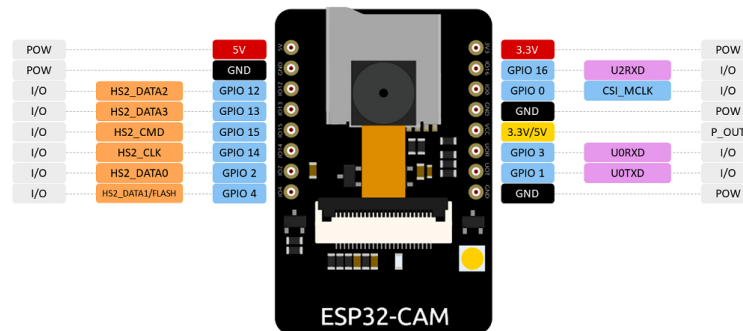
Figura 13 – ESP32-CAM



Fonte: Adaptado de Espressif Systems (2022b).

féricos externos, permitindo que ele interaja com o ambiente e execute tarefas específicas de acordo com as necessidades do projeto.

Figura 14 – GPIO disponíveis do ESP32-CAM



Fonte: Adaptado de Random Nerd Tutorials (2023).

Esses pinos são necessários para a comunicação com sensores, atuadores, dispositivos de armazenamento, *displays* e muitos outros componentes eletrônicos, tornando o ESP32-CAM adequado para inúmeras aplicações, desde sistemas de segurança e monitoramento, até projetos de automação residencial.

No quadro a seguir (Quadro 1), são detalhadas as funcionalidades dos 16 pinos de Entrada/Saída (E/S) disponíveis no ESP32-CAM. O entendimento destes pinos, possibilita aos desenvolvedores a flexibilidade de personalizar e expandir suas aplicações de acordo com suas necessidades específicas.

3.2 Interface gráfica

O *Display* LCD TFT (*Thin-Film Transistor*) de 1.8 polegadas, com resolução de 128x160 *pixels* e *driver* ST7735 (Figura 15) é um componente popular utilizado em uma variedade de aplicações eletrônicas, devido à sua capacidade de fornecer uma interface visual clara e interativa. Esse tipo de *display* é frequentemente empregado em projetos que requerem a exibição de informações, gráficos e interação direta com o usuário.

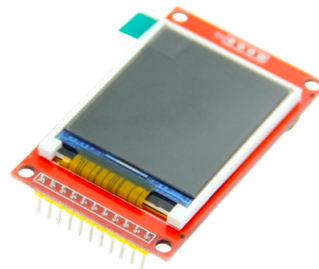
Neste contexto, o uso deste *display* no projeto tem o propósito de aprimorar a expe-

Quadro 1 – Portas de Entrada/Saída ESP32-CAM

Pinos	Descrição
5V	Pino de entrada para alimentação do circuito do ESP32.
3GND	3 pinos de aterramento, usado para referência de potencial zero.
GPIO12	Pino de propósito geral.
GPIO13	Pino de propósito geral.
GPIO15	Pino de propósito geral.
GPIO14	Pino de propósito geral.
GPIO2	Pino de propósito geral.
GPIO4	Pino de propósito geral e pode ser utilizado para acionar o Flash do ESP32.
3.3V	Pino de fornecimento de energia de 3,3V.
GPIO16	Este pino sempre fica em nível lógico alto e é utilizado para alimentar o circuito de PSRAM.
GPIO0	Pino de propósito geral, entretanto este pino é responsável pelo clock da câmera.
3.3V/5V	Pode fornecer energia de 3,3V ou 5V para outros dispositivos.
GPIO3	Pino de entrada de dados UART (RX) para comunicação serial.
GPIO1	Pino de saída de dados UART (TX) para comunicação serial.

Fonte: Autoria própria (2023).

riência do usuário, fornecendo uma representação visual clara da aplicação. Além disso, sua biblioteca de fácil utilização simplifica o processo de desenvolvimento do projeto, tornando-o mais acessível e eficaz.

Figura 15 – Display LCD TFT 1.8"

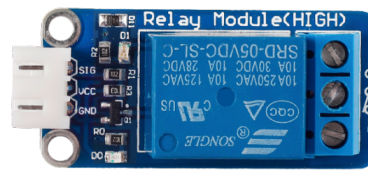
Fonte: Adaptado de LCD wiki (2023).

3.3 Módulo de acionamento

Um módulo relé, como apresentado na Figura 16, é um componente eletrônico utilizado em projetos que envolvem controle e automação de dispositivos elétricos. Ele desempenha um papel importante ao permitir o controle de circuitos de alta potência por meio de sinais de baixa potência.

Quando se trata de fechaduras eletrônicas, os módulos relé desempenham um papel crucial, facilitando o funcionamento e a segurança do sistema de controle de acesso. Geralmente, essas fechaduras incluem um mecanismo de trinco que pode ser controlado eletronicamente. O módulo relé é utilizado para controlar a ativação e desativação desse mecanismo,

Figura 16 – Módulo Relé



Fonte: Sunfounder (2022)

quando um usuário autorizado fornece uma credencial válida (como uma senha, cartão RFID ou impressão digital), o sistema eletrônico de controle gera um sinal de baixa potência para acionar o módulo relé. O relé fecha seu contato, permitindo a passagem de energia para o mecanismo de destravamento da fechadura, liberando assim o acesso.

3.4 Desenvolvimento do *software*

Para o desenvolvimento do *software*, primeiro foi selecionada uma plataforma capaz de programar e gravar códigos no ESP32. Das opções conhecidas eram o Arduino IDE e o PlatformIO. Além disso, o desenvolvimento exigiu um estudo aprofundado das bibliotecas disponíveis para atender aos requisitos do projeto, tais como o reconhecimento facial, transmissão em tempo real de imagens em um *display* LCD, implementação de um *timer* e a manipulação dos dados de entrada e saída.

3.4.1 PlatformIO

A plataforma escolhida para esse projeto foi o PlatformIO, que é uma alternativa para quem trabalha com dispositivos microcontrolados, como o ESP32. Projetado para simplificar o processo de desenvolvimento e programação de microcontroladores, o PlatformIO oferece diversos recursos e uma abordagem unificada que facilita o trabalho com diferentes plataformas de *hardware* e ambientes de desenvolvimento.

O PlatformIO destaca-se especialmente ao lidar com o ESP32, oferecendo aos desenvolvedores a capacidade de criar suas aplicações de forma eficiente, inclusive de maneira remota. Essa ferramenta foi projetada para funcionar com diversos editores de código, como *Visual Studio Code* (VSCode), Atom e CLion, permitindo que os desenvolvedores escolha a IDE que melhor se adapte às suas preferências. Um ponto forte da ferramenta é que a plataforma possui um gerenciador de bibliotecas interno que facilita a pesquisa, instalação e atualização direto com o código-fonte, contribuindo com a reutilização e organização do código. Além disso, outro ponto foi a simplicidade e a rapidez no processo de compilação e carregamento do código no ESP32.

Para estabelecer a comunicação com o PlatformIO, tanto para gravação quanto para a leitura de dados no ESP32-CAM, a utilização do adaptador ESP32-CAM MB (conforme repre-

sentado na Figura 17) revelou-se necessário. Este módulo foi muito importante, pois simplificou a conexão do ESP32-CAM com o computador.

Figura 17 – Módulo Adaptador ESP32-CAM MB



Fonte: Adaptado de Robo Builders (2023).

3.4.2 Biblioteca ESP-DL

A biblioteca mais importante nesse projeto foi a ESP-DL (2023), uma ferramenta que disponibiliza Interfaces de Programação de Aplicações (APIs) para tarefas como inferência de redes neurais (NN), processamento de imagens, operações matemáticas e alguns modelos de aprendizado profundo. Essa biblioteca contribuiu tanto na implementação da identificação de rostos, quanto na execução do processo de reconhecimento facial.

Dentre os recursos disponíveis do ESP-DL, se encontra o ESP-Face, componente que fornece funções de detecção e reconhecimento facial e também operações de rede neural. O método utilizado para detecção facial foi o *face detect* (como pode ser visto na Listagem 1) e que tem como base o modelo de redes neurais MTMN.

Listagem 1 – Função de detecção facial

```

1 #include "fd_forward.h"
2
3 box_array_t *run_face_detect(camera_fb_t *fb, dl_matrix3du_t *image_matrix)
4 {
5     fmt2rgb888(fb->buf, fb->len, fb->format, image_matrix->item);
6     return face_detect(image_matrix, &mtmn_config);
7 }

```

Fonte: Autoria própria (2023).

No contexto do reconhecimento facial, uma vez que um rosto humano tenha sido detectado por meio do procedimento mencionado anteriormente, é possível realizar uma verificação comparando-o com os rostos previamente cadastrados. A entrada para esse processo é a imagem original juntamente com os resultados obtidos na etapa de detecção facial.

O método de reconhecimento facial, denominado *recognize face* (como pode ser visto na Listagem 2) faz uso do modelo FRMN.

Listagem 2 – Função de reconhecimento facial

```

1 #include "fr_forward.h"
2 #define FACE_WIDTH 56
3 #define FACE_HEIGHT 56
4 #define MAX_NUMBER_USER 5
5
6 int run_face_recognition(face_id_list *id_list , dl_matrix3du_t *image_matrix
7   , box_array_t *net_boxes, int *user_number, bool *enroll_enabled)
8 {
9     dl_matrix3du_t *aligned_face = NULL;
10    int matched_id = -1;
11    aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
12    if (align_face(net_boxes, image_matrix, aligned_face) == ESP_OK)
13    {
14        if (*enroll_enabled)
15        {
16            if (MAX_NUMBER_USER == *user_number)
17            {
18                dl_matrix3du_free(aligned_face);
19                return matched_id;
20            }
21            else
22            {
23                int8_t number_file = enroll_face(id_list, aligned_face);
24            }
25        }
26        else
27        {
28            matched_id = recognize_face(id_list, aligned_face);
29        }
30    }
31    dl_matrix3du_free(aligned_face);
32    return matched_id;
33 }

```

Fonte: Autoria própria (2023).

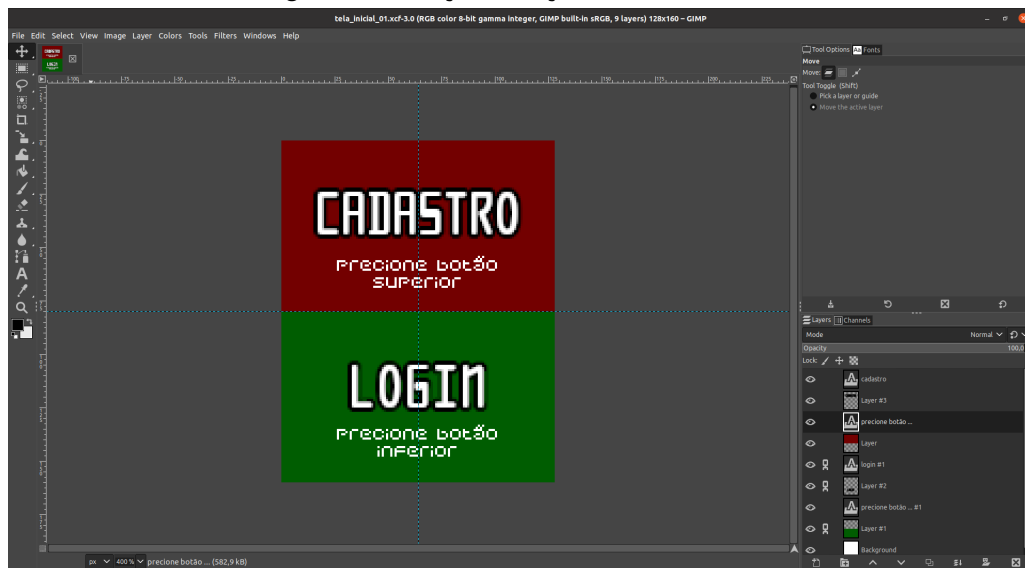
3.4.3 Fluxo de telas

Os recursos gráficos fornecidos pelo *display* LCD melhorou de forma substancial a usabilidade da aplicação. Nesse contexto, a elaboração do fluxo de telas se tornou indispensável para proporcionar aos usuários uma experiência eficiente. Todos os fluxos foram planejados e projetados com o objetivo de guiar o usuário por meio de diferentes interações e funcionalidades.

A criação dos fluxos de telas foi realizada por meio do *software* GIMP (*GNU Image Manipulation Program* ou Programa de Manipulação de Imagem do GNU). Trata-se de uma aplicação de código aberto dedicada à criação e edição de imagens, conforme ilustrado na

Figura 18.

Figura 18 – Criação e edição da tela do menu



E para exibir as imagens do fluxo de telas no *Display LCD*, optou-se pelo uso da biblioteca *TFT_eFEX*, conforme ilustrado no trecho de código apresentado na Listagem 3.

Listagem 3 – Função de exibição do menu

```

1 #include "TFT_eFEX.h"
2 TFT_eFEX fex = TFT_eFEX(&tft);
3
4 void display_menu()
5 {
6     fex.drawJpgFile(SPIFFS, "/_initial.jpeg", 0, 0);
7     delay(200); // Debounce
8 }

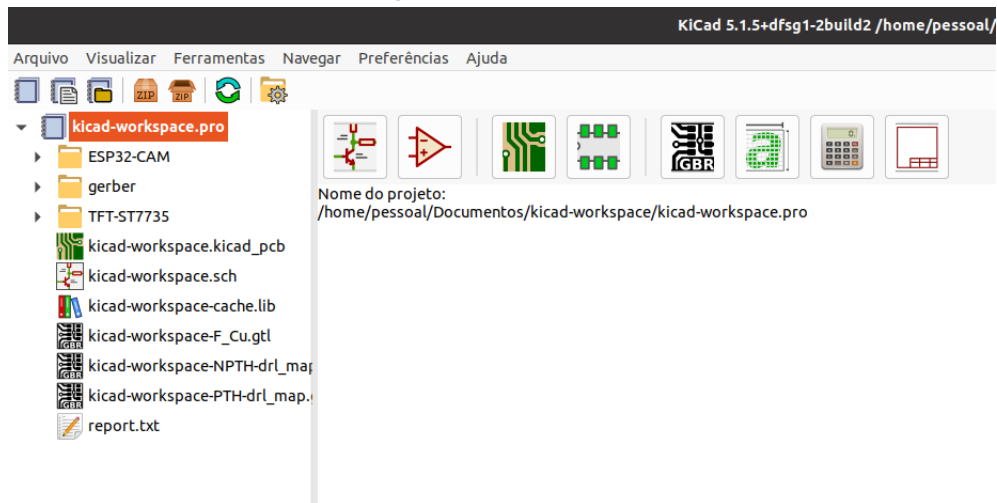
```

Fonte: Autoria própria (2023).

3.5 Desenvolvimento do protótipo

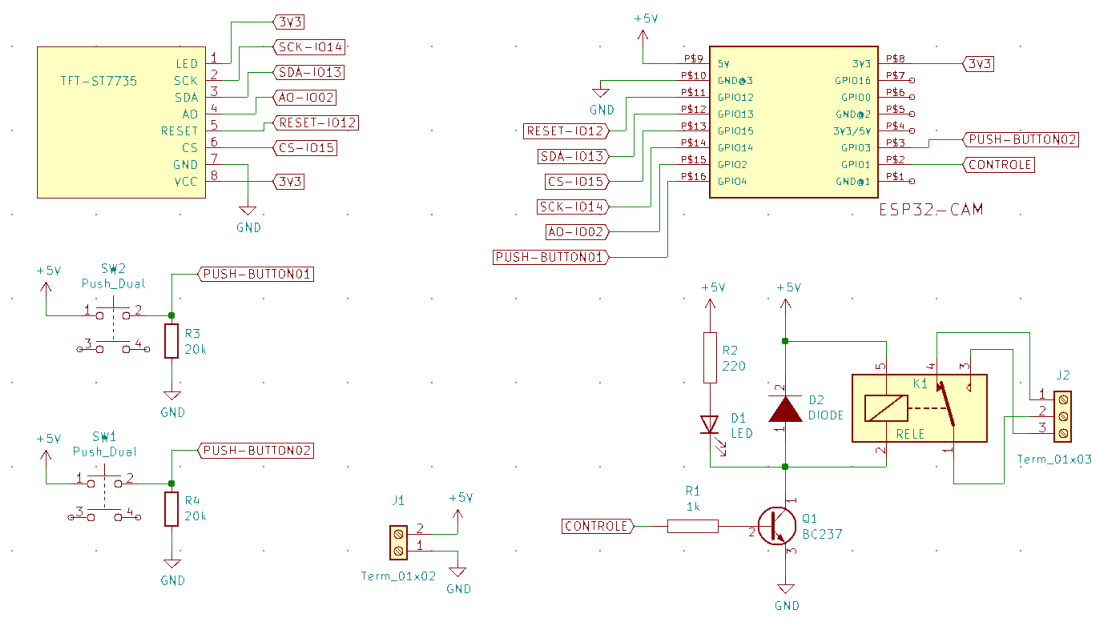
Para o desenvolvimento do protótipo, utilizou-se o *software KiCad* (Figura 19), uma aplicação de código aberto destinada a projetos eletrônicos. Sua finalidade é simplificar a criação de layouts e a conversão destes para placas de circuito impresso (PCB). O *KiCad* apresenta ferramentas abrangentes para a elaboração da estrutura do produto, criação de arte final e visualizações tridimensionais da PCB e seus componentes. Por meio dele, foi criado um diagrama elétrico detalhado, conforme ilustrado na Figura 20. Este diagrama foi construído com base em um circuito previamente montado e testado em uma placa de prototipagem ou *protoboard*.

Figura 19 – Menu KiCad



Fonte: Autoria própria (2023).

Figura 20 – Diagrama elétrico do protótipo



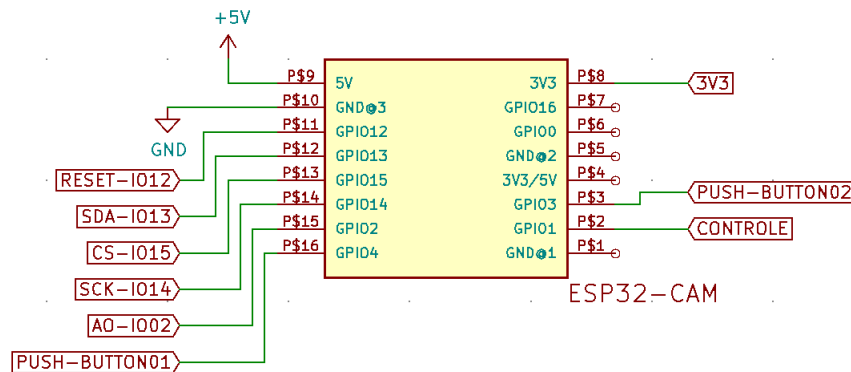
Fonte: Autoria própria (2023).

O diagrama elétrico do protótipo divide o circuito em quatro blocos principais: o ESP32-CAM, o TFT ST7735, o módulo de relé e o módulo de botões (*push button*). O bloco principal, que é o ESP32-CAM, possui 16 pinos de entrada/saída (GPIOs), como representado na Figura 21 e desempenha o papel de controle e coordenação do sistema como um todo. Dos 16 pinos disponíveis, 7 são dedicados ao *display* TFT (GPIOs 2, 12, 13, 14 e 15), 2 são reservados para os botões (GPIOs 4 e 3) e 1 pino é utilizado como saída para controlar o módulo do relé (GPIO 1).

Importante mencionar que algumas GPIOs têm funções específicas. Por exemplo, a

GPIO 16 permanece em nível lógico alto e é utilizada para habilitar a memória PSRAM. Além disso, a GPIO 0 é designada como *clock* da câmera e não pode ser realocada para outras finalidades, uma vez que isso afetaria o funcionamento adequado da captura de imagens. Essas atribuições de GPIOs foram cuidadosamente planejadas para garantir o correto funcionamento de cada componente do protótipo.

Figura 21 – Diagrama elétrico do ESP32-CAM



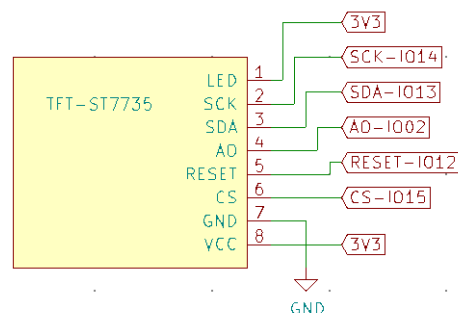
Fonte: A autoria própria (2023).

O segundo bloco corresponde ao controlador ST7735 (Figura 22), um componente projetado para o gerenciamento de *displays* TFT de tamanho reduzido e com capacidade de exibir cores. A principal função desse controlador é administrar a apresentação de informações na tela, permitindo a criação de gráficos e a exibição de texto colorido.

A tela em si é composta por uma matriz de *pixels* coloridos, em que cada *pixel* é formado por subpixels nas cores vermelha, verde e azul (RGB), possibilitando a exibição de uma vasta gama de cores. Os pinos SCK (*Serial Clock*), SDA e CS (*Chip Select*) são utilizados para a comunicação serial com o microcontrolador. Por sua vez, o pino de *reset* é designado para reiniciar o *display* em caso de falhas ou erros.

Adicionalmente, o pino A0 indica se os dados transmitidos se referem a comandos de controle (estado *LOW*) ou a dados de *pixel* (estado *HIGH*), desempenhando um papel crucial no funcionamento adequado do controlador ST7735.

Figura 22 – Diagrama elétrico do *Display* TFT

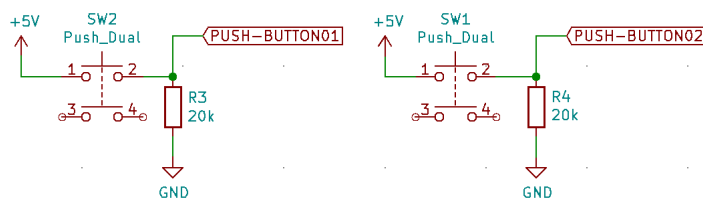


Fonte: A autoria própria (2023).

O bloco que compreende os botões é importante na interatividade do projeto, pois permite a seleção de itens no menu e a inserção dos valores da senha. A estrutura dos botões é ilustrada no circuito apresentado na Figura 23, e cada botão é capaz de enviar um sinal de nível lógico baixo quando não pressionado e de nível lógico alto quando acionado, transmitindo esses sinais ao microcontrolador.

Devido a mudança de estado do botões ao alterar o sinal elétrico conforme são pressionados ou soltos, permite ao microcontrolador reconhecer as ações do usuário, possibilitando a navegação no aplicativo de forma eficiente e prática. Isso faz com que esses componentes contribuam significativamente com a interatividade do protótipo e a experiência do usuário.

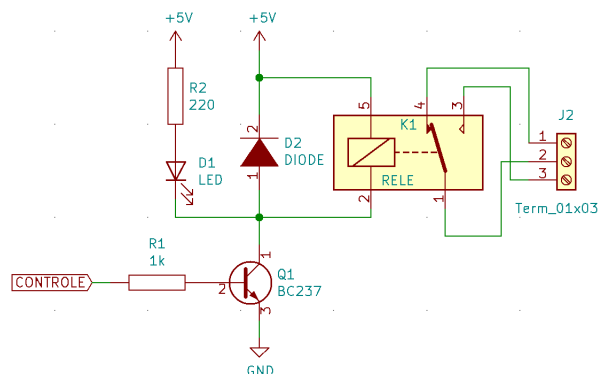
Figura 23 – Diagrama elétrico dos botões



Fonte: Autoria própria (2023).

O módulo de relé, conforme apresentado na (Figura 24), consiste em um circuito que viabiliza que um microcontrolador envie um sinal de baixa potência para a base de um transistor. A conexão do coletor do transistor à bobina do relé viabiliza o seu acionamento quando o transistor entra em estado de saturação. Isso, por sua vez, permite que o ESP32 controle dispositivos de alta potência por meio de sinais de baixa potência. O diodo em paralelo ao relé, também conhecido como diodo de roda livre, desempenha o papel de proteger o circuito de picos ou sobretensão reversa.

Figura 24 – Diagrama elétrico do módulo relé



Fonte: Autoria própria (2023).

A escolha dos conectores do tipo borne KRE para as conexões de entrada e saída do circuito se justifica pela praticidade e segurança que eles oferecem, devido ao seu encaixe lateral que simplifica a conexão com fios e outros módulos eletrônicos.

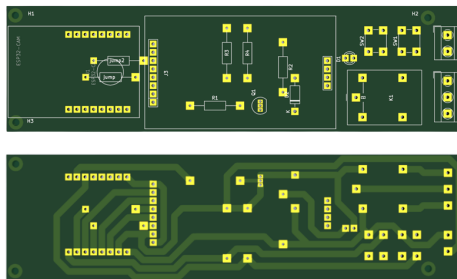
4 RESULTADOS E DISCUSSÃO

De forma geral, o trabalho atingiu todos os objetivos estabelecidos. Desde a elaboração do *hardware* para aquisição dos sinais, como também a implementação de um código otimizado e organizado (detalhado no Apêndice A). Adicionalmente, a construção de uma interface gráfica e física, além da implementação de um módulo para cadastrar e excluir usuários.

4.1 Protótipo *hardware*

Conforme pode ser observado na Figura 25 o projeto da PCB (Placa de Circuito Impresso) foi construído com base no esquema elétrico da Figura 20.

Figura 25 – Projeto da placa de circuito impresso do protótipo



Fonte: Autoria própria (2023).

Após a conclusão da PCB foi feita a montagem da placa, como ilustrado na Figura 26. Nela foi adicionada barras de pinos fêmea no lugar dos conectores do ESP32-CAM e do *Display* TFT, para facilitar a remoção e a substituição desses componentes. Esse recurso é especialmente útil para o ESP32, permitindo a conexão com o gravador sempre que for necessário.

Figura 26 – Placa manufaturada



Fonte: Autoria própria (2023).

E assim que a placa foi confeccionada, projetou-se uma caixa para cumprir dois propósitos essenciais: proteger o circuito eletrônico e oferecer um acabamento esteticamente agradável para o usuário. Após a realização da modelagem 3D e da impressão, o protótipo da caixa foi

acoplada à placa conforme ilustrada na Figura 27. Além de proteger o *hardware*, a caixa contribuiu com uma apresentação mais amigável, melhorando a experiência de uso.

Figura 27 – Placa montada

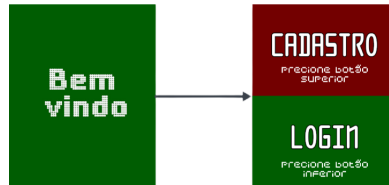


Fonte: Autoria própria (2023).

4.2 Interface gráfica e física

As interfaces gráficas foram elaboradas para auxiliar os usuários na utilização protótipo. Desta forma, o primeiro conjunto de telas (Figura 28) são referente à inicialização dos recursos utilizados no protótipo (câmera, sensores e o sistema de arquivos *flash*).

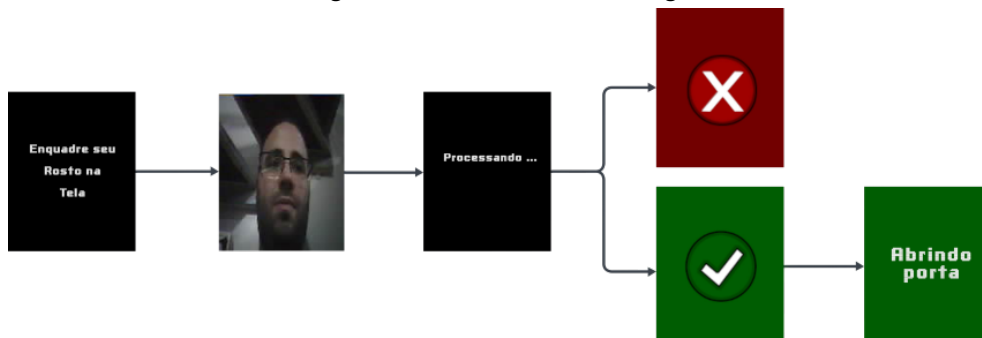
Figura 28 – Telas de inicialização



Fonte: Autoria própria (2023).

Já as telas da Figura 29 correspondem ao fluxo do login, iniciando com a exibição de uma mensagem ao usuário, em seguida a câmera é ligada para iniciar o processo de reconhecimento facial. Se o usuário estiver cadastrado, uma mensagem de sucesso é apresentada; no entanto, se não estiver, é exibida uma mensagem de erro.

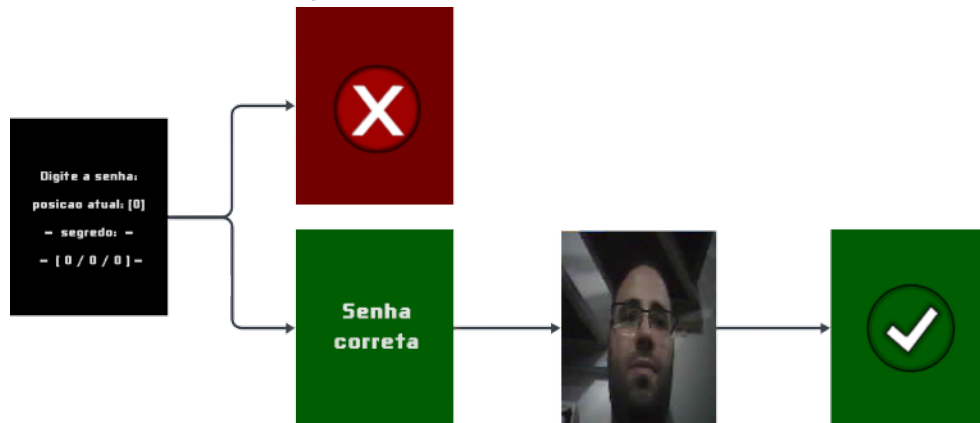
Figura 29 – Fluxo de telas do login



Fonte: Autoria própria (2023).

Outro fluxo semelhante é o do cadastro, conforme ilustrado na Figura 30. Em contraste ao processo de login, a primeira tela solicita uma senha, e as telas subsequentes dependem do valor inserido pelo usuário. Se o usuário inserir corretamente a senha, a tela seguinte exibirá a mensagem "Senha correta", seguida pela ativação da câmera para iniciar o processo de reconhecimento facial. Após a conclusão do cadastro, os dados do reconhecimento são armazenados na memória *flash*, como indicado na Figura 31, e uma mensagem de sucesso é exibida. No caso de algum erro, uma mensagem de erro é apresentada.

Figura 30 – Fluxo de telas do cadastro



Fonte: Autoria própria (2023).

Figura 31 – Lista de arquivos salvos na memória *flash*

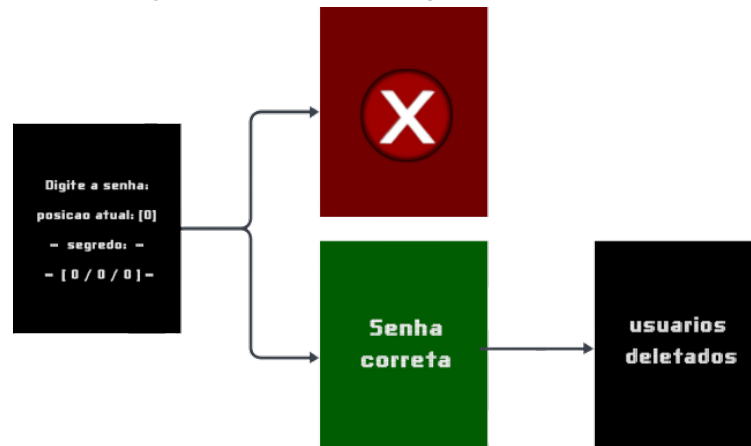
```

Listing SPIFFS files:
=====
File name                               Size
=====
/_count2.jpg                             6996 bytes
/_count3.jpg                             7706 bytes
/_count1.jpg                             5491 bytes
/selfie_t_1.jpg                          3463 bytes
/selfie_f_1.jpg                          36080 bytes
/selfie_t_2.jpg                          3301 bytes
/selfie_f_2.jpg                          42439 bytes
/selfie_t_3.jpg                          3070 bytes
/selfie_f_3.jpg                          36270 bytes
/selfie_t_4.jpg                          2971 bytes
/selfie_f_4.jpg                          35891 bytes
/usuario_1/dados_4.obj                   20 bytes
/usuario_1/dados_3.obj                   20 bytes
/usuario_1/dados_2.obj                   20 bytes
/usuario_1/dados_1.obj                   20 bytes
/usuario_1/dados_0.obj                   20 bytes
=====
  
```

Fonte: Autoria própria (2023).

O último fluxo é o processo de exclusão de usuários (Figura 32), que se assemelha ao fluxo do cadastro. Como pode ser observado, em ambos os casos, a primeira tela solicita a inserção de uma senha e o usuário tem aproximadamente 40 segundos digitar a senha correta. Durante esse tempo, pode ser exibida uma mensagem de "usuários deletados" ou exibido uma mensagem de erro.

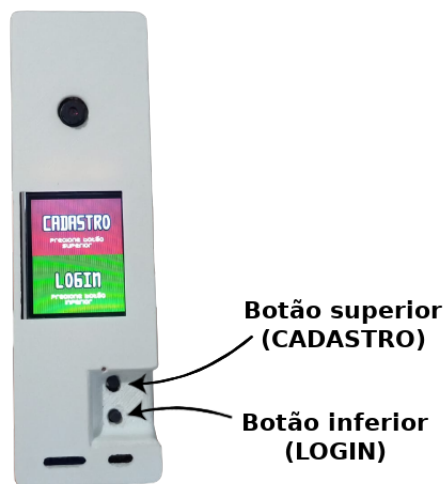
Figura 32 – Fluxo de telas para deletar usuário



Fonte: Autoria própria (2023).

Dentre os fluxos já apresentados, aqueles que requerem senhas correspondem ao módulo do administrador, onde somente ele detém a capacidade de cadastrar ou excluir usuários. No decorrer desses fluxos, destaca-se a relevância da interface física (conforme ilustrado na Figura 33), composta pelos botões. Esses elementos possibilitam ao usuário a seleção tanto para acessar o login quanto para acessar o cadastro. Além disso, esses mesmos botões desempenham um papel importante para inserir os valores na tela de senha.

Figura 33 – Interface física do protótipo



Fonte: Autoria própria (2023).

4.3 Melhorias

Embora todos os objetivos estabelecidos tenham sido alcançados, ainda foi possível identificar algumas possibilidades para aprimorar o protótipo. Por exemplo, alterar o número máximo de usuários que podem se cadastrar. Também a inclusão de IDs para cada usuário, possibilitando a exclusão por meio desses identificadores. Outra consideração é permitir que o administrador altere as senhas associadas ao protótipo.

Uma melhoria interessante seria a ativação do módulo Wi-Fi, permitindo a integração do protótipo com aplicativos ou serviços web.

Por último, uma aprimoramento de grande impacto envolveria a criação de uma placa personalizada, inicialmente utilizando apenas o *chip* ESP32-S. Essa abordagem implicaria na liberação do I2C para uso geral, ampliando a disponibilidade de portas (GPIO) e possibilitando a incorporação de novos recursos.

5 CONCLUSÃO

O trabalho de desenvolvimento de um protótipo de reconhecimento facial alcançou todos os objetivos estabelecidos. O hardware foi projetado e construído com sucesso, utilizando componentes de baixo custo e disponíveis no mercado. O código foi implementado de forma otimizada e organizada, utilizando técnicas de aprendizado de máquina para realizar o reconhecimento facial.

No que diz respeito à interface gráfica e física, a elaboração de telas específicas para a inicialização, login, cadastro e exclusão de usuários foi cuidadosamente projetada para fornecer uma experiência intuitiva ao usuário. Os fluxos de interação, incluindo o módulo do administrador com senhas, destacam a importância dos botões na interface física, desempenhando um papel significativo na navegação e inserção de valores.

Após a avaliação do protótipo, foram identificadas algumas possibilidades de melhoria. Uma delas seria aumentar o número máximo de usuários que podem ser cadastrados. Outra seria incluir IDs para cada usuário, o que permitiria excluí-los por meio desses identificadores. Também seria permitir que o administrador altere as senhas associadas ao protótipo.

Outra melhoria interessante seria a ativação do módulo Wi-Fi, o que permitiria a integração do protótipo com aplicativos ou serviços web. Por fim, uma melhoria de grande impacto envolveria a criação de uma placa personalizada, utilizando apenas o chip ESP32-S. Essa abordagem implicaria na liberação do I2C para uso geral, ampliando a disponibilidade de portas (GPIO) e possibilitando a incorporação de novos recursos.

No geral, o trabalho foi um sucesso e o protótipo desenvolvido apresenta um bom potencial de aplicação em diversos contextos. Com uma abordagem contínua de desenvolvimento e aprimoramento, esse sistema tem o potencial de se tornar uma ferramenta poderosa para controle de acesso em diversas aplicações.

REFERÊNCIAS

- ABDA. **TDah NO ADULTO – ALGUMAS ESTRATÉGIAS PARA O DIA A DIA**. 2023. Disponível em: <https://tdah.org.br/tdah-no-adulto-algumas-estrategias-para-o-dia-a-dia/>. Acesso em: 23 nov. 2023.
- BOECHAT, G. C. **INVESTIGAÇÃO DE UM MODELO DE ARQUITETURA BIOMETRICA MULTIMODAL PARA IDENTIFICAÇÃO PESSOAL**. 2008. Dissertação (Mestrado) — Universidade Federal do Pernambuco, Ciências da Computação, Campinas, 2008.
- CANALTECH. **Flipper Zero: Anatel barra dispositivo que clona radi-ofrequências**. 2023. Disponível em: <https://canaltech.com.br/seguranca/flipper-zero-anatel-barra-dispositivo-que-clona-radiofrequencias-244192>. Acesso em: 22 nov. 2023.
- CAVALCANTI, G. D. da C. **Composição de biometria para sistemas multimodais de verificação de identidade pessoal**. 2005. Tese (Doutorado) — Programa de Pós-Graduação em Ciência da Computação - PPGCC, Universidade Federal de Pernambuco, Pernambuco, 2005.
- ESP-DL. **ESP-DL User Guide**. 2023. Disponível em: <https://docs.espressif.com/projects/esp-dl/en/latest/esp32/index.html>. Acesso em: 23 set. 2023.
- ESPRESSIF SYSTEMS. **FRMN**. 2019. Disponível em: https://github.com/espressif/esp-dl/blob/v0.6.0/face_recognition/README.md. Acesso em: 25 out. 2023.
- ESPRESSIF SYSTEMS. **MTMN**. 2019. Disponível em: https://github.com/espressif/esp-dl/blob/v0.6.0/face_detection/README.md. Acesso em: 25 out. 2023.
- ESPRESSIF SYSTEMS. **ESP32: A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications**. 2022. Disponível em: <https://www.espressif.com/en/products/socs/esp32>. Acesso em: 26 nov. 2022.
- ESPRESSIF SYSTEMS. **ESP32-CAM and Other Cool Projects on RNT**. 2022. Disponível em: https://www.espressif.com/en/news/ESP32_CAM. Acesso em: 04 nov. 2022.
- ESPRESSIF SYSTEMS. **ESP32-S Series**. 2022. Disponível em: <https://www.espressif.com/en/products/devkits>. Acesso em: 04 nov. 2022.
- FERREIRA, A. B. de H. **Dicionário Aurélio Eletrônico**. 5. ed. Curitiba: Positivo, 2009.
- GALIMBERTI, L. H. de O. **ESTUDO COMPARATIVO DE ALGORITMOS DE BIOMETRIA FACIAL DISPONIBILIZADOS PELA BIBLIOTECA OPENCV PARA CONTROLE DE ACESSO**. 2018. 16 p. Dissertação (Mestrado) — UNIVERSIDADE DO VALE DO TAQUARI, Engenharia da Computação, Lajeado, 2018.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. 3. ed. São Paulo: Pearson, 2010.
- LCD WIKI. **1.8inch Arduino SPI Module ST7735S**. 2023. Disponível em: http://www.lcdwiki.com/1.8inch_Arduino_SPI_Module_ST7735S_SKU:MAR1801. Acesso em: 12 out. 2023.
- LECUN, Y. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Acesso em: 27 nov. 2023.

- LUNA, J. I. V. *et al.* System for opening cabinets and equipment containers based on facial recognition. **International Research Journal of Advanced Engineering and Science**, v. 7, p. 7, 7 2022. Disponível em: <https://irjaes.com/wp-content/uploads/2022/07/IRJAES-V7N3P58Y22.pdf>. Acesso em: 22 out. 2023.
- MORAES, J. L. de. **Controle de acesso baseado em biometria facial**. 2010. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática - PPGI, Universidade Federal do Espírito Santo, Vitória, 2010.
- MUKHERJEE, S. *et al.* In: **2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)**. [S.l.: s.n.], 2017. p. 1–5.
- NOE, I. T. **Redes neurais convolucionais aplicadas ao reconhecimento facial em indivíduos com máscara**. 2021. TCC (Graduação) — Engenharia de Computação, Departamento de Computação e Sistemas, Universidade Federal de Ouro Preto, João Monlevade, 2021.
- OPENCV. **Face landmark detection in an image**. 2023. Disponível em: https://docs.opencv.org/4.x/d2/d42/tutorial_face_landmark_detection_in_an_image.html. Acesso em: 27 set. 2023.
- OPENCV. **Introduction to Java Development**. 2023. Disponível em: https://docs.opencv.org/4.x/d9/d52/tutorial_java_dev_intro.html. Acesso em: 26 set. 2023.
- PENIDO Édilus de C. C.; TRINDADE, R. S. **Microcontroladores**. 1. ed. Ouro Preto: Rede e-Tec Brasil, 2013.
- RANDOM NERD TUTORIALS. **ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained**. 2023. Disponível em: <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>. Acesso em: 28 set. 2023.
- ROBO BUILDERS. **Módulo Adaptador Esp32 Cam MB Com Conversor Usb Serial**. 2023. Disponível em: <https://www.robobuilders.com.br/produto/modulo-adaptador-esp32-cam-mb-com-conversor-usb-serial-nfe.html>. Acesso em: 29 out. 2023.
- ROUHANI, S. **Reconhecimento de face e de “prova de vida” com Tensorflow para criação de um sistema de segurança voltado a residências e a ambientes de acesso restrito**. 2019. Dissertação (Mestrado) — Programa de Pós-Graduação em Matemática Estatística e Computação Aplicadas à Indústria - PPGMECAI, Universidade de São Paulo, São Paulo, 2019.
- SUNFOUNDER. **Relay(HIGH) for Arduino and Raspberry Pi**. 2022. Disponível em: [http://wiki.sunfounder.cc/index.php?title=Relay\(HIGH\)_for_Arduino_and_Raspberry_Pi](http://wiki.sunfounder.cc/index.php?title=Relay(HIGH)_for_Arduino_and_Raspberry_Pi). Acesso em: 22 nov. 2022.
- TEIXEIRA, R. F. da S. **Transformação Multi-escala para Segmentação de Impressões digitais**. 2011. Dissertação (Mestrado) — Universidade Estadual de Campinas, Instituto de Computação, Campinas, 2011.
- TERRA. **Soluções de reconhecimento facial ajudarão na reabertura da economia após a pandemia**. 2020. Disponível em: <https://www.terra.com.br/noticias/dino/solucoes-de-reconhecimento-facial-ajudarao-na-reabertura-da-economia-apos-a-pandemia,e5458ffd1885519a92fec6be1b64671fnn9amkk.html>. Acesso em: 19 set. 2022.
- TIINSIDE. **Além da redução de custos: como a automação transforma a cadeia produtiva?** 2023. Disponível em: <https://tiinside.com.br/12/04/2023/>

alem-da-reducao-de-custos-como-a-automacao-transforma-a-cadeia-produtiva/. Acesso em: 23 nov. 2023.

VIEIRA, M. L.; OLIVA, A. D. **Evolução, Cultura e Comportamento Humano**. 1. ed. Florianópolis: Edições do Bosque/CFH/UFSC, 2017.

VIOLA, P.; JONES, M. J. Face recognition: Features versus templates. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 15, p. 1, 11 1993. Disponível em: <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>. Acesso em: 27 set. 2023.

VIOLA, P.; JONES, M. J. Robust real-time face detection. **International Journal of Computer Vision**, v. 57, p. 137–154, 1 2004. Disponível em: <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>. Acesso em: 04 nov. 2022.

ZAFEIRIOU, S.; ZHANG, C.; ZHANG, Z. A survey on face detection in the wild: Past, present and future. **Computer Vision and Image Understanding**, v. 138, p. 24, 9 2015. Disponível em: <https://doi.org/10.1016/j.cviu.2015.03.015>. Acesso em: 28 nov. 2023.

ZHANG, K. *et al.* Joint face detection and alignment using multi-task cascaded convolutional networks. **IEEE Signal Processing Letters**, v. 23, p. 5, 10 2016. Disponível em: <https://doi.org/10.1109/LSP.2016.2603342>. Acesso em: 28 nov. 2023.

ZHAO, W.; CHELLAPPA, R.; PHILLIPS, J. Face recognition: a literature survey. **ACM Computing Survey**, v. 35, p. 61, 12 2003. Disponível em: <https://dl.acm.org/doi/10.1145/954339.954342>. Acesso em: 04 nov. 2022.

APÊNDICE A – Código principal

Abaixo, apresenta-se o código do arquivo principal (main.cpp), fornecendo uma visão dos principais métodos empregados no projeto. Para acessar a implementação completa, por favor, visite o repositório no *GitHub* disponível em: <https://github.com/JoseBarreto1/projeto-tcc-software>.

```

1  #include "face_recognition.h"
2  #include "face_detect.h"
3  #include "constants.h"
4  #include "display.h"
5  #include "camera.h"
6  #include "debug.h"
7  #include "timer.h"
8  #include "user.h"
9
10 // variaveis globais
11 static face_id_list id_list = {0};
12 static bool enroll_enabled = true;
13 static bool reset_password = true;
14 static int user_number = 0;
15 static int selected_option = MENU_OPTION;
16 camera_fb_t *fb = NULL;
17
18 void setup()
19 {
20     // inicializando o pinos de entrada e saída
21     pinMode(RELAY_PIN, OUTPUT);
22     pinMode(PUSH_BUTTON_1, INPUT);
23     pinMode(PUSH_BUTTON_2, INPUT_PULLDOWN);
24     digitalWrite(RELAY_PIN, LOW);
25     debug_init();
26     camera_init();
27     init_sensor_camera();
28     display_init();
29     init_SPIFFS();
30     user_number = find_last_number_user_save();
31     init_face_id(&id_list, user_number);
32 }
33
34 void return_menu(void *arg)
35 {
36     selected_option = MENU_DEFAULT;
37     delay(100);
38     if (fb != NULL)
39     {
40         esp_camera_fb_return(fb);
41         fb = NULL;
42     }
43     digitalWrite(FLASH_PIN, LOW);
44     display_error();
45     pinMode(PUSH_BUTTON_2, INPUT_PULLDOWN);
46     selected_option = MENU_OPTION;

```



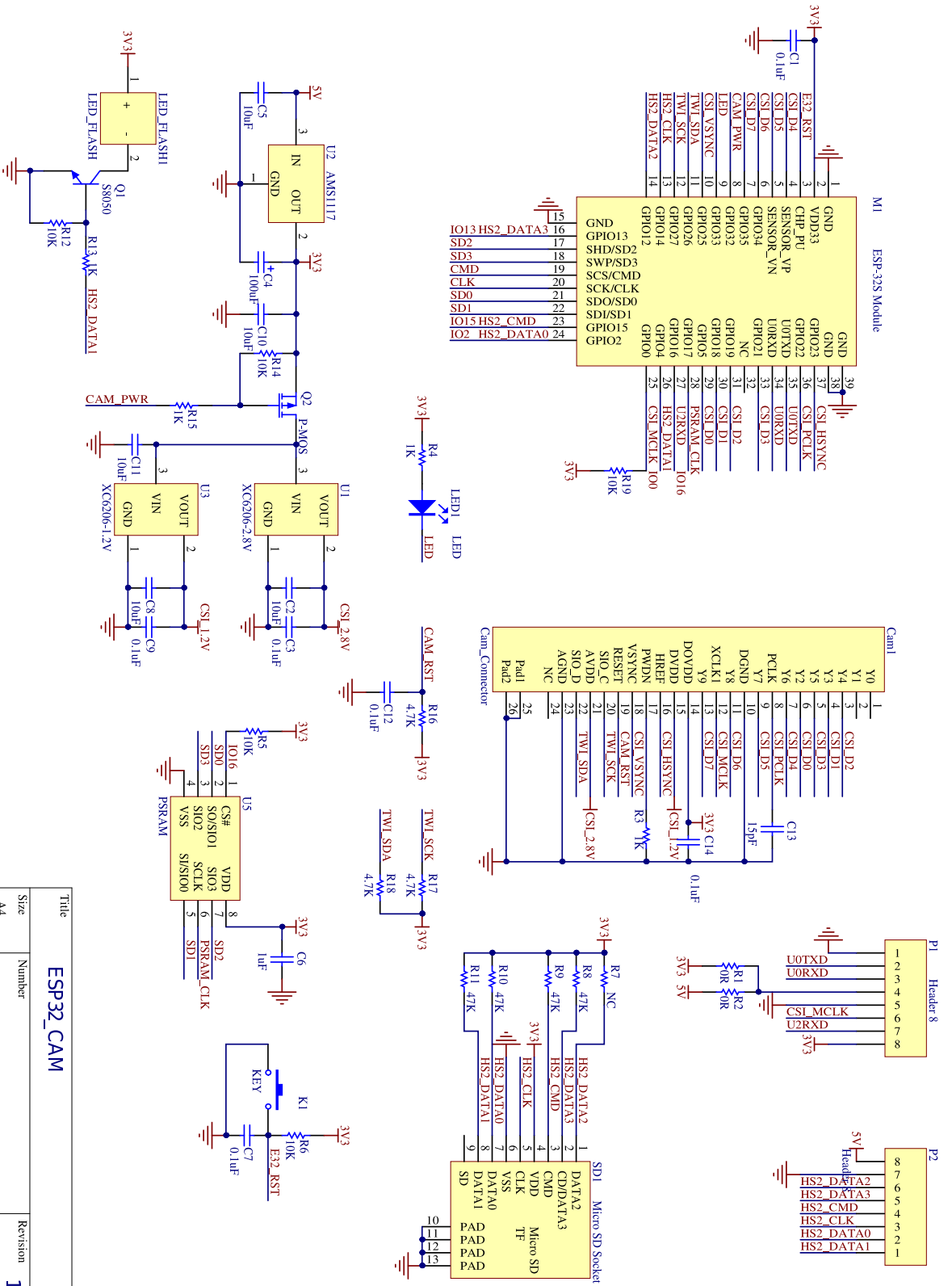
```

47     return;
48 }
49
50 void open_door()
51 {
52     if (digitalRead(RELAY_PIN) == LOW)
53     {
54         digitalWrite(RELAY_PIN, HIGH);
55         display_write_string("Abrindo \n \n porta ", TEXT_SIZE_MEDIUM,
TFT_DARKGREEN);
56         delay(2000);
57         digitalWrite(RELAY_PIN, LOW);
58     }
59     selected_option = MENU_OPTION;
60     return;
61 }
62 void success_face_detected()
63 {
64     stop_timer();
65     display_success();
66     digitalWrite(FLASH_PIN, LOW);
67     pinMode(PUSH_BUTTON_2, INPUT_PULLDOWN);
68     selected_option = OPEN_DOOR;
69     return;
70 }
71
72 void menu_option()
73 {
74     display_menu();
75     if (digitalRead(PUSH_BUTTON_1) == HIGH)
76     {
77         init_timer(TIME_DEFAULT, &return_menu);
78
79         display_write_string("Enquadre seu\n \n Rosto na \n \n
Tela \n", TEXT_SIZE_DEFAULT);
80         delay(4000);
81         pinMode(FLASH_PIN, OUTPUT);
82         digitalWrite(FLASH_PIN, HIGH);
83
84         selected_option = LOGIN_OPTION;
85         enroll_enabled = false;
86         return;
87     }
88     else if (digitalRead(PUSH_BUTTON_2) == HIGH)
89     {
90         init_timer(TIME_DEFAULT, &return_menu);
91         reset_password = true;
92         selected_option = REGISTER_OPTION;
93         return;
94     }
95 }

```

```
96
97 void login()
98 {
99     if (!processing_face(fb, &id_list, &user_number, &enroll_enabled, &
100 success_face_detected) && selected_option != MENU_DEFAULT)
101     {
102         camera_live_stream(fb);
103     }
104 }
105 void registration()
106 {
107     bool push_button_up = digitalRead(PUSH_BUTTON_1) == HIGH;
108     bool push_button_down = digitalRead(PUSH_BUTTON_2) == HIGH;
109     bool is_correct = enter_password(push_button_up, push_button_down, &
110 reset_password);
111     if (is_correct)
112     {
113         init_timer(TIME_DEFAULT, &return_menu);
114         display_initial_count();
115         pinMode(FLASH_PIN, OUTPUT);
116         digitalWrite(FLASH_PIN, HIGH);
117         selected_option = LOGIN_OPTION;
118         enroll_enabled = true;
119         return;
120     }
121 }
122 void loop()
123 {
124     switch (selected_option)
125     {
126     case MENU_OPTION:
127         menu_option();
128         break;
129     case LOGIN_OPTION:
130         login();
131         break;
132     case REGISTER_OPTION:
133         registration();
134         break;
135     case OPEN_DOOR:
136         open_door();
137         break;
138     default:
139         break;
140     }
141 }
```

APÊNDICE B – Esquema elétrico ESP32-CAM



Title		ESP32_CAM	
Size	Number	Revision	
A4		1.6	
Date:	2018/1/30	Sheet	of